KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT DER TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven-Heverlee

COMPUTING ZETA FUNCTIONS OF CURVES OVER FINITE FIELDS

Promotoren:
Prof. dr. ir. B. Preneel
Prof. dr. J. Denef

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Frederik Vercauteren

November 2003

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT DER TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven-Heverlee

COMPUTING ZETA FUNCTIONS OF CURVES OVER FINITE FIELDS

Jury:
Prof. E. Aernoudt, voorzitter
Prof. B. Preneel, promotor
Prof. J. Denef, promotor
Prof. A. Barbé
Dr. R. Lercier (CELAR)
Prof. J.-J. Quisquater (UCL)
Dr. N. Smart (Bristol Univ.)
Prof. M. Van Barel
Prof. J. Vandewalle

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Frederik Vercauteren

U.D.C. 512.75

November 2003

D/2003/7515/43

ISBN 90-5682-433-3

Acknowledgments

Four years have passed since I started doing research; a time during which I have visited many places and have met an even greater number of very interesting people. It truly is a pleasure for me to express my gratitude to some of those who made a real difference.

First of all I would like to thank my promoters, Prof. Bart Preneel and Prof. Jan Denef, both of whom have contributed greatly to this thesis. Prof. Bart Preneel allowed me to do research as a member of the COSIC group; funded the many conferences and workshops I have been to; introduced me to important people and always offered excellent career advice. Prof. Jan Denef answered all of my ill-posed mathematical questions; spent several afternoons (and evenings and nights) explaining his ideas on the blackboard and sacrificed many hours correcting preprints of articles and earlier drafts of this thesis. I can only hope that for every thousand things he taught me, he learned one from me.

I also would like to thank Prof. André Barbé and Prof. Joos Vandewalle for proof-reading this thesis, Dr. Reynald Lercier, Prof. Jean-Jacques Quisquater, Dr. Nigel Smart and Prof. Marc Van Barel for kindly accepting to be a member of my jury and finally, Prof. Etienne Aernoudt for chairing the jury.

The first two years of my time as a Ph.D. student, I spent at the COSIC group and I would like to thank all my colleagues for the excellent working atmosphere. Special thanks to Danny De Cock, Claudia Diaz, Vincent Rijmen, Antoon Bosselaers, Joris Claessens, Jasper Scholten, Siddika Berna Örs, Karel Wouters, Thomas Herlea, Michaël Quisquater, Stefaan Seys, Jorge Nakahara and Bart Van Rompay. Often forgotten, but so indispensable for the proper functioning of the COSIC group; a big thanks to Péla Noë and Elvira Wouters.

The financial support of the F.W.O. (Fund for Scientific Research - Flanders) during this period is greatly appreciated.

At the end of my second year, I moved to the Crypto group at the University of Bristol. I'm especially grateful to Nigel Smart for offering and convincing me to take up a lecturer position at the computer science department; for making me a co-founder and director of Argelcom; for providing a very stimulating working environment and for being a master in sarcasm. Special thanks to all those colleagues who have made my life at least vaguely interesting: Katharina and Florian, John, Matt, Dan, Rob, Peter, Tim, Petrus, Amoss, Angus, Henk, Rich, Oli, Kate, Paul, Sarah and Simone.

Over the years, several people kindly answered my many and sometimes stupid questions. A sincere "thank you" to all of them: Florian Hess, Nigel Smart, Takakazu Satoh, Berit Skjernaa, Pierrick Gaudry, Robert Harley, Nicolas Gürel, Mireille Fouquet, Andreas Enge, François Morain, Steven Galbraith, Jasper Scholten, Reynald Lercier, David Lubicz, Robert Carls, Kiran Kedlaya, Bas Edixhoven, Gerhard Frey and Martin Sombra.

Thanks to all my friends who are always available for a good old chat: Niels and Tinnie, Jakke, Koen, Werner and Brunie, Poulie, Raf and Belle, Beirre, Pierre, Sigi, Arnoud and Kristel.

Finally, I would like to thank my parents for giving me the freedom to do what I like best and for supporting me in all my decisions. Last but not least a big thanks to my brother and sisters and especially to my nephews and nieces. Although they are too young to realize, they have kept me from going insane by reminding me what life is all about: sheer unadulterated fun. I think the time has finally come to have some ...

Abstract

This thesis focuses on p-adic algorithms to compute zeta functions of curves over finite fields. The zeta function contains important arithmetic and geometric information about the curve and its Jacobian. Motivated by practical applications such as cryptography and coding theory, efficient algorithms to compute the zeta function have become increasingly important. Due to their p-adic nature, these algorithms are only practical in small characteristic p.

The first part of this thesis deals with p-adic algorithms to count the number of points on an elliptic curve over a finite field following the work of Satoh. The main idea of Satoh's algorithm is to lift both the curve and the Frobenius endomorphism to a p-adic ring and to recover the number of points directly from the lifted data. The principles of Satoh's original algorithm are explained and a new memory efficient variant is described. Finally, a detailed overview of the more recent developments in elliptic curve p-adic point counting is given.

The second part contains algorithms to compute the zeta function of hyperelliptic curves over finite fields using Monsky-Washnitzer cohomology, i.e. the de Rham cohomology of a ring of overconvergent p-adic power series. The zeta function is recovered from the characteristic polynomial of a Frobenius operator acting on the first Monsky-Washnitzer cohomology group. This line of research was initiated by Kedlaya, who described such an algorithm for hyperelliptic curves over finite fields of odd characteristic. After presenting an overview of Kedlaya's algorithm, a new extension to characteristic two is described.

The final part presents a new algorithm to compute the zeta function of C_{ab} curves over any finite field of small characteristic. This algorithm is again based on Monsky-Washnitzer cohomology, but is much more general than the algorithm for hyperelliptic curves. Furthermore, several lifting and proof techniques are valid for general curves.

Samenvatting

In dit proefschrift bestuderen en ontwerpen we p-adische algoritmen om de zetafunctie van een kromme over een eindig lichaam te berekenen. De zetafunctie bevat belangrijke aritmetische en meetkundige informatie over de kromme en de Jacobiaan van de kromme. Efficiënte algoritmen om de zetafunctie te berekenen hebben praktische toepassingen in cryptografie en codetheorie. Wegens hun p-adische aard zijn deze algoritmen enkel efficiënt voor krommen over eindige lichamen van kleine karakteristiek p.

In het eerste deel van dit proefschrift bestuderen we het algoritme van Satoh om het aantal punten op een elliptische kromme over een eindig lichaam te berekenen. Het basisidee van Satoh's algoritme bestaat erin om tegelijkertijd de kromme en het Frobenius endomorfisme te liften naar een p-adische ring en het aantal punten af te leiden van de gelifte informatie. Na een analyse van Satoh's origineel algoritme, beschrijven we een geoptimaliseerde variant met een lagere geheugencomplexiteit. Tenslotte geven we een gedetailleerd overzicht van de meer recente ontwikkelingen op het gebied van p-adische algoritmen voor elliptische krommen.

Het tweede deel van dit proefschrift behandelt algoritmen om de zetafunctie van een hyperelliptische kromme over een eindig lichaam te berekenen. Deze algoritmen zijn gebaseerd op Monsky-Washnitzer cohomologie, i.e. de de Rham cohomologie van een ring van overconvergente p-adische machtreeksen. De zetafunctie wordt berekend aan de hand van de karakteristieke veelterm van een Frobenius operator die werkt op de eerste cohomologie groep. Deze strategie werd voorgesteld door Kedlaya, die een algoritme gaf voor hyperelliptische krommen over eindige lichamen van oneven karakteristiek. Na een overzicht van Kedlaya's algoritme, beschrijven we een uitbreiding naar karakteristiek twee.

In het derde en laatste deel van dit proefschrift ontwerpen we een nieuw algoritme om de zetafunctie van een C_{ab} kromme over een eindig lichaam van kleine karakteristiek te berekenen. Dit algoritme is eveneens gebaseerd op Monsky-Washnitzer cohomologie, maar is veel algemener dan het algoritme voor hyperelliptische krommen. Bovendien zijn verscheidene stellingen en bewijsvoeringen geldig voor willekeurige krommen.

Contents

List of Symbols

A

$\overline{a}$	reduction of element a
A^{∞}	p-adic completion of ring A
$A^{\dagger}$	dagger ring or weak completion of ring A
$\mathbb{A}^n(\mathbb{K})$	affine space over $\mathbb{K}$ of dimension n
$A[n]$	n-torsion subgroup of abelian variety A

B

$\binom{n}{k}$	binomial coefficient

C

$\cong$	isomorphic with
$\mathcal{C}$	ciphertext space
$\mathbb{C}$	field of complex numbers
C	curve, possibly singular
$\widetilde{C}$	the non-singular model of curve C
χ_{α}	characteristic polynomial of α
$\mathrm{Cl}(S)$	ideal class group of S
$\mathrm{char}\,\mathbb{K}$	characteristic of $\mathbb{K}$

D

D_k	decryption under key k
D^1	universal module of differentials
$\deg(\phi)$	degree of rational map ϕ
$\deg(D)$	degree of divisor D
dim	dimension of vector space
$\mathrm{Div}(C)$	divisor group on curve C
$\mathrm{Div}_{\mathbb{K}}(C)$	$\mathbb{K}$-rational divisor group on curve C
$\mathrm{Div}^0(C)$	degree 0 subgroup of $\mathrm{Div}(C)$
$\mathrm{Div}^0_{\mathbb{K}}(C)$	degree 0 subgroup of $\mathrm{Div}_{\mathbb{K}}(C)$
$\mathrm{div}(f)$	divisor of function f
$\mathrm{div}_0(f)$	zero divisor of function f
$\mathrm{div}_{\infty}(f)$	pole divisor of function f
$\mathrm{div}(\omega)$	divisor of differential form ω

E

$\bigwedge^i$	i-th exterior product
$\equiv$	equivalence relation
ε	strictly positive real number
E	elliptic curve, mostly over characteristic zero field
$\overline{E}$	elliptic curve over finite field
E_k	encryption under key k
$\mathrm{End}(A)$	endomorphism ring of abelian variety A
exp	exponential function

F

$f\|k$	truncated polynomial of degree k
f^h	homogenization of polynomial f
$\overline{F}$	q-th power Frobenius over finite field
$\mathcal{F}$	lift of q-th power Frobenius to characteristic zero
$\mathbb{F}_q$	finite field with q elements

G

g	genus of curve or generator of group
G	abelian group
$\|G\|$	group order of G
$G/\mathbb{K}$	group variety over $\mathbb{K}$
$\mathrm{Gal}(\mathbb{L}/\mathbb{K})$	Galois group of $\mathbb{L}/\mathbb{K}$

H

h_{aff}	affine Hilbert function
h_{proj}	projective Hilbert function
$H^i_{DR}(C/\mathbb{Q}_q)$	i-th algebraic de Rham cohomology group of C
$H^i(\overline{C}/\mathbb{Q}_q)$	i-th Monsky-Washnitzer cohomology group of $\overline{C}$
$H^i(\overline{C}/\mathbb{Q}_q)^-$	anti-invariant part of $H^i(\overline{C}/\mathbb{Q}_q)$
$H^i(\overline{C}/\mathbb{Q}_q)^+$	invariant part of $H^i(\overline{C}/\mathbb{Q}_q)$
$\mathrm{Hom}(A,B)$	ring of homomorphisms from A to B

I

$\imath$	hyperelliptic involution
$\mathcal{I}$	isogeny
$\widehat{\mathcal{I}}$	dual of isogeny $\mathcal{I}$
Id	identity map
$\mathrm{Im}(\phi)$	image of map ϕ
$I(V)$	ideal of algebraic set V
$(h_0,\ldots,f_n)$	ideal generated by $h_0,\ldots,h_n$

J

$j(E)$	j-invariant of elliptic curve E
J_C	Jacobian variety of curve C

K

$\mathcal{K}$	key space
$\mathbb{K}$	perfect field
$\overline{\mathbb{K}}$	algebraic closure of $\mathbb{K}$
$\mathbb{K}[V]$	coordinate ring of variety V
$\mathbb{K}(V)$	function field of variety V
$\mathbb{K}[V]_P$	local ring of variety V at place P
$\mathrm{Ker}(\alpha)$	kernel of homomorphism α
$\mathbb{K}[x_1, \ldots, x_n]$	polynomial ring over $\mathbb{K}$ in n variables
$\mathbb{K}[[x_1, \ldots, x_n]]$	power series ring over $\mathbb{K}$ in n variables

L

$\mathcal{L}(D)$	Riemann-Roch space of divisor D
$l(D)$	dimension of Riemann-Roch space $\mathcal{L}(D)$
$\log r$	natural logarithm of r
$\log_a b$	(discrete) logarithm of b to the base a
$\lim$	direct limit
$\varprojlim$	projective limit

M

$\mathcal{M}$	message space
$\mathfrak{m}$	maximal ideal of local ring
μ	complexity of multiplying two M-bit integers is $O(M^\mu)$
mod	remainder by division

N

$\lvert\cdot\rvert_p$	p-adic norm
n_A	multiplication-by-n on abelian variety A
$\mathbb{N}$	set of natural numbers
$\mathcal{N}(f)$	Newton polytope of polynomial f
$\mathrm{N}_{\mathbb{L}/\mathbb{K}}(\alpha)$	relative norm of α

O

ord_p	p-adic valuation
ord_P	discrete valuation of local ring at P
$\mathcal{O}$	point at infinity on elliptic curve
ω	differential form
Ω_C	vector space of differential forms on curve C

P

p	prime, mostly characteristic of field
$\overline{\pi}$	projection on x-axis
$\pi(a)$	reduction of a
P_∞	place at infinity

$\mathbb{P}^n(\mathbb{K})$	projective space over $\mathbb{K}$ of dimension n
$\mathrm{Pic}(C)$	Picard group of curve C
$\mathrm{Pic}^0(C)$	degree zero part of Picard group of curve C
$\mathrm{Pr}(C)$	set of principal divisors on curve C
ϕ	morphism between curves
ϕ^*	pull-back defined by ϕ
Φ_l	l-th modular polynomial
Ψ_l	l-th division polynomial

Q

q	n-th power of prime p
$\mathbb{Q}$	field of rational numbers
$\mathbb{Q}_p$	field of p-adic numbers
$\mathbb{Q}_q$	unramified extension of p-adic numbers
$\mathbb{Q}_q^{\mathrm{ur}}$	maximal unramified extension of $\mathbb{Q}_q$

R

$\mathbb{R}$	field of real numbers
$R^\times$	set of units in ring R
$\mathrm{Res}(f, g)$	resultant of f and g

S

$\simeq$	approximately equal to
$\sim$	equivalence modulo exact differentials
σ	p-th power Frobenius over finite field
Σ	lift of p-th power Frobenius to characteristic zero
S^*	non-zero elements in set S
$\#S$	number of elements in set S
$S_r(\mathbb{K})$	homogeneous polynomials over $\mathbb{K}$ of degree r
Supp	support of divisor

T

$\otimes$	tensor product
$\mathrm{Tr}(\alpha)$	trace of endomorphism α
$\mathrm{Tr}_{\mathbb{L}/\mathbb{K}}(\alpha)$	relative trace of α
$T_l(A)$	l-adic Tate module of A

V

V_I	algebraic set of ideal I
$\overline{V}$	Verschiebung on elliptic curve over finite field
$\mathcal{V}$	canonical lift of Verschiebung

X

$\lfloor x \rfloor$	largest integer smaller than or equal to x
$\lceil x \rceil$	smallest integer larger than or equal to x

Y

Υ_l	l-th modified modular polynomial

Z

$\mathbb{Z}$	ring of integers
$\mathbb{Z}_p$	ring of p-adic integers
$\mathbb{Z}_q$	valuation ring of unramified extension $\mathbb{Q}_q$
$\mathbb{Z}_q^{\mathrm{ur}}$	valuation ring of maximal unramified extension of $\mathbb{Q}_q$
$Z(X/\mathbb{F}_q;t)$	zeta function of variety X

Chapter 1

Introduction

1.1 Cryptography as Enabling Technology

In the current society, more and more information is stored, transmitted and processed in digital form. This trend is mainly caused by the widespread use of computers and local area networks, the rapid growth of the Internet and the increasing popularity of wireless communications and electronic commerce. Digital information has many advantages over its physical counterpart: huge amounts of data can be stored easily and accessed instantly using databases and data can be communicated fast, cheaply and reliably. However, the very nature of digital information and communication over open networks implies several questions and specific threats that should be addressed if society is to benefit from these advantages.

Unlike paper documents, a copy of an electronic document is identical to the original and an alteration leaves no physical trace. To secure information in electronic form, we therefore need methods which do not depend on the physical medium recording or conveying it, but rely solely on the digital information itself.

Furthermore, appending a handwritten signature to a document uniquely identifies the signer and often serves as a means to validate and authorize its contents. In an electronic setting, a signature cannot simply be something unique to the signer and independent of the document signed, since it would be trivial to copy this signature and append it to any other document.

The field of information security addresses these questions by providing an elaborated set of technical and legal mechanisms to achieve many security objectives such as privacy, authentication, signatures, authorization, validation, access control, certification, witnessing, anonymity, voting, non-repudiation, time-stamping, ...

The technical means to secure information in electronic form are provided by cryptography. The following four security objectives constitute the goals of cryptography and form a framework upon which the others will be derived [82]:

- *Confidentiality* is a service to keep information secret from all but those who are authorized to access it.
- *Data integrity* is a service that addresses the unauthorized alteration of data such as insertion, deletion and substitution and provides mechanisms to detect data manipulation.
- *Authentication* is generally subdivided into
  - *Entity authentication* enables two entities entering into communication to identify each other.
  - *Data origin authentication* provides identification information about the sender or source of the data.
- *Non-repudiation* is a service which prevents an entity from denying previous commitments or actions.

In order to achieve the above objectives, cryptography provides a number of security primitives which can be used as building blocks for a cryptographic protocol, i.e. a distributed algorithm that specifies a sequence of actions required of two or more entities to achieve a security objective [82].

The best known cryptographic primitive is that of an encryption scheme; it can be defined by two families of keyed algorithms representing invertible transformations:

- a set $\{E_k : \mathcal{M} \to \mathcal{C} | k \in \mathcal{K}\}$ of encryption functions,
- a set $\{D_k : \mathcal{C} \to \mathcal{M} | k \in \mathcal{K}\}$ of decryption functions

with $\mathcal{M}$ the message space, $\mathcal{C}$ the ciphertext space and $\mathcal{K}$ the key space. Furthermore, for each key $e \in \mathcal{K}$ there exists a unique key $d \in \mathcal{K}$ such that $D_d = E_e^{-1}$, i.e. $D_d(E_e(m)) = m$ for all $m \in \mathcal{M}$. The keys e and d are called a key pair.

An encryption scheme can be used to achieve confidentiality as follows. Two entities A and B who want to communicate over an insecure channel, first secretly exchange a key pair (e, d). If entity A wants to send a message m to entity B, A computes $c = E_e(m)$ and send this to B. Upon receiving c, B computes $D_d(c) = m$ and recovers the original message m. Note that c can be read by an unauthorized entity E, but since d is secret, it should be impossible for E to recover any information about m.

A fundamental assumption in cryptography, formulated by Kerckhoffs [62] in 1883, is that the encryption and decryption functions should be publicly known and that the security of the system should reside only in the key chosen. The main reason for this assumption is that history has shown that it is very difficult to keep a scheme secret. Furthermore, it also facilitates independent security analysis.

Until the mid 1970's, the only cryptographic techniques available were all symmetric key based, i.e. for each associated key pair (e, d) it is computationally easy to determine d knowing only e and to determine e from d. In most practical symmetric key schemes, $e = d$ which explains the terminology. The best known symmetric key encryption algorithms are the Data Encryption Standard (DES) [35, 36] and its replacement, the Advanced Encryption Standard (AES) [20, 39]. Symmetric key based primitives can be used to achieve confidentiality, digital signatures, data integrity and data origin authentication. The main disadvantage of symmetric key cryptography is the establishment of pairwise secret keys. In a large scale system this typically requires the use of a central server which shares a common key with every user. Note that this introduces a single point of failure, i.e. if the server is compromised all communications are insecure.

In 1976, Diffie and Hellman [26] revolutionized cryptography by introducing a totally new concept called public key (or asymmetric) cryptography. Each entity I now has a pair of keys (e_I, d_I) with the property that knowing E_{e_I}, it is computationally infeasible to find a message $m \in \mathcal{M}$ such that $E_{e_I}(m) = c$ for a given random ciphertext $c \in \mathcal{C}$. This implies that given the public key e_I, it is infeasible to compute the corresponding private key d_I and that the public key e_I can be published without compromising the private key d_I. If entity A now wants to send a message m to entity B, B first sends e_B to A, but keeps d_B secret. A then computes $c = E_{e_B}(m)$ and sends this to B. Finally, B decrypts the ciphertext c by applying the inverse transformation D_{d_B}. Note that it is no longer necessary for each pair of entities who want to communicate to share a common secret key. This makes public key schemes much more scalable than symmetric key schemes and the preferred choice for large, flexible user groups.

Another immediate application of public key cryptography are digital signatures, which provide authentication and non-repudiation. A digital signature on a message m depends on the secret key d_I and m itself. Anyone can verify the signature, since only the public key e_I is needed. To illustrate this concept, consider a reversible public key encryption scheme, i.e. $E_e(D_d(m)) = m$ for all public/private key pairs (e, d). Note that this implicitly assumes that $\mathcal{M} = \mathcal{C}$. To sign a message m it suffices to compute the signature $s = D_d(P(m))$, where P is an appropriate padding scheme. Since the encryption scheme is assumed to be reversible, everyone can verify the signature by testing if $E_e(s) = P(m)$. Informally a digital signature can be seen as proof of knowledge of the secret key d_I, i.e. it should be impossible to devise a valid signature without knowledge of the secret key d_I. Thus, if only entity I knows d_I, then entity I must have produced the signature and cannot deny doing so.

Although Diffie and Hellman described the above principles of public key cryptography, they did not provide an implementation of their ideas. In 1978, Rivest, Shamir and Adleman [103] proposed the first practical public key scheme called RSA. Since the RSA encryption function is reversible, it can be used both for encryption and for digital signatures. The security of RSA is related to the hardness of factoring a large integer n, which acts as the public key e. The prime factors of n constitute the private key d. It has been conjectured that breaking RSA is equivalent to factoring, but this remains unproven and Boneh and Venkatesan [11] showed that RSA is probably not equivalent to factoring when small encryption exponents are used. A good overview of the various attacks on RSA can be found in Boneh [10]. Despite 25 years of research, RSA remains secure (if properly used) and continues to dominate the public key cryptography market; maybe even more so ever since the RSA patent expired in September of 2000.

The only widespread alternative to RSA are public key schemes based on the discrete logarithm problem, most notably in the multiplicative group of finite fields. Diffie and Hellman introduced this problem in their seminal paper [26] and used it as the basis for their well known key agreement protocol. However, it was not until 1985 that ElGamal [28, 29] invented the first public key encryption and signature scheme based on the discrete logarithm problem. Another example is the Digital Signature Algorithm [37], proposed and standardized by the U.S. government's National Institute of Standards and Technology in 1994.

It would appear that public key cryptography is the ideal solution, since it provides all the services set out by information security. Unfortunately, public key cryptography also has several disadvantages which complicate the picture. The first and foremost disadvantage is the necessity to authenticate public keys in order to achieve data origin authentication of the public keys themselves. Indeed, if entity A wants to send an encrypted message to entity B using a public key e, A must be convinced that e really belongs to B and not to an adversary who is impersonating B. This problem can be solved by means of a public key certificate, which provides integrity and authenticity of a public key. A public key certificate consists of a data part, containing the public key and an identity string, and a signature of a Certification Authority on the data part. In order to verify the authenticity of a public key, all that is needed is an authentic copy of the Certification Authority's public key, which is usually provided by non-cryptographic means. A second disadvantage of public key cryptography is that encryption is several orders of magnitude slower than most symmetric key ciphers. Therefore, most systems combine both techniques: public key cryptography is used for key establishment and symmetric key cryptography for secure and efficient communication.

In conclusion, cryptography acts as a fundamental security enabler in the current information society providing indispensable services such as privacy, authentication and non-repudiation. Applications vary from e-commerce and on-line payment systems to mobile phone communications and secure data transmission over open networks.

1.2 This Thesis

In this thesis we propose efficient algorithms to construct abelian groups in which the discrete logarithm problem is believed to be hard. Although the original schemes [26, 28, 29] are all formulated in the setting of the multiplicative group of a finite field $\mathbb{F}_p$, they can be easily generalized to work in any finite cyclic group G. Let g be a generator of G and let h be an element of G. The *discrete logarithm* of h to the base g, denoted $\log_g h$, is the unique integer $0 \leq x < |G|$ such that $h = g^x$. The problem of finding the integer x given g and h is called the *discrete logarithm problem* (DLP). For a group G to be suitable for cryptography it should satisfy the following two conditions:

1. Security: the DLP in G is computationally infeasible,
2. Efficiency: elements of G admit a compact representation and the group operation is efficient.

An easy application of the Chinese remainder theorem and p-adic expansion, as described by Pohlig and Hellman [100], shows that the security of the DLP in G can be reduced to the difficulty of computing discrete logarithms in prime order subgroups of G. An immediate consequence of this reduction is that the order of G should contain a large enough prime factor to preclude solving the DLP in the prime order subgroup.

Given any cyclic group G of prime order p, we can always solve the DLP by exhaustive search using $O(p)$ group operations and storing $O(1)$ group elements, simply by computing $g, g^2, \ldots$ until $h = g^x$ is found. However, there exist algorithms that perform much better: Shanks' baby-step giant-step algorithm [115], which only needs $O(\sqrt{p})$ group operations, but requires storage for $O(\sqrt{p})$ group elements and two randomized algorithms proposed by Pollard [101], called the rho and lambda method respectively, which have the same expected running time as Shanks' algorithm, but need negligible storage. These algorithms are called generic algorithms, since they work in any cyclic group and do not depend on the presentation of the group. A remarkable result by Nechaev [97] and Shoup [116] shows that $\Omega(\sqrt{p})$ group operations is in fact a lower bound for the complexity of generic algorithms to solve the DLP, which means that the above algorithms are essentially optimal within the class of generic algorithms. Since it is widely accepted that 2^{80} operations is currently infeasible, these attacks imply that p should be at least 2^{160}.

In practice however, we always have to use a concrete presentation of the group G, which in many instances allows for faster attacks. The best known example is the index calculus method and often leads to a sub-exponential time algorithm to compute discrete logarithms. The index calculus method requires a relatively small subset $S \subset G$, called the factor base, such that with high probability every element of G can be efficiently expressed as a product of elements in S. By taking discrete logarithms of these expressions, we obtain linear relations involving logarithms of elements of S. When more than $|S|$ such relations have been found, we apply linear algebra over $\mathbb{F}_p$ to recover the discrete logarithms of all elements in S. To compute the discrete logarithm of a particular group element $h = g^x$ we proceed as follows: write h as a product of elements in S, take logarithms of both sides of this expression and recover x as a linear combination of logarithms of elements in S. The complexity of the index calculus method is determined by the size of the factor base: if $|S|$ is too small, the generation of relations involving factor base logarithms will dominate the complexity, whereas if $|S|$ is too large, the linear algebra over $\mathbb{F}_p$ becomes infeasible. Suitable factor bases leading to a sub-exponential algorithm are known for several cyclic groups including the widely used multiplicative groups of the finite fields $\mathbb{F}_{2^n}$ [18], $\mathbb{F}_p$ [52] and $\mathbb{F}_q$ [1]. For the discrete logarithm problem to be computationally infeasible, one therefore has to work in the multiplicative group of fields of size at least 2^{1024}. Note that this is much larger than the theoretical lower bound of 2^{160}, so the question remains if there exist groups that attain this lower bound.

In the last 20 years, several candidates have been proposed for which no attacks faster than the generic ones, are known. All these groups are examples of abelian varieties over finite fields, i.e. the group is the set of solutions to a system of multivariate polynomial equations over a finite field and the group operation is given by rational formulae. However, general abelian varieties are too complicated to work with, since both the number of coordinates to represent an element and the number of operations to compute the group operation grow exponentially with the dimension [95]. In cryptography, one therefore only uses Jacobians of algebraic curves, which do allow an efficient representation and group operation.

Inspired by Lenstra's factoring algorithm [73], Koblitz [65] and Miller [86] independently proposed to replace the multiplicative group $\mathbb{F}_q^*$ by the group of points on an elliptic curve E defined over $\mathbb{F}_q$. In this case the Jacobian $J_E(\mathbb{F}_q)$ is isomorphic to $E(\mathbb{F}_q)$ itself, so the points on E form an abelian group. The main advantage of elliptic curve cryptography (ECC) is that there is no known sub-exponential algorithm to solve the DLP in $E(\mathbb{F}_q)$, except for a few special classes such as supersingular curves [42, 81] and anomalous curves [108, 113, 119]. The best known algorithm to solve the DLP in $E(\mathbb{F}_q)$ for random E is Pollard's rho algorithm [101] and its parallelization by van Oorschot and Wiener [128], both of which work in any group. An immediate consequence is that elliptic curve cryptography provides a similar level of security as traditional cryptosystems, but with the benefit of much smaller key sizes. This makes it the preferred choice for constrained environments such as smart cards or hand-held devices. Furthermore, to aid interoperability, several primitives based on the elliptic curve discrete logarithm problem have now been standardized [5, 6, 38, 53].

A few years later, Koblitz [66] proposed hyperelliptic curve cryptography, which is based on the DLP in the Jacobian $J_C(\mathbb{F}_q)$ of a hyperelliptic curve C over a finite field $\mathbb{F}_q$. The most important invariant of such a curve C is its genus, which is the dimension of the Jacobian variety J_C. Elliptic curves simply are hyperelliptic curves of genus one. However, for genus larger than one, the points on the curve no longer form a group and we really need to work with $J_C(\mathbb{F}_q)$. Group elements are represented by pairs of polynomials over $\mathbb{F}_q$ of degree less than g and the group operation can be computed efficiently using algorithms due to Cantor [14] and Koblitz [66]. The two main advantages over elliptic curves are as follows: firstly, the size of $J_C(\mathbb{F}_q)$ is about q^g with g the genus of C, so for $g \geq 2$ the finite field $\mathbb{F}_q$ can be smaller, possibly allowing faster arithmetic on 64-bit platforms; secondly, it is much easier to implement a true hyperelliptic curve cryptosystem without infringing any patents, than it is for an elliptic curve cryptosystem.

In [87], Miura introduced a very wide class of curves, the so called C_{ab} curves, which generalize elliptic and hyperelliptic curves. The Jacobian $J_C(\mathbb{F}_q)$

of a C_{ab} curve C is isomorphic to the ideal class group of its coordinate ring. Using ideal reduction techniques, Arita [7] and Harasawa and Suzuki [54] gave efficient algorithms for arithmetic in $J_C(\mathbb{F}_q)$.

Apart from the reduction to prime order subgroups by Pohlig and Hellman and the generic algorithms by Shanks and Pollard, there currently exist four classes of Jacobians for which attacks faster than the generic ones are known.

- Large genus. The DLP in the Jacobian $J_C(\mathbb{F}_q)$ of a genus g hyperelliptic curve C can be solved in sub-exponential time if $2g + 2 \geq \log q$ using an index calculus algorithm due to Adleman, DeMarris and Huang [2], Müller, Stein and Thiel [94], Enge [34] and Bauer [8]. Recent work by Hess [58] shows that similar results hold for arbitrary curves of large genus. Gaudry [47] modified these algorithms using the knowledge of $|J_C(\mathbb{F}_q)|$ and showed that for small fixed genus the complexity is $O(q^2)$. Thériault [125] improved Gaudry's algorithm and obtained a running time of $O(g^5 q^\nu)$ with $\nu = 2 - \frac{4}{2g+1} + \varepsilon$. This is still exponential, but faster than the generic attacks for $g \geq 3$.

- Additive reduction. If C is an algebraic curve over a prime field $\mathbb{F}_p$ such that $|J_C(\mathbb{F}_p)| = p$, then the DLP in $J_C(\mathbb{F}_p)$ can be reduced to a DLP in the additive group of the $\mathbb{F}_p$-vector space of holomorphic differentials on C. For elliptic curves, this reduction was described by Semaev [113], Satoh and Araki [108] and Smart [119] and for the general case by Rück [104]. This reduction and the resulting DLP can be computed in time $O(\log p)$.

- Multiplicative reduction. For any algebraic curve C over a finite field $\mathbb{F}_q$, the Tate pairing can be used to reduce the DLP in the Jacobian $J_C(\mathbb{F}_q)$ to the DLP in the multiplicative group of an extension field $\mathbb{F}_{q^k}$. Menezes, Okamoto and Vanstone [81] presented an algorithm specific for elliptic curves, whereas Frey and Rück [42] described the general case. The extension degree k is the order of q in $\mathbb{F}_p^*$, where p is the largest prime factor of $|J_C(\mathbb{F}_q)|$. For almost all curves, k is much too large for this attack to be sub-exponential. Only very special curves, such as supersingular curves, are vulnerable to this attack.

- Weil descent. Every d-dimensional abelian variety V over $\mathbb{F}_{q^n}$ can also be viewed as an nd-dimensional abelian variety over $\mathbb{F}_q$. This latter variety is called the Weil restriction of V with respect to $\mathbb{F}_{q^n}/\mathbb{F}_q$. In 1998, Frey suggested the following strategy to solve the DLP in the Jacobian $J_C(\mathbb{F}_{q^n})$ of an algebraic curve C over $\mathbb{F}_{q^n}$: construct the Weil restriction A of J_C with respect to $\mathbb{F}_{q^n}/\mathbb{F}_q$, find a curve C' defined over $\mathbb{F}_q$ which lies on A and transfer the DLP from $J_C(\mathbb{F}_{q^n})$ to $J_{C'}(\mathbb{F}_q)$. This strategy was worked out for elliptic curves by Gaudry, Hess and Smart [51] and for Jacobians of arbitrary curves by Galbraith [45]. Weil descent cannot be applied to curves over $\mathbb{F}_p$ with p a large prime and currently fails for finite fields $\mathbb{F}_{p^n}$ where n is prime, but not a Fermat prime.

The above attacks can be easily avoided if the Jacobian $J_C(\mathbb{F}_{p^n})$ satisfies the following conditions: the genus of the curve C is small enough ($g \leq 3$), the group order $|J_C(\mathbb{F}_{p^n})|$ is divisible by a large prime $l > 2^{160}$ with $l \neq p$, the smallest $k \in \mathbb{N}$ such that $p^{nk} \equiv 1 \bmod l$ is large enough ($k > 50$) and n is either prime or 1. In conclusion, the hardness of the DLP in the Jacobian $J_C(\mathbb{F}_q)$ mainly depends on the group order $|J_C(\mathbb{F}_q)|$.

For an elliptic curve E, computing the group order of $J_E(\mathbb{F}_q)$ is equivalent to computing the number of points on $E(\mathbb{F}_q)$. In general, the group order of $J_C(\mathbb{F}_q)$ is determined by the number of points on C with coordinates in the g extension fields $\mathbb{F}_{q^k}$ for $k = 1, \ldots, g$. Let $N_k = |C(\mathbb{F}_{q^k})|$ for $k \in \mathbb{N}$, then the zeta function $Z(C/\mathbb{F}_q; t)$ of the curve C is defined as the generating function

$$Z(C/\mathbb{F}_q; t) = \exp\left(\sum_{k=1}^{\infty} \frac{N_k}{k} t^k\right).$$

Weil [136] proved that the zeta function of a smooth projective curve C has the following remarkable properties: let g be the genus of C, then $Z(C/\mathbb{F}_q; t)$ is a rational function of the form

$$Z(C/\mathbb{F}_q; t) = \frac{L(t)}{(1-t)(1-qt)},$$

with $L(t) = 1 + c_1 t + \cdots + c_{2g-1} t^{2g-1} + q^g t^{2g} \in \mathbb{Z}[t]$ and if $L(t) = \prod_{i=1}^{2g}(1 - \alpha_i t)$ then the α_i are algebraic integers of absolute value $\sqrt{q}$. Furthermore, the group order $|J_C(\mathbb{F}_q)|$ is given by $L(1)$.

In this thesis, we present algorithms to compute the zeta function $Z(C/\mathbb{F}_q; t)$ for all types of curves used in cryptography, i.e. elliptic, hyperelliptic and C_{ab} curves. As a byproduct we also obtain the group order of the Jacobian $J_C(\mathbb{F}_q)$, which determines the hardness of the DLP in $J_C(\mathbb{F}_q)$. These algorithms are practical when the characteristic p of $\mathbb{F}_q$ is small, with $p = 2$ being the most important for applications in cryptography. Note that as recent as 1999, it was still completely impossible to compute the zeta function of hyperelliptic and C_{ab} curves for cryptographic applications, thereby hampering the take-up of cryptosystems based on the DLP in the Jacobian of these curves.

1.3 Outline and Main Contributions

This thesis is based on a number of papers and preprints describing algorithms to compute zeta functions of elliptic, hyperelliptic and C_{ab} curves; each type of curve representing one major chapter. The outline and main contributions are as follows.

Chapter 2 - Mathematical Background. This chapter briefly recalls the definitions and properties of the mathematical objects that will be used throughout the thesis. The most important topics are curves, function fields, differentials and the Riemann-Roch theorem; the construction of the Jacobian of a curve, especially elliptic, hyperelliptic and C_{ab} curves; isogenies and the modular equations $\Phi_l(X,Y)$; the structure of the torsion subgroups of abelian varieties; the zeta function and the Weil conjectures; the Frobenius morphism and its characteristic polynomial; p-adic fields and finally, complexity theory.

Chapter 3 - Elliptic Curves. This chapter gives a detailed overview of all existing p-adic algorithms to count the number of points on an elliptic curve over a finite field $\mathbb{F}_{p^n}$ of small characteristic p. This line of research was initiated by the breakthrough work of Satoh [105] at the end of 1999. The key idea of Satoh's algorithm is to lift both the curve and the Frobenius endomorphism to a p-adic ring and to recover the number of points from the lifted data. The original algorithm only worked for $p \geq 5$, but was shortly after extended to the cases $p = 2$ and $p = 3$ by Fouquet, Gaudry and Harley [41] and independently by Skjernaa [118] to the case $p = 2$. For fixed p, the time complexity of all these algorithms is $O(n^{3+\varepsilon})$ bit-operations using fast arithmetic and the space complexity amounts to $O(n^3)$. This is asymptotically faster than the l-adic approach by Schoof, Elkies and Atkin [30, 111, 112] which needs $O((\log q)^{4+\varepsilon})$ bit-operations using fast arithmetic and $O((\log q)^2)$ memory where $q = p^n$.

The main contribution of this chapter is a memory efficient version of Satoh's algorithm. More precisely, we propose an algorithm to count the number of points on an elliptic curve over a finite field $\mathbb{F}_{p^n}$ which still runs in $O(n^{3+\varepsilon})$ bit-operations, but only needs $O(n^2)$ memory. The algorithm works for any small characteristic and is even faster than the original by a factor of about 1.5. The second contribution of this chapter is a detailed description of the more recent developments in p-adic point counting for elliptic curves, culminating in an algorithm by Harley which runs in $O(n^{2+\varepsilon})$ time and requires $O(n^2)$ memory. Parts of this chapter have been published in the article [133] and an abbreviated version will appear as a contributed chapter in the books [9] and [17].

Chapter 4 - Hyperelliptic Curves. In this chapter we present several algorithms to compute the zeta function of a hyperelliptic curve following the work of Kedlaya [61]. These algorithms are all based on Monsky-Washnitzer cohomology and compute the zeta function from the characteristic polynomial of a Frobenius operator acting on p-adic cohomology groups, i.e. finite dimensional vector spaces. Kedlaya's algorithm works for hyperelliptic curves over finite fields $\mathbb{F}_{p^n}$ of small, odd characteristic. For fixed p, the time complexity for a genus g hyperelliptic curve over $\mathbb{F}_{p^n}$ is $O(g^{4+\varepsilon}n^{3+\varepsilon})$ bit-operations and the space complexity is $O(g^3n^3)$.

The contribution of this chapter is an extension of Kedlaya's algorithm to characteristic 2, which for cryptography is the most important case. To ease the exposition, we proceed in two steps: the first algorithm is limited to hyperelliptic curves of rank 0 or 1, whereas the second algorithm works for arbitrary hyperelliptic curves over $\mathbb{F}_{2^n}$. The time complexity for a genus g curve is $O(g^{4+\varepsilon}n^{3+\varepsilon})$ in the average case and $O(g^{5+\varepsilon}n^{3+\varepsilon})$ in the worst case and the space complexity is $O(g^3n^3)$ and $O(g^4n^3)$ respectively. The results of this chapter are joint work with Jan Denef and have been published in the articles [23, 24, 131] and will appear in abbreviated form in the book [17].

Chapter 5 - C_{ab} Curves. The contribution of this chapter is a new algorithm to compute the zeta function of a C_{ab} curve over any finite field of small characteristic. The algorithm requires $O(g^{5+\varepsilon}n^{3+\varepsilon})$ bit-operations and $O(g^3n^3)$ space for a C_{ab} curve of genus g defined over a finite field $\mathbb{F}_{p^n}$, with p fixed.

The algorithm is again based on Monsky-Washnitzer cohomology, but is much more general than the algorithm described in Chapter 4. Furthermore, the techniques used to lift the Frobenius endomorphism and to prove a tight bound on the convergence rate of the resulting power series are valid for any non-singular affine curve. Only the explicit description of a basis for the first Monsky-Washnitzer cohomology group and the reduction formulae to express any differential on this basis are limited to C_{ab} curves. The results of this chapter include some proof techniques by Jan Denef and will be published in the article [132].

Chapter 6 - Conclusions. This chapter concludes the thesis and lists several open problems related to the computation of zeta functions of curves over finite fields.

Chapter 2

Mathematical Background

This chapter briefly recalls the definitions and theorems that will be needed throughout the thesis. More details can be found in the main references: Danilov and Shokurov [21], Silverman [117], Cornell and Silverman [19] and Serre [114].

In Section 2.1 we introduce the Picard group of a curve and construct a unique representative for each divisor class using the Riemann-Roch theorem. This construction is then specialized to elliptic, hyperelliptic and C_{ab} curves. In Section 2.2 we identify the Picard group with an abelian variety, called the Jacobian variety, introduce the zeta function and the Weil conjectures and show how the order of the Picard group can be recovered from the characteristic polynomial of Frobenius on the Tate modules. In Section 2.3 we recall the basic properties of p-adic fields and finally, in Section 2.4 we introduce some complexity theoretic notation.

2.1 Curves and Jacobians

Let $\mathbb{K}$ be a perfect field, typically a finite field or a field of characteristic 0. Let $\overline{\mathbb{K}}$ be a fixed algebraic closure of $\mathbb{K}$ and let $\mathrm{Gal}(\overline{\mathbb{K}}/\mathbb{K})$ denote the Galois group of $\overline{\mathbb{K}}/\mathbb{K}$.

2.1.1 Affine Varieties

Let $\mathbb{A}^n$ denote affine n-space over $\overline{\mathbb{K}}$, i.e. the set of n-tuples with coordinates in $\overline{\mathbb{K}}$ and let $I \subset \overline{\mathbb{K}}[X_1, \ldots, X_n]$ an ideal. To each such ideal I we associate the affine algebraic set

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

The ideal $I(V)$ of an affine algebraic set V is the ideal of polynomials that are zero on V.

DEFINITION 2.1.1 *An affine algebraic set V is called an affine variety if $I(V)$ is a prime ideal in $\overline{\mathbb{K}}[X_1, \ldots, X_n]$. The affine variety V is defined over $\mathbb{K}$, denoted with $V/\mathbb{K}$ if $I(V)$ can be generated by polynomials in $\mathbb{K}[X_1, \ldots, X_n]$.*

The affine coordinate ring of $V/\mathbb{K}$ is defined by

$$\mathbb{K}[V] = \mathbb{K}[X_1, \ldots, X_n]/I(V/\mathbb{K}).$$

Since $I(V/\mathbb{K})$ is prime, $\mathbb{K}[V]$ is an integral domain and its quotient field $\mathbb{K}(V)$ is called the function field of $V/\mathbb{K}$. The dimension $\dim(V)$ of V is the transcendence degree of $\overline{\mathbb{K}}(V)$ over $\overline{\mathbb{K}}$.

Let $f_1, \ldots, f_m \in \overline{\mathbb{K}}[X_1, \ldots, X_n]$ be generators for $I(V)$. A point $P \in V$ is called non-singular if the $m \times n$ matrix

$$\left(\frac{\partial f_i}{\partial X_j}\right)_{1 \le i \le m, 1 \le j \le n}$$

has rank $n - \dim(V)$. If V is non-singular at every point, then V is called non-singular or smooth.

DEFINITION 2.1.2 *The local ring of V at P, denoted by $\overline{\mathbb{K}}[V]_P$, is defined as*

$$\overline{\mathbb{K}}[V]_P = \{F \in \overline{\mathbb{K}}(V) | F = \frac{f}{g} \text{ with } f, g \in \overline{\mathbb{K}}[V] \text{ and } g(P) \neq 0\}.$$

The elements of $\overline{\mathbb{K}}[V]_P$ are called regular at P.

2.1.2 Projective Varieties

Let $\mathbb{P}^n$ denote projective n-space, i.e. the non-zero points in $\mathbb{A}^{n+1}$ modulo the equivalence relation given by multiplication with a unit. Note that for each $0 \le i \le n$ we get a decomposition $\mathbb{P}^n = \mathbb{A}_i^n \cup \mathbb{P}_i^{n-1}$ where the affine part $\mathbb{A}_i^n$ is obtained by taking the i-th coordinate 1 and the projective part $\mathbb{P}_i^{n-1}$ by setting the i-th coordinate 0. Since elements of $\mathbb{P}^n$ are defined up to multiplication by a unit, it only makes sense to look at the zero sets of homogeneous polynomials. For an ideal $I \subset \overline{\mathbb{K}}[X_0, \ldots, X_n]$ generated by homogeneous polynomials, we define the projective algebraic set V_I as in the affine case. If V is a projective algebraic set, then $I(V)$ is the ideal generated by the homogeneous polynomials which vanish on V.

DEFINITION 2.1.3 *A projective algebraic set V is called a projective variety if the homogeneous ideal $I(V)$ is prime in $\overline{\mathbb{K}}[X_0, \ldots, X_n]$. The projective variety V is defined over $\mathbb{K}$, denoted with $V/\mathbb{K}$ if $I(V)$ can be generated by homogeneous polynomials in $\mathbb{K}[X_0, \ldots, X_n]$.*

Given a projective variety V, we can consider the decomposition

$$V = (V \cap \mathbb{A}_i^n) \cup (V \cap \mathbb{P}_i^{n-1})$$

for a fixed i. The part $V \cap \mathbb{A}_i^n$ is called the affine part and is in fact an affine variety and the second part $V \cap \mathbb{P}_i^{n-1}$ is called the set of points at infinity. Let $V/\mathbb{K}$ be a projective variety and choose i such that $V \cap \mathbb{A}_i^n \neq \emptyset$, then the dimension of V is defined as $\dim(V \cap \mathbb{A}_i^n)$ and the function field of V is the function field of $V \cap \mathbb{A}_i^n$. Note that this is well defined since all the function fields $\mathbb{K}(V \cap \mathbb{A}_i^n)$ are isomorphic. Given an affine variety V, we can consider the $n+1$ embeddings in $\mathbb{P}^n$ given by the map

$$V \subset \mathbb{A}^n \to \mathbb{P}^n : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_{i-1}, 1, x_i, \ldots, x_n).$$

The projective closure of V, denoted by $\overline{V}$, is the projective algebraic set such that the homogeneous polynomials in $I(\overline{V})$ are zero on the image of V under the above map.

Let V be a projective variety and $P \in V$. Choose an i such that $P \in (V \cap \mathbb{A}_i^n)$, then P is called non-singular if P is non-singular on $V \cap \mathbb{A}_i^n$. The local ring of V at P is the local ring of $V \cap \mathbb{A}_i^n$ at P.

A rational map between two varieties V_1 and $V_2 \subset \mathbb{P}^n$ is a map of the form $\phi = [f_0, \ldots, f_n]$ with $f_0, \ldots, f_n \in \overline{\mathbb{K}}(V_1)$ such that for every point $P \in V_1$ at which $f_0, \ldots, f_n$ are all defined $\phi(P) = [f_0(P), \ldots, f_n(P)] \in V_2$. A rational map which is regular (i.e. defined) at every point is called a morphism. The varieties V_1 and V_2 are called isomorphic if there are morphisms $\phi : V_1 \to V_2$ and $\psi : V_2 \to V_1$ such that $\psi \circ \phi$ and $\phi \circ \psi$ are the identity maps on V_1 and V_2 respectively.

2.1.3 Curves and Divisors

A curve C is a projective variety of dimension 1. In what follows we assume that the curve is smooth, although we often work with singular models. Given a non-singular point, we can study the behavior of functions in this point. The following theorem shows that the local ring of a non-singular point has a very simple structure.

THEOREM 2.1.1 *Let C be a curve and $P \in C$ a smooth point, then $\overline{\mathbb{K}}[C]_P$ is a discrete valuation ring. The unique maximal ideal $M_P \in \overline{\mathbb{K}}[C]_P$ consists of the functions which are zero in P.*

For $\overline{\mathbb{K}}[C]_P$ we define the normalized valuation ord_P by

$$\mathrm{ord}_P : \overline{\mathbb{K}}[C]_P \to \mathbb{N} \cup \{\infty\} : f \mapsto \mathrm{ord}_P(f) = \max\{d \in \mathbb{Z} | f \in M_P^d\}.$$

The valuation ord_P can be easily extended to the whole of $\overline{\mathbb{K}}(C)$ by defining $\mathrm{ord}_P(f/g) = \mathrm{ord}_P(f) - \mathrm{ord}_P(g)$. A uniformizer for C at P is a function $t \in \overline{\mathbb{K}}(C)$ with $\mathrm{ord}_P(t) = 1$. Note that if $\mathrm{ord}_P(f) < 0$ then f has a pole in P and if $\mathrm{ord}_P(f) > 0$ then f has a zero in P. Furthermore, the unique maximal ideal satisfies $M_P = \{f \in \overline{\mathbb{K}}(C) | \mathrm{ord}_P(f) > 0\} = t\overline{\mathbb{K}}[C]_P$.

The above theorem associates to each non-singular point $P \in C$ a unique discrete valuation ring $\overline{\mathbb{K}}[C]_P$ such that for all functions $f \in \overline{\mathbb{K}}(C)$ either we have $f \in \overline{\mathbb{K}}[C]_P$ or $1/f \in \overline{\mathbb{K}}[C]_P$. Given the function field $\overline{\mathbb{K}}(C)$ of a possibly singular curve, there always exists a projective non-singular curve $\widetilde{C}$, called the non-singular model, such that its function field $\overline{\mathbb{K}}(\widetilde{C})$ is isomorphic to $\overline{\mathbb{K}}(C)$. This abstract curve is defined as the set of all valuation rings $\mathcal{O}$ in $\overline{\mathbb{K}}(C)$, i.e. $\overline{\mathbb{K}} \subsetneq \mathcal{O} \subsetneq \overline{\mathbb{K}}(C)$ and for each $f \in \overline{\mathbb{K}}(C)$ either $f \in \mathcal{O}$ or $1/f \in \mathcal{O}$. The unique maximal ideal of $\mathcal{O}$ is called a place of $\overline{\mathbb{K}}(C)$ and corresponds to a point P on $\widetilde{C}$. From now on we will always assume that the curve C is projective and non-singular.

Definition 2.1.4 *The group of divisors* $\mathrm{Div}(C)$ *on a curve* C *is the free abelian group generated by the points of* C*, i.e. a divisor is a formal sum* $\sum_{P_i \in C} n_i P_i$ *with only finitely many* $n_i \neq 0$.

The degree of a divisor $D = \sum_{P_i \in C} n_i P_i$ is given by $\deg(D) = \sum_{P_i \in C} n_i$ and the support of D is the set of points $P_i \in C$ for which $n_i \neq 0$. A divisor D is called effective if all $n_i \geq 0$. This defines a partial ordering $\geq$ on $\mathrm{Div}(C)$ such that $D \leq D'$ if $D - D'$ is effective. The subgroup of divisors of degree 0 is denoted $\mathrm{Div}^0(C)$.

Every element of the Galois group $\mathrm{Gal}(\overline{\mathbb{K}}/\mathbb{K})$ acts on $D = \sum_{P_i \in C} n_i P_i$ by acting on the coordinates of the points P_i. The subgroup of $\mathrm{Div}(C)$ fixed by $\mathrm{Gal}(\overline{\mathbb{K}}/\mathbb{K})$ is denoted $\mathrm{Div}_{\mathbb{K}}(C)$ and a divisor $D \in \mathrm{Div}_{\mathbb{K}}(C)$ is called $\mathbb{K}$-rational. Note that the points P_i in a $\mathbb{K}$-rational divisor $D = \sum_{P_i \in C} n_i P_i$ do not have to be defined over $\mathbb{K}$. It is sufficient that the Galois actions leave the divisor as a whole unchanged, e.g. by permuting points P_i with equal n_i. A $\mathbb{K}$-rational divisor D is called prime if D is effective and for any effective $\mathbb{K}$-rational divisor $D' \leq D$, either $D' = 0$ or $D' = D$.

2.1.4 Principal Divisors and Picard Group

The next theorem associates in a natural way a degree 0 divisor to a function $f \in \overline{\mathbb{K}}(C)$.

Theorem 2.1.2 *Let* $f \in \overline{\mathbb{K}}(C)$ *be a non-zero function on* C*, then there exists only a finite number of points* P *with* $\mathrm{ord}_P(f) \neq 0$*. Furthermore, the divisor*

$$\mathrm{div}(f) = \sum_{P_i \in C} \mathrm{ord}_{P_i}(f) P_i$$

has degree 0.

A divisor of the form $\mathrm{div}(f)$ is called a principal divisor and the set of principal divisors is denoted $\mathrm{Pr}(C)$. Note that $\mathrm{Pr}(C)$ is a subgroup of $\mathrm{Div}^0(C)$, since $\mathrm{div}(f) + \mathrm{div}(g) = \mathrm{div}(fg)$. A principal divisor is normally written as the difference of the zero divisor $\mathrm{div}_0(f) \geq 0$ and the pole divisor $\mathrm{div}_\infty(f) \geq 0$.

Definition 2.1.5 *The Picard group or divisor class group of* C *is defined as the quotient* $\mathrm{Pic}(C) = \mathrm{Div}(C)/\mathrm{Pr}(C)$.

Two divisors D_1 and D_2 whose images in $\mathrm{Pic}(C)$ are equal are called linearly equivalent, i.e. there exists a function $f \in \overline{\mathbb{K}}(C)$ such that $D_1 = D_2 + \mathrm{div}(f)$. The $\mathbb{K}$-rational Picard group $\mathrm{Pic}_{\mathbb{K}}(C)$ is the subgroup of $\mathrm{Pic}(C)$ fixed by $\mathrm{Gal}(\overline{\mathbb{K}}/\mathbb{K})$. Considering the kernel of the degree map, we obtain the degree 0 part of the divisor class group $\mathrm{Pic}^0(C) = \mathrm{Div}^0(C)/\mathrm{Pr}(C)$ and also $\mathrm{Pic}^0_{\mathbb{K}}(C)$ which is the subgroup of $\mathrm{Pic}^0(C)$ stable under $\mathrm{Gal}(\overline{\mathbb{K}}/\mathbb{K})$. In general, $\mathrm{Pic}^0_{\mathbb{K}}(C)$ is not the quotient of $\mathrm{Div}^0_{\mathbb{K}}(C)$ by its subgroup of principal divisors $\mathrm{Pr}_{\mathbb{K}}(C)$, but the next theorem shows that $C(\mathbb{K}) \neq \emptyset$ is sufficient.

THEOREM 2.1.3 *Let C be a curve defined over $\mathbb{K}$ with a $\mathbb{K}$-rational point and let $D \in \mathrm{Div}^0_{\mathbb{K}}(C)$. If there exists a function $f \in \overline{\mathbb{K}}(C)$ such that $D' = D + \mathrm{div}(f)$ is defined over $\mathbb{K}$, then there also exists a function $g \in \mathbb{K}(C)$ with $\mathrm{div}(g) = \mathrm{div}(f)$.*

2.1.5 Differentials and the Riemann-Roch Theorem

Let C be a curve with function field $\overline{\mathbb{K}}(C)$, then the space of differential forms on C, denoted Ω_C, is the $\overline{\mathbb{K}}(C)$-vector space generated by symbols df for $f \in \overline{\mathbb{K}}(C)$ satisfying the usual rules for differentiation, especially the so-called Leibniz rule $d(fg) = fdg + gdf$. Since C has dimension one, Ω_C is a one-dimensional vector space over $\overline{\mathbb{K}}(C)$.

Let $P \in C$ be a non-singular point on C and let $t \in \overline{\mathbb{K}}(C)$ be a uniformizer at P. For every $\omega \in \Omega_C$ there exists a unique function $g \in \overline{\mathbb{K}}(C)$ depending on ω and t such that $\omega = g\,dt$. The function g is denoted by ω/dt. The quantity $\mathrm{ord}_P(\omega/dt)$ is well defined and depends only on ω and P and not on the choice of the uniformizer t. This value is called the order of ω at P and is denoted by $\mathrm{ord}_P(\omega)$.

For all but finitely many $P \in C$ we have that $\mathrm{ord}_P(\omega) = 0$, so we can associate a divisor to ω defined by $\mathrm{div}(\omega) = \sum_{P \in C} \mathrm{ord}_P(\omega)P$. A differential ω for which $\mathrm{div}(\omega)$ is effective is called regular or holomorphic. Since Ω_C is a one-dimensional vector space over $\overline{\mathbb{K}}(C)$ it immediately follows that any differential maps to the same divisor class in $\mathrm{Pic}(C)$. Any divisor in this class is called a canonical divisor.

For any divisor D, we can consider the vector space of functions

$$\mathcal{L}(D) = \{f \in \overline{\mathbb{K}}(C)^* | \mathrm{div}(f) + D \geq 0\} \cup \{0\}\,.$$

For example, if $D = nP$ for a point $P \in C$, then $\mathcal{L}(D)$ consists of the functions which are regular everywhere except possibly in P, where it can have a pole of order at most n. The $\overline{\mathbb{K}}$-vector space $\mathcal{L}(D)$ is finite dimensional and its dimension is denoted by $l(D) = \dim_{\overline{\mathbb{K}}} \mathcal{L}(D)$.

Let ω be a differential, then $\mathcal{L}(\mathrm{div}(\omega))$ consists of the functions $f \in \overline{\mathbb{K}}(C)$ such that $\mathrm{div}(f) + \mathrm{div}(\omega) = \mathrm{div}(f\omega) \geq 0$, i.e. $f\omega$ is a holomorphic differential. Furthermore, let ω' be a holomorphic differential, then we can write it as $\omega' = g\omega$ with $g \in \overline{\mathbb{K}}(C)$ and $g \in \mathcal{L}(\mathrm{div}(\omega))$. This shows that $\mathcal{L}(\mathrm{div}(\omega))$ is isomorphic to the space of holomorphic differentials on C.

DEFINITION 2.1.6 *The genus of C is $g = l(\mathrm{div}(\omega)) = \dim(\mathcal{L}(\mathrm{div}(\omega))$ for any non-zero divisor $\omega \in \Omega_C$.*

The next theorem allows us to adequately describe the functions on a curve C having prescribed zeros and poles.

THEOREM 2.1.4 (RIEMANN-ROCH) *Let C be a smooth curve of genus g and ω any non-zero differential on C. Then for every divisor $D \in \mathrm{Div}(C)$ we have*

$$l(D) - l(\mathrm{div}(\omega) - D) = \deg D - g + 1\,.$$

An immediate consequence of the Riemann-Roch theorem is the existence of a unique representative for each divisor class in $\mathrm{Pic}^0_{\mathbb{K}}(C)$. A proof of the following theorem can be found in Galbraith, Paulus and Smart [46].

THEOREM 2.1.5 *Let C be a non-singular curve over $\mathbb{K}$ of genus g with a given $\mathbb{K}$-rational point P_0. Let $D \in \mathrm{Div}^0_{\mathbb{K}}(C)$, then there exists a unique effective divisor E over $\mathbb{K}$ of minimal degree $0 \leq m \leq g$ such that $E - mP_0$ is linearly equivalent to D.*

A divisor of the form $E - rP_0$ with $r \leq g$, but not necessarily minimal is called semi-reduced. The divisor $E - mP_0$, with m minimal is called reduced and m is the weight of the divisor.

2.1.6 Elliptic, Hyperelliptic and C_{ab}-curves

Elliptic Curves

DEFINITION 2.1.7 *An elliptic curve is a pair $(E, \mathcal{O})$ where E is a curve of genus 1 and $\mathcal{O} \in E$. The elliptic curve E is defined over $\mathbb{K}$, written $E/\mathbb{K}$, if E is defined over $\mathbb{K}$ as a curve and $\mathcal{O} \in E(\mathbb{K})$.*

Using the Riemann-Roch theorem we can easily embed every elliptic curve $E/\mathbb{K}$ in $\mathbb{P}^2$. The result is as follows.

PROPOSITION 2.1.1 *Let $E/\mathbb{K}$ be an elliptic curve defined over $\mathbb{K}$, then there exists functions $x, y \in \mathbb{K}(E)$ such that the map $\phi = [x, y, 1] : E \to \mathbb{P}^2$ gives an isomorphism of $E/\mathbb{K}$ onto a curve given by a Weierstrass equation*

$$C : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\,.$$

From now on, we will always assume that E is given by a Weierstrass equation. Theorem 2.1.5 implies that for every divisor $D \in \mathrm{Div}^0_{\mathbb{K}}(E)$ there exists a unique effective divisor D' of degree ≤ 1 such that D is linearly equivalent to $D' - \mathcal{O}$. Considering the map $E(\mathbb{K}) \to \mathrm{Pic}^0_{\mathbb{K}}(E) : P \mapsto [P - \mathcal{O}]$, with $[P - \mathcal{O}]$ the class of $P - \mathcal{O}$, we see that $E(\mathbb{K})$ and $\mathrm{Pic}^0_{\mathbb{K}}(E)$ are isomorphic and that $E(\mathbb{K})$ has a natural group law. Note that this group law coincides with the classical chord and tangent law since the sum of three collinear points is zero in $\mathrm{Pic}^0_{\mathbb{K}}(E)$.

To every elliptic curve E we can associate an invariant $j(E)$, the so-called j-invariant. If $E/\mathbb{K}$ is given by a Weierstrass equation, then $j(E) \in \mathbb{K}$ is a simple algebraic function of the coefficients $a_1, a_2, a_3, a_4, a_6 \in \mathbb{K}$. The j-invariant of a curve only depends on its isomorphism class and not on the particular equation chosen. Indeed, two elliptic curves are isomorphic over $\overline{\mathbb{K}}$ if and only if they have the same j-invariant.

Hyperelliptic Curves

Let $C_1/\mathbb{K}$ and $C_2/\mathbb{K}$ be two curves and ϕ a map from C_1 to C_2. If ϕ is constant we define the degree of ϕ to be zero; otherwise ϕ is called finite and its degree is defined by $\deg \phi = [\mathbb{K}(C_1) : \phi^* \mathbb{K}(C_2)]$ with $\phi^* f = f \circ \phi$ for $f \in \mathbb{K}(C_2)$.

Definition 2.1.8 *A curve C is called hyperelliptic if there exists a morphism of degree 2 from C to $\mathbb{P}^1$.*

Clearly, an ellipitic curve E is hyperelliptic: assume that E is given by a Weierstrass equation, then the map $x : E \to \mathbb{P}^1 : (\overline{x}, \overline{y}) \mapsto \overline{x}$ is a morphism of degree 2. The existence of a morphism φ of degree 2 immediately implies the existence of a hyperelliptic involution $\imath : C \to C$, such that $P \in C$ and $\imath(P) \in C$ have the same image under φ.

The function field of $\mathbb{P}^1(\mathbb{K})$ is simply the rational function field $\mathbb{K}(x)$. Since φ has degree 2, we conclude that the function field $\mathbb{K}(C)$ of a hyperelliptic curve C is a quadratic extension of $\mathbb{K}(x)$. Therefore, there exists a function $y \in \mathbb{K}(C)$ which satisfies

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

with $h(x), f(x) \in \mathbb{K}[x]$ and the hyperelliptic involution $\imath$ is given by

$$\imath : (x, y) \mapsto (x, -y - h(x))\,.$$

The following theorem provides normal forms for hyperelliptic curves; the case $\operatorname{char} \mathbb{K} \neq 2$ is fairly easy (see Stichtenoth [122, Proposition VI.2.3]), whereas the case $\operatorname{char} \mathbb{K} = 2$ needs a little more work (see Enge [33]).

Theorem 2.1.6 *Every hyperelliptic curve $C/\mathbb{K}$ of genus g admits an affine model of the form:*

- $\operatorname{char} \mathbb{K} \neq 2$: *$C/\mathbb{K}$ admits a model with equation*

$$y^2 = f(x)\,,$$

  with $f(x) \in \mathbb{K}[x]$ square-free. The degree of $f(x)$ is either $2g+1$ and the model is called imaginary quadratic or it is $2g+2$ and the model is called real quadratic.

- $\operatorname{char} \mathbb{K} = 2$: *$C/\mathbb{K}$ admits a model with equation*

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

with $h(x), f(x) \in \mathbb{K}[x]$ and every irreducible polynomial dividing $h(x)$ is a simple divisor of $f(x)$. The degrees of $h(x)$ and $f(x)$ are one of the following:

- $\deg f(x) = 2g+1$ *and* $\deg h(x) \leq g$, *then C is called imaginary quadratic.*
- $\deg h(x) = g+1$ *and* $\deg f(x) < 2g+2$ *or* $\deg f(x) = 2g+2$ *and the inverse of the leading coefficient of $h(x)$ equals z^2+z for some element $z \in \mathbb{K}$, then C is called real quadratic.*
- $\deg h(x) = g+1$ *and* $\deg f(x) = 2g+2$ *and the inverse of the leading coefficient of $h(x)$ cannot be written as z^2+z for any element $z \in \mathbb{K}$, then C is called inert.*

Let x_∞ be the prime divisor at infinity of $\mathbb{K}(x)$. The terminology in the above theorem is such that $C/\mathbb{K}$ is called imaginary, real or inert depending on whether x_∞ ramifies, splits or is inert in $\mathbb{K}(C)$.

In cryptography we normally only work with $\mathrm{Pic}^0_{\mathbb{K}}(C)$ for C an imaginary quadratic hyperelliptic curve, since the arithmetic in $\mathrm{Pic}^0_{\mathbb{K}}(C)$ is particularly simple.

THEOREM 2.1.7 *Let $C/\mathbb{K}$ be a smooth projective curve and $\phi : C \to \mathbb{P}^1$ a morphism of degree d. Assume that there exists a $\mathbb{K}$-rational point $P_\infty \in C$ which is totally ramified and maps to the point at infinity x_∞ of $\mathbb{P}^1$. Let S be the integral closure of $\mathbb{K}[x]$ in $\mathbb{K}(C)$ and let $\mathrm{Cl}(S)$ be the ideal class group of S, then $\mathrm{Cl}(S) \simeq \mathrm{Pic}^0_{\mathbb{K}}(C)$ and the isomorphism is given by*

$$\Phi : \mathrm{Pic}^0_{\mathbb{K}}(C) \to \mathrm{Cl}(S)\,,$$
$$[\sum_{P \neq P_\infty \in C} n_p P - nP_\infty] \mapsto [\mathcal{L}(\infty P_\infty - \sum_{P \neq P_\infty \in C} n_P P)]\,.$$

For $C/\mathbb{K}$ an imaginary quadratic hyperellipitc curve, the arithmetic in $\mathrm{Pic}^0_{\mathbb{K}}(C)$ reduces to ideal arithmetic in the integral closure S of $\mathbb{K}[x]$ in $\mathbb{K}(C)$. If $C(x,y)$ is an equation for $C/\mathbb{K}$ which is non-singular in the affine plane, then

$$S = \mathbb{K}[x,y]/(C(x,y))\,.$$

Let $D = \sum_{P \neq P_\infty} n_P P - nP_\infty$ be a semi-reduced divisor, then D can be uniquely represented by a pair of polynomials $u, v \in \mathbb{K}[x]$ with

$$u(x) = \prod_{P \neq P_\infty} (x - x(P))^{n_P}$$

and $v(x)$ the unique polynomial with

$$\deg v(x) < \deg u(x) \quad \text{and} \quad u(x) | v(x)^2 + v(x)h(x) - f(x)\,.$$

Note that for each point $P \neq P_\infty \in C$ with $n_P \neq 0$ we have $v(x(P)) = y(P)$. The pair $(u(x), v(x))$ is called Mumford's representation [96] and corresponds to the integral ideal $(u(x), y - v(x)) \in S$. Adding two reduced divisors D_1 and D_2 represented by the ideals

$$I_1 = (u_1(x), y - v_1(x)) \quad \text{and} \quad I_2 = (u_2(x), y - v_2(x))$$

proceeds in two steps: the first step is called composition and computes a semi-reduced ideal in the class of $I_1 I_2$. The second step is called reduction and computes the unique reduced ideal I_3 in the ideal class of $I_1 I_2$. Efficient algorithms for composition and reduction have been proposed by Cantor [14] (for odd characteristic only) and Koblitz [66]. Stein [120] and Enge [32] showed that a generic addition takes $17g^2 + O(g)$ operations in $\mathbb{F}_q$, whereas a doubling takes $16g^2 + O(g)$ operations. More recently, several authors [68, 98] have presented explicit formulae for performing the group law in the Jacobian of small genus hyperelliptic curves.

C_{ab} Curves

The class of C_{ab} curves was first introduced by Miura [87] in the setting of algebraic geometric codes and generalizes hyperelliptic curves.

Let $C/\mathbb{K}$ be an algebraic curve over $\mathbb{K}$ and assume that C has at least one non-singular $\mathbb{K}$-rational point P. Consider the ring $\mathcal{L}(\infty P)$ of functions on C which are holomorphic away from P

$$\mathcal{L}(\infty P) = \{f \in \mathbb{K}(C) \,|\, \forall Q \neq P : \mathrm{ord}_Q(f) \geq 0\} .$$

The pole orders $-\mathrm{ord}_P(f)$ of all functions in $\mathcal{L}(\infty P)$ become a monoid M

$$M = \{-\mathrm{ord}_P(f) \,|\, f \in \mathcal{L}(\infty P)\} .$$

DEFINITION 2.1.9 *A curve C with a non-singular $\mathbb{K}$-rational point P is called a C_{ab} curve if M is generated by two positive integers a and b.*

Note that a and b are automatically coprime. Given a C_{ab} curve C, there exists a function $x \in \mathcal{L}(\infty P)$ with pole order a and a function $y \in \mathcal{L}(\infty P)$ with pole order b. Using these two functions we obtain an affine model of the form

$$C : y^a + \sum_{i=1}^{a-1} f_i(x) y^i + f_0(x) = 0 ,$$

with $f_i(x) \in \mathbb{K}[x]$ for $i = 0, \ldots, a-1$, $\deg f_0 = b$ and $a \deg f_i + b\,i < ab$ for $i = 1, \ldots, a-1$. For a C_{ab} curve, the above model will be non-singular as an affine curve and the place at infinity $x_\infty \in \mathbb{K}(x)$ ramifies in $\mathbb{K}(C)$, giving rise to one place P_∞ at infinity. The genus of a C_{ab} curve is equal to $(a-1)(b-1)/2$ and the integral closure of $\mathbb{K}[x]$ in $\mathbb{K}(C)$ is given by the coordinate ring $\mathbb{K}[x,y]/(C(x,y))$ with $C(x,y)$ the defining equation.

Note that a C_{ab} curve satisfies Theorem 2.1.7, which maps arithmetic in the Jacobian to the ideal class group of the coordinate ring of the curve. Arita [7] and later, Harasawa and Suzuki [54], gave efficient algorithms to compute in the Jacobian of a C_{ab} curve. Divisors are represented as ideals and the group law is computed using ideal reduction techniques analogous to the number field case [16, Section 4.7.1].

2.2 Abelian Varieties

2.2.1 Definition and Jacobian Variety

A group variety $G/\mathbb{K}$ is an algebraic variety endowed with a group structure compatible with the structure of the variety. This means that the group law map $m : G \times G \to G$ and the inverse map $\mathrm{inv} : G \to G$ are regular and that there exists an identity element $e \in G(\mathbb{K})$.

DEFINITION 2.2.1 *An abelian variety is a non-singular projective commutative group variety.*

Informally, an abelian variety is a non-singular projective variety such that the group law can be expressed locally by fractions of multivariate polynomials.

Let C be a genus g curve defined over $\mathbb{K}$ with a $\mathbb{K}$-rational point P_0, then there exists an abelian variety J_C of dimension g, called the Jacobian variety of C, such that for every field $\mathbb{K} \subset \mathbb{L} \subset \overline{\mathbb{K}}$ the set $J_C(\mathbb{L})$ is in bijection with $\mathrm{Pic}^0(C)^{\mathrm{Gal}(\mathbb{L}/\mathbb{K})}$. Furthermore, there exists a morphism $C \to J_C$ which sends P_0 to the identity element of J_C. This morphism canonically injects C into J_C by mapping a point P to the class of $P - P_0$. Note that the very natural addition law on $\mathrm{Pic}^0(C)$ corresponds to the group law morphism on $J_C(\overline{\mathbb{K}})$.

2.2.2 Isogenies

Let A and B be two abelian varieties and $\mathcal{I} : A \to B$ a homomorphism. The kernel of $\mathcal{I}$ is defined as $\mathrm{Ker}\mathcal{I} = \{a \in A | \mathcal{I}(a) = e_B \in B\}$, with e_B the neutral element of B.

PROPOSITION 2.2.1 *A homomorphism $\mathcal{I} : A \to B$ of abelian varieties is called an isogeny if it is surjective and has finite kernel. Furthermore, the following are equivalent:*

1. *$\mathcal{I}$ is an isogeny;*
2. *$\dim A = \dim B$ and $\mathcal{I}$ is surjective;*
3. *$\dim A = \dim B$ and $\mathrm{Ker}(\mathcal{I})$ is finite.*

The set of isogenies from A to B forms a group under the natural addition law $(\mathcal{I}_1 + \mathcal{I}_2)(a) = \mathcal{I}_1(a) + \mathcal{I}_2(a)$ and is denoted by $\mathrm{Hom}(A, B)$. Furthermore, the set of endomorphisms $\mathrm{End}(A)$ forms a ring with multiplication given by composition. The subgroup of isogenies which are defined over $\mathbb{K}$, i.e. commute with the action of $\mathrm{Gal}(\overline{\mathbb{K}}/\mathbb{K})$, is denoted by $\mathrm{Hom}_{\mathbb{K}}(A, B)$.

The degree of an isogeny $\mathcal{I}$ is defined as the degree of the field extension $[\mathbb{K}(A) : \mathcal{I}^*\mathbb{K}(B)]$, where $\mathcal{I}^*(f)$ is defined as $f \circ \mathcal{I}$ for $f \in \mathbb{K}(B)$. The isogeny $\mathcal{I}$ is called separable (purely inseparable) if the extension $\mathbb{K}(A) : \mathcal{I}^*\mathbb{K}(B)$ has the corresponding property. An isogeny $\mathcal{I}$ factors as the composition of a separable and purely inseparable part and $\#\mathrm{Ker}(\mathcal{I}) = \deg_s \mathcal{I}$, with $\deg_s \mathcal{I}$ the degree of the separable part.

The simplest example of an isogeny is multiplication by an integer n, denoted n_A and defined by

$$n_A : A \to A : a \mapsto na = a + \cdots + a \quad (n \text{ times}).$$

Two abelian varieties are called isogenous if there exists a non-constant isogeny between them. The next theorem shows that being isogenous is in fact a symmetric relation.

Theorem 2.2.1 *Let $\mathcal{I}$ be an isogeny of degree d between two abelian varieties A and B. Then there exists an isogeny $\widehat{\mathcal{I}} : B \to A$, called the dual isogeny, such that*

$$\widehat{\mathcal{I}} \circ \mathcal{I} = d_A \quad \textit{and} \quad \mathcal{I} \circ \widehat{\mathcal{I}} = d_B .$$

2.2.3 Modular Polynomials and Vélu's Formulae

Let E_1 and E_2 be elliptic curves over a field $\mathbb{K}$, which is either a finite field or such that it can be embedded in $\mathbb{C}$, then we can easily test if there exists a cyclic isogeny of degree n between E_1 and E_2.

Theorem 2.2.2 *Let $n \in \mathbb{N}$ and define $d = n \prod_{p|n}(1 + 1/p)$ then there exists a symmetric polynomial $\Phi_n(X, Y) \in \mathbb{Z}[X, Y]$ of degree d such that E_1 and E_2 are related via a cyclic isogeny of degree n if and only if $\Phi_n(j(E_1), j(E_2)) = 0$.*

The polynomial $\Phi_n(X, Y)$ is called the n-th modular polynomial and can be computed using the Fourier expansion of the modular function $j(\tau)$. If n is a prime l, then the degree of Φ_l equals $d = l + 1$. Furthermore, $\Phi_l(X, Y)$ satisfies the so-called Kronecker relation

$$\Phi_l(X, Y) \equiv (X^l - Y)(X - Y^l) \mod l .$$

Let E_1, E_2, E_3 be elliptic curves over a perfect field $\mathbb{K}$ and let $\phi : E_1 \to E_2$ and $\psi : E_1 \to E_3$ be non-constant isogenies. If $\mathrm{Ker}\,\phi \subset \mathrm{Ker}\,\psi$ and if ϕ is separable then there always exists a unique isogeny λ such that the following diagram commutes.

$$\begin{tikzcd}
E_1 \arrow[rr, "\psi"] \arrow[dr, "\phi"'] & & E_3 \\
& E_2 \arrow[ur, dotted, "\lambda"'] &
\end{tikzcd} \tag{2.1}$$

Let Φ be a finite subgroup of E_1, then we can consider the quotient group E_1/Φ. Clearly, E_1/Φ is a group itself, but amazingly enough E_1/Φ also corresponds to the group of points on some unique elliptic curve $E_2 \cong E_1/\Phi$. Furthermore, there exists a separable isogeny $\phi : E_1 \to E_1/\Phi$ with $\operatorname{Ker}\phi = \Phi$. Assume that E_1 is defined by the Weierstrass equation

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\,,$$

then Vélu [129] showed how to construct a Weierstrass equation for E_1/Φ and how to find the isogeny ϕ. Write

$$\Phi = S\cup(-S)\cup R\cup\{\mathcal{O}\} \text{ with } R = (E[2]\cap\Phi)\setminus\{\mathcal{O}\} \text{ and } S\cup(-S) = \Phi\setminus(R\cup\{\mathcal{O}\})$$

then E_1/Φ is given by the Weierstrass equation

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + (a_4 - 5t)X + (a_6 - (a_1^2 + 4a_2)t - 7w)\,,$$

with

$$t = \sum_{Q\in S\cup R} t_Q\,, \quad w = \sum_{Q\in S\cup R} (u_Q + x_Qt_Q)\,,$$

$$u_Q = \beta_Q^2\,, \quad t_Q = \begin{cases} \alpha_Q & \text{if} \quad Q \in R \\ 2\alpha_Q - a_1\beta_Q & \text{if} \quad Q \notin R \end{cases}$$

$$Q = (x_Q, y_Q)\,, \quad \alpha_Q = 3x_Q^2 + 2a_2x_Q + a_4 - a_1y_Q\,, \quad \beta_Q = -2y_Q - a_1x_Q - a_3\,.$$

Furthermore, the isogeny ϕ constructed by Vélu satisfies $\phi^*(\omega_\Phi) = \omega_1$ with ω_1 and ω_Φ the invariant differentials on E_1 and E_1/Φ respectively. This property will be used frequently in Chapter 3.

2.2.4 Tate Modules

Let A be an abelian variety over $\mathbb{K}$ of dimension g and consider the multiplication by n map n_A. The kernel of n_A (over $\overline{\mathbb{K}}$) is called the n-torsion and is denoted by $A[n]$. The following theorem determines the structure of $A[n]$.

THEOREM 2.2.3 *Let A be an abelian variety of dimension g defined over $\mathbb{K}$ of characteristic $p \geq 0$, then n_A is an isogeny of degree n^{2g}.*

- *Let $l \neq p$ be any prime, then l_A^k is separable and the l^k-torsion $A[l^k]$ is isomorphic to $(\mathbb{Z}/(l^k\mathbb{Z}))^{2g}$.*
- *The map p_A^k is inseparable and $A[p^k]$ is isomorphic to $(\mathbb{Z}/(p^k\mathbb{Z}))^r$ for some integer r independent of k and $0 \leq r \leq g$. The integer r is called the p-rank of the abelian variety A.*

The abelian variety is called ordinary if its p-rank is equal to g and very special if its p-rank is 0.

DEFINITION 2.2.2 *Fix a prime $l \neq p$, then the Tate module $T_l(A)$ is defined as the projective limit of the l-power torsion (over $\overline{\mathbb{K}}$), i.e.*

$$T_l(A) = \varprojlim A[l^k] .$$

Informally, an element of $T_l(A)$ is an infinite sequence $(a_1, a_2, \dots, a_k, \dots)$ with $a_k \in A[l^k]$ and $la_k = a_{k-1}$, $la_1 = e_A$. Since $A[l^k]$ is isomorphic to $(\mathbb{Z}/(l^k\mathbb{Z}))^{2g}$, $T_l(A)$ is isomorphic to $\mathbb{Z}_l^{2g}$, with $\mathbb{Z}_l$ the l-adic integers (see Section 2.3). Note that the action of $\mathrm{Gal}(\overline{\mathbb{K}}/\mathbb{K})$ commutes with multiplication by l, so $\mathrm{Gal}(\overline{\mathbb{K}}/\mathbb{K})$ acts on $T_l(A)$.

Let A and B be abelian varieties over $\mathbb{K}$ and let $\alpha \in \mathrm{Hom}(A, B)$. Since α is a homomorphism, it commutes with multiplication by l and thus induces a $\mathbb{Z}_l$-linear map

$$\alpha_l : T_l(A) \to T_l(B) .$$

Therefore we obtain a homomorphism $\mathrm{Hom}(A, B) \to \mathrm{Hom}(T_l(A), T_l(B))$. Let $\mathrm{Hom}_{\mathbb{K}}(T_l(A), T_l(B))$ be the group of $\mathbb{Z}_l$-linear maps from $T_l(A)$ to $T_l(B)$ which commute with the action of $\mathrm{Gal}(\overline{\mathbb{K}}/\mathbb{K})$. If $\mathbb{K}$ is a finite field, Tate [124] proved the following result.

THEOREM 2.2.4 (TATE) *Let $\mathbb{K}$ be a finite field, then the natural map*

$$\mathrm{Hom}_{\mathbb{K}}(A, B) \otimes \mathbb{Z}_l \to \mathrm{Hom}_{\mathbb{K}}(T_l(A), T_l(B))$$

is an isomorphism.

Any endomorphism $\alpha \in \mathrm{End}(A)$ acts $\mathbb{Z}_l$-linear on $T_l(A)$ and once a basis for the $\mathbb{Z}_l$-module $T_l(A)$ is chosen, α_l can be represented by a $2g \times 2g$ matrix M_l. The characteristic polynomial $\chi_\alpha(x)$ of α is defined to be the characteristic polynomial of M_l. One can show that $\chi_\alpha \in \mathbb{Z}[x]$ and the above theorem implies that χ_α is independent of l. Alternatively, $\chi_\alpha(x)$ can be defined as the unique polynomial in $\mathbb{Z}[x]$ such that $\deg(\alpha - n) = |\chi_\alpha(n)|$ for all $n \in \mathbb{N}$.

2.2.5 The Weil Conjectures

Let $\mathbb{F}_q$ be a finite field with $q = p^n$ and p prime. For any algebraic set X defined over $\mathbb{F}_q$, let N_k denote the number of $\mathbb{F}_{q^k}$-rational points on X.

DEFINITION 2.2.3 *The zeta function $Z(X/\mathbb{F}_q; t)$ of X over $\mathbb{F}_q$ is the generating function*

$$Z(X/\mathbb{F}_q; t) = \exp\left(\sum_{k=1}^{\infty} \frac{N_k}{k} t^k\right) .$$

A fortiori, the zeta function should be interpreted as a formal power series with coefficients in $\mathbb{Q}$. In 1949, Weil [137] stated the following conjectures, all of which have now been proven.

THEOREM 2.2.5 (WEIL CONJECTURES) *Let X be a smooth projective variety of dimension d defined over a finite field with q elements.*

1. *Rationality: $Z(X/\mathbb{F}_q;t) \in \mathbb{Q}[[t]]$ is a rational function.*
2. *Functional equation: $Z(t) = Z(X/\mathbb{F}_q;t)$ satisfies*
$$Z(\frac{1}{q^d t}) = \pm q^{d\chi/2} t^{\chi} Z(t)\,,$$
with χ equal to the Euler-Poincaré characteristic of X, i.e. the intersection number of the diagonal with itself in the product $X \times X$.
3. *Riemann hypothesis: there exists polynomials $P_i(t) \in \mathbb{Z}[t]$ for $i = 0, \dots, 2d$, such that*
$$Z(X/\mathbb{F}_q;t) = \frac{P_1(t)P_3(t)\cdots P_{2d-1}}{P_0(t)P_2(t)\cdots P_{2d}(t)}$$
with $P_0(t) = 1-t$, $P_{2d}(t) = 1 - q^d t$ and for $1 \le r \le 2d-1$
$$P_r(t) = \prod_{i=1}^{\beta_r}(1 - \alpha_{r,i}t)$$
where the $\alpha_{r,i}$ are algebraic integers of absolute value $q^{r/2}$ and the β_r are the Betti numbers of the variety X.

Weil [136] proved these conjectures for curves and abelian varieties. The rationality of the zeta function in general was settled in 1960 by Dwork [27] using p-adic methods. Soon after, the Grothendieck school developed l-adic cohomology and gave another proof of the rationality and the functional equation. Finally, in 1973, Deligne [22] proved the Riemann hypothesis.

Applying these conjectures to a smooth projective curve $C/\mathbb{F}_q$ of genus g, gives $d = 1$, $\chi = 2 - 2g$ and $\beta_1 = 2g$, so
$$Z(C/\mathbb{F}_q;t) = \frac{L(t)}{(1-t)(1-qt)}\,,$$
with $L(t) = c_0 + c_1 t + \cdots + c_{2g}t^{2g} \in \mathbb{Z}[t]$ and $c_0 = 1$. The functional equation shows that $c_{2g-i} = q^{g-i}c_i$ for $i = 0, \dots, g$. If we write $L(t) = \prod_{i=1}^{2g}(1 - \alpha_i t)$, then the Riemann hypothesis implies $|\alpha_i| = \sqrt{q}$. Furthermore, we can label the α_i such that $\alpha_i \alpha_{i+g} = q$ for $i = 0, \dots, g$. Taking the logarithm of both expressions for the zeta function leads to
$$\log Z(C/\mathbb{F}_q;t) = \sum_{k=1}^{\infty} \frac{N_k}{k} t^k = \sum_{i=1}^{2g} \log(1 - \alpha_i t) - \log(1-t) - \log(1-qt)\,.$$
Since $\log(1 - st) = -\sum_{i=1}^{\infty} \frac{(st)^k}{k}$, we conclude that for all $k \in \mathbb{N}_0$
$$N_k = q^k + 1 - \sum_{i=1}^{2g} \alpha_i^k\,.$$

Let $\mathbb{F}_q$ be a finite field with $q = p^n$ elements and fix an algebraic closure $\overline{\mathbb{F}}_q$. The Galois group $\mathrm{Gal}(\overline{\mathbb{F}}_q/\mathbb{F}_q)$ is generated by the arithmetic Frobenius automorphism $\overline{F} : \overline{\mathbb{F}}_q \to \overline{\mathbb{F}}_q : x \mapsto x^q$. Note that $\mathbb{F}_q$ is exactly the fixed field of $\overline{F}$. Let $C/\mathbb{F}_q$ be a smooth projective curve of genus g with Jacobian variety J_C, then $\overline{F}$ induces a geometric Frobenius map on C and J_C, also denoted with $\overline{F}$, by acting on the coordinates of points. The geometric Frobenius on J_C is in fact a purely inseparable isogeny of degree q^g. Since $\overline{F}$ generates $\mathrm{Gal}(\overline{\mathbb{F}}_q/\mathbb{F}_q)$, a point on J_C will be $\mathbb{F}_q$-rational if and only if $\overline{F}(P) = P$, so

$$\mathrm{Ker}(\overline{F} - \mathrm{Id}) = J_C(\mathbb{F}_q)\,.$$

Let $\chi_{\overline{F}} \in \mathbb{Z}[x]$ be the characteristic polynomial of $\overline{F} \in \mathrm{End}(J_C)$. The map $\overline{F} - \mathrm{Id}$ is separable, which implies that $|J_C(\mathbb{F}_q)| = \deg(\overline{F} - \mathrm{Id})$ and by the second definition of the characteristic polynomial we conclude

$$|J_C(\mathbb{F}_q)| = \chi_{\overline{F}}(1)\,.$$

Let $Z(C/\mathbb{F}_q;t) = L(t)/(1-t)(1-qt)$ be the zeta function of C, then one can prove [57, Appendix C] that

$$L(t) = t^{2g}\chi_{\overline{F}}(1/t)\,.$$

The underlying reason is that for an abelian variety A the first étale cohomology group $H^1(A_{et}, \mathbb{Z}_l)$ is the dual of $T_l(A)$.

The zeta function $Z(C/\mathbb{F}_q;t)$ of a curve C contains important geometric information about C and its Jacobian J_C. For example, Stichtenoth [121] proved that the p-rank of J_C is equal to

$$\max\{i \,|\, c_i \not\equiv 0 \bmod p\}$$

with $L(t) = c_0 + \cdots + c_{2g}t^{2g}$. Furthermore, Stichtenoth and Xing [123] showed that J_C is supersingular, i.e. isogenous over $\overline{\mathbb{F}}_q$ to a product of supersingular elliptic curves, if and only if $p^{\lceil nk/2 \rceil} \,|\, c_k$ for all $1 \leq k \leq g$ with $q = p^n$.

2.3 p-adic Rings and Fields

In this section we give a very brief introduction to p-adic fields. More details can be found in the books by Koblitz [64] and Serre [114].

2.3.1 The Field $\mathbb{Q}_p$ of p-adic Numbers

Fix a prime p and let $a \neq 0 \in \mathbb{Z}$, then we define

$$\mathrm{ord}_p(a) = \max\{i \in \mathbb{N} \,|\, a \equiv 0 \bmod p^i\}\,.$$

By definition we set $\mathrm{ord}_p(0) = \infty$, so $\mathrm{ord}_p : \mathbb{Z} \to \mathbb{N} \cup \infty$. The function ord_p can be easily extended to $\mathbb{Q}$ by defining

$$\mathrm{ord}_p(\frac{a}{b}) = \mathrm{ord}_p(a) - \mathrm{ord}_p(b) \quad \text{for } a, b \in \mathbb{Z}, b \neq 0\,.$$

An easy calculation shows that ord_p satisfies $\mathrm{ord}_p(rs) = \mathrm{ord}_p(r) + \mathrm{ord}_p(s)$ and $\mathrm{ord}_p(r+s) \geq \min\{\mathrm{ord}_p(r), \mathrm{ord}_p(s)\}$ for $r, s \in \mathbb{Q}$. These properties imply that the function $|\cdot|_p$

$$|x|_p = p^{-\mathrm{ord}_p(x)} \text{ for } x \in \mathbb{Q},$$

defines a non-Archimedean or ultrametric norm on $\mathbb{Q}$.

DEFINITION 2.3.1 *The field $\mathbb{Q}_p$ of p-adic numbers is the completion of $\mathbb{Q}$ with respect to the p-adic norm $|\cdot|_p$. The ring $\mathbb{Z}_p$ of p-adic integers is defined as*

$$\mathbb{Z}_p = \{x \in \mathbb{Q}_p \,|\, |x|_p \leq 1\} = \{x \in \mathbb{Q}_p \,|\, \mathrm{ord}_p(x) \geq 0\}\,.$$

This makes $\mathbb{Z}_p$ into a local ring with discrete valuation ord_p and unique maximal ideal $M = \{x \in \mathbb{Z}_p \,|\, \mathrm{ord}_p(x) > 0\}$. The residue field $\mathbb{Z}_p/M$ is isomorphic to $\mathbb{F}_p$ and we denote by $\overline{a} \in \mathbb{F}_p$ the reduction of an element $a \in \mathbb{Z}_p$. Note that the units in $\mathbb{Z}_p$ are precisely the elements with valuation zero, i.e. $\mathbb{Z}_p^\times = \mathbb{Z}_p \setminus M$ and thus $\overline{u} \neq 0$ for $u \in \mathbb{Z}_p^\times$. In practice, an element of $\mathbb{Q}_p$ can be approximated by a truncated Laurent series as shown by the next theorem.

THEOREM 2.3.1 *Every element $a \in \mathbb{Q}_p$ can be represented uniquely as a Laurent series $a = \sum_{i=m}^{\infty} a_i p^i$ with $0 \leq a_i < p$, $a_m \neq 0$ and $m = \mathrm{ord}_p(a)$.*

2.3.2 Extensions of $\mathbb{Q}_p$

Let $\mathbb{K}$ be a finite extension of $\mathbb{Q}_p$ of degree n. A fundamental theorem of p-adic analysis states that the norm $|\cdot|_p$ can be extended uniquely from $\mathbb{Q}_p$ to $\mathbb{K}$. This norm also defines a valuation on $\mathbb{K}$ given by $\mathrm{ord}_p(\alpha) = -\log_p(|\alpha|_p)$ for $\alpha \in \mathbb{K}$. The integral closure of $\mathbb{Z}_p$ is given by $R = \{x \in K \,|\, |x|_p \leq 1\}$ and $M = \{x \in K \,|\, |x|_p < 1\}$ is the unique maximal ideal of R. The ring R is called the valuation ring of $\mathbb{K}$ and the residue field of $\mathbb{K}$ is defined as R/M.

DEFINITION 2.3.2 *Let $\mathbb{K}$ be a finite extension of $\mathbb{Q}_p$ of degree n, then $\mathbb{K}$ is called unramified if its residue field R/M is isomorphic to $\mathbb{F}_{p^n}$.*

A degree n unramified extension of $\mathbb{Q}_p$ is unique up to isomorphism and is denoted by $\mathbb{Q}_q$ with $q = p^n$. The valuation ring of $\mathbb{Q}_q$ is denoted by $\mathbb{Z}_q$.

Let $\mathbb{F}_{p^n} \simeq \mathbb{F}_p[x]/(\overline{f}(x))$ with $\overline{f}(x) \in \mathbb{F}_p[x]$ a monic irreducible polynomial of degree n. The field $\mathbb{Q}_q$ can be represented as $\mathbb{Q}_p[x]/(f(x))$, with $f(x) \in \mathbb{Z}_p[x]$ an arbitrary monic lift of $\overline{f}(x)$. Thus, every element $\alpha \in \mathbb{Q}_q$ can be represented uniquely as $\sum_{i=0}^{n-1} \alpha_i x^i$ with $\alpha_i \in \mathbb{Q}_p$. The sum and product in $\mathbb{Q}_q$ are simply the sum and product of polynomials modulo $f(x)$. The valuation on $\mathbb{Q}_p$ extends naturally to $\mathbb{Q}_q$ by defining $\mathrm{ord}_p(\alpha) = \min\{\mathrm{ord}_p(a_i)\}$. Note that this implies that $\mathbb{Z}_q \simeq \mathbb{Z}_p[x]/(f(x))$ and $\mathbb{Z}_q^\times = \mathbb{Z}_q \setminus p\mathbb{Z}_q$.

The Galois group $\mathrm{Gal}(\mathbb{Q}_q/\mathbb{Q}_p)$ is isomorphic to the Galois group $\mathrm{Gal}(\mathbb{F}_q/\mathbb{F}_p)$. Indeed, every automorphism $\Lambda \in \mathrm{Gal}(\mathbb{Q}_q/\mathbb{Q}_p)$ satisfies $\Lambda(p\mathbb{Z}_q) = p\mathbb{Z}_q$ and thus defines an automorphism of $\mathbb{F}_q$ by reduction modulo $p\mathbb{Z}_q$. Since $\overline{f}(x)$ is square-free, we conclude that this mapping is an isomorphism. The Galois group

$\mathrm{Gal}(\mathbb{F}_q/\mathbb{F}_p)$ is cyclic and is generated by the p-th power Frobenius endomorphism $\sigma : \mathbb{F}_q \to \mathbb{F}_q : x \mapsto x^p$.

DEFINITION 2.3.3 *The unique automorphism $\Sigma \in \mathrm{Gal}(\mathbb{Q}_q/\mathbb{Q}_p)$ which reduces to the p-th power Frobenius endomorphism $\sigma \in \mathrm{Gal}(\mathbb{F}_q/\mathbb{F}_p)$ is called the Frobenius substitution on $\mathbb{Q}_q$.*

2.3.3 Hensel's Lemma

Let $\mathbb{K}$ be a complete field with non-Archimedian norm $|\cdot|$ and valuation ring R. Given an initial approximation $r_0 \in R$ to a root of a polynomial $f \in R[x]$, then Hensel's lemma shows that there exists a unique root $r \in R$ of f close to r_0.

LEMMA 2.3.1 (HENSEL'S LEMMA) *Let $\mathbb{K}$ be a complete ultrametric field with norm $|\cdot|$ and valuation ring R. Let $f \in R[x]$ and assume that r_0 satisfies*

$$|f(r_0)| < |f'(r_0)|^2 .$$

Then there exists a unique element $r \in R$ such that

$$f(r) = 0 \quad \text{and} \quad |r - r_0| \leq \left|\frac{f(r_0)}{f'(r_0)}\right| < |f'(r_0)| .$$

The proof of Hensel's lemma is constructive in the sense that it provides an algorithm to compute r starting from r_0. Define the sequence $\{r_k\}_{k\in\mathbb{N}}$ by

$$r_{k+1} := r_k - \frac{f(r_k)}{f'(r_k)} ,$$

then one easily proves by induction that

$$|f(r_k)| \leq c^{2^k-1}|f(r_0)| \quad \text{with} \quad c = \left|\frac{f(r_0)}{f'(r_0)^2}\right| < 1 .$$

This also shows that the convergence of the sequence $\{r_k\}_{k\in\mathbb{N}}$ is quadratic.

2.4 Complexity of Algorithms

2.4.1 Notation

For every algorithm described in this thesis, we will estimate its cost, i.e. the number of bit-operations as a function of the size of its input. However, most of the time we are not interested in the exact cost, but merely in the growth rate of the cost.

Let $g : \mathbb{N} \to \mathbb{R}$ be a function which is strictly positive for large enough $n \in \mathbb{N}$. Then $O(g)$ is the set of all functions $f : \mathbb{N} \to \mathbb{R}$ for which there exists constants $c, N \in \mathbb{N}$ such that $f(n) \leq cg(n)$ for all $n \geq N$, i.e. eventually $cg(n)$ acts as an upper bound on $f(n)$.

If were are only interested in lower bounds, the following definition is useful. Let $g : \mathbb{N} \to \mathbb{R}$ be a function which is strictly positive for large enough $n \in \mathbb{N}$. Then $\Omega(g)$ is the set of all functions $f : \mathbb{N} \to \mathbb{R}$ for which there exists constants $c, N \in \mathbb{N}$ such that $f(n) \geq cg(n)$ for all $n \geq N$.

2.4.2 Complexity of Multiplication

The overall cost of most algorithms in this thesis is ultimately determined by the cost of multiplying two large integers. There exist many different multiplication routines and each has its own complexity. To make abstraction of the specific algorithm used, we introduce a constant μ such that multiplication of two N-bit integers takes $O(N^\mu)$ time. The most commonly used multiplication algorithms and their values of μ are as follows.

- Classical multiplication: $\mu = 2$.
- Karatsuba multiplication [60]: $\mu = \log_2 3$.
- Degree-3 Toom multiplication [126]: $\mu = \log_3 5$.
- Schönhage and Strassen multiplication [110]: $\mu = 1 + \varepsilon$, with $\epsilon \in \mathbb{R}_{>0}$.

The complexity of Schönhage and Strassen multiplication is more precisely given by $O(N \log N \log\log N)$ which is an element of $O(N^{1+\varepsilon})$ for any $\varepsilon \in \mathbb{R}_{>0}$, since $\log N \log\log N$ grows slower than any strictly positive power of N.

Fast multiplication of large integers can also be used to obtain a fast multiplication algorithm for polynomials over the integers. Indeed, let

$$f := \sum_{i=0}^{d-1} f_i x^i \quad \text{and} \quad g := \sum_{i=0}^{d-1} g_i x^i \qquad f_i, g_i \in \mathbb{N}$$

be polynomials of degree $< d$ and assume that $0 \leq f_i, g_i \leq 2^B$ for $i = 0, \ldots, d-1$. An easy calculation shows that the largest coefficient of the product $f \cdot g$ is bounded by $2^{2B}d$. Define $b = \lceil 2B + \log_2 d \rceil$ and consider the integers

$$F := f(2^b) = \sum_{i=0}^{d-1} 2^{bi} f_i \quad \text{and} \quad G := g(2^b) = \sum_{i=0}^{d-1} 2^{bi} g_i\,,$$

then it is clear that the coefficients of $f \cdot g$ can be recovered from the integer product $F \cdot G$. Therefore, the product of the polynomials f and g can be computed in time $O(bd \log bd \log\log bd)$.

Chapter 3

Elliptic Curves

Let $\overline{E}$ be an elliptic curve over $\mathbb{F}_q$, with $q = p^n$. The number of $\mathbb{F}_q$-rational points $\#\overline{E}(\mathbb{F}_q)$ satisfies the relation $\#\overline{E}(\mathbb{F}_q) = q + 1 - t$, where t is the trace of the Frobenius endomorphism $\overline{F} : \overline{E} \to \overline{E} : (x, y) \mapsto (x^q, y^q)$. By Hasse's theorem [117, Theorem V.1.1] we have $|t| \leq 2\sqrt{q}$, so it suffices to compute $t \bmod B$ with $B > 4\sqrt{q}$.

Schoof [111] described the first polynomial time algorithm to compute $\#\overline{E}(\mathbb{F}_q)$ using an l-adic approach. The key idea of Schoof's algorithm is to compute t modulo primes $l \leq l_m$ where l_m is the smallest prime with $\prod_{l \leq l_m} l > 4\sqrt{q}$. The time complexity of Schoof's algorithm is $O((\log q)^{3\mu+2})$ and the space complexity amounts to $O((\log q)^3)$. Several improvements mainly due to Elkies and Atkin resulted in a heuristically estimated time complexity of $O((\log q)^{2\mu+2})$ and space complexity of $O((\log q)^2)$. Detailed descriptions of this so called SEA algorithm can be found in articles by Schoof [112] and Elkies [30] and the doctoral theses of Müller [93] and Lercier [74].

At the end of 1999, Satoh [105] proposed a completely different strategy based on p-adic methods. The main idea is to lift the curve and the Frobenius endomorphism to a p-adic ring and to recover $t \bmod p^N$ with $p^N > 4\sqrt{q}$ directly from the lifted data. Satoh's original algorithm only worked for $p \geq 5$, but was shortly after extended to the cases $p = 2$ and $p = 3$ by Fouquet, Gaudry and Harley [41] and independently by Skjernaa [118] for $p = 2$. For fixed p, the time complexity of these algorithms is $O(n^{2\mu+1})$ and the space complexity is $O(n^3)$. Since the O-constant of the time complexity grows as $O(p^2)$, these algorithms are only practical for small p.

This chapter is organized as follows: after reviewing Satoh's algorithm and the extensions to $p = 2, 3$ in Sections 3.1-3.3, we present a memory efficient version which still runs in $O(n^{2\mu+1})$ time, but only requires $O(n^2)$ memory. The results of Section 3.4 have been published in [133]. More recent developments in p-adic point counting for elliptic curves, culminating in an $O(n^{2\mu} \log n)$ time and $O(n^2)$ space algorithm by Harley, are analysed in Sections 3.5 to 3.10. An abbreviated version of this chapter will appear in the books [9] and [17].

3.1 Satoh's Algorithm

The basic idea of Satoh's algorithm is to lift both the curve $\overline{E}$ and the Frobenius endomorphism $\overline{F}$ to the valuation ring $\mathbb{Z}_q$ of a degree n unramified extension $\mathbb{Q}_q$ of the p-adic field $\mathbb{Q}_p$. Since this lifting is done in a canonical way, the trace of the lifted Frobenius $\mathcal{F}$ equals the trace of Frobenius t. However, it turns out that the Frobenius endomorphism $\overline{F}$ itself is difficult to lift since it is inseparable and its kernel is trivial. Therefore one actually works with the dual of $\overline{F}$, called the Verschiebung $\overline{V}$. This Verschiebung is separable if and only if $\overline{E}$ is ordinary and can be lifted explicitly by lifting its kernel. Analysing the action of the lift $\mathcal{V}$ of $\overline{V}$ on the invariant differential of the canonical lift E, we obtain an expression for the trace of $\mathcal{V}$ which equals the trace of Frobenius t.

3.1.1 The Canonical Lift of an Ordinary Elliptic Curve

The main step in Satoh's algorithm is lifting the curve $\overline{E}$ and the Verschiebung $\overline{V}$ to the valuation ring $\mathbb{Z}_q$ of a degree n unramified extension $\mathbb{Q}_q$ of $\mathbb{Q}_p$. Among the many possible lifts of $\overline{E}$ from $\mathbb{F}_q$ to $\mathbb{Z}_q$ there is one which has particularly nice properties, called the canonical lift.

Definition 3.1.1 *The canonical lift E of an ordinary elliptic curve $\overline{E}$ over $\mathbb{F}_q$ is an elliptic curve over $\mathbb{Q}_q$ which satisfies:*

- *the reduction modulo p of E equals $\overline{E}$,*
- $\mathrm{End}(\overline{E}) \cong \mathrm{End}(E)$ *as a ring.*

Deuring [25] has shown that the canonical lift E always exists and is unique up to isomorphism. Furthermore, a theorem by Lubin, Serre and Tate [77] provides an effective algorithm to compute the j-invariant of E given the j-invariant of $\overline{E}$.

Theorem 3.1.1 (Lubin-Serre-Tate) *Let $\overline{E}$ be an ordinary elliptic curve over $\mathbb{F}_q$ with j-invariant $j(\overline{E}) \in \mathbb{F}_q \setminus \mathbb{F}_{p^2}$. Denote with Σ the Frobenius substitution on $\mathbb{Z}_q$ and with $\Phi_p(X,Y)$ the p-th modular polynomial. Then the system of equations*

$$\Phi_p(X, \Sigma(X)) = 0 \quad \text{and} \quad X \equiv j(\overline{E}) \bmod p, \tag{3.1}$$

has a unique solution $J \in \mathbb{Z}_q$, which is the j-invariant of the canonical lift E of $\overline{E}$ (defined up to isomorphism).

The hypothesis $j(\overline{E}) \notin \mathbb{F}_{p^2}$ in Theorem 3.1.1 is necessary to ensure that a certain partial derivative of Φ_p does not vanish modulo p and guarantees the uniqueness of the solution of equation (3.1). The case $j(\overline{E}) \in \mathbb{F}_{p^2}$ can be handled very easily using the Weil conjectures [137]: since $j(\overline{E}) \in \mathbb{F}_{p^2}$ there exists an elliptic curve $\overline{E}'$ defined over $\mathbb{F}_{p^m}$ with $m = 1$ or $m = 2$, which is isomorphic to $\overline{E}$ over $\mathbb{F}_q$. Let $t_k = p^{mk} + 1 - \#\overline{E}'(\mathbb{F}_{p^{mk}})$ then $t_{k+1} = t_1 t_k - p^m t_{k-1}$ with $t_0 = 2$ and therefore $\#\overline{E}(\mathbb{F}_q) = p^n + 1 - t_{n/m}$. So in the following we will assume $j(\overline{E}) \notin \mathbb{F}_{p^2}$ and in particular that $\overline{E}$ is ordinary [117, Theorem V.3.1].

3.1.2 Isogeny Cycles

Let $\sigma : \overline{E} \to \overline{E}^{\sigma} : (x, y) \mapsto (x^p, y^p)$ be the p-th power Frobenius isogeny, where $\overline{E}^{\sigma}$ is the curve obtained by raising each coefficient of $\overline{E}$ to the p-th power and let $\hat{\sigma}$ be the dual of σ. Repeatedly applying $\hat{\sigma}$ gives rise to the following cycle

$$\overline{E}_0 \xrightarrow{\hat{\sigma}_0} \overline{E}_1 \xrightarrow{\hat{\sigma}_1} \cdots \xrightarrow{\hat{\sigma}_{n-2}} \overline{E}_{n-1} \xrightarrow{\hat{\sigma}_{n-1}} \overline{E}_0 \ ,$$

with $\overline{E}_{(n-i)} = \overline{E}^{\sigma^i}$ and $\hat{\sigma}_i$ the dual of $\sigma_i : \overline{E}_{i+1} \to \overline{E}_i : (x, y) \mapsto (x^p, y^p)$. Composing these isogenies, we can express the Verschiebung as

$$\overline{V} = \hat{\overline{F}} = \hat{\sigma}_{n-1} \circ \hat{\sigma}_{n-2} \circ \ldots \circ \hat{\sigma}_0 \, .$$

Instead of lifting $\overline{E}$ and $\overline{V}$ directly, Satoh lifts the whole cycle $(\overline{E}_0, \overline{E}_1, \ldots, \overline{E}_{n-1})$ simultaneously leading to the diagram

$$\begin{tikzcd}
E_0 \arrow[r, "\widehat{\Sigma}_0"] \arrow[d, "\pi"'] & E_1 \arrow[r, "\widehat{\Sigma}_1"] \arrow[d, "\pi"'] & \cdots \arrow[r, "\widehat{\Sigma}_{n-2}"] & E_{n-1} \arrow[r, "\widehat{\Sigma}_{n-1}"] \arrow[d, "\pi"'] & E_0 \arrow[d, "\pi"'] \\
\overline{E}_0 \arrow[r, "\hat{\sigma}_0"] & \overline{E}_1 \arrow[r, "\hat{\sigma}_1"] & \cdots \arrow[r, "\hat{\sigma}_{n-2}"] & \overline{E}_{n-1} \arrow[r, "\hat{\sigma}_{n-1}"] & \overline{E}_0 \, ,
\end{tikzcd} \tag{3.2}$$

with E_i the canonical lift of $\overline{E}_i$ and $\widehat{\Sigma}_i$ the corresponding lift of $\hat{\sigma}_i$. The existence of a lift of $\hat{\sigma}_i$ is not trivial and follows from the following theorem which can be found in Messing [83, Corollary 3.4].

THEOREM 3.1.2 *Let $\overline{E}_1, \overline{E}_2$ be ordinary elliptic curves over $\mathbb{F}_q$ and E_1, E_2 their respective canonical liftings, then* $\mathrm{Hom}(\overline{E}_1, \overline{E}_2) \cong \mathrm{Hom}(E_1, E_2)$.

3.1.3 Lifting the j-invariants

The theorem of Lubin, Serre and Tate implies that the j-invariants of E_i satisfy

$$\Phi_p(j(E_i), j(E_{i+1})) = 0 \quad \text{and} \quad j(E_i) \equiv j(\overline{E}_i) \bmod p,$$

for $i = 0, \ldots, n-1$. Define $\Theta : \mathbb{Z}_q^n \to \mathbb{Z}_q^n$ by

$$\Theta(x_0, x_1, \ldots, x_{n-1}) = (\Phi_p(x_0, x_1), \Phi_p(x_1, x_2), \ldots, \Phi_p(x_{n-1}, x_0)),$$

then clearly we have $\Theta(j(E_0), j(E_1), \ldots, j(E_{n-1})) = (0, 0, \ldots, 0)$. Using a multivariate Newton iteration on Θ, we can lift the cycle $(j(\overline{E}_0), j(\overline{E}_1), \ldots, j(\overline{E}_{n-1}))$ to $\mathbb{Z}_q^n$ with arbitrary precision. The iteration is given by

$$(x_0, x_1, \ldots, x_{n-1}) \leftarrow (x_0, x_1, \ldots, x_{n-1}) - ((D\Theta)^{-1}\Theta)(x_0, x_1, \ldots, x_{n-1}), \tag{3.3}$$

with $(D\Theta)(x_0,\ldots,x_{n-1})$ the Jacobian matrix

$$\begin{pmatrix} \Phi_p'(x_0,x_1) & \Phi_p'(x_1,x_0) & 0 & \cdots & 0 \\ 0 & \Phi_p'(x_1,x_2) & \Phi_p'(x_2,x_1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \Phi_p'(x_{n-1},x_{n-2}) \\ \Phi_p'(x_0,x_{n-1}) & 0 & 0 & \cdots & \Phi_p'(x_{n-1},x_0) \end{pmatrix},$$

where $\Phi_p'(X,Y)$ denotes the partial derivative with respect to X. Note that $\Phi_p'(Y,X)$ is the partial derivative of $\Phi_p(X,Y)$ with respect to Y since $\Phi_p(X,Y)$ is symmetric. The p-th modular polynomial satisfies the Kronecker relation

$$\Phi_p(X,Y) \equiv (X^p - Y)(X - Y^p) \bmod p \tag{3.4}$$

and since $j(\overline{E}_i) \notin \mathbb{F}_{p^2}$ and $j(\overline{E}_i) = j(\overline{E}_{i+1})^p$, we have

$$\begin{cases} \Phi_p'(j(\overline{E}_i), j(\overline{E}_{i+1})) \equiv j(\overline{E}_{i+1})^{p^2} - j(\overline{E}_{i+1}) \not\equiv 0 \bmod p, \\ \Phi_p'(j(\overline{E}_{i+1}), j(\overline{E}_i)) \equiv j(\overline{E}_{i+1})^p - j(\overline{E}_{i+1})^p \equiv 0 \bmod p. \end{cases}$$

The above equations imply that $(D\Theta)(x_0,\ldots,x_{n-1}) \bmod p$ is a diagonal matrix with non-zero diagonal elements. Therefore, the Jacobian matrix is invertible over $\mathbb{Z}_q$ and we conclude that $\delta = ((D\Theta)^{-1}\Theta)(x_0,x_1,\ldots,x_{n-1}) \in \mathbb{Z}_q^n$. Note that we can simply apply Gauss elimination to solve

$$(D\Theta)^{-1}(x_0,\ldots,x_{n-1})\delta = \Theta(x_0,\ldots,x_{n-1})$$

since the diagonal elements are all invertible. Using row operations we move the bottom left element $\Phi_p'(x_0,x_{n-1})$ towards the right. After k row operations this element becomes

$$(-1)^k \Phi_p'(x_0,x_{n-1}) \prod_{i=0}^{k-1} \frac{\Phi_p'(x_{i+1},x_i)}{\Phi_p'(x_i,x_{i+1})}$$

which clearly is divisible by p^k since $\Phi_p'(x_{i+1},x_i) \equiv 0 \bmod p$. This procedure is summarized in Algorithm 3.1.1.

3.1.4 The Trace of Frobenius

The canonical lift E of an ordinary elliptic curve $\overline{E}$ over $\mathbb{F}_q$ has the property that $\mathrm{End}(\overline{E}) \cong \mathrm{End}(E)$. Let $\overline{F}$ be the Frobenius endomorphism on $\overline{E}$ and $\mathcal{F}$ the image of $\overline{F}$ under this ring isomorphism, then we clearly have $\mathrm{Tr}(\overline{F}) = \mathrm{Tr}(\mathcal{F})$. Furthermore, since $\mathbb{Q}_q$ has characteristic zero, we will be able to compute the exact value of $\mathrm{Tr}(\overline{F})$ and not just modulo p. The following proposition gives an easy relation between the trace of an endomorphism $f \in \mathrm{End}_{\mathbb{Q}_q}(E)$ and its action on the invariant differential of E.

ALGORITHM 3.1.1 (LIFT_J_INVARIANTS)

```
INPUT:   Cycle j_i ∈ F_q \ F_{p^2} with Φ_p(j_i, j_{i+1}) ≡ 0 mod p for 0 ≤ i < n
         and precision N.
OUTPUT:  Cycle J_i ∈ Z_q with π(J_i) = j_i and Φ_p(J_i, J_{i+1}) ≡ 0 mod p^N
         for all 0 ≤ i < n.

  1.  If N = 1 Then
      1.1.  For i = 0,...,n-1 do J_i = j_i;
  2.  Else
      2.1.  N' = ⌈N/2⌉; M = N - N';
      2.2.  (J_0,...,J_{n-1}) = Lift_j_Invariants((j_0,...,j_{n-1}), N');
      2.3.  For i = 0,...,n-2 do
            2.3.1.  t ≡ Φ'_p(J_i, J_{i+1})^{-1} mod p^M;
            2.3.2.  D_i ≡ tΦ'_p(J_{i+1}, J_i) mod p^M;
            2.3.3.  P_i ≡ t(((Φ_p(J_i, J_{i+1}) mod p^N)/p^{N'}) mod p^M;
      2.4.  R ≡ Φ'_p(J_0, J_{n-1}) mod p^M;
      2.5.  S ≡ (((Φ_p(J_{n-1}, J_0) mod p^N))/p^{N'}) mod p^M;
      2.6.  For i = 0,...,min(M, n-2) do
            2.6.1.  S ≡ S - RP_i mod p^M;
            2.6.2.  R ≡ -RD_i mod p^M;
      2.7.  R ≡ R + Φ'_p(J_{n-1}, J_0) mod p^M; P_{n-1} ≡ SR^{-1} mod p^M;
      2.8.  For i = n-2,...,0 do P_i ≡ P_i - D_i P_{i+1} mod p^M;
      2.9.  For i = 0,...,n-2 do J_i ≡ J_i - p^{N'} P_i mod p^N;
  3.  Return (J_0,...,J_{n-1});
```

Proposition 3.1.1 (Satoh) *Let E be an elliptic curve over $\mathbb{Q}_q$ having good reduction modulo p and let $f \in \mathrm{End}_{\mathbb{Q}_q}(E)$ of degree d. Let ω be the invariant differential on E and let $f^*(\omega) = c\omega$ be the action of f on ω, then*

$$\mathrm{Tr}(f) = c + \frac{d}{c}.$$

Note that if we apply the above proposition to the Frobenius endomorphism $\mathcal{F}$, we get $\mathrm{Tr}(\mathcal{F}) = \gamma + q/\gamma$ with $\mathcal{F}^*(\omega) = \gamma\omega$. Since $\overline{E}$ is ordinary it follows that either γ or q/γ is a unit in $\mathbb{Z}_q$. However, $\overline{F}$ is inseparable and thus $\gamma \equiv 0 \bmod p$, which implies that $\gamma \equiv 0 \bmod q$, since q/γ has to be a unit in $\mathbb{Z}_q$. So if we would like to compute $\mathrm{Tr}(\mathcal{F}) \bmod p^N$, then we would have to compute $\gamma \bmod p^{n+N}$. Furthermore, it turns out to be quite difficult to compute γ directly. As will become clear later, we would need to know $\mathrm{Ker}(\Sigma_i)$ which is subgroup of $E_{i+1}[p]$ of order p. However, since $\mathrm{Ker}(\sigma_i)$ is trivial we cannot simply lift $\mathrm{Ker}(\sigma_i)$ to $\mathrm{Ker}(\Sigma_i)$, but would need to factor the p division polynomial of E.

To avoid these problems, Satoh works with the Verschiebung $\overline{V}$ which is separable since $\overline{E}$ is ordinary and thus $\mathrm{Ker}(\overline{V})$ can be easily lifted to $\mathrm{Ker}(\mathcal{V})$. Furthermore, the trace of an endomorphism equals the trace of its dual, so we have $\mathrm{Tr}(\mathcal{F}) = \mathrm{Tr}(\mathcal{V}) = c + q/c$ with $\mathcal{V}^*(\omega) = c\,\omega$ and c a unit in $\mathbb{Z}_q$. Diagram (3.2) shows that $\mathcal{V}$ can be written as $\mathcal{V} = \widehat{\Sigma}_{n-1} \circ \widehat{\Sigma}_{n-2} \circ \cdots \circ \widehat{\Sigma}_0$ and therefore we can compute c from the action of $\widehat{\Sigma}_i$ on the invariant differential ω_{i+1} of E_{i+1} for $i = 0, \ldots, n-1$. More precisely, let c_i be defined by

$$\widehat{\Sigma}_i^*(\omega_{i+1}) = c_i\,\omega_i\,,$$

with ω_i the invariant differential on E_i, then $c = \prod_{0 \le i < n} c_i$. Since $\overline{V}$ is separable, c will be non-zero modulo p and we conclude

$$\mathrm{Tr}(\overline{F}) \equiv \prod_{0 \le i < n} c_i \bmod q\,.$$

Remark 3.1.1 Since all commutative squares in diagram (3.2) are conjugates of each other, we can also recover the trace of Frobenius as the norm of c_0, i.e.

$$\mathrm{Tr}(\overline{F}) \equiv \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(c_0) \bmod q\,.$$

3.1.5 Computing the c_i

The final step in Satoh's algorithm is to compute the coefficients c_i, based on the equations for E_i and E_{i+1} and the kernel of $\widehat{\Sigma}_i$, using Vélu's formulae [129].

The equations for E_i and E_{i+1} can be easily obtained using a univariate Newton iteration, since we already know their j-invariants. Satoh restricted $p \ge 5$, so we can assume that the elliptic curve $\overline{E}$ over $\mathbb{F}_q$ has an equation of the form $y^2 = x^3 + ax + b$ and that we want to find a lift $E : y^2 = x^3 + Ax + B$ with $A, B \in \mathbb{Z}_q$. Taking an arbitrary lift A of a, we compute B using the following formula for the j-invariant

$$j(E) = \frac{6912A^3}{4A^3 + 27B^2}\,.$$

ALGORITHM 3.1.2 (LIFT_CURVE)

```
INPUT:   Coefficients a, b ∈ F_q of curve Ē : y^2 = x^3 + ax + b,
         j-invariant J ∈ Z_q with j(Ē) ≡ J mod p, precision N.
OUTPUT:  Coefficients A, B ∈ Z_q of curve E : y^2 = x^3 + Ax + B
         with j(E) ≡ J mod p^N.

   1.  If N = 1 Then
       1.1.  Take A, B ∈ Z_q with A ≡ a mod p and B ≡ b mod p;
   2.  Else
       2.1.  N' = ⌈N/2⌉;  M = N − N';
       2.2.  A, B = Lift_Curve(a, b, J, N');
       2.3.  E ≡ (J(4A^3 + 27B^2) − 6912A^3 mod p^N)/p^N' mod p^M;
       2.4.  Δ ≡ E/(54JB) mod p^M;
       2.5.  B ≡ B − p^N' Δ mod p^N;
   3.  Return A, B;
```

Algorithm 3.1.2 provides a detailed description of the resulting Newton iteration. Note that since $p \geq 5$, the denominator in step 2.4 will be non-zero modulo p.

REMARK 3.1.2 Skjernaa [118] suggested to simply take $A = 3\gamma$ and $B = 2\gamma$ with

$$\gamma = \frac{j(E)}{1728 - j(E)} .$$

An easy calculation shows that this model is minimal, has the correct j-invariant and reduces to $\overline{E}$. Furthermore, only one inversion in $\mathbb{Z}_q$ is needed, so this method is much faster than Algorithm 3.1.2.

Once we have computed E_i and E_{i+1} we consider the following diagram:

$$\begin{tikzcd} E_i \arrow[rr, "\widehat{\Sigma}_i"] \arrow[dr, "\nu_i"'] & & E_{i+1} \\ & E_i/\mathrm{Ker}(\widehat{\Sigma}_i) \arrow[ur, dotted, "\lambda_i"'] & \end{tikzcd} \tag{3.5}$$

Given $\mathrm{Ker}(\widehat{\Sigma}_i)$, Satoh uses Vélu's formulae [129] to compute an equation for the curve $E_i/\mathrm{Ker}(\widehat{\Sigma}_i)$ and the isogeny ν_i. Since ν_i and $\widehat{\Sigma}_i$ are both separable, $\mathrm{Ker}(\nu_i) = \mathrm{Ker}(\widehat{\Sigma}_i)$ and $\deg(\nu_i) = \deg(\widehat{\Sigma}_i)$, there exists an isomorphism $\lambda_i : E_i/\mathrm{Ker}(\widehat{\Sigma}_i) \to E_{i+1}$ which makes the above diagram commutative. Due to

Vélu's construction, the action of ν_i on the invariant differential is trivial, i.e. $\nu_i^*(\omega_{i,K}) = \omega_i$ with $\omega_{i,K}$ the invariant differential on $E_i/\mathrm{Ker}(\widehat{\Sigma}_i)$. Therefore it is sufficient to compute the action of λ_i on ω_{i+1}.

Note that $\mathrm{Ker}\widehat{\Sigma}_i$ is a subgroup of order p of $E_i[p]$. Let $H_i(x)$ be defined as

$$H_i(x) = \prod_{P \in (\mathrm{Ker}\widehat{\Sigma}_i \setminus \{\mathcal{O}\})/\pm} (x - x(P)),$$

then $H_i(x)$ divides the p-division polynomial $\Psi_{p,i}(x)$ of E_i. To find the correct factor of $\Psi_{p,i}(x)$ Satoh proves the following lemma.

LEMMA 3.1.1 (SATOH) *Let $p \geq 3$, then $\mathrm{Ker}\widehat{\Sigma}_i = E_i[p] \cap E_i(\mathbb{Z}_q^{\mathrm{ur}})$, with $\mathbb{Z}_q^{\mathrm{ur}}$ the valuation ring of the maximal unramified extension $\mathbb{Q}_q^{\mathrm{ur}}$ of $\mathbb{Q}_q$.*

The above lemma implies that $H_i(x) \in \mathbb{Z}_q[x]$ is the unique monic polynomial of degree $(p-1)/2$ which divides $\Psi_{p,i}(x)$ and such that $H_i(x) \bmod p$ is square-free. Since $\overline{E}_i$ is ordinary, $\mathrm{Ker}\,\hat{\sigma}_i = \overline{E}_i[p]$ and $\psi_{p,i}(x) \bmod p$ has inseparable degree p. Therefore, $H_i(x)^p \equiv \psi_{p,i}(x) \bmod p$. This implies that we cannot apply Hensel's lemma since $H_i(x) \bmod p$ and $\psi_{p,i}(x)/H_i(x) \bmod p$ are not coprime. To solve this problem, Satoh [105, Lemma 2.1] devises a modified Hensel lifting, which also has quadratic convergence.

LEMMA 3.1.2 (SATOH) *Let $p \geq 3$ be a prime and $\Psi(x) \in \mathbb{Z}_q[x]$ satisfying $\Psi'(x) \equiv 0 \bmod p$ and $\Psi'(x) \not\equiv 0 \bmod p^2$. Let $h(x) \in \mathbb{Z}_q[x]$ be a monic polynomial such that*

1. *$h(x) \bmod p$ is square-free and coprime to $(\Psi'(x)/p) \bmod p$,*
2. *$\Psi(x) \equiv q(x)h(x) \bmod p^{n+1}$,*

then the polynomial

$$H(x) = h(x) + \left(\frac{\Psi(x)}{\Psi'(x)} h'(x) \bmod h(x)\right)$$

satisfies $H(x) \equiv h(x) \bmod p^n$ and $\Psi(x) \equiv Q(x)H(x) \bmod p^{2n+1}$.

This immediately leads to the Newton iteration given in Algorithm 3.1.3.

REMARK 3.1.3 Note that Lemma 3.1.1 no longer holds for $p = 2$. Indeed, the Newton polygon of the 2-division polynomial shows that there are two non-trivial points in $E_i[p] \cap E_i(\mathbb{Z}_q^{\mathrm{ur}})$, whereas $\mathrm{Ker}\widehat{\Sigma}_i$ has only one non-trivial point. The main problem in extending Satoh's algorithm to characteristic 2 therefore lies in choosing the correct 2-torsion point.

Let E_i be defined by the equation $y^2 = x^3 + A_i x + B_i$, then using Vélu's formulae, Satoh [105, Proposition 4.3] shows that $E_i/\mathrm{Ker}(\widehat{\Sigma})$ is given by the equation $y^2 = x^3 + \alpha_i x + \beta_i$ with

$$\begin{aligned}
\alpha_i &= (6 - 5p)A_i - 30(h_{i,1}^2 - 2h_{i,2}) \\
\beta_i &= (15 - 14p)B_i - 70(-h_{i,1}^3 + 3h_{i,1}h_{i,2} - 3h_{i,3}) + 42A_i h_{i,1}
\end{aligned}$$

ALGORITHM 3.1.3 (LIFT_KERNEL)

```
INPUT:    The p-division polynomial Ψ_p(x) of elliptic curve E
          over Z_q/(p^N Z_q), precision N.
OUTPUT:   H(x) = ∏_{P∈(Ker Σ̂_i \ {O})/±} (x − x(P)) mod p^{N−1}.

   1.  If N = 1 Then
       1.1.  H(x) = h(x) with Ψ_p(x) ≡ h(x)^p mod p;
   2.  Else
       2.1.  N' = ⌈(N−1)/2⌉;
       2.2.  H(x) = Lift_Kernel(Ψ_p(x), N');
       2.3.  H(x) ≡ H(x) + (H'(x)Ψ_p(x)/Ψ'_p(x) mod H(x)) mod p^N;
   3.  Return H(x);
```

---

where $h_{i,k}$ denotes the coefficient of $x^{(p-1)/2-k}$ in $H_i(x)$ and we define $h_{i,k} = 0$ for $(p-1)/2 - k < 0$.

Given the equation for $E_i/\mathrm{Ker}(\widehat{\Sigma}_i)$ we can now compute the isomorphism λ_i to $E_{i+1} : y^2 = x^3 + A_{i+1}x + B_{i+1}$. The only change of variables preserving the form of these equations is $\lambda_i : (x, y) \to (u_i^2 x, u_i^3 y)$ with

$$u_i^2 = \frac{\alpha_i}{\beta_i}\frac{B_{i+1}}{A_{i+1}}.$$

The action of λ_i on ω_{i+1} is given by $\lambda_i^*(\omega_{i+1}) = u_i^{-1}\omega_{i,K}$ and therefore

$$c_i^2 = \frac{\beta_i}{\alpha_i}\frac{A_{i+1}}{B_{i+1}}.$$

Computing $c^2 = \prod_{i=0}^{n-1} c_i^2$ and taking the square root gives the trace of Frobenius up to the sign. As shown in the proof of [117, Theorem V.4.1], we have

$$t \equiv \gamma\gamma^{\sigma}\cdots\gamma^{\sigma^{n-1}} \mod p$$

where γ is the coefficient of x^{p-1} in the polynomial $(x^3 + ax + b)^{(p-1)/2}$. This finally leads to Algorithm 3.1.4.

3.1.6 Complexity

According to Hasse's theorem we have $|t| \leq 2\sqrt{q}$. Therefore it suffices to lift all the data with precision $N \simeq n/2$. Since elements of $\mathbb{Z}_q/(p^N\mathbb{Z}_q)$ are represented as polynomials of degree less than n with coefficients in $\mathbb{Z}/(p^N\mathbb{Z})$, every element

---

ALGORITHM 3.1.4 (SATOH)

INPUT: Elliptic curve $\overline{E} : y^2 = x^3 + ax + b$ over $\mathbb{F}_{p^n}$, $j(\overline{E}) \notin \mathbb{F}_{p^2}$.
OUTPUT: The number of points on $\overline{E}(\mathbb{F}_{p^n})$.

1. $N = \lceil \log_p 4 + n/2 \rceil$; $S = 1$; $T = 1$;
2. $a_0 = a_n = a$; $b_0 = b_n = b$; $j_0 = j_n = j(\overline{E})$;
3. For $i = n - 1$ to 0 do $a_i = a_{i+1}^p$; $b_i = b_{i+1}^p$; $j_i = j_{i+1}^p$;
4. $(J_0, \ldots, J_{n-1}) =$ Lift_j_Invariants$((j_0, \ldots, j_{n-1}), N)$;
5. For $i = 0$ to $n - 1$ do
   - 5.1. $\gamma \equiv \frac{J_i}{1728 - J_i} \bmod p^N$; $A \equiv 3\gamma \bmod p^N$; $B \equiv 2\gamma \bmod p^N$;
   - 5.2. $\Psi_p(x) = p$-division polynomial of $y^2 = x^3 + Ax + B$;
   - 5.3. $H(x) =$ Lift_Kernel$(\Psi_p(x), N + 1)$;
   - 5.4. For $j = 0$ to 3 do $h_j =$ Coeff$(H(x), (p - 1)/2 - j)$;
   - 5.5. $\alpha_i = (6 - 5p)A_i - 30(h_1^2 - 2h_2)$;
   - 5.6. $\beta_i = (15 - 14p)B_i - 70(-h_1^3 + 3h_1h_2 - 3h_3) + 42A_ih_1$;
   - 5.7. $S = \beta AS$; $T = \alpha BT$;
6. $c =$ Sqrt$(S/T, N)$; $C =$ Coeff$((x^3 + ax + b)^{(p-1)/2}, p - 1)$;
7. If $c \not\equiv CC^\sigma \cdots C^{\sigma^{n-1}} \bmod p$ Then $c \equiv -c \bmod p^N$;
8. If $c^2 > 4p^n$ Then $c = c - p^N$;
9. Return $p^n + 1 - c$;

---

will take $O(n^2)$ memory for fixed p. Multiplication and division in $\mathbb{Z}_q/(p^N\mathbb{Z}_q)$ can be performed using $O(n^{2\mu})$ bit-operations.

For each curve $\overline{E}_i$ with $0 \leq i < n$ we need $O(1)$ elements of $\mathbb{Z}_q/(p^N\mathbb{Z}_q)$, so the total memory needed is $O(n^3)$ bits. Lifting the cycle of j-invariants to precision N, requires $O(\log N)$ iterations. In every iteration the precision of the computations almost doubles so the complexity is determined by the last iteration which takes $O(n^{2\mu+1})$ bit-operations. Computing one coefficient c_i^2 requires $O(1)$ multiplications, so to compute all c_i we also need $O(n^{2\mu+1})$ bit-operations. This leads to the following theorem.

THEOREM 3.1.3 (SATOH) *Let $\overline{E}$ be an elliptic curve over a finite field $\mathbb{F}_q$ with $q = p^n$ $(p \geq 5)$ and $j(\overline{E}) \notin \mathbb{F}_{p^2}$. Then there exists a deterministic algorithm to compute the number of points $\#\overline{E}(\mathbb{F}_q)$ which requires $O(n^3 \log n \log\log n)$ bit-operations and $O(n^3)$ memory for fixed p.*

Algorithm 3.1.4 was extended to $p = 2$ by Skjernaa [118] and independently to $p = 2, 3$ by Fouquet, Gaudry and Harley [41], who also reported on the first large scale computations in characteristic 2. The main difference with Algorithm 3.1.4 is the lifting of the kernel of Verschiebung: for $p = 2, 3$ it suffices to lift a single non-trivial p-torsion point to $\mathbb{Z}_q$. In the next two sections we will briefly review both algorithms.

3.2 Fouquet-Gaudry-Harley's Algorithm

Shortly after Satoh described his algorithm, Fouquet, Gaudry and Harley [41] extended it to characteristic 2 and 3. Furthermore, they provided a detailed description of an implementation for the characteristic 2 case and computed the number of points on an elliptic curve over $\mathbb{F}_{2^{8009}}$ in 13 days on an Alpha EV6 750 MHz using 16.9 GB of memory. This should be compared with the previous record: using an optimized implementation of the SEA-algorithm, we computed the number of points on a curve over $\mathbb{F}_{2^{1999}}$ [130] in 65 days on a PII 400 MHz.

3.2.1 Characteristic Two

Let $\overline{E}$ be an ordinary elliptic curve over a finite field $\mathbb{F}_q$ with $q = 2^n$ elements and assume that $j(\overline{E}) \notin \mathbb{F}_4$. It is well known that either $\overline{E}$ or its quadratic twist is isomorphic over $\mathbb{F}_q$ with an elliptic curve given by an equation of the form $y^2 + xy = x^3 + a$, with $a \in \mathbb{F}_q^*$.

Fouquet, Gaudry and Harley also use Algorithm 3.1.1 to lift the cycle of j-invariants and an analogon of Algorithm 3.1.2 to lift the equation of the curve. Assuming that E is given by the equation $y^2 + xy = x^3 + A$, we can compute $A \in \mathbb{Z}_q$ using the following formula for the j-invariant

$$j(E) = -\frac{1}{A + 432A^2}.$$

Given $j(E) \bmod 2^N$, we compute $A \bmod 2^N$ as a root of the polynomial $f(x) = 1 + J(x + 432x^2)$ starting from the initial approximation $-1/j(E) \bmod 16$. An easy calculation shows that in each step of the iteration the precision of the approximation increases from N to $2N + 4$.

ALGORITHM 3.2.1 (LIFT_CURVE_2)

INPUT: `Coefficient` $a \in \mathbb{F}_q$ `of curve` $\overline{E} : y^2 + xy = x^3 + a$,
j`-invariant` $J \in \mathbb{Z}_q$ `with` $j(\overline{E}) \equiv J \bmod 2$, `precision` N.
OUTPUT: `Coefficients` $A \in \mathbb{Z}_q$ `of curve` $E : y^2 + xy = x^3 + A$
`with` $j(E) \equiv J \bmod 2^N$.

1. `If` $N \leq 4$ `Then`
   1.1. $A \equiv -1/J \bmod 16$;
2. `Else`
   2.1. $N' = \lceil \frac{N-4}{2} \rceil$; $M = N - N'$;
   2.2. $A =$ `Lift_Curve_2`(a, N');
   2.3. $E \equiv ((1 + J(A + 432A^2)) \bmod 2^N)/2^{N'} \bmod 2^M$;
   2.4. $\Delta \equiv \frac{E}{J(1+864A)} \bmod 2^M$;
   2.5. $A \equiv A - 2^{N'}\Delta \bmod 2^N$;
3. `Return` A;

Let $\overline{E}_i$ be given by the equation $y^2 + xy = x^3 + a_i$ with $a \in \mathbb{F}_q^*$ and E_i by $y^2 + xy = x^3 + A_i$ with $A_i \in \mathbb{Z}_q^\times$. Since $\widehat{\Sigma}_i$ is separable and of degree 2, $\mathrm{Ker}\widehat{\Sigma}_i = \{\mathcal{O}, P_i\}$. To lift the non-trivial 2-torsion point $(0, \sqrt{a_i})$ to P_i, Fouquet, Gaudry and Harley use the fact that $\mathrm{Ker}\widehat{\Sigma} \subset E_i[2]$. The x-coordinate of P_i thus satisfies the 2-division polynomial $4x^3 + x^2 + 4A_i$. Since $x(P_i) \equiv 0 \bmod 2$, they compute $z_i = x(P_i)/2$ as a zero of the modified division polynomial $8z^3 + z^2 + A$ starting from the initial approximation $1/\sqrt{j(E_i)} \equiv 1/j(E_{i+1}) \bmod 8$. The equality $j(E_i/\mathrm{Ker}(\Sigma_i)) = j(E_{i+1})$ proves that this is indeed the correct starting value giving $x(P_i)$. In each step of the iteration the precision increases from N to $2N - 2$.

Once we have computed $x(P_i)$ we can recover P_i since $P_i = -P_i$ implies that $y(P_i) = -x(P_i)/2$. Again we consider diagram (3.5) and using Vélu's formulae, Fouquet, Gaudry and Harley prove that $E_i/\mathrm{Ker}\widehat{\Sigma}_i$ is given by the equation $y^2 + xy = x^3 + \alpha_i x + \beta_i$ with

$$\alpha_i = -5\gamma_i \quad \text{and} \quad \beta_i = A_i - \gamma_i - 7\delta_i,$$

where $\gamma_i = 3x(P_i)^2 + x(P_i)/2$ and $\delta_i = 7x(P_i)^3 + 3x(P_i)^2/2 + 4A_i$. The isomor-

ALGORITHM 3.2.2 (LIFT_Z)

```
INPUT:   Coefficient A ∈ F_q of curve E : y^2 + xy = x^3 + A,
         its j-invariant J ∈ Z_q, precision N.
OUTPUT:  Z ≡ x(P)/2 mod p^N with KerΣ̂ = {O, P}.

   1.  If N ≤ 3 Then
       1.1.  Z ≡ 1/√J mod 8;
   2.  Else
       2.1.  N' = ⌈(N+2)/2⌉;  M = N − N';
       2.2.  Z = Lift_Z(A, J, N');
       2.3.  E ≡ ((8Z^3 + Z^2 + A) mod p^N)/p^N' mod p^M;
       2.4.  Δ ≡ E/(12Z^2+Z) mod p^M;
       2.3.  Z ≡ Z − 2^(N'−1) Δ mod 2^N;
   3.  Return Z;
```

phism λ_i has the general form

$$(x, y) \to (u_i^2 x + r_i, u_i^3 y + u_i^2 s_i x + t_i), \quad (u_i, r_i, s_i, t_i) \in \mathbb{Q}_q^* \times \mathbb{Q}_q^3,$$

but a simple calculation shows that $c_i^2 = u_i^{-2}$. Solving the equations satisfied by (u_i, r_i, s_i, t_i) given in [117, Table 1.2] finally leads to

$$c_i^2 = -\frac{864\beta_i - 72\alpha_i + 1}{(48\alpha_i - 1)(1 + 864A_{i+1})}. \tag{3.6}$$

Substituting the expressions for α_i and β_i in the above equation and reducing modulo the minimal polynomial of $x(P_i)$ proves the following proposition.

PROPOSITION 3.2.1 *Let ω_i be the invariant differential on E_i and define c_i as $\widehat{\Sigma}_i^*(\omega_{i+1}) = c_i\omega_i$. Denote the non-trivial point in* $\mathrm{Ker}(\widehat{\Sigma}_i)$ *by $P_i = (x_i, y_i)$ then*

$$c_i^2 = \frac{1 - 252x_i + 19008A_i}{(1 + 120(x_i + 6x_i^2))(1 + 864A_{i+1})}.$$

Since $(\sqrt[4]{a}, \sqrt{a})$ is a point of order 4, we have $t \equiv 1 \bmod 4$ and we can decide which square root is the correct one. Taking into account that we lose 1 bit of precision in the computation of the square root, we need to compute c^2 with precision $\lceil n/2 \rceil + 3$. This finally leads to Algorithm 3.2.3.

ALGORITHM 3.2.3 (FOUQUET_GAUDRY_HARLEY_2)

INPUT: Elliptic curve $\overline{E}: y^2 + xy = x^3 + a$ over $\mathbb{F}_{2^n}$ with $j(\overline{E}) \notin \mathbb{F}_4$.
OUTPUT: The number of points on $\overline{E}(\mathbb{F}_{2^n})$.

1. $N = \lceil \frac{n}{2} \rceil + 3$; $S = 1$; $T = 1$;
2. $j_0 = j_n = j(\overline{E})$; $a_0 = a_n = a$;
3. For $i = n-1$ to 0 do $j_i = j_{i+1}^p$; $a_i = a_{i+1}^p$;
4. $(J_0, \ldots, J_{n-1}) =$ Lift_j_Invariants$((j_0, \ldots, j_{n-1}), N)$;
5. For $i = 0$ to $n-1$ do
   - 5.1. $A =$ Lift_Curve_2(a_i, N);
   - 5.2. $Z =$ Lift_Z(J_i, A, N);
   - 5.3. $S \equiv S(1 - 504Z + 19008A) \mod 2^N$;
   - 5.4. $T \equiv T((1 + 240(Z + 12Z^2))(1 + 864A)) \mod 2^N$;
6. $c =$ Sqrt$(S/T, N) \mod 2^{N-1}$;
7. If $c \not\equiv 1 \mod 4$ Then $c \equiv -c \mod 2^{N-1}$;
7. If $c^2 > 2^{n+2}$ Then $c = c - 2^{N-1}$;
9. Return $2^n + 1 - c$;

3.2.2 Implementation and Numerical Results

In this section we present running times of an implementation of Algorithm 3.2.3 in the `C` programming language. The basic operations on integers modulo 2^N for $N \leq 256$ are implemented in assembly language. Elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo a degree n irreducible polynomial, which is either a trinomial or a pentanomial. Multiplication of two elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ is implemented using Karatsuba's trick [60] in the polynomial dimension and classical multiplication for the coefficients.

Table 3.1 contains running times and memory usages of the algorithm by Fouquet, Gaudry and Harley for elliptic curves over various finite fields of characteristic 2. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1.

Table 3.1: Running time (s) and memory usage (KB) of Fouquet-Gaudry-Harley algorithm for elliptic curves over $\mathbb{F}_{2^n}$

n	j-inv	Curve	Torsion	Total	Mem
120	0.86	0.34	0.40	1.67	363
144	2.25	0.67	0.87	3.99	766
168	3.44	1.00	1.33	6.04	1034
192	5.26	1.50	1.99	9.29	1784
240	9.25	2.62	3.59	16.5	2768
288	22.1	6.55	8.13	38.6	4954
336	35.4	10.4	13.1	62.5	8064
384	58.4	16.2	20.1	101.1	12258
480	116.1	32.8	41.9	205.3	21824

3.2.3 Characteristic Three

Let $\overline{E}$ be an ordinary elliptic curve over $\mathbb{F}_q$ with $q = 3^n$ and $j(\overline{E}) \notin \mathbb{F}_9$. Without loss of generality we can assume that $\overline{E}$ has the equation $y^2 = x^3 + a_2x^2 + a_6$ with $a_2, a_6 \in \mathbb{F}_q^*$. The j-invariant is given by $j(\overline{E}) = -a_2^3/a_6$. The algorithm for $p = 3$ is very similar to Satoh's original algorithm since Lemma 3.1.1 remains valid. The main difference is the application of Vélu's formulae.

Given the j-invariant of the canonical lift E, we can easily find a Weierstrass model for E of the form $y^2 = x^3 + A_2x^2 + A_6$. It suffices to lift a_2 arbitrarily to A_2 and to compute A_6 using a Newton iteration based on the formula for the j-invariant $j(E) = -256A_2^6/(4A_2^3A_6 + 27A_6^2)$. Note that in each iteration the precision increases from N to $2N + 3$. The pseudo-code for this Newton iteration is given in Algorithm 3.2.4.

Since $\widehat{\Sigma}$ is separable and of degree 3, $\mathrm{Ker}\widehat{\Sigma} = \{\mathcal{O}, P, -P\}$ and Lemma 3.1.1 shows that $\mathrm{Ker}\widehat{\Sigma} = E[3] \cap E(\mathbb{Z}_q^{\mathrm{ur}})$. Therefore, we can compute $x(P)$ using a

---

ALGORITHM 3.2.4 (LIFT_CURVE_3)

```
INPUT:   Coefficients a_2, a_6 ∈ F_q of curve Ē : y^2 = x^3 + a_2 x^2 + a_6,
         j-invariant J ∈ Z_q with j(Ē) ≡ J mod 3, element A_2 ∈ Z_q
         with A_2 ≡ a_2 mod 3, precision N.
OUTPUT:  Coefficient A_6 ∈ Z_q of curve E : y^2 = x^3 + A_2 x^2 + A_6
         with j(E) ≡ J mod 3^N.
```

1. If $N \leq 3$ Then
   1.1. $A_6 \equiv -64A_2^3/J \bmod 27$;
2. Else
   2.1. $N' = \lceil \frac{N-3}{2} \rceil$; $M = N - N'$;
   2.2. $A_6 =$ Lift_Curve_3(a_2, a_6, J, A_2, N');
   2.3. $E \equiv (J(4A_2^3A_6 + 27A_6^2) + 256A_2^6 \bmod 3^N)/3^{N'} \bmod 3^M$;
   2.4. $\Delta \equiv \frac{E}{J(4A_2^3+54A_6)} \bmod 3^M$;
   2.5. $A_6 \equiv A - 3^{N'}\Delta \bmod 3^N$;
3. Return A_6;

Newton iteration on the 3-division polynomial

$$\Psi_3(x) = 3x^4 + 4A_2x^3 + 12A_6x + 4A_2A_6\,,$$

starting from the initial approximation $x_0 \equiv -\sqrt[3]{A_6} \bmod 9$. An easy calculation shows that $\mathrm{ord}_3(\Psi_3(x_0)) = 2$ and $\mathrm{ord}_3(\Psi_3'(x_0)) = 1$ and that in each step the precision increases from N to $2N - 1$, which leads to Algorithm 3.2.5.

Let E_i be given by the equation $y^2 = x^3 + A_ix^2 + B_i$ and assume we have computed $x_i = x(P_i)$ with $\mathrm{Ker}\widehat{\Sigma}_i = \{\mathcal{O}, P_i, -P_i\}$. Again we consider diagram (3.5) and using Vélu's formulae, Fouquet, Gaudry and Harley prove that $E_i/\mathrm{Ker}\widehat{\Sigma}_i$ is given by the equation $y^2 = x^3 + \alpha_ix^2 + \beta_ix + \gamma_i$ with

$$\alpha_i = A_i, \beta_i = -5t, \gamma_i = B_i - 4A_it - 7w\,,$$

with $t = 6x_i^2 + 4A_ix_i$ and $w = 10x_i^3 + 8A_ix_i^2 + 4B_i$. The expression for c_i^2 follows from an easy calculation of the isomorphism λ_i and is given by

$$c_i^2 = \frac{A_{i+1}^2(-1890x_i^3 - 1890A_ix_i^2 - 252A_i^2x_i + 2A_i^3 - 729B_i)}{(2A_{i+1}^3 + 27B_{i+1})(90x_i^2 + 60A_ix_i + A_i^2)}\,. \tag{3.7}$$

Finally, we recover $\mathrm{Tr}\overline{F} \equiv \pm\prod_{i=0}^{n-1} c_i \bmod q$. To determine the correct sign, note that $\mathrm{Tr}\overline{F} \equiv \mathrm{N}_{\mathbb{F}_{3^n}/\mathbb{F}_3}(a_2) \bmod 3$. The complexity of Algorithms 3.2.3 and 3.2.6 is easily seen to be the same as Satoh's original algorithm.

ALGORITHM 3.2.5 (LIFT_X)

INPUT: Coefficients $A_2, A_6 \in \mathbb{Z}_q$ of curve $E : y^2 = x^3 + A_2x^2 + A_6$, its j-invariant $J \in \mathbb{Z}_q$, precision N.
OUTPUT: $X \equiv x(P) \mod p^N$ with $\mathrm{Ker}\widehat{\Sigma} = \{\mathcal{O}, P, -P\}$.

1. If $N \leq 2$ Then
   1.1. $X \equiv -\sqrt[3]{A_6} \mod 9$;
2. Else
   2.1. $N' = \lceil \frac{N+1}{2} \rceil$; $M = N - N'$;
   2.2. $X =$ Lift_X(A_2, A_6, J, N');
   2.3. $E \equiv 3X^4 + 4A_2X^3 + 12A_6X + 4A_2A_6 \mod 3^N$;
   2.4. $E \equiv E/3^{N'} \mod 3^M$;
   2.5. $\Delta \equiv \frac{E}{4(X^3+A_2X^2+A_6)} \mod 3^M$;
   2.6. $Z \equiv Z - 3^{N'-1}\Delta \mod 3^N$;
3. Return Z;

THEOREM 3.2.1 (FOUQUET-GAUDRY-HARLEY) *Let $\overline{E}$ be an elliptic curve over a finite field $\mathbb{F}_{p^n}$ with $p = 2, 3$ and $j(\overline{E}) \notin \mathbb{F}_{p^2}$. Then there exists a deterministic algorithm to compute the number of points $\#\overline{E}(\mathbb{F}_{p^n})$ which requires $O(n^3 \log n \log\log n)$ bit-operations and $O(n^3)$ memory.*

3.3 Skjernaa's Algorithm

In [118], Skjernaa described an extension of Satoh's algorithm to characteristic 2. The main difference with Algorithm 3.1.4 is that in this case $\mathrm{Ker}\widehat{\Sigma}_i$ is a subgroup of $E_i[2]$ of order 2 and thus contains only one non-trivial point $P_i \in \mathrm{Ker}\widehat{\Sigma}_i$. Instead of lifting this 2-torsion point from $\mathbb{F}_q$ to $\mathbb{Z}_q$, Skjernaa gives an explicit formula for the coordinates of P as a function of $j(E_i)$ and $j(E_{i+1})$. Applying Vélu's formulae then leads again to an expression for c_i^2.

As in the previous section, we will assume that $\overline{E}$ is an ordinary elliptic curve over $\mathbb{F}_q$ with $q = 2^n$ and that $j(\overline{E}) \notin \mathbb{F}_4$. Furthermore, $\overline{E}$ is given by an equation of the form $y^2 + xy = x^3 + a$, with $a \in \mathbb{F}_q^*$.

Lifting the cycle of j-invariants is done using Algorithm 3.1.1. Given an approximation to $j(E_i)$, Skjernaa does not lift the equation of the curve as in Algorithm 3.1.2, but simply uses the Weierstrass form

$$E_i : y^2 + xy = x^3 - \frac{36}{j(E_i) - 1728}x - \frac{1}{j(E_i) - 1728}, \tag{3.8}$$

ALGORITHM 3.2.6 (FOUQUET_GAUDRY_HARLEY_3)

```
INPUT:   Elliptic curve E̅ : y^2 = x^3 + a_2 x^2 + a_6 over F_{3^n}, j(E̅) ∉ F_9.
OUTPUT:  The number of points on E̅(F_{3^n}).
```

INPUT: Elliptic curve $\overline{E} : y^2 = x^3 + a_2x^2 + a_6$ over $\mathbb{F}_{3^n}$, $j(\overline{E}) \notin \mathbb{F}_9$.
OUTPUT: The number of points on $\overline{E}(\mathbb{F}_{3^n})$.

1. $N = \lceil \frac{n}{2} \rceil + 2$; $S = 1$; $T = 1$;
2. $j_0 = j_n = j(\overline{E})$; $a_{2,0} = a_{2,n} = a_2$; $a_{6,0} = a_{6,n} = a_6$;
3. For $i = n-1$ to 0 do $j_i = j_{i+1}^3$; $a_{2,i} = a_{2,i+1}^3$; $a_{6,i} = a_{6,i+1}^3$;
4. $(J_0, \ldots, J_{n-1}) =$ Lift_j_Invariants($(j_0, \ldots, j_{n-1}), N$);
5. For $i = 0$ to $n-1$ do
   5.1. $A \equiv a_2 \mod 3$;
   5.2. $B =$ Lift_Curve_3(a_2, a_6, J_i, A, N);
   5.3. $X =$ Lift_X(A, B, J_i, N);
   5.4. $S \equiv SA^2(-1890X^2(X+A) - 252A^2X + 2A^3 - 729B) \mod 3^N$;
   5.5. $T \equiv T((2A^3 + 27B)(90X^2 + 60AX + A^2)) \mod 3^N$;
6. $c \equiv$ Sqrt($S/T, N$) $\mod 3^N$;
7. If $c \not\equiv \mathrm{N}_{\mathbb{F}_{3^n}/\mathbb{F}_3}(a_2) \mod 3$ Then $c \equiv -c \mod 3^N$;
7. If $c^2 > 4 \cdot 3^n$ Then $c = c - 3^N$;
9. Return $3^n + 1 - c$;

---

which has the correct j-invariant, is defined over $\mathbb{Z}_q$ and reduces to $\overline{E}_i$.

Assuming $\mathrm{Ker}\widehat{\Sigma}_i = \{\mathcal{O}, P_i\}$ has been computed, we can again consider diagram (3.5). Using Vélu's formulae Skjernaa shows that $E_i/\mathrm{Ker}\widehat{\Sigma}_i$ is given by the equation $y^2 + xy = x^3 + \alpha_i x + \beta_i$ with

$$\alpha_i = -\frac{36}{j(E_i) - 1728} - 5\gamma_i\,,$$

$$\beta_i = -\frac{1}{j(E_i) - 1728} - (1 + 7x_i)\gamma_i\,,$$

where x_i is the x-coordinate of P_i and $\gamma_i = 3x_i^2 - 36/(j(E_i) - 1728) + x_i/2$. The isomorphism λ_i now has the general form

$$(x, y) \to (u_i^2 x + r_i, u_i^3 y + u_i^2 s_i x + t_i), \quad (u_i, r_i, s_i, t_i) \in \mathbb{Q}_q^* \times \mathbb{Q}_q^3,$$

but an easy calculation shows that $c_i^2 = u_i^{-2}$. Solving the equations satisfied by (u_i, r_i, s_i, t_i) given in [117, Table 1.2] finally leads to

$$c_i^2 = -\frac{864\beta_i - 72\alpha_i + 1}{48\alpha_i - 1}\,.$$

Substituting the expressions for α_i and β_i in the above equation proves the following proposition.

PROPOSITION 3.3.1 *Let ω_i be the invariant differential on E_i and define c_i as $\widehat{\Sigma}_i^*(\omega_{i+1}) = c_i\omega_i$. Denote the non-trivial point in $\mathrm{Ker}(\widehat{\Sigma}_i)$ by $P_i = (x_i, y_i)$ and let $z_i = x_i/2$ and $v_i = (12z_i^2 + z_i)(j(E_i) - 1728) - 36$, then*

$$c_i^2 = \frac{j(E_i) - (504 + 12096z_i)v_i}{j(E_i) + 240v_i}\,. \tag{3.9}$$

In order to compute c_i^2, we need an expression for the x-coordinate of the non-trivial point $P_i \in \mathrm{Ker}(\widehat{\Sigma}_i)$. Since P_i is a 2-torsion point, it follows that $2y_i + x_i = 0$. Substituting y_i in the equation of the curve and using the equality $j(E_{i+1}) = j(E_i/\mathrm{Ker}\widehat{\Sigma}_i)$, Skjernaa finally deduces an explicit expression for x_i. A proof of the following proposition can be found in [118, Lemma 4.1].

PROPOSITION 3.3.2 *Let $P_i = (x_i, y_i)$ be the non-trivial point in $\mathrm{Ker}(\widehat{\Sigma}_i)$ and let $z_i = x_i/2$, then*

$$z_i = -\frac{j(E_{i+1})^2 + 195120j(E_{i+1}) + 4095j(E_i) + 660960000}{8(j(E_{i+1})^2 + j(E_{i+1})(563760 - 512j(E_i)) + 372735j(E_i) + 8981280000)}\,.$$

Skjernaa shows that the 2-adic valuation of both the numerator and denominator is 12, so we have to compute $j(E_i) \bmod 2^{N+12}$ to recover $z_i \bmod 2^N$.

Since $(\sqrt[4]{a}, \sqrt{a})$ is a point of order 4, we have $t \equiv 1 \bmod 4$ and we can decide which square root is the correct one. Taking into account that we lose 1 bit of precision in the computation of the square root, we need to compute c^2 with precision $\lceil n/2 \rceil + 3$. Substituting the expressions for z_i and t_i in c_i^2, we see that we have to determine the j-invariants $j(E_i)$ with precision $\lceil n/2 \rceil + 13$. This finally leads to Algorithm 3.3.1 and also proves the following theorem.

THEOREM 3.3.1 (SKJERNAA) *Let $\overline{E}$ be an elliptic curve over a finite field $\mathbb{F}_q$ with $q = 2^n$ and $j(\overline{E}) \notin \mathbb{F}_4$. Then there exists a deterministic algorithm to compute the number of points $\#\overline{E}(\mathbb{F}_q)$ which requires $O(n^3 \log n \log\log n)$ bit-operations and $O(n^3)$ memory.*

ALGORITHM 3.3.1 (SKJERNAA)

INPUT: Elliptic curve $\overline{E} : y^2 + xy = x^3 + a$ over $\mathbb{F}_{2^n}$ with $j(\overline{E}) \notin \mathbb{F}_4$.
OUTPUT: The number of points on $\overline{E}(\mathbb{F}_{2^n})$.

1. $N = \lceil \frac{n}{2} \rceil + 13$; $N' = N - 10$; $S = 1$; $T = 1$;
2. $j_0 = j_n = j(\overline{E})$;
3. For $i = n-1$ to 0 do $j_i = j_{i+1}^p$;
4. $(J_0, \ldots, J_{n-1}) =$ Lift_j_Invariants$((j_0, \ldots, j_{n-1}), N)$;
5. For $i = 0$ to $n-1$ do
   - 5.1. $Z \equiv -\frac{J_{i+1}^2 + 195120 J_{i+1} + 4095 J_i + 660960000}{8(J_{i+1}^2 + J_{i+1}(563760 - 512 J_i) + 372735 J_i + 8981280000)} \mod p^N$;
   - 5.2. $V \equiv (12Z^2 + Z)(J_i - 1728) - 36 \mod p^{N'}$;
   - 5.3. $S \equiv S(J_i - (504 + 12096Z)V) \mod p^{N'}$;
   - 5.4. $T \equiv T(240V + J_i) \mod p^{N'}$;
6. $c \equiv$ Sqrt$(S/T, N') \mod 2^{N'-1}$;
7. If $c \not\equiv 1 \mod 4$ Then $c \equiv -c \mod 2^{N'-1}$;
7. If $c^2 > 2^{n+2}$ Then $c = c - 2^{N'-1}$;
9. Return $2^n + 1 - c$;

3.3.1 Implementation and Numerical Results

In this section we present running times of an implementation of Algorithm 3.3.1 in the `C` programming language. The basic operations on integers modulo 2^N for $N \leq 256$ are implemented in assembly language. Elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo a degree n irreducible polynomial, which is either a trinomial or a pentanomial. Multiplication of two elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ is implemented using Karatsuba's trick [60] in the polynomial dimension and classical multiplication for the coefficients.

Table 3.2 contains running times and memory usages of Skjernaa's algorithm for elliptic curves over various finite fields of characteristic 2. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1.

Table 3.2: Running time (s) and memory usage (KB) of Skjernaa's algorithm for elliptic curves over $\mathbb{F}_{2^n}$

n	j-inv	Torsion	Total	Mem
120	1.41	0.23	1.77	537
144	2.30	0.37	2.89	766
168	3.85	0.65	4.93	1372
192	5.40	0.93	6.95	1784
240	13.9	2.26	17.7	3453
288	22.1	3.25	27.8	4954
336	35.8	6.03	46.4	8064
384	58.9	10.3	77.1	12258
480	117.2	22.1	157.9	21824

3.4 Vercauteren's Algorithm

In this section, we present a new version of Satoh's algorithm and its extensions to characteristic 2 and 3, which requires only $O(n^2)$ memory, but still runs in $O(n^{2\mu+1})$ time. The basic idea is very simple: the trace of Frobenius t can be computed as $t \equiv \prod_{0 \le i < n} c_i \bmod q$ and the c_i only depend on the curves E_i and E_{i+1}. So the main problem of Satoh's algorithm is that it lifts all j-invariants simultaneously, instead of lifting one j-invariant at a time. Note however that lifting all j-invariants simultaneously is exactly what makes Satoh's algorithm efficient, since this avoids costly Frobenius computations in $\mathbb{Z}_q$. Thus if we want our algorithm to run in $O(n^{2\mu+1})$ bit-operations and only use $O(n^2)$ memory, we have to devise a method to lift one j-invariant without using Frobenius computations.

3.4.1 Lifting One j-invariant

Our strategy is as follows: the j-invariants $j(E_i)$ and $j(E_{i+1})$ satisfy the following relations

$$\Phi_p(j(E_{i+1}), j(E_i)) = 0, \ j(E_i) \equiv j(\overline{E}_i) \bmod p, \ j(E_{i+1}) \equiv j(\overline{E}_{i+1}) \bmod p.$$

Suppose we have $J_{i+1} \equiv j(E_{i+1}) \bmod p^N$ at our disposal, then we can compute $J_i \equiv j(E_i) \bmod p^N$ using a univariate Newton iteration on $\Phi_p(J_{i+1}, Y)$. This iteration is given by

$$Z \leftarrow Z - \frac{\Phi_p(J_{i+1}, Z)}{\frac{\partial \Phi_p}{\partial Y}(J_{i+1}, Z)}, \tag{3.10}$$

and we can use $j(\overline{E}_i) \equiv j(E_i) \bmod p$ as an initial approximation. Note that since $\Phi_p(X, Y)$ satisfies the Kronecker relation, $\frac{\partial \Phi_p}{\partial Y}(J_{i+1}, j(\overline{E}_i))$ will be invertible in $\mathbb{Z}_q$.

Applying this method repeatedly, we conclude that it suffices to compute one j-invariant to precision N. The following proposition proves that the above iteration has a quite remarkable property: let J_i be the zero of $\Phi_p(J_{i+1}, Y)$, then J_i equals $j(E_i)$ up to precision $N+1$ and not just up to precision N as one would expect.

PROPOSITION 3.4.1 *Let $\mathbb{Q}_q$ be an unramified extension of $\mathbb{Q}_p$ and denote with $\mathbb{Z}_q$ its valuation ring. Let $g \in \mathbb{Z}_q[X,Y]$ and assume that $x_0, y_0 \in \mathbb{Z}_q$ satisfy*

$$g(x_0, y_0) \equiv 0 \bmod p, \; \frac{\partial g}{\partial X}(x_0, y_0) \equiv 0 \bmod p, \; \frac{\partial g}{\partial Y}(x_0, y_0) \not\equiv 0 \bmod p.$$

Then the following properties hold:

1. *For every $x \in \mathbb{Z}_q$ with $x \equiv x_0 \bmod p$ there exists a unique $y \in \mathbb{Z}_q$ such that $y \equiv y_0 \mod p$ and $g(x,y) = 0$.*
2. *Let $x' \in \mathbb{Z}_q$ with $x \equiv x' \bmod p^M$, $M \geq 1$ and let $y' \in \mathbb{Z}_q$ be the unique element with $y' \equiv y_0 \bmod p$ and $g(x', y') = 0$. Then $y' \equiv y \bmod p^{M+1}$.*

PROOF: Define $h \in \mathbb{Z}_q[Y]$ by $h(Y) = g(x, Y)$. Then $h(y_0) \equiv 0 \mod p$ and $h'(y_0) \equiv \frac{\partial g}{\partial Y}(x_0, y_0) \bmod p$. Therefore, $h'(y_0) \not\equiv 0 \bmod p$ and Hensel's lemma guarantees the existence of a unique $y \in \mathbb{Z}_q$ such that $h(y) = g(x,y) = 0$ and $y \equiv y_0 \bmod p$, which proves the first part of the proposition. Furthermore, given x, one can compute y with arbitrary precision using a univariate Newton iteration on $g(x, Y)$ with $y_0 \bmod p$ as an initial approximation. For the second part, define $\delta_x = x' - x$ and $\delta_y = y' - y$. Clearly $\delta_x \equiv \delta_y \equiv 0 \bmod p^M$. Writing out the Taylor series of $g(X,Y) = \sum_{i,j} g_{i,j} X^i Y^i$ leads to

$$\begin{aligned} 0 = g(x', y') = g(x + \delta_x, y + \delta_y) &= \sum_{i,j} g_{i,j} (x + \delta_x)^i (y + \delta_y)^j \\ &= \sum_{i,j} g_{i,j} (x^i + i x^{i-1} \delta_x + \delta_x^2 R_x(x))(y^j + j y^{j-1} \delta_y + \delta_y^2 R_y(y)), \end{aligned} \tag{3.11}$$

with R_x, R_y polynomials with coefficients in $\mathbb{Z}_q$. Since $\delta_x^2 \equiv \delta_y^2 \equiv 0 \bmod p^{2M}$ and $M \geq 1$ we get

$$0 \equiv \frac{\partial g}{\partial X}(x,y)(x - x') + \frac{\partial g}{\partial Y}(x,y)(y - y') \bmod p^{M+1}.$$

The above equation implies that $y \equiv y' \bmod p^{M+1}$, since $\delta_x \equiv 0 \bmod p^M$, $\frac{\partial g}{\partial X}(x,y) \equiv 0 \bmod p$ and $\frac{\partial g}{\partial Y}(x,y) \not\equiv 0 \bmod p$. □

Repeatedly applying Proposition 3.4.1 leads to a very simple algorithm to compute $J_0 \equiv j(E_0) \bmod p^N$. Starting from $j(\overline{E}_{N-1}) \equiv j(E_{N-1}) \bmod p$, we determine $J_{N-i} \equiv j(E_{N-i}) \bmod p^i$ as the zero of $\Phi_p(J_{N-i+1}, Y)$ with $J_{N-i+1} \equiv j(E_{N-i+1}) \bmod p^{i-1}$. After $N-1$ steps we reach $J_0 \equiv j(E_0) \bmod p^N$. The details of this procedure are summarized in Algorithms 3.4.1, 3.4.2 and 3.4.3.

ALGORITHM 3.4.1 (LIFT_FIRST_J_INVARIANT)

INPUT: A j-invariant $j \in \mathbb{F}_{p^n} \setminus \mathbb{F}_{p^2}$ and a precision N.
OUTPUT: $J \in \mathbb{Z}_q$ with $J \equiv j \mod p$ and $\Phi_p(J, \Sigma(J)) \equiv 0 \mod p^N$.

1. $J \equiv j^{p^{(n-N+1)}} \mod p$;
2. For $i = 2$ To N Do
   2.1. $J =$ Lift_Previous_J_Invariant(J, i);
3. Return J.

ALGORITHM 3.4.2 (LIFT_PREVIOUS_J_INVARIANT)

INPUT: $J \in \mathbb{Z}_q$ with $\Phi_p(J, \Sigma(J)) \equiv 0 \mod p^{N-1}$, precision N.
OUTPUT: $J' \in \mathbb{Z}_q$ with $\Phi_p(J', \Sigma(J')) \equiv 0 \mod p^N$.

1. Compute $h(Y) \equiv \Phi_p(J, Y) \mod p^N$;
2. $J' =$ Lift_Previous_J_Invariant_Rec$(J, h(Y), N)$;
3. Return J'.

ALGORITHM 3.4.3 (LIFT_PREVIOUS_J_INVARIANT_REC)

INPUT: $J \in \mathbb{Z}_q$, $h(Y) \equiv \Phi_p(J, Y) \mod p^N$ and precision N.
OUTPUT: An element $J' \in \mathbb{Z}_q$ with $h(J') \equiv 0 \mod p^N$.

1. If $N = 1$ Then
   1.1. $J' \equiv J^p \mod p$;
2. Else
   2.1. $N' = \lceil \frac{N}{2} \rceil$;
   2.2. $J' =$ Lift_Previous_J_Invariant_Rec$(J, h(Y), N')$;
   2.3. $J' \equiv J' - \frac{h(J')}{h'(J')} \mod p^N$;
3. Return J'.

3.4.2 An $O(n^2)$ Memory Algorithm

Using the ideas of the previous section, we obtain an $O(n^2)$ memory algorithm by interleaving the lifting of the j-invariants with the computation of the trace of Frobenius.

Analysing step 5 of Algorithm 3.1.4 and Algorithm 3.2.3, we see that in each iteration of the `For`-loop we only need one j-invariant (two in Skjernaa's algorithm) to compute the value of c_i^2. Thus instead of lifting the whole cycle of j-invariants simultaneously, we lift the first j-invariant using Algorithm 3.4.1 and every other j-invariant using Algorithm 3.4.2. We summaries this strategy in Algorithm 3.4.4 for $p \geq 5$ and Algorithm 3.4.5 for $p = 2$. Note that the algorithm by Fouquet, Gaudry and Harley can also be easily adapted.

The complexity of Algorithms 3.4.4 and 3.4.5 can be determined as follows: every element of $\mathbb{Z}_q/(p^N\mathbb{Z}_q)$ is represented by a polynomial of degree $\leq n$ with coefficients in $\mathbb{Z}/(p^N\mathbb{Z})$ and since $N \simeq n/2$, every element will take $O(n^2)$ space for fixed p. Since at every stage in the algorithm we only need $O(1)$ elements of $\mathbb{Z}_q/(p^N\mathbb{Z}_q)$ the total memory complexity is $O(n^2)$.

Lifting one j-invariant to precision N takes $O(n^{2\mu})$ bit-operations. Indeed, in every recursive call of Algorithm 3.4.3 the precision almost doubles, so the overall complexity is determined by the last call. The loop in step 3 has to be executed $O(n)$ times, so the total complexity of step 3 amounts to $O(n^{2\mu+1})$ bit-operations. Since the j-invariant in step 2 is computed using N Newton iterations with varying precision $i = 2, \ldots, N$, the total cost of step 1 is trivially bounded by $O(n^{2\mu+1})$ bit-operations. This proves the following theorem.

THEOREM 3.4.1 (VERCAUTEREN) *Let $\overline{E}$ be an elliptic curve over a finite field $\mathbb{F}_q$ with $q = p^n$ and $j(\overline{E}) \notin \mathbb{F}_{p^2}$. Then there exists a deterministic algorithm to compute the number of points $\#\overline{E}(\mathbb{F}_q)$ which requires $O(n^3 \log n \log\log n)$ bit-operations and $O(n^2)$ memory for fixed p.*

3.4.3 Implementation and Numerical Results

In this section we present running times of an implementation of Algorithm 3.4.5 in the `C` programming language. The basic operations on integers modulo 2^N for $N \leq 256$ are implemented in assembly language. Elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo a degree n irreducible polynomial, which is either a trinomial or a pentanomial. Multiplication of two elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ is implemented using Karatsuba's trick [60] in the polynomial dimension and classical multiplication for the coefficients.

Table 3.3 contains running times and memory usages of our algorithm for elliptic curves over various finite fields of characteristic 2. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1.

REMARK 3.4.1 Note that the loop in step 3 of Algorithm 3.4.5 can be replaced by a norm computation as explained in Remark 3.1.1. The norm computation algorithms described in Sections 3.6.2 and 3.10.3 would lead to a further speed-up by a factor of 4 to 5.

Table 3.3: Running time (s) and memory usage (KB) of Vercauteren's algorithm for elliptic curves over $\mathbb{F}_{2^n}$

n	First j-inv	Other j-inv	Torsion	Total	Mem
120	0.25	0.59	0.22	1.17	26
144	0.43	0.78	0.35	1.91	32
168	0.70	1.68	0.64	3.22	48
192	1.02	2.07	0.85	4.62	55
240	2.02	5.30	2.23	10.9	84
288	3.62	8.31	3.41	17.6	103
336	6.13	13.5	6.05	30.2	143
384	9.54	21.4	10.3	49.3	189
480	21.1	44.4	22.3	106.5	270

3.5 Mestre's AGM Algorithm

In a letter to Gaudry and Harley, Mestre [85] described a very elegant method based on the Arithmetic Geometric Mean (AGM) to count the number of points on ordinary curves of genus 1 and 2 over $\mathbb{F}_{2^n}$. The resulting algorithm runs in $O(n^{2\mu+1})$ bit-operations and requires $O(n^2)$ memory, which is the same as our algorithm. However, the AGM algorithm is easier to implement and runs faster by a constant. Using their highly optimized implementation, Gaudry and Harley computed the number of points on an elliptic curve over $\mathbb{F}_{2^{16001}}$ in 5 days on an Alpha EV6 750 MHz using 500 MB of memory.

The AGM algorithm was reformulated in the language of schemes and extended to characteristic 3 by Carls [15]. In 2003, Kohel [67] deduced explicit equations for characteristics $p = 2, 3, 5, 7, 13$ by lifting Heegner points on the modular curves $X_0(N)$. In this section we give a detailed description of the bivariate version of the AGM as originally proposed by Mestre [85]. In practice one would use the univariate version which can be found in Section 3.8.

For $a_0, b_0 \in \mathbb{R}$ with $a_0 \geq b_0 > 0$, define the AGM iteration for $k \in \mathbb{N}$ as

$$(a_{k+1}, b_{k+1}) = \left(\frac{a_k + b_k}{2}, \sqrt{a_k b_k}\right) .$$

Then $b_k \leq b_{k+1} \leq a_{k+1} \leq a_k$ and $0 \leq a_{k+1} - b_{k+1} \leq (a_k - b_k)/2$, therefore $\lim_{k\to\infty} a_k = \lim_{k\to\infty} b_k$ exists. This common value is called the AGM of a_0 and b_0 and is denoted as $\mathrm{AGM}(a_0, b_0)$. An easy calculation shows that

$$\frac{a_k}{b_k} - 1 \leq \frac{a_0 - b_0}{2^k b_k} \leq \frac{1}{2^k}\left(\frac{a_0}{b_0} - 1\right) ,$$

so after a logarithmic number of steps we have $a_k/b_k = 1 + \varepsilon_k$ with $\varepsilon_k < 1$. The

---

ALGORITHM 3.4.4 (VERCAUTEREN_CHAR_5)

```
INPUT:   Elliptic curve $\overline{E} : y^2 = x^3 + ax + b$ over $\mathbb{F}_{p^n}$ with $j(\overline{E}) \notin \mathbb{F}_{p^2}$.
OUTPUT:  The number of points on $\overline{E}(\mathbb{F}_{p^n})$.

  1.  $N = \lceil \log_p 4 + n/2 \rceil$; $S = 1$; $T = 1$;
  2.  $J =$ Lift_First_J_Invariant($j(\overline{E}), N$);
  3.  For $i = 0$ to $n - 1$ do
      3.1.  $A, B =$ Lift_Curve($a^{p^{n-i}}, b^{p^{n-i}}, J, N$);
      3.2.  $\Psi_p(x) =$ $p$-division polynomial of $y^2 = x^3 + Ax + B$;
      3.3.  $H(x) =$ Lift_Kernel($\Psi_p(x), N + 1$);
      3.4.  For $j = 0$ to 3 do $h_j =$ Coeff($H(x), (p-1)/2 - j$);
      3.5.  $\alpha = (6 - 5p)A - 30(h_1^2 - 2h_2)$;
      3.6.  $\beta = (15 - 14p)B - 70(-h_1^3 + 3h_1h_2 - 3h_3) + 42Ah_1$;
      3.7.  $S = \beta AS$; $T = \alpha BT$;
      3.8.  $J =$ Lift_Previous_J_Invariant($J, N$);
  4.  $c =$ Sqrt($S/T, N$); $C =$ Coeff($(x^3 + ax + b)^{(p-1)/2}, p - 1$);
  5.  If $c \neq CC^\sigma \cdots C^{\sigma^{n-1}} \bmod p$ Then $c = -c \bmod p^N$;
  6.  If $c^2 > 4p^n$ Then $c = c - p^N$;
  7.  Return $p^n + 1 - c$;
```

Algorithm 3.4.5 (Vercauteren_Char_2)

INPUT: Elliptic curve $\overline{E} : y^2 + xy = x^3 + a$ over $\mathbb{F}_{2^n}$ with $j(\overline{E}) \notin \mathbb{F}_4$.
OUTPUT: The number of points on $\overline{E}(\mathbb{F}_{2^n})$.

1. $N = \lceil \frac{n}{2} \rceil + 13$; $N' = N - 10$; $S = 1$; $T = 1$;
2. $J =$ `Lift_First_J_Invariant`$(1/a, N)$;
3. For $i = 0$ to $n - 1$ do
    - 3.1. $J' =$ `Lift_Previous_J_Invariant`(J, N);
    - 3.2. $Z \equiv -\frac{(J^2+195120J+4095J'+660960000)}{8(J^2+J(563760-512J')+372735J'+8981280000)} \mod 2^N$;
    - 3.3. $V \equiv (12Z^2 + Z)(J' - 1728) - 36 \mod 2^{N'}$;
    - 3.4. $S \equiv S(J' - (504 + 12096Z)V) \mod 2^{N'}$;
    - 3.5. $T \equiv T \times (240V + J') \mod 2^{N'}$;
    - 3.6. $J = J'$;
4. $c =$ `Sqrt`$(S/T, N') \mod 2^{N'-1}$;
5. If $c \not\equiv 1 \mod 4$ Then $c = -c \mod 2^{N'-1}$;
6. If $c^2 > 2^{n+2}$ Then $c = c - 2^{N'-1}$;
7. Return $2^n + 1 - c$;

Taylor series expansion of $1/\sqrt{1+\varepsilon_k}$ shows that convergence is even quadratic:

$$\frac{a_{k+1}}{b_{k+1}} = \frac{a_k+b_k}{2\sqrt{a_k b_k}} = \frac{2+\varepsilon_k}{2\sqrt{1+\varepsilon_k}} = 1 + \frac{\varepsilon_k^2}{8} - \frac{\varepsilon_k^3}{8} + \frac{15\varepsilon_k^4}{128} - \frac{7\varepsilon_k^5}{64} + O(\varepsilon_k^6)\,. \quad (3.12)$$

3.5.1 The AGM Isogeny and the Canonical Lift

Let $\mathbb{Q}_q$ be the unramified extension of $\mathbb{Q}_2$ of degree n with valuation ring $\mathbb{Z}_q$ and residue field $\mathbb{F}_q$. For $c \in 1+8\mathbb{Z}_q$, denote by $\sqrt{c}$ the unique element $d \in 1+4\mathbb{Z}_q$ with $d^2 = c$. Given $a, b \in \mathbb{Z}_q$ with $a/b \in 1+8\mathbb{Z}_q$, then $a' = (a+b)/2$ and $b' = b\sqrt{a/b}$ also belong to $\mathbb{Z}_q$ and $a'/b' \in 1+8\mathbb{Z}_q$. Furthermore, if $a, b \in 1+4\mathbb{Z}_q$, then also $a', b' \in 1+4\mathbb{Z}_q$. However, as can be seen from equation (3.12) the AGM sequence will only converge if and only if $a/b \in 1+16\mathbb{Z}_q$. For $a/b \in 1+8\mathbb{Z}_q$ the AGM sequence will not converge at all.

Let $a, b \in 1+4\mathbb{Z}_q$ with $a/b \in 1+8\mathbb{Z}_q$ and $E_{a,b}$ the elliptic curve defined by $y^2 = x(x-a^2)(x-b^2)$. Note that $E_{a,b}$ is not a minimal Weierstrass model and that its reduction modulo 2 is singular. The following lemma gives an isomorphism from $E_{a,b}$ to a minimal model. Our approach is somewhat different from the ones in Fouquet [40] and Satoh [107], so we also provide a proof.

Lemma 3.5.1 *Let $a, b \in 1+4\mathbb{Z}_q$ with $a/b \in 1+8\mathbb{Z}_q$ and $E_{a,b}$ the elliptic curve defined by $y^2 = x(x-a^2)(x-b^2)$. Then the isomorphism*

$$(x, y) \mapsto (\frac{x-ab}{4}, \frac{y-x+ab}{8})$$

transforms $E_{a,b}$ in $y^2 + xy = x^3 + rx^2 + sx + t$ with

$$\begin{aligned} r &= \frac{-a^2+3ab-b^2-1}{4}\,, \\ s &= \frac{-a^3b+2a^2b^2-ab^3}{8}\,, \\ t &= \frac{-a^4b^2+2a^3b^3-a^2b^4}{64}\,. \end{aligned} \quad (3.13)$$

Furthermore, $r \in 2\mathbb{Z}_q$, $s \in 8\mathbb{Z}_q$, $t \equiv -\left(\frac{a-b}{8}\right)^2 \bmod 16$ and the equation defines a minimal Weierstrass model.

Proof: Substituting the inverse transformation $(x, y) \mapsto (4x+ab, 8y+4x)$ in the equation of $E_{a,b}$ and dividing by 64 gives the values of r, s, t. Since $a \equiv b \bmod 8$ and $a \equiv b \equiv 1 \bmod 4$, we can write $a = 1+4u+8\alpha$ and $b = 1+4u+8\beta$, with $u, \alpha, \beta \in \mathbb{Z}_q$. Substituting this in equation (3.13) shows that $r \in 2\mathbb{Z}_q$, $s \in 8\mathbb{Z}_q$ and

$$t \equiv -(\alpha-\beta)^2 \equiv -\left(\frac{a-b}{8}\right)^2 \bmod 16\,.$$

To prove that this is a minimal Weierstrass model, it suffices to show [117, VII.1.1] that $\mathrm{ord}_2(c_4) < 4$. Since $c_4 = (1+4r)^2 - 48s$, we conclude that the model is indeed minimal. □

The next proposition shows that the AGM iteration constructs a sequence of elliptic curves all of which are 2-isogenous. This will provide the missing link between the AGM iteration and the canonical lift.

PROPOSITION 3.5.1 *Let $a, b \in 1 + 4\mathbb{Z}_q$ with $a/b \in 1 + 8\mathbb{Z}_q$ and $E_{a,b}$ the elliptic curve defined by the equation $y^2 = x(x-a^2)(x-b^2)$. Let $a' = (a+b)/2$, $b' = \sqrt{ab}$ and $E_{a',b'} : y^2 = x(x-a'^2)(x-b'^2)$, then $E_{a,b}$ and $E_{a',b'}$ are 2-isogenous. The isogeny is given by*

$$\varphi : E_{a,b} \to E_{a',b'} : (x, y) \mapsto \left(\frac{(x+ab)^2}{4x}, y\frac{(x-ab)(x+ab)}{8x^2} \right),$$

and the kernel of φ is $\langle(0,0)\rangle$. Furthermore, the action of φ on the invariant differential is

$$\varphi^* \left(\frac{dx}{y} \right) = 2\frac{dx}{y}.$$

Let $\overline{E}$ be an ordinary elliptic curve defined by $y^2 + xy = x^3 + c$ with $c \in \mathbb{F}_q^*$ and let E be its canonical lift. Take any $r \in \mathbb{Z}_q$ such that $r^2 \equiv c \bmod 2$ and let $a_0 = 1 + 4r$ and $b_0 = 1 - 4r$, then Lemma 3.5.1 shows that E_{a_0,b_0} is isomorphic to a lift of $\overline{E}$ to $\mathbb{Z}_q$ and thus $j(E_{a_0,b_0}) \equiv j(\overline{E}) \bmod 2$.

Let $(a_k, b_k)_{k=0}^{\infty}$ be the AGM sequence and consider the elliptic curves E_{a_k,b_k}. Proposition 3.5.1 implies that $\Phi_2\left(j(E_{a_k,b_k}), j(E_{a_{k+1},b_{k+1}})\right) = 0$ and an easy computation shows that $j(E_{a_{k+1},b_{k+1}}) \equiv j(E_{a_k,b_k})^2 \bmod 2$. Note that $\Phi_2(X, Y)$ and $(j(E_{a_{k+1},b_{k+1}}), j(E_{a_k,b_k}))$ satisfy the conditions of Proposition 3.4.1, so we conclude

$$j\left(E_{a_k,b_k}\right) \equiv \Sigma^k\left(j(E)\right) \bmod 2^{k+1}.$$

REMARK 3.5.1 Note that although the AGM sequence $(a_k, b_k)_{k=0}^{\infty}$ in itself does not converge, the sequence of elliptic curves $(E_{a_{nk},b_{nk}})_{k=0}^{\infty}$ does in the sense that $\lim_{k\to\infty} j(E_{a_{nk},b_{nk}})$ exists and is equal to the j-invariant of the canonical lift of $\overline{E}$.

3.5.2 Computing the Trace of Frobenius

The AGM not only provides an efficient algorithm to compute the j-invariant of the canonical lift E, but also leads to an elegant formula for the trace of Frobenius.

Assume we have computed $a, b \in 1 + 4\mathbb{Z}_q$ with $a/b \in 1 + 8\mathbb{Z}_q$ such that $j(E_{a,b}) = j(E)$. Let $(a', b') = ((a + b)/2, \sqrt{ab})$, $\varphi : E_{a,b} \to E_{a',b'}$ the AGM isogeny and $\Sigma : E_{a,b} \to E_{\Sigma(a),\Sigma(b)}$ the lift of the 2-nd power Frobenius, then we have the following diagram:

$$\begin{tikzcd} E_{a,b} \arrow[rr, "\varphi"] \arrow[dr, "\Sigma"'] & & E_{a',b'} \arrow[dl, dotted, "\lambda"] \\ & E_{\Sigma(a),\Sigma(b)} & \end{tikzcd} \tag{3.14}$$

The kernel of the Frobenius isogeny $\Sigma : E_{a,b} \to E_{\Sigma(a),\Sigma(b)}$ is a subgroup of order 2 of

$$E_{a,b}[2] = \{\mathcal{O}, (0,0), (a^2, 0), (b^2, 0)\}\,.$$

Using the isomorphism given in Lemma 3.5.1, we now analyse the reduction of the 2-torsion on a minimal model. An easy calculation shows that $(0,0)$ is mapped onto $\mathcal{O}$ whereas $(a^2, 0)$ and $(b^2, 0)$ are mapped to $(0, (a-b)/8 \bmod 2)$. Therefore we conclude that $\mathrm{Ker}\Sigma = \{\mathcal{O}, (0,0)\}$. Proposition 3.5.1 shows that $\mathrm{Ker}\Sigma = \mathrm{Ker}\varphi$ and since both isogenies are separable, there exists an isomorphism $\lambda : E_{a',b'} \to E_{\Sigma(a),\Sigma(b)}$, such that $\Sigma = \lambda \circ \varphi$. The following lemma shows that this isomorphism has a very simple form.

Proposition 3.5.2 *Given two elliptic curves $E_{a,b} : y^2 = x(x-a^2)(x-b^2)$ and $E_{c,d} : y'^2 = x'(x'-c^2)(x'-d^2)$ over $\mathbb{Q}_q$ with $a, b, c, d \in 1 + 4\mathbb{Z}_q$ and $a/b, c/d \in 1 + 8\mathbb{Z}_q$, then $E_{a,b}$ and $E_{c,d}$ are isomorphic if and only if $x' = u^2 x$ and $y' = u^3 y$ with $u^2 = \frac{c^2+d^2}{a^2+b^2}$. Furthermore, we have $\left(\frac{a}{b}\right)^2 = \left(\frac{c}{d}\right)^2$ or $\left(\frac{a}{b}\right)^2 = \left(\frac{d}{c}\right)^2$.*

Proof: Let $\lambda : E_{a,b} \to E_{c,d}$ be an isomorphism from $E_{a,b}$ to $E_{c,d}$, then λ has the general form

$$(x, y) \to (u^2 x + r, u^3 y + u^2 s x + t), \quad (u, r, s, t) \in \mathbb{Q}_q^* \times \mathbb{Q}_q^3\,,$$

where the (u, r, s, t) satisfy the equations given in [117, Table 1.2]. The first and third equation immediately lead to $s = t = 0$. The remaining equations give

$$u^2(a^2 + b^2) = c^2 + d^2 - 3r\,, \tag{3.15}$$

$$0 = r(r - c^2)(r - d^2)\,. \tag{3.16}$$

Equation (3.16) shows that the only possibilities for r are $r = 0$, $r = c^2$ or $r = d^2$. Assume that $r \neq 0$, then equation (3.15) immediately leads to a contradiction since $\mathrm{ord}_2(u^2(a^2+b^2))$ is odd, whereas $\mathrm{ord}_2(c^2+d^2-3r)$ would be zero. Therefore we conclude that $r = 0$ and $u^2 = \frac{c^2+d^2}{a^2+b^2}$. Substituting $x' = u^2 x$ and $y' = u^3 y$ in the equation of $E_{c,d}$ and dividing by u^6, we see that

$$(x - a^2)(x - b^2) = (x - \frac{c^2}{u^2})(x - \frac{d^2}{u^2})\,.$$

There are two possible ways to assign the linear factors to each other, which leads to $\left(\frac{a}{b}\right)^2 = \left(\frac{c}{d}\right)^2$ or $\left(\frac{a}{b}\right)^2 = \left(\frac{d}{c}\right)^2$. $\square$

Let ω and ω' be the invariant differentials on $E_{a,b}$ and $E_{\Sigma(a),\Sigma(b)}$ respectively, then

$$\Sigma^*(\omega') = (\lambda \circ \varphi)^*(\omega') = 2u^{-1}\omega\,,$$

with $u^2 = \frac{\Sigma(a)^2+\Sigma(b)^2}{a'^2+b'^2}$. Define $\zeta = a/b = 1+8c$ and $\zeta' = a'/b' = 1+8c'$, then Lemma 3.5.2 also implies that

$$\zeta'^2 = \Sigma(\zeta)^2 \quad \text{or} \quad \zeta'^2 = \frac{1}{\Sigma(\zeta)^2}\,.$$

Substituting $\zeta = 1+8c$ and $\zeta' = 1+8c'$ in the above equation and dividing by 16, we conclude that $c' \equiv \Sigma(c) \bmod 4$ or $c' \equiv -\Sigma(c) \bmod 4$. The Taylor expansion of $1+8c' = (1+4c)/\sqrt{1+8c}$ modulo 32 shows that $c' \equiv c^2 \bmod 4$. Since after the first iteration, c itself is a square α^2 modulo 4 and $\Sigma(\alpha^2) \equiv \alpha^4 \bmod 4$, we conclude that $\zeta'^2 = \Sigma(\zeta)^2$. Substituting $b'^2 = a'^2\Sigma(b)/\Sigma(a)^2$ in the expression for u^2 and taking square roots leads to

$$u = \pm\frac{\Sigma(a)}{a'}\,. \tag{3.17}$$

Let $(a_k, b_k)_{k=0}^{\infty}$ be the AGM sequence with $a_0 = a$ and $b_0 = b$ and consider the following diagram where $E_k = E_{\Sigma^k(a),\Sigma^k(b)}$ and $\Sigma_k : E_k \to E_{k+1}$ the lift of the 2-nd power Frobenius isogeny.

$$\begin{tikzcd}
E_{a,b} \arrow[r, "\varphi_0"] \arrow[d, "\mathrm{Id}"'] & E_{a_1,b_1} \arrow[r, "\varphi_1"] \arrow[d, "\lambda_1"'] & E_{a_2,b_2} \arrow[r, "\varphi_2"] \arrow[d, "\lambda_2"'] & \cdots \arrow[r, "\varphi_{n-1}"] & E_{a_n,b_n} \arrow[d, "\lambda_n"'] \\
E_0 \arrow[r, "\Sigma_0"] & E_1 \arrow[r, "\Sigma_1"] & E_2 \arrow[r, "\Sigma_2"] & \cdots \arrow[r, "\Sigma_{n-1}"] & E_n = E_0
\end{tikzcd} \tag{3.18}$$

Since $\mathrm{Ker}(\Sigma_k \circ \lambda_k) = \mathrm{Ker}(\varphi_k)$ for $k \in \mathbb{N}$, we can repeat the same argument as for diagram (3.14) and find an isomorphism λ_{k+1} such that $\Sigma_k = \lambda_{k+1} \circ \varphi_k \circ \lambda_k^{-1}$. Since $E_{a,b}$ is isomorphic to the canonical lift E of $\overline{E}$, we conclude that

$$\mathrm{Tr}(\Sigma_{n-1} \circ \cdots \circ \Sigma_0) = \mathrm{Tr}\,\mathcal{F} = \mathrm{Tr}\,\overline{F}\,.$$

The above diagram shows that $\Sigma_{n-1} \circ \cdots \circ \Sigma_0 = \lambda_n \circ \varphi_{n-1} \circ \cdots \circ \varphi_0$ and since φ_k acts on the invariant differential ω as multiplication by 2 and λ_n as multiplication by $\pm a_n/a_0$, we conclude that

$$\mathcal{F}^*(\omega) = \pm 2^n \frac{a_n}{a_0}(\omega)\,.$$

The Weil conjectures imply that the product of the roots of the characteristic polynomial of Frobenius is 2^n and thus

$$\mathrm{Tr}\,\mathcal{F} = \mathrm{Tr}\,\overline{F} = \pm\frac{a_0}{a_n} \pm 2^n \frac{a_n}{a_0}\,. \tag{3.19}$$

Remark 3.5.2 If the curve $\overline{E}$ is defined by the equation $y^2 + xy = x^3 + a$ with $a \in \mathbb{F}_q^*$, then $(\sqrt[4]{a}, \sqrt{a})$ is a point of order 4, which implies that $\mathrm{Tr}\overline{F} \equiv 1 \bmod 4$. Since $a_0/a_n \in 1 + 4\mathbb{Z}_q$, we can choose the correct sign in equation (3.19) and conclude that $\mathrm{Tr}\overline{F} \equiv a_0/a_n \bmod q$.

3.5.3 The AGM Algorithm

The only remaining problem is that we only have approximations to a and b and not the exact values as assumed in the previous section.

Let $\overline{E}$ be an ordinary elliptic curve defined by $y^2 + xy = x^3 + c$ with $c \in \mathbb{F}_q^*$. Take any $r \in \mathbb{Z}_q$ such that $r^2 \equiv c \bmod 2$ and let $a_0 = 1 + 4r$ and $b_0 = 1 - 4r$. In Section 3.5.1 we showed that if $(a_k, b_k)_{k=0}^{\infty}$ is the AGM sequence, then

$$j\left(E_{a_k,b_k}\right) \equiv j(E_k) \bmod 2^{k+1},$$

where E_k is the canonical lift of $\sigma^k(\overline{E})$. Expressing $j(a_k, b_k)$ as a function of a_k and b_k shows that a_k and b_k must be correct modulo 2^{k+3}. Therefore, we conclude that

$$\operatorname{Tr} \overline{F} \equiv \frac{a_{N-3}}{a_{N-3+n}} + 2^n \frac{a_{N-3+n}}{a_{N-3}} \bmod 2^N .$$

ALGORITHM 3.5.1 (AGM)

INPUT: Elliptic curve $\overline{E} : y^2 + xy = x^3 + c$ over $\mathbb{F}_{2^n}$ with $j(\overline{E}) \notin \mathbb{F}_4$.
OUTPUT: The number of points on $\overline{E}(\mathbb{F}_{2^n})$.

1. $N = \lceil \frac{n}{2} \rceil + 2$;
2. Take $r \in \mathbb{Z}_q$ with $r \equiv \sqrt{c} \bmod 2$;
3. $a = 1 + 4r$; $b = 1 - 4r$;
4. For $i = 4$ to N do
   4.1 $(a, b) \equiv \left((a + b)/2, \sqrt{ab}\right) \bmod 2^i$;
5. $a_0 = a$;
6. For $i = 0$ to $n - 1$ do
   6.1. $(a, b) \equiv \left((a + b)/2, \sqrt{ab}\right) \bmod 2^N$;
7. $t \equiv \frac{a_0}{a} \bmod 2^N$;
8. If $t^2 > 2^{n+2}$ Then $t = t - 2^N$;
9. Return $2^n + 1 - t$;

The complexity of Algorithm 3.5.1 is determined by steps 4 and 6, which require $O(n)$ square root computations to precision $O(n)$. Since each square root computation takes $O(1)$ multiplications at full precision, the complexity of Algorithm 3.5.1 is $O(n^{2\mu+1})$ bit-operations. The space complexity clearly is $O(n^2)$, since only $O(1)$ elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are required.

REMARK 3.5.3 Note that it is possible to replace step 6 of Algorithm 3.5.1 by one AGM iteration and a norm computation. Indeed, equation (3.17) shows that

$$\Sigma_0^* \left(\frac{dx}{y}\right) = \pm 2 \frac{a_1}{\Sigma(a_0)} \left(\frac{dx}{y}\right) ,$$

and since all curves E_k are conjugates, we have

$$\mathcal{F}^* \left(\frac{dx}{y}\right) = \pm 2^n \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p} \left(\frac{a_1}{\Sigma(a_0)}\right) \left(\frac{dx}{y}\right) = \pm 2^n \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p} \left(\frac{a_1}{a_0}\right) \left(\frac{dx}{y}\right) .$$

Therefore, it suffices to compute a_1 with one AGM iteration in step 6 and to set $t \equiv \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(a_0/a_1) \bmod 2^N$ in step 7. We refer to Sections 3.6.2, 3.7.2 and 3.10.3 for efficient norm computation algorithms.

3.5.4 Implementation and Numerical Results

In this section we present running times of an implementation of Algorithm 3.5.1 in the `C` programming language. The basic operations on integers modulo 2^N for $N \leq 256$ are implemented in assembly language. Elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo a degree n irreducible polynomial, which is either a trinomial or a pentanomial. Multiplication of two elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ is implemented using Karatsuba's trick [60] in the polynomial dimension and classical multiplication for the coefficients.

Table 3.4 contains running times and memory usages of the bivariate AGM algorithm for elliptic curves over various finite fields of characteristic 2. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1. The columns that are labeled 1st AGM and 2nd AGM contain the time taken to execute step 4 and step 6 in Algorithm 3.5.1.

Table 3.4: Running time (s) and memory usage (KB) of the bivariate AGM algorithm for elliptic curves over $\mathbb{F}_{2^n}$

n	1st AGM	2nd AGM	Total	Mem
120	0.07	0.22	0.29	14
144	0.12	0.44	0.57	23
168	0.21	0.67	0.88	26
192	0.32	1.06	1.39	38
240	0.64	2.16	2.81	47
288	1.20	4.22	5.44	69
336	2.08	7.18	9.28	97
384	3.29	12.1	15.4	128
480	7.32	27.1	34.5	181

3.6 Satoh-Skjernaa-Taguchi's Algorithm

The previous point counting algorithms all avoided the computation of the Frobenius substitution since this was assumed to be very costly. In [109] Satoh, Skjernaa and Taguchi (SST) proposed an alternative representation of $\mathbb{Z}_q$ that allows fast Frobenius substitutions. Let $\overline{E}$ be an ordinary elliptic curve over $\mathbb{F}_{p^n}$, with $j(\overline{E}) \notin \mathbb{F}_{p^2}$. Satoh, Skjernaa and Taguchi obtain an algorithm to compute the j-invariant $j(E) \bmod p^{O(n)}$ of the canonical lift E of $\overline{E}$ in time $O(n^{2\mu+1/(1+\mu)})$, not including precomputations, and using $O(n^2)$ memory. They also describe a fast norm computation algorithm which runs in $O(n^{2\mu+0.5})$ bit-operations and uses $O(n^2)$ memory. As a result, they obtain a point counting algorithm which requires $O(n^{2\mu+0.5})$ bit-operations and $O(n^2)$ space, after a precomputation phase which only depends on the finite field and which takes $O(n^{2\mu+1})$ bit-operations.

3.6.1 Computing the Canonical Lift

Let $\overline{E}$ be an ordinary elliptic curve over $\mathbb{F}_q$ with $q = p^n$ and $j(\overline{E}) \in \mathbb{F}_q \setminus \mathbb{F}_{p^2}$. Denote with $\mathbb{Q}_q$ the degree n unramified extension of $\mathbb{Q}_p$ with valuation ring $\mathbb{Z}_q$. In the previous sections, $\mathbb{Q}_q$ is represented as $\mathbb{Q}_p[x]/(f(x))$ with $f(x) \in \mathbb{Z}_q[x]$ a sparse, monic polynomial of degree n whose reduction modulo p is irreducible in $\mathbb{F}_p[x]$. Using this representation, reductions modulo $f(x)$ are very efficient, but Frobenius substitutions are slow.

Satoh, Skjernaa and Taguchi use an alternative representation of $\mathbb{Q}_q$ that allows fast Frobenius substitutions, but needs some precomputation to set up. Let $\overline{\theta} \in \mathbb{F}_q$ be a zero of $f(x) \bmod p$, such that $\mathbb{F}_q = \mathbb{F}_p[\overline{\theta}]$. It is clear that $\overline{\theta}$ is a $(q-1)$-th root of unity, since $\overline{\theta} \in \mathbb{F}_q$. Let θ be the Teichmüller lift of $\overline{\theta}$, i.e. the unique $(q-1)$-th root of unity in $\mathbb{Z}_q$ with $\overline{\theta} \equiv \theta \bmod p$ and let $m(x) \in \mathbb{Z}_p[x]$ be its minimal polynomial. Note that $m(x) \equiv f(x) \bmod p$, which implies that we can also represent $\mathbb{Q}_q$ as $\mathbb{Q}_p[x]/(m(x))$. Since $\Sigma(\theta) \equiv \theta^p \bmod p$ and $\Sigma(\theta)$ is a $(q-1)$-th root of unity, we conclude that $\Sigma(\theta) = \theta^p$. We can therefore compute the Frobenius substitution Σ very efficiently as

$$\Sigma\left(\sum_{i=0}^{n-1} a_i \theta^i\right) = \sum_{i=0}^{n-1} a_i \theta^{ip},$$

where we reduce the result modulo $m(x)$.

Instead of avoiding Frobenius substitutions, we can now try to solve the system of equations in Theorem 3.1.1 directly using a variant of Proposition 3.4.1. Assume we have $J \equiv j(E) \bmod p^k$ and would like to compute $j(E) = J + p^k e$. Substituting this in the p-th modular polynomial leads to

$$\begin{aligned}0 &= \Phi_p(J + p^k e, \Sigma(J) + p^k \Sigma(e)) \\ &= \Phi_p(J, \Sigma(J)) + p^k e \frac{\partial \Phi_p}{\partial X}(J, \Sigma(J)) + p^k \Sigma(e) \frac{\partial \Phi_p}{\partial Y}(J, \Sigma(J)) + p^{2k} r(e, \Sigma(e)),\end{aligned}$$

with $r \in \mathbb{Z}_q[X,Y]$. Since $\Phi_p(J, \Sigma(J)) \equiv 0 \bmod p^k$, we can divide this equation by p^k and obtain a relation for e modulo p. The Kronecker relation (3.4) implies that $\frac{\partial \Phi_p}{\partial X}(J, \Sigma(J)) \equiv 0 \bmod p$ and that $\frac{\partial \Phi_p}{\partial Y}(J, \Sigma(J)) \not\equiv 0 \bmod p$. Finally, since $\Sigma(e) \equiv e^p \bmod p$ we get

$$e^p \equiv -\frac{\Phi_p(J, \Sigma(J))}{p^k \frac{\partial \Phi_p}{\partial Y}(J, \Sigma(J))} \bmod p\,.$$

Taking the unique p-th root $e' \in \mathbb{F}_q$ leads to a better approximation of $j(E)$ given by $J + p^k e' \equiv j(E) \bmod p^{k+1}$.

To avoid computing the p-th root in the above equation, Satoh, Skjernaa and Taguchi replace the system of equations in Theorem 3.1.1 by

$$\Phi_p(\Sigma^{-1}(X), X) = 0 \quad \text{and} \quad X \equiv j(\overline{E}) \bmod p\,.$$

It is clear that the solution to the above equations also gives the j-invariant of the canonical lift E. Substituting $J + p^k e$ in these equations leads to

$$e \equiv -\frac{\Phi_p(\Sigma^{-1}(J), J)}{p^k \frac{\partial \Phi_p}{\partial Y}(\Sigma^{-1}(J), J)} \bmod p\,.$$

Note that since $\Phi_p(\Sigma^{-1}(J), J) \equiv 0 \bmod p^k$, we only need to compute the inverse of $\frac{\partial \Phi_p}{\partial Y}(\Sigma^{-1}(J), J)$ modulo p. The only remaining problem is computing Σ^{-1}, but this can be done efficiently as

$$\Sigma^{-1}\left(\sum_{i=0}^{n-1} a_i \theta^i\right) = \sum_{j=0}^{p-1}\left(\sum_{0 \le pk+j < n} a_{pk+j}\theta^k\right) C_j(\theta),$$

where $C_j(\theta) = \Sigma^{-1}(\theta^j) = \theta^{jp^{n-1}}$. If we precompute $C_j(\theta)$ for $j = 0, \ldots, p-1$, computing $\Sigma^{-1}(z)$ for $z \in \mathbb{Z}_q/(p^N \mathbb{Z}_q)$ only takes $p-1$ multiplications over $\mathbb{Z}_q/(p^N \mathbb{Z}_q)$. Implementing these ideas naively leads to Algorithm 3.6.1, which can replace Algorithm 3.4.1.

The main problem of Algorithm 3.6.1 is that in each step it recomputes $\Phi_p(x,y)$ although the values of x and y in step $i+1$ are very close to the ones in step i. Assume we have computed $y \equiv j(E) \bmod p^W$ and $x \equiv \Sigma^{-1}(y) \bmod p^W$ for some W. Observe that for $m \ge 1$ and $i \ge 0$

$$\begin{aligned}&\Phi_p(x + p^{mW+i}\Delta_x, y + p^{mW+i}\Delta_y)\\ &\equiv \Phi_p(x,y) + p^{mW+i}\left(\frac{\partial \Phi_p}{\partial X}(x,y)\Delta_x + \frac{\partial \Phi_p}{\partial Y}(x,y)\Delta_y\right) \bmod p^{(m+1)W}\,, \quad (3.20)\end{aligned}$$

and that it is sufficient to know $\frac{\partial \Phi_p}{\partial X}(x,y)$ and $\frac{\partial \Phi_p}{\partial Y}(x,y)$ modulo p^W. Note that we can use the above equation to compute $\Phi_p(x + p^{mW+i}\Delta_x, y + p^{mW+i}\Delta_y)$ from $\Phi_p(x,y)$ as long as $i < W$. Algorithm 3.6.2 thus consists of two stages: in the first stage we compute $y \equiv j(E) \bmod p^W$ using Algorithm 3.6.1 and in the second stage we use equation (3.20) to update $\Phi_p(x,y)$.

ALGORITHM 3.6.1 (LIFT_J_INVARIANT_NAIVE)

INPUT: A j-invariant $j \in \mathbb{F}_{p^n} \setminus \mathbb{F}_{p^2}$ and a precision N.
OUTPUT: $J \in \mathbb{Z}_q$ with $J \equiv j \bmod p$ and $\Phi_p(\Sigma^{-1}(J), J) \equiv 0 \bmod p^N$.

1. $d \equiv \left(\frac{\partial \Phi_p}{\partial Y}(\Sigma^{-1}(j), j)\right)^{-1} \bmod p$;
2. $y \equiv j \bmod p$;
3. For $i = 2$ To N Do
   3.1. $x \equiv \Sigma^{-1}(y) \bmod p^i$;
   3.2. $y \equiv y - d\Phi_p(x, y) \bmod p^i$;
4. Return y.

Satoh, Skjernaa and Taguchi [109, Theorem 2.2] prove that for $W \simeq n^{\mu/(1+\mu)}$ and $N \simeq n/2$, Algorithm 3.6.2 runs in time $O(n^{2\mu+1/(1+\mu)})$. Although the value for W minimizes the asymptotic running time, in practice it is faster to take W a multiple of the CPU word size. The above complexity does not include the precomputation of the field polynomial $m(x)$ and the elements $C_j(\theta)$ for $j = 0, \dots, p-1$, which can be obtained in time $O(n^{2\mu+1})$ as shown by Satoh [106]. We do not elaborate on this method since $m(x)$ needs to be computed only once for each field. Furthermore, in Section 3.10.2 we describe an algorithm to compute $m(x)$ in $O(n^{2\mu} \log n)$ bit-operations.

To devise a complete point counting algorithm, one could proceed in the same way as in Algorithm 3.4.4. However, as observed in Remark 3.1.1 we can recover the trace of Frobenius as

$$t \equiv \prod_{i=0}^{n-1} \Sigma^i(c_0) \equiv \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(c_0) \bmod q\,.$$

3.6.2 Norm Computation I

In the same paper [109], Satoh, Skjernaa and Taguchi also describe a fast norm computation based on an analytic method. Let $\exp(x) = \sum_{i=0}^{\infty} x^i/i!$ be the p-adic exponential function and $\log(x) = \sum_{i=1}^{\infty} (-1)^{i-1}(x-1)^i/i$ the p-adic logarithmic function. Note that $\exp(x)$ converges for $\mathrm{ord}_p(x) > 1/(p-1)$ and $\log(x)$ converges for $\mathrm{ord}_p(x-1) > 0$.

Assume first that a is close to unity, i.e. $\mathrm{ord}_p(a-1) > 1/(p-1)$, then

$$\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(a) = \exp\left(\mathrm{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\log(a))\right)\,,$$

since Σ is continuous and both series converge. The dominant step in the above formula is the evaluation of the logarithm. Using a naive algorithm, this would take $O(N)$ multiplications over $\mathbb{Z}_q/(p^N\mathbb{Z}_q)$ or $O(n^{2\mu+1})$ bit-operations.

ALGORITHM 3.6.2 (LIFT_J_INVARIANT)

`INPUT:` A j-invariant $j \in \mathbb{F}_{p^n} \setminus \mathbb{F}_{p^2}$ and a precision N.
`OUTPUT:` $J \in \mathbb{Z}_q$ with $J \equiv j \bmod p$ and $\Phi_p(\Sigma^{-1}(J), J) \equiv 0 \bmod p^N$.

1. $d \equiv \left(\frac{\partial \Phi_p}{\partial Y}(\Sigma^{-1}(j), j)\right)^{-1} \bmod p$;
2. $y \equiv j \bmod p$;
3. For $i = 2$ To W Do
   - 3.1. $x \equiv \Sigma^{-1}(y) \bmod p^i$;
   - 3.2. $y \equiv y - d\Phi_p(x, y) \bmod p^i$;
4. $x \equiv \Sigma^{-1}(y) \bmod p^W$;
5. $D_x \equiv \frac{\partial \Phi_p}{\partial X}(x, y) \bmod p^W$;
6. $D_y \equiv \frac{\partial \Phi_p}{\partial Y}(x, y) \bmod p^W$;
7. For $m = 1$ To $\lfloor (N-1)/W \rfloor$ Do
   - 7.1. $x \equiv \Sigma^{-1}(y) \bmod p^{(m+1)W}$;
   - 7.2. $V \equiv \Phi_p(x, y) \bmod p^{(m+1)W}$;
   - 7.3. For $i = 0$ To $W - 1$ Do
     - 7.3.1. $\Delta_y \equiv -dp^{-(mW+i)}V \bmod p$;
     - 7.3.2. $\Delta_x \equiv \Sigma^{-1}(\Delta_y) \bmod p^{W-i}$;
     - 7.3.3. $y \equiv y + p^{mW+i}\Delta_y \bmod p^{(m+1)W}$;
     - 7.3.4. $V \equiv V + p^{(mW+i)}(D_x\Delta_x + D_y\Delta_y) \bmod p^{(m+1)W}$;
8. Return y.

Satoh, Skjernaa and Taguchi solve this problem by noting that a^{p^m} for $m \in \mathbb{N}$ is very close to the unity, i.e. $\mathrm{ord}_p(a^{p^m} - 1) > m + \frac{1}{p-1}$. If $a \in \mathbb{Z}_q/(p^N\mathbb{Z}_q)$, then a^{p^m} is well defined in $\mathbb{Z}_q/(p^{N+m}\mathbb{Z}_q)$ and it can be computed with $O(m)$ multiplications in $\mathbb{Z}_q/(p^{N+m}\mathbb{Z}_q)$. Furthermore, note that

$$\log(a) \equiv p^{-m}(\log(a^{p^m}) \bmod p^{N+m}) \bmod p^N$$

and that $\log(a^{p^m}) \bmod p^{N+m}$ can be computed with $O(N/m)$ multiplications over $\mathbb{Z}_q/(p^{N+m}\mathbb{Z}_q)$. So if we take $m \simeq \sqrt{N}$, then $\log(a) \bmod p^N$ can be computed in $O(n^{2\mu+0.5})$ bit-operations.

In characteristic 2, we can use an extra trick. Since $\mathrm{ord}_2(a-1) > 1$, we have $a \equiv 1 \bmod 2^\nu$ for $\nu \geq 2$. Let $z = a - 1 \in 2^\nu\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ and define $\gamma = \frac{z}{2+z} \in 2^{\nu-1}\mathbb{Z}_q/(2^{N-1}\mathbb{Z}_q)$, then $a = 1 + z = \frac{1+\gamma}{1-\gamma}$ and thus

$$\log(a) = \log(1+z) = 2\sum_{j=1}^{\infty} \frac{\gamma^{2j-1}}{2j-1} .$$

Note that all the denominators in the above formula are odd. Reducing this equation modulo 2^N therefore leads to

$$\log(a) \equiv \log(1+z) \equiv 2 \sum_{1 \leq (\nu-1)(2j-1) < N-1} \frac{\gamma^{2j-1}}{2j-1} \bmod 2^N .$$

Define $a_i \in \mathbb{Z}$ by $\log(a) \equiv \sum_{i=0}^{n-1} a_i\theta^i \bmod p^N$, then

$$\mathrm{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\log(a)) \equiv \sum_{i=0}^{n-1} a_i \mathrm{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\theta^i) \bmod p^N$$

and each $\mathrm{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\theta^i)$ for $i = 0, \ldots, n-1$ can be precomputed using Newton's formula:

$$\mathrm{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\theta^i) + \sum_{j=1}^{i-1} \mathrm{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\theta^{i-j})m_{n-j} + i m_{n-i} \equiv 0 \bmod p^N ,$$

with $m(x) = \sum_{i=0}^{n} m_i x^i$ the minimal polynomial of θ.

Let $t_a = \mathrm{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\log(a)) \in \mathbb{Z}_p$, then $\mathrm{ord}_p(t_a) > 1/(p-1)$, so $t_a \in p\mathbb{Z}_p$ for $p \geq 3$ and $t_a \in 4\mathbb{Z}_2$ for $p = 2$. If we precompute $\exp(p) \bmod p^N$ for $p \geq 3$ or $\exp(4) \bmod 2^N$ for $p = 2$, then

$$\begin{aligned} \exp(t_a) &\equiv \exp(p)^{t_a/p} \bmod p^N \quad \text{for } p \geq 3, \\ \exp(t_a) &\equiv \exp(4)^{t_a/4} \bmod 2^N \quad \text{for } p = 2, \end{aligned}$$

and we can use a simple square and multiply algorithm to compute $\exp(t_a)$. Therefore, if a is close to unity then $\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(a)$ can be computed in $O(n^{2\mu+0.5})$ bit-operations and $O(n^2)$ space. Algorithm 3.6.3 computes the norm of an

---

ALGORITHM 3.6.3 (NORM_CHAR_2)

```
INPUT:   An element a ∈ 1 + 2^ν Z_q with ν ≥ 2 and a precision N.
OUTPUT:  The norm N_{Q_q/Q_p}(a) mod 2^N.

  1.  s = ⌊√N/2⌋;
  2.  z ≡ a^(2^s) − 1 mod 2^(N+s);
  3.  w ≡ log(1 + z) mod 2^(N+s);
  4.  w ≡ 2^(−s) w mod 2^N;
  5.  u ≡ 2^(−ν) Tr_{Q_q/Q_p}(w) mod 2^(N−ν);
  8.  Return exp(4)^u mod 2^N.
```

---

element in $1 + 2^\nu \mathbb{Z}_q$, with $q = 2^n$, assuming that $\exp(4)$ and $\mathrm{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\theta^i)$ for $i = 0, \ldots, n-1$ are precomputed.

Note that for $p = 2$, $c_i^2 \equiv 1 \bmod 8$ by equation (3.9) or (3.6), but that for $p \geq 3$ this is no longer valid. So for odd p, we need to consider the more general situation where a is not close to unity. Define $\overline{\alpha} \in \mathbb{F}_q$ by $\overline{\alpha} \equiv a \bmod p$ and let $\alpha \in \mathbb{Z}_q$ be the Teichmüller lift of $\overline{\alpha}$. Then

$$\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(a) = \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha)\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha^{-1}a)$$

and $\mathrm{ord}_p(\alpha^{-1}a - 1) \geq 1 > 1/(p-1)$ since p is odd. Furthermore, note that $\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha)$ is equal to the Teichmüller lift of $\mathrm{N}_{\mathbb{F}_q/\mathbb{F}_p}(\overline{\alpha})$. The main problem therefore is computing the Teichmüller lift of $\overline{\alpha}$. Using a Newton iteration on $X^{1-q} - 1 = 0$, $\alpha \bmod p^N$ can be computed in $O(n^{2\mu+1})$ bit-operations and $O(n^2)$ space. However, Satoh [107] pointed out that the Teichmüller lift α satisfies

$$\alpha = \Sigma^{-1}(\alpha^p) \quad \text{and} \quad \alpha \equiv \overline{\alpha} \bmod p\,,$$

so we can apply the same trick as in Algorithm 3.6.2. Assume we have computed $x \equiv \alpha \bmod p^W$, then for $m > 0$ and $i \geq 0$

$$\begin{aligned} x + p^{mW+i}\Delta_x - \Sigma^{-1}\left((x + p^{mW+i}\Delta_x)^p\right) \\ \equiv x - \Sigma^{-1}(x^p) + p^{mW+i}\left(\Delta_x - p\Sigma^{-1}(x^{p-1}\Delta_x)\right) \bmod p^{(m+1)W}\,, \end{aligned}$$

and we only need to compute $\Delta_x - p\Sigma^{-1}(x^{p-1}\Delta_x)$ modulo p^{W-i}.

For $W \simeq n^{\mu/(1+\mu)}$ and $N \simeq n/2$, Algorithm 3.6.4 runs in $O(n^{2\mu+1/(1+\mu)})$ bit-operations, not including precomputation, and needs $O(n^2)$ space.

3.6.3 An $O(n^{2.5+\varepsilon})$ Time Algorithm

Combining the algorithms of the last two subsections leads to Algorithm 3.6.5. The time complexity is determined by the complexity of the norm computation.

ALGORITHM 3.6.4 (TEICHMULLER_LIFT)

INPUT: A non-zero element $\overline{\alpha} \in \mathbb{F}_{p^n}$ and a precision N.
OUTPUT: The Teichmüller lift $\alpha \bmod p^N$.

1. $x \equiv \overline{\alpha} \bmod p$;
2. For $i = 2$ To W Do
   - 2.1. $x \equiv \Sigma^{-1}(x^p) \bmod p^i$;
3. $D_x \equiv p\Sigma^{-1}(x^{p-1}) \bmod p^W$;
7. For $m = 1$ To $\lfloor (N-1)/W \rfloor$ Do
   - 7.2. $V \equiv x - \Sigma^{-1}(x^p) \bmod p^{(m+1)W}$;
   - 7.3. For $i = 0$ To $W - 1$ Do
     - 7.3.1. $\Delta_x \equiv (-p^{-(mW+i)}V) \bmod p$;
     - 7.3.2. $\Delta_V \equiv \Delta_x - D_x\Sigma^{-1}(\Delta_x) \bmod p^{W-i}$;
     - 7.3.3. $x \equiv x + p^{(mW+i)}\Delta_x \bmod p^{(m+1)W}$;
     - 7.3.4. $V \equiv V + p^{(mW+i)}\Delta_V \bmod p^{(m+1)W}$;
8. Return x.

THEOREM 3.6.1 (SATOH-SKJERNAA-TAGUCHI) *Let $\overline{E}$ be an elliptic curve over a finite field $\mathbb{F}_q$ with $q = p^n$ and $j(\overline{E}) \notin \mathbb{F}_{p^2}$. Then there exists a deterministic algorithm for computing the number of points $\#\overline{E}(\mathbb{F}_q)$ which requires, for p fixed, $O(n^{2.5} \log n \log\log n)$ bit-operations and $O(n^2)$ memory, after a precomputation phase which only depends on the finite field and takes $O(n^3 \log n \log\log n)$ bit-operations.*

ALGORITHM 3.6.5 (SATOH_SKJERNAA_TAGUCHI)

INPUT: Elliptic curve $\overline{E} : y^2 + xy = x^3 + a$ over $\mathbb{F}_{2^n}$ with $j(\overline{E}) \notin \mathbb{F}_4$.
OUTPUT: The number of points on $\overline{E}(\mathbb{F}_{2^n})$.

1. $N = \lceil \frac{n}{2} \rceil + 13$; $N' = N - 10$;
2. $J =$ `Lift_J_Invariant` $(j(\overline{E}), N)$; $J' = \Sigma^{-1}(J) \mod p^N$;
3. $Z \equiv -\frac{J'^2+195120J'+4095J+660960000}{8(J'^2+J'(563760-512J)+372735J'+8981280000)} \mod p^{N'}$;
4. $V \equiv (12Z^2 + Z)(J - 1728) - 36 \mod p^{N'}$;
5. $S =$ `Norm_Char_2` $(\frac{(J-(504+12096Z)V)}{240V+J}, N')$;
6. $c \equiv$ `Sqrt`$(S, N') \mod 2^{N'-1}$;
7. If $c \not\equiv 1 \mod 4$ Then $c \equiv -c \mod 2^{N'-1}$;
8. If $c^2 > 2^{n+2}$ Then $c = c - 2^{N'-1}$;
9. Return $2^n + 1 - c$;

3.6.4 Implementation and Numerical Results

In this section we present running times of an implementation of Algorithm 3.6.5 in the `C` programming language. The basic operations on integers modulo 2^N for $N \leq 256$ are implemented in assembly language. Elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo the minimal polynomial of a $(q-1)$-th root of unity. Multiplication of two elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ is implemented using Karatsuba's trick [60] in the polynomial dimension and classical multiplication for the coefficients. Since the minimal polynomial is dense we need to use a fast reduction algorithm such as the one described in the book by von zur Gathen and Gerhard [134, Algorithm 9.5].

Table 3.5 contains running times and memory usages of Satoh-Skjernaa-Taguchi algorithm for elliptic curves over various finite fields of characteristic 2. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1.

Table 3.5: Running time (s) and memory usage (KB) of Satoh-Skjernaa-Taguchi algorithm for elliptic curves over $\mathbb{F}_{2^n}$

n	Lift	Norm	Total	Mem
120	0.09	0.015	0.12	41
144	0.10	0.017	0.13	48
168	0.21	0.023	0.26	75
192	0.22	0.038	0.29	82
240	0.48	0.105	0.65	131
288	0.54	0.107	0.72	150
336	0.86	0.185	1.17	208
384	1.29	0.295	1.76	276
480	2.57	0.610	3.56	393

3.7 Kim et al.'s Algorithm

To speed up the Frobenius substitution, Kim, Park, Cheon, Park, Kim and Hahn [63] proposed to use finite fields with a Gaussian Normal Basis (GNB) of small type. Such a basis can be lifted trivially to $\mathbb{Z}_q$ and allows efficient computation of arbitrary iterates of Frobenius. Combined with a norm computation algorithm suggested by Kedlaya [61], they obtained a point counting algorithm for elliptic curves over $\mathbb{F}_{p^n}$ with Gaussian Normal Basis which requires $O(n^{2\mu+1/(1+\mu)})$ bit-operations and $O(n^2)$ space, without precomputation.

3.7.1 Gaussian Normal Basis

Let $\mathbb{L}/\mathbb{K}$ be a finite Galois extension of degree n. If there exists an element $\alpha \in \mathbb{L}$ such that the set $B = \{\sigma(\alpha) \,|\, \sigma \in \mathrm{Gal}(\mathbb{L}/\mathbb{K})\}$ is a basis of $\mathbb{L}$ over $\mathbb{K}$, then B is called a normal basis. The element α is called a normal element. A proof of the following proposition for finite fields can be found in Menezes et al. [80] and its generalization to p-adic fields in Kim et al. [63].

Proposition 3.7.1 *Let q be a prime power and n, t positive integers such that $nt+1$ is a prime not dividing q. Let γ be a primitive $(nt+1)$-th root of unity in some extension field of $\mathbb{Q}_q$. If $\gcd(nt/e, n) = 1$, with e the order of q modulo $nt+1$, then for any primitive t-th root of unity τ in $\mathbb{Z}/(nt+1)\mathbb{Z}$*

$$\alpha = \sum_{i=0}^{t-1} \gamma^{\tau^i} \tag{3.21}$$

is a normal element and $[\mathbb{Q}_q(\alpha) : \mathbb{Q}_q] = n$. Such a basis is called a Gaussian Normal Basis of type t.

Kim et al. represent elements of $\mathbb{Z}_q$ as elements of the ring

$$\mathbb{Z}_p[x]/(x^{nt+1} - 1)\,.$$

Multiplication of two elements in $\mathbb{Z}_q/(p^N\mathbb{Z}_q)$ therefore requires $O((nNt)^\mu)$ bit-operations, so one normally restricts $t \leq 2$.

For $t = 1$ we have $\alpha = \gamma$ and the minimal polynomial of α therefore is

$$f(x) = \frac{x^{n+1} - 1}{x - 1} = x^n + x^{n-1} + \cdots + x + 1 .$$

To speed up modular multiplication we use the fact that $f(x) | (x^{n+1} - 1)$. Let $a, b \in \mathbb{Z}_q$ be represented as elements of $\mathbb{Z}_p[x]/(f(x))$, then clearly

$$ab \bmod f(x) \equiv (ab \bmod x^{n+1} - 1) \bmod f(x) . \tag{3.22}$$

Since the reduction modulo $x^{n+1} - 1$ is simply an addition and the final reduction modulo $f(x)$ a subtraction, we obtain a linear time modular reduction.

To speed up the Frobenius substitution, Kim et al. use a redundant representation, i.e. they embed $\mathbb{Z}_q$ in $\mathbb{Z}_p[x]/(x^{n+1} - 1)$ by mapping $a = \sum_{i=0}^{n-1} a_i \alpha^i$ to $a(x) = \sum_{i=0}^{n-1} a_i x^i + 0x^n$. Since $\Sigma^k(\alpha) = \alpha^{p^k}$, we have

$$\Sigma^k(a(x)) = \sum_{i=0}^{n} a_i x^{ip^k} = a_0 + \sum_{j=1}^{n} a_{j/p^k \bmod (n+1)} x^j . \tag{3.23}$$

So we can compute $\Sigma^k(a)$ by a simple permutation of the coefficients of $a(x)$, which only requires $O(n)$ bit-operations.

3.7.2 Norm Computation II

Kedlaya [61] suggested a square and multiply approach to compute the norm

$$\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(M) = \Sigma^{n-1}(M) \cdots \Sigma(M) M$$

of a matrix $M \in \mathbb{Q}_q^{d \times d}$. For $d = 1$ this reduces to the usual norm on $\mathbb{Q}_q$. The main idea is to compute

$$M_{i+1} = \Sigma^{2^i}(M_i)\, M_i \quad \text{for} \quad i = 0, \ldots, \lfloor \log_2 n \rfloor , \tag{3.24}$$

with $M_0 = M$ and to combine these to recover $\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(M)$. Let $n = \sum_{i=0}^{l} n_i 2^i$, with $n_i \in \{0, 1\}$ and $n_l = 1$, then we can write

$$\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(M) = \prod_{i=0}^{l} \Sigma^{2^{i+1} + 2^{i+2} + \cdots + 2^l}(M_i^{n_i}) , \tag{3.25}$$

where the sum $2^{i+1} + \cdots 2^l$ is zero for $i \geq l$ and the product is taken from left to right. This formula immediately leads to Algorithm 3.7.1 which differs from Kim et al.'s algorithm since it only uses repeated Frobenius substitutions of the form Σ^{2^i} for $i = 0, \ldots, l$. This algorithm clearly requires $O(\log n)$ repeated Frobenius substitutions and $O(\log n)$ multiplications in $\mathbb{Z}_q/(p^N\mathbb{Z}_q)$, so the time complexity amounts to $O(n^\mu N^\mu \log n)$ bit-operations and the space complexity is $O(nN)$.

Combining this fast norm computation algorithm with the SST algorithm of the previous section proves the following theorem.

ALGORITHM 3.7.1 (NORM_GNB)

INPUT: An element $a \in \mathbb{Z}_q$ with $q = p^n$ and a precision N.
OUTPUT: The norm $\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(a) \bmod p^N$.

1. $i = n$; $j = 0$; $r = 1$; $s = a$;
2. While $i > 0$ Do
   - 2.1. If $i \equiv 1 \bmod 2$ Then $r \equiv \Sigma^{2^j}(r) s \bmod p^N$;
   - 2.2. If $i > 1$ Then $s = \Sigma^{2^j}(s) s \bmod p^N$;
   - 2.3. $j = j + 1$; $i = \lfloor i/2 \rfloor$;
3. Return r.

THEOREM 3.7.1 (KIM ET AL.) *Let $\overline{E}$ be an elliptic curve over a finite field $\mathbb{F}_q$ with $q = p^n$ and $j(\overline{E}) \notin \mathbb{F}_{p^2}$. Assume that $\mathbb{F}_q$ has a Gaussian Normal Basis of small type t, then there exists a deterministic algorithm for computing the number of points $\#\overline{E}(\mathbb{F}_q)$ which requires, for p fixed, $O(n^{2\mu+1/(1+\mu)})$ bit-operations and $O(n^2)$ memory.*

3.7.3 Implementation and Numerical Results

In this section we present running times of an implementation of Kim et al.'s algorithm for GNB of type 1 in the C programming language. The basic operations on integers modulo 2^N for $N \leq 256$ are implemented in assembly language. Elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo the irreducible polynomial $(x^{n+1} - 1)/(x - 1)$. Multiplication of two elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ is implemented using Karatsuba's trick [60] in the polynomial dimension and classical multiplication for the coefficients.

Table 3.6 contains running times and memory usages of Kim et al.'s algorithm for elliptic curves over various finite fields of characteristic 2 with GNB of type 1. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1.

3.8 Gaudry's MSST Algorithm

At first sight, it seems difficult to combine the AGM and the SST algorithm. However, Gaudry [48] devised a simplified modular polynomial $\Upsilon_2(x, y)$ based on the AGM iteration and then used the SST algorithm to find a solution of $\Upsilon_2(x, \Sigma(x)) = 0$. The time and space complexities are those of the SST algorithm, i.e. $O(n^{2\mu+0.5})$ bit-operations, after a $O(n^{2\mu+1})$ precomputation phase which only depends on the finite field, and $O(n^2)$ space respectively. However,

Table 3.6: Running time (s) and memory usage (KB) of Kim et al.'s algorithm for elliptic curves over $\mathbb{F}_{2^n}$ with GNB of type 1

n	Lift	Norm	Total	Mem
130	0.028	0.002	0.031	38
148	0.037	0.003	0.040	44
172	0.058	0.004	0.062	66
196	0.074	0.007	0.081	75
268	0.161	0.017	0.178	127
292	0.186	0.018	0.206	139
346	0.306	0.035	0.341	196
388	0.448	0.047	0.496	256
466	0.782	0.094	0.872	349

the algorithm is much faster since $\Upsilon_2(x, y)$ has lower degree than $\Phi_2(x, y)$.

3.8.1 Modified Modular Polynomial

Given the bivariate AGM sequence $(a_k, b_k)_{k=0}^{\infty}$ with $a_k \equiv b_k \equiv 1 \bmod 4$ and $a_k \equiv b_k \bmod 8$, we can easily define a univariate AGM sequence $(\lambda_k)_{k=0}^{\infty}$ by $\lambda_k = b_k/a_k$, which corresponds to the curves

$$E_{\lambda_k} : y^2 = x(x-1)(x-\lambda_k^2)\,.$$

Since $(a_{k+1}, b_{k+1}) = ((a_k + b_k)/2, \sqrt{a_k b_k})$, we can compute λ_{k+1} from λ_k by

$$\lambda_{k+1} = \frac{2\sqrt{\lambda_k}}{1+\lambda_k}\,. \tag{3.26}$$

In Section 3.5.1 we showed that for an ordinary curve $\overline{E} : y^2 + xy = x^3 + \overline{c}$ with $\overline{c} \in \mathbb{F}_q^*$, the AGM sequence can be initialized as $(a_0, b_0) = (1+4u, 1-4u)$ where $u \in \mathbb{Z}_q$ satisfies $u^2 \equiv c \bmod 2$. However, we cannot use these values to initialize the univariate AGM sequence, since (a_0, b_0) is only correct modulo 8, which would lead to $\lambda_0 \equiv 1 \bmod 8$. However, this problem can be easily solved by computing $(a_1, b_1) \equiv (1, 1-8u^2) \bmod 16$ and starting from

$$\lambda_1 \equiv 1 + 8u^2 \equiv 1 + 8c \bmod 16\,,$$

with $c \in \mathbb{Z}_q$ and $c \equiv \overline{c} \bmod 2$. Unlike the bivariate AGM sequence, the univariate AGM sequence does converge, in the sense that $\lambda_k \equiv \lambda_{k+n} \bmod 2^{k+3}$. In Section 3.5.2 we proved that

$$\frac{b_{k+1}}{a_{k+1}} \equiv \Sigma(\frac{b_k}{a_k}) \bmod 2^{k+3}\,,$$

and thus $\lambda_{k+1} \equiv \Sigma(\lambda_k) \bmod 2^{k+3}$. Substituting this in equation (3.26) shows that λ_k satisfies

$$\Sigma(Z)^2(1+Z)^2 - 4Z \equiv 0 \bmod 2^{k+3} \quad \text{and} \quad Z \equiv 1 + 8\Sigma^{k-1}(c) \bmod 16\,.$$

Let $\Lambda_2(X,Y) = Y^2(1+X)^2 - 4X$, then λ_k satisfies $\Lambda_2(X, \Sigma(X)) \equiv 0 \bmod 2^{k+3}$. Note that both partial derivatives vanish modulo 2, so we cannot directly use the SST algorithm of Section 3.6 to compute a solution to this equation. Therefore, we make the change of variables $X \leftarrow 1 + 8X$ and $Y \leftarrow 1 + 8Y$ to obtain the modified modular polynomial

$$\Upsilon_2(X,Y) = (X + 2Y + 8XY)^2 + Y + 4XY\,,$$

Let γ_k be defined by $\lambda_k = 1 + 8\gamma_k$, then γ_k satisfies $\Upsilon_2(X, \Sigma(X)) \equiv 0 \bmod 2^k$ and $\gamma_k \equiv \sigma^{k-1}(c) \bmod 2$. The partial derivatives of Υ_2 are

$$\begin{aligned}
\frac{\partial \Upsilon_2}{\partial X}(X,Y) &= 2(X + 2Y + 8XY)(1 + 8Y) + 4Y\,, \\
\frac{\partial \Upsilon_2}{\partial Y}(X,Y) &= (4(X + 2Y + 8XY) + 1)(1 + 4X)\,,
\end{aligned}$$

which shows that $\frac{\partial \Upsilon_2}{\partial X} \equiv 0 \bmod 2$, but $\frac{\partial \Upsilon_2}{\partial Y} \equiv 1 \bmod 2$.

3.8.2 The MSST Algorithm

In Section 3.5.3 we showed that given the bivariate AGM sequence $(a_k, b_k)_{k=0}^{\infty}$ we can compute the trace of Frobenius as

$$\operatorname{Tr} \overline{F} \equiv t_k + \frac{q}{t_k} \bmod 2^{k+3}\,,$$

with $t_k = \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(a_k/a_{k+1})$. Substituting $a_{k+1} = (a_k + b_k)/2$, $\lambda_k = b_k/a_k$ and $\lambda_k = 1 + 8\gamma_k$ in the expression for t_k, we conclude that

$$t_k = \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}\left(\frac{1}{1 + 4\gamma_k}\right).$$

Combining the above formula for the trace of Frobenius with the SST algorithm to compute γ_k finally leads to Algorithm 3.8.1.

Algorithm 3.8.1 requires $O(n^{2\mu+0.5})$ bit-operations, after an $O(n^{2\mu+1})$ pre-computation phase and requires $O(n^2)$ space. However, the constants hidden in the big-O notation are smaller than for the SST algorithm, since $\Upsilon_2(X,Y)$ can be evaluated with 1 multiplication and 1 squaring, whereas the evaluation of $\Phi_2(X,Y)$ needs 3 multiplications and 2 squarings.

3.8.3 Implementation and Numerical Results

In this section we present running times of an implementation of Algorithm 3.8.1 in the `C` programming language. The basic operations on integers modulo 2^N

ALGORITHM 3.8.1 (MSST)

INPUT: Elliptic curve $\overline{E}: y^2 + xy = x^3 + c$ over $\mathbb{F}_{2^n}$ with $j(\overline{E}) \notin \mathbb{F}_4$.
OUTPUT: The number of points on $\overline{E}(\mathbb{F}_{2^n})$.

1. $N = \lceil \frac{n}{2} \rceil + 2$;
2. $y \equiv c \mod 2$;
3. For $i = 2$ To W Do
   - 3.1. $x \equiv \Sigma^{-1}(y) \mod 2^i$;
   - 3.2. $y \equiv y - \Upsilon_2(x, y) \mod 2^i$;
4. $x \equiv \Sigma^{-1}(y) \mod 2^W$;
5. $D_x \equiv \frac{\partial \Upsilon_2}{\partial X}(x, y) \mod 2^W$; $D_y \equiv \frac{\partial \Phi_2}{\partial Y}(x, y) \mod 2^W$;
6. For $m = 1$ To $\lfloor (N-1)/W \rfloor$ Do
   - 6.1. $x \equiv \Sigma^{-1}(y) \mod 2^{(m+1)W}$;
   - 6.2. $V \equiv \Upsilon_2(x, y) \mod 2^{(m+1)W}$;
   - 6.3. For $i = 0$ To $W - 1$ Do
     - 6.3.1. $\Delta_y \equiv -2^{-(mW+i)} V \mod 2$;
     - 6.3.2. $\Delta_x \equiv \Sigma^{-1}(\Delta_y) \mod 2^{W-i}$;
     - 6.3.3. $y \equiv y + 2^{mW+i} \Delta_y \mod 2^{(m+1)W}$;
     - 6.3.4. $V \equiv V + 2^{(mW+i)}(D_x \Delta_x + D_y \Delta_y) \mod 2^{(m+1)W}$;
7. $t =$ Norm_Char_2 $(\frac{1}{1+4y}, N)$;
8. If $t^2 > 2^{n+2}$ Then $t = t - 2^N$;
9. Return $2^n + 1 - t$;

---

for $N \leq 256$ are implemented in assembly language. Elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo the minimal polynomial of a $(q-1)$-th root of unity. Multiplication of two elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ is implemented using Karatsuba's trick [60] in the polynomial dimension and classical multiplication for the coefficients. Since the minimal polynomial is dense we need to use a fast reduction algorithm such as the one described in the book by von zur Gathen and Gerhard [134, Algorithm 9.5].

Table 3.7 contains running times and memory usages of Gaudry's modified Satoh-Skjernaa-Taguchi algorithm for elliptic curves over various finite fields of characteristic 2. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1.

Table 3.7: Running time (s) and memory usage (KB) of modified Satoh-Skjernaa-Taguchi algorithm for elliptic curves over $\mathbb{F}_{2^n}$

n	Lift	Norm	Total	Mem
120	0.023	0.015	0.039	25
144	0.042	0.017	0.059	42
168	0.061	0.023	0.085	48
192	0.095	0.038	0.134	70
240	0.160	0.105	0.265	87
288	0.280	0.107	0.389	128
336	0.452	0.185	0.640	177
384	0.677	0.295	0.974	235
480	1.423	0.610	2.035	333

3.9 Lercier-Lubicz' Algorithm

Lercier and Lubicz [75] presented a p-adic point counting algorithm without precomputation which runs in $O(n^{2\mu}\log n)$ bit-operations and uses $O(n^2)$ space for finite fields with Gaussian Normal Basis.

3.9.1 Solving Artin-Schreier Equations

Recall that if $\mathbb{F}_q$ is a field of characteristic p, an Artin-Schreier equation is an equation of the form $x^p - x + \alpha = 0$ with $\alpha \in \mathbb{F}_q$. The additive version of Hilbert Satz 90 states that such an equation has a solution in $\mathbb{F}_q$ if and only if $\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(\alpha) = 0$. Since $\sigma(x) = x^p$, we can generalize this type of equation to $\mathbb{Z}_q$ by considering

$$\Sigma(x) - ax - b = 0\,, \tag{3.27}$$

with $a, b \in \mathbb{Z}_q$. Let $a_1 = a$, $b_1 = b$ and define $\Sigma^k(x) = a_k x + b_k$ for $k = 2, \ldots, n$ by induction. Since $\Sigma^n(x) = x$, we conclude that the unique solution to the above equation is given by $b_n/(1-a_n)$. To compute the $a_k, b_k \in \mathbb{Z}_q$ Lercier and Lubicz use a simple square and multiply algorithm based on the formula

$$\Sigma^{k+l}(x) = \Sigma^l(a_k x + b_k) = \Sigma^l(a_k)(a_l x + b_l) + \Sigma^l(b_k)\,.$$

This immediately leads to Algorithm 3.9.1 which is somewhat different from the algorithm given in [75]. To find a solution to $\Sigma(x) \equiv ax + b \bmod p^N$ we simply call ARTIN_SCHREIER_ROOT(a, b, n, N) which returns a_n and b_n. The complexity of Algorithm 3.9.1 is determined by steps 2.3 and 2.4 which need $O(1)$ multiplications and $O(1)$ repeated Frobenius substitutions in $\mathbb{Z}_q/(p^N\mathbb{Z}_q)$. For fields with a Gaussian Normal Basis, the Frobenius substitution takes $O(n)$ bit-operations as shown in Section 3.7, and a multiplication takes $O(n^{2\mu})$ bit-operations for $N \simeq n/2$. Since the algorithm needs $O(\log n)$ recursive calls, the time and space complexity for fields with Gaussian Normal Basis are $O(n^{2\mu}\log n)$ and $O(n^2)$ respectively.

ALGORITHM 3.9.1 (ARTIN_SCHREIER_ROOT)

INPUT: Elements $a, b \in \mathbb{Z}_q$, power m, precision N.
OUTPUT: Elements $a_m, b_m \in \mathbb{Z}_q$ such that $\Sigma^m(x) \equiv a_m x + b_m \mod p^N$ and $\Sigma(x) \equiv ax + b \mod p^N$.

1. If $m = 1$ Then
   - 1.1. $a_m \equiv a \mod p^N, b_m \equiv b \mod p^N$;
2. Else
   - 2.1. $m' = \lfloor \frac{m}{2} \rfloor$;
   - 2.2. $a_{m'}, b_{m'} =$ Artin_Schreier_Root(a, b, m', N);
   - 2.3. $a_m \equiv a_{m'}\Sigma^{m'}(a_{m'}) \mod p^N$;
   - 2.4. $b_m \equiv b_{m'}\Sigma^{m'}(a_{m'}) + \Sigma^{m'}(b_{m'}) \mod p^N$;
   - 2.5. If $m \equiv 1 \mod 2$ Then
     - 2.5.1. $a_m \equiv a\,\Sigma(a_m) \mod p^N$;
     - 2.5.2. $b_m \equiv b\,\Sigma(a_m) + \Sigma(b_m) \mod p^N$;
3. Return a_m, b_m;

3.9.2 Generalized Newton's Algorithm I

In Sections 3.6 and 3.8, we used the SST algorithm to solve an equation of the form $\phi(X, \Sigma(X)) = 0$ with $\phi(X, Y) \in \mathbb{Z}_q[X, Y]$. Let $x \in \mathbb{Z}_q$ be a root of this equation and assume we know $x_m \equiv x \bmod p^m$. Define $\delta_m = (x - x_m)/p^m$, then the Taylor expansion around x_m gives

$$\begin{aligned} 0 = \phi(x, \Sigma(x)) &= \phi(x_m + p^m \delta_m, \Sigma(x_m + p^m \delta_m)) \\ &\equiv \phi(x_m, \Sigma(x_m)) + p^m(\delta_m \Delta_x + \Sigma(\delta_m)\Delta_y) \bmod p^{2m}\,, \end{aligned} \tag{3.28}$$

with $\Delta_x \equiv \frac{\partial \phi}{\partial X}(x_m, \Sigma(x_m)) \mod p^m$ and $\Delta_y \equiv \frac{\partial \phi}{\partial Y}(x_m, \Sigma(x_m)) \mod p^m$. This implies that δ_m has to be a solution of

$$-\frac{\phi(x, \Sigma(x))}{p^m} \equiv \delta \Delta_x + \Sigma(\delta)\Delta_y \bmod p^m\,.$$

Let $k = \mathrm{ord}_p(\Delta_y)$, then if $\mathrm{ord}_p(\Delta_x) \geq k$ and $\mathrm{ord}_p(\phi(x_m, \Sigma(x_m))) \geq k + m$ and $m > k$, we recover the Artin-Schreier equation $\Sigma(\delta) \equiv a\delta + b \bmod p^{m-k}$ with $a = -\Delta_x/\Delta_y$ and $b = -\phi(x, \Sigma(x))/(p^m \Delta_y)$. Since a and b are integral and Σ is an isometry, i.e. preserves the valuation, any solution $\delta' \in \mathbb{Z}_q$ to the above equation satisfies $\delta' \equiv \delta_m \bmod p^{m-k}$. Let $x'_m = x_m + p^m \delta'$, then $x'_m \equiv x \bmod p^{2m-k}$ and equation (3.28) implies that $\phi(x'_m, \Sigma(x'_m)) \equiv 0 \bmod p^{2m}$. Furthermore, since we assumed that $m > k$, we have

$$\mathrm{ord}_p(\frac{\partial \phi}{\partial Y}(x'_m, \Sigma(x'_m)) = k \quad \text{and} \quad \mathrm{ord}_p(\frac{\partial \phi}{\partial X}(x'_m, \Sigma(x'_m)) \geq k\,, \tag{3.29}$$

so we can repeat the same procedure to find a solution $x_N \equiv x \bmod p^N$ up to arbitrary precision. This strategy is formalized in Algorithm 3.9.2. Since the precision of the computations almost doubles in every step, the complexity of Algorithm 3.9.2 is the same as the complexity of Algorithm 3.9.1, i.e. $O(n^{2\mu} \log n)$ bit-operations using $O(n^2)$ space for fields with Gaussian Normal Basis.

3.9.3 An $O(n^{2+\varepsilon})$ Time Algorithm for Fields with GNB

In the SST algorithm, we computed the j-invariant of the canonical lift E of an elliptic curve $\overline{E}$ as the solution of

$$\Phi_p(x, \Sigma(x)) = 0 \quad \text{and} \quad j(E) \equiv j(\overline{E}) \bmod p\,,$$

with $\Phi_p(X, Y)$ the p-th modular equation. The Kronecker relation implies that

$$\frac{\partial \Phi_p}{\partial X}(j(E), \Sigma(j(E))) \equiv 0 \bmod p \quad \text{and} \quad \frac{\partial \Phi_p}{\partial Y}(j(E), \Sigma(j(E))) \not\equiv 0 \bmod p\,,$$

so we can use Algorithm 3.9.2 to compute $j(E)$ and adapt the SST algorithm to obtain a point counting algorithm.

In characteristic 2, we can speed up the computations by using the modified modular polynomial $\Lambda_2(X, Y) = Y^2(1 + X)^2 - 4X$ introduced in Section 3.8.1. Recall that for an elliptic curve $\overline{E} : y^2 + xy = x^3 + \overline{a}$ with $\overline{a} \in \mathbb{F}_q^*$, the univariate AGM sequence $\{\lambda_k\}_{k=1}^{\infty}$ is defined as $\lambda_{k+1} = 2\sqrt{\lambda_k}/(1 + \lambda_k)$ with $\lambda_1 = 1 + 8a$

ALGORITHM 3.9.2 (GEN_NEWTON_LIFT)

```
INPUT:   Polynomial φ(X,Y) ∈ Z_q, element x_0 ∈ Z_q satisfying
         φ(x_0, Σ(x_0)) ≡ 0 mod p^{2k+1} with k = ord_p(∂φ/∂Y(x_0, Σ(x_0))),
         precision N.
OUTPUT:  Element x_N ∈ Z_q, with φ(x_N, Σ(x_N)) ≡ 0 mod p^{N+k} and
         x_N ≡ x_0 mod p^{k+1}.
```

1. If $N \leq 2k+1$ Then
   1.1. $x := x_0$;
2. Else
   2.1. $N' = \lceil \frac{N}{2} \rceil + k$; $M = N' - k$;
   2.2. $x' =$ Gen_Newton_Lift(ϕ, x_0, N');
   2.3. $y' = \Sigma(x') \mod p^N$;
   2.4. $V \equiv \phi(x', y') \mod p^N$;
   2.5. $\Delta_x \equiv \frac{\partial \phi}{\partial X}(x', y') \mod p^{N'}$;
   2.6. $\Delta_y \equiv \frac{\partial \phi}{\partial Y}(x', y') \mod p^{N'}$;
   2.7. $a_n, b_n =$ Artin_Schreier_Root $(-\frac{V}{(p^M \Delta_y)}, -\frac{\Delta_x}{\Delta_y}, M, n)$;
   2.8. $x = x' + p^M \frac{b_n}{1-a_n}$;
3. Return x;

---

and $a \in \mathbb{Z}_q$ such that $a \equiv \overline{a} \bmod 2$. In Section 3.8.1 we showed that λ_k satisfies $\Lambda_2(x, \Sigma(x)) \equiv 0 \bmod 2^{k+3}$. Since

$$\frac{\partial \Lambda_2}{\partial X}(\lambda_k, \Sigma(\lambda_k)) \equiv 0 \bmod 2^4 \quad \text{and} \quad \frac{\partial \Lambda_2}{\partial Y}(\lambda_k, \Sigma(\lambda_k)) \not\equiv 0 \bmod 2^4 ,$$

we can again use Algorithm 3.9.2 to solve for λ_k. Note that it is no longer necessary to apply the transformation $X \to 1 + 8X$ and $Y \to 1 + 8Y$ as in Section 3.8.1. Given $\lambda_k \bmod 2^{k+3}$ we can recover the trace of Frobenius as

$$\operatorname{Tr} \overline{F} \equiv t_k + \frac{q}{t_k} \bmod 2^{k+3} \quad \text{with} \quad t_k = \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}\left(\frac{2}{1+\lambda_k}\right) .$$

ALGORITHM 3.9.3 (LERCIER_LUBICZ)

`INPUT:` `Elliptic curve` $\overline{E} : y^2 + xy = x^3 + a$ `with` $a \in \mathbb{F}_{2^n}^*$.
`OUTPUT:` `The number of points on` $\overline{E}(\mathbb{F}_{2^n})$ `.`

1. $N = \lceil \frac{n}{2} \rceil + 2$;
2. $x \equiv 1 + 8a \bmod 16$;
3. $y =$ `Gen_Newton_Lift` (Λ_2, x, N);
4. $t =$ `Norm_Char_2` $(\frac{2}{1+y}, N)$;
5. `If` $t^2 > 2^{n+2}$ `Then` $t = t - 2^N$;
6. `Return` $2^n + 1 - t$;

Algorithm 3.9.3 summarizes the complete point counting algorithm. The complexity is determined by steps 3 and 4, which leads to the following theorem.

THEOREM 3.9.1 (LERCIER-LUBICZ) *Let $\overline{E}$ be an elliptic curve over a finite field $\mathbb{F}_q$ with $q = p^n$ and $j(\overline{E}) \notin \mathbb{F}_{p^2}$. Assume that $\mathbb{F}_q$ has a Gaussian Normal Basis, then there exists a deterministic algorithm to compute the number of points $\#\overline{E}(\mathbb{F}_q)$ which requires, for fixed p, $O(n^2 (\log n)^2 \log\log n)$ bit-operations and $O(n^2)$ space.*

3.9.4 Implementation and Numerical Results

In this section we present running times of an implementation of the Lercier-Lubicz algorithm for GNB of type 1 in the `C` programming language. The basic operations on integers modulo 2^N for $N \leq 256$ are implemented in assembly language. Elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo the irreducible polynomial $(x^{n+1} - 1)/(x - 1)$. Multiplication of two elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ is implemented using Karatsuba's trick [60] in the polynomial dimension and classical multiplication for the coefficients.

Table 3.8 contains running times and memory usages of Lercier-Lubicz algorithm for elliptic curves over various finite fields of characteristic 2. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1.

Table 3.8: Running time (s) and memory usage (KB) of Lercier-Lubicz algorithm for elliptic curves over $\mathbb{F}_{2^n}$

n	Lift	Norm	Total	Mem
130	0.013	0.002	0.016	24
148	0.019	0.003	0.023	28
172	0.025	0.004	0.029	32
196	0.030	0.007	0.038	48
268	0.079	0.017	0.096	79
292	0.092	0.018	0.111	87
346	0.139	0.035	0.174	123
388	0.172	0.047	0.220	161
466	0.278	0.095	0.374	219

3.10 Harley's Algorithm

In a December 2002 e-mail to the NMBRTHRY list [55], Harley announced an algorithm to count the number of points on an ordinary elliptic curve over $\mathbb{F}_{p^n}$, which runs in $O(n^{2\mu}\log n)$ bit-operations and uses $O(n^2)$ space for fixed p. However, the e-mail only contained the code for the main function and several routines, e.g. lifting the field polynomial, were missing. In this section we provide a detailed description of all routines in Harley's algorithm and explain why they work.

3.10.1 Generalised Newton's Algorithm II

In Section 3.9.2 we considered the following problem: given $\phi(X,Y) \in \mathbb{Z}_q[X,Y]$, find a root $x \in \mathbb{Z}_q$ of

$$\phi(X,\Sigma(X)) = 0\,. \tag{3.30}$$

Assume we know $x_m \equiv x \bmod p^m$ and let $\delta_m = (x - x_m)/p^m$, then we already showed that δ_m is determined as the solution of

$$-\frac{\phi(x,\Sigma(x))}{p^m} \equiv \delta\Delta_x + \Sigma(\delta)\Delta_y \bmod p^m\,,$$

with $\Delta_x \equiv \frac{\partial\phi}{\partial X}(x_m,\Sigma(x_m)) \mod p^m$ and $\Delta_y \equiv \frac{\partial\phi}{\partial Y}(x_m,\Sigma(x_m)) \mod p^m$. Let $k = \mathrm{ord}_p(\Delta_y)$, then if $\mathrm{ord}_p(\Delta_x) \geq k$ and $\mathrm{ord}_p(\phi(x_m,\Sigma(x_m))) \geq k+m$ and

$m > k$, we get an equation of the form

$$\alpha\Sigma(\delta) + \beta\delta + \gamma \equiv 0 \bmod p^t\,, \tag{3.31}$$

with $t = m - k$, $\alpha, \beta, \gamma \in \mathbb{Z}_q$ and α a unit in $\mathbb{Z}_q$. Note that since Σ preserves the valuation, this equation determines $\delta_m \bmod p^{m-k}$ uniquely and $x \equiv x_m + p^m \delta_m \bmod p^{2m-k}$. Assume we have an algorithm that returns a zero $\delta_{t'}$ of equation (3.31) to precision $t' = \lceil t/2 \rceil$, then we can use the same algorithm to find a zero δ_t to precision t. Indeed, substituting $\delta_t = \delta_{t'} + p^{t'}\Delta_t$ in equation (3.31) leads to

$$\alpha\Sigma(\Delta_t) + \beta\Delta_t + \frac{\alpha\Sigma(\delta_{t'}) + \beta\delta_{t'} + \gamma}{p^{t'}} \equiv 0 \bmod p^{t-t'}\,,$$

and since $t - t' \leq t'$ we can use the same algorithm to determine $\Delta_t \bmod p^{t-t'}$ and therefore δ_t. This immediately leads to a recursive algorithm if we can solve the base case, i.e. find a solution to equation (3.31) modulo p. If we assume that $\mathrm{ord}_p(\Delta_x) > \mathrm{ord}_p(\Delta_y)$, then $\mathrm{ord}_p(\beta) > 0$ and the base case reduces to solving $\alpha\Sigma(\delta) + \gamma \equiv 0 \bmod p$. Since α is a unit, this uniquely determines $\delta \bmod p$. Algorithms 3.10.1 and 3.10.2 contain pseudo-code to solve equations (3.31) and (3.30) respectively. Note that Algorithm 3.10.2 is identical to Algorithm 3.9.2, but uses a different subroutine to solve the Artin-Schreier equation.

Remark 3.10.1 Note that the condition $\mathrm{ord}_p(\beta) > 0$ implies that the unique solution to equation (3.31) is integral. Indeed, let $b = -\beta/\alpha$ and $c = -\gamma/\alpha$, then δ is also the solution of $\Sigma(\delta) = b\delta + c$. Repeatedly applying Σ to both sides of this equation and using the fact that $\Sigma^n(\delta) = \delta$, leads to the following explicit expression

$$\delta = \frac{\sum_{i=0}^{n-1} \Sigma^i(c) \cdot \prod_{j=i+1}^{n-1} \Sigma^j(b)}{1 - \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(b)}\,.$$

Since $\mathrm{ord}_p(b) = \mathrm{ord}_p(\beta) > 0$, we conclude that $1 - \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(b)$ is a unit and therefore $\delta \in \mathbb{Z}_q$.

Remark 3.10.2 The p-th root in step 1.1 of Algorithm 3.10.1 should not be computed by naively taking the p^{n-1}-th power. Instead, let $\mathbb{F}_q = \mathbb{F}_p[\overline{\theta}]$, then

$$\left(\sum_{i=0}^{n-1} \overline{a}_i \overline{\theta}^i\right)^{1/p} = \sum_{j=0}^{p-1} \left(\sum_{0 \leq pk+j < n} \overline{a}_{pk+j} \overline{\theta}^k\right) C_j(\overline{\theta}),$$

with $C_j(\overline{\theta}) = (\overline{\theta}^j)^{1/p} = \overline{\theta}^{jp^{n-1}}$. This shows that for $\overline{z} \in \mathbb{F}_q$, we can compute $\overline{z}^{1/p}$ with $p - 1$ multiplications over $\mathbb{F}_q$. Note that we could avoid computing the p-th root altogether if we replaced equation (3.30) by $\phi(\Sigma^{-1}(X), X) = 0$.

ALGORITHM 3.10.1 (ARTIN_SCHREIER_ROOT_II)

```
INPUT:   Elements α, β, γ ∈ Z_q, α ∈ Z_q^×, ord_p(β) > 0, precision N.
OUTPUT:  Element x ∈ Z_q such that αΣ(x) + βx + γ ≡ mod p^N.

  1.  If N = 1 Then
      1.1.  x ≡ (−γ/α)^{1/p} mod p;
  2.  Else
      2.1.  N′ = ⌈N/2⌉;  M = N − N′;
      2.2.  x′ = Artin_Schreier_Root_II(α, β, γ, N′);
      2.3.  γ′ ≡ (αΣ(x′) + βx′ + γ)/p^{N′} mod p^M;
      2.4.  Δ′ = Artin_Schreier_Root_II(α, β, γ′, M);
      2.5.  x ≡ x′ + p^{N′}Δ′ mod p^N;
  3.  Return x;
```

The complexity of Algorithm 3.10.1 is determined by the recursive calls in steps 2.2 and 2.4, the $O(1)$ multiplications in step 2.3 and the Frobenius substitution in step 2.3. If we assume that $\mathbb{Z}_q$ is represented using the Teichmüller lift of the field polynomial, the Frobenius substitution in $\mathbb{Z}_q/(p^N\mathbb{Z}_q)$ can be computed using $O((nN)^\mu)$ bit-operations. If $T(N)$ is the running time of Algorithm 3.10.1 for precision N, then we have

$$T(N) \leq 2T(\lceil N/2 \rceil) + c((nN)^\mu)\,,$$

for some constant c. The above relation implies by induction that the complexity of Algorithm 3.10.1 is $O((nN)^\mu \log N)$.

The complexity of Algorithm 3.10.2 can now be easily determined and is also given by $O((nN)^\mu \log N)$ bit-operations. Indeed, since the algorithm calls itself only once and the precision almost doubles in every iteration, the complexity is determined by the call to ARTIN_SCHREIER_ROOT_II in step 2.7.

3.10.2 Lifting the Field Polynomial

Let $\overline{\theta} \in \mathbb{F}_q$ be such that $\mathbb{F}_q = \mathbb{F}_p[\overline{\theta}]$ and let $\theta \in \mathbb{Z}_q$ be the Teichmüller lift of $\overline{\theta}$. In the previous section we assumed that $\mathbb{Z}_q$ was represented as $\mathbb{Z}_p[x]/(f(x))$ with $f(x)$ the minimal polynomial of θ. Since θ is a $(q-1)$-th root of unity, we have that $\Sigma(\theta) = \theta^p$ and $f(x) = \prod_{i=0}^{n-1}(x - \theta^{p^i})$. Let ζ_p be a formal p-th root of unity, then

$$f(x^p) = \prod_{i=0}^{p-1} f(\zeta_p^i x)\,. \tag{3.32}$$

ALGORITHM 3.10.2 (GEN_NEWTON_LIFT_II)

INPUT: Polynomial $\phi(X,Y) \in \mathbb{Z}_q$, element $x_0 \in \mathbb{Z}_q$ satisfying $\phi(x_0, \Sigma(x_0)) \equiv 0 \mod p^{2k+1}$ and $\operatorname{ord}_p(\frac{\partial \phi}{\partial X}(x_0, \Sigma(x_0))) > k$, with $k = \operatorname{ord}_p(\frac{\partial \phi}{\partial Y}(x_0, \Sigma(x_0)))$, precision N.

OUTPUT: Element $x_N \in \mathbb{Z}_q$, with $\phi(x_N, \Sigma(x_N)) \equiv 0 \mod p^{N+k}$ and $x_N \equiv x_0 \mod p^{k+1}$.

```
1.  If N ≤ 2k + 1 Then
    1.1.  x := x_0;
2.  Else
    2.1.  N' = ⌈N/2⌉ + k;  M = N' − k;
    2.2.  x' = Gen_Newton_Lift(φ, x_0, N');
    2.3.  y' = Σ(x') mod p^N;
    2.4.  V ≡ (φ(x', y') mod p^N)/p^{N'} mod p^M;
    2.5.  Δ_x ≡ (∂φ/∂X(x', y') mod p^{N'})/p^k mod p^M;
    2.6.  Δ_y ≡ (∂φ/∂Y(x', y') mod p^{N'})/p^k mod p^M;
    2.7.  Δ' = Artin_Schreier_Root_II(Δ_y, Δ_x, V, M);
    2.8.  x ≡ x' + p^M Δ' mod p^N;
3.  Return x;
```

Indeed, for $i = 0, \ldots, n-1$ each factor $(x^p - \theta^{p^{i+1}})$ of $f(x^p)$ splits as

$$(x - \theta^{p^i})(\zeta_p x - \theta^{p^i}) \cdots (\zeta_p^{p-1} x - \theta^{p^i}) .$$

Write $f(x)$ as $f(x) = \sum_{i=0}^{p-1} f_i(x^p) x^i$ with $f_i(x) \in \mathbb{Z}_q[x]$, then equation (3.32) can be rewritten as

$$f(x^p) = \prod_{j=0}^{p-1} \left(\sum_{i=0}^{p-1} \zeta_p^{ij} f_i(x^p) x^i \right) = \sum_{k=0}^{p-1} h_k(f_0(x^p), \ldots, f_{p-1}(x^p)) x^{pk} ,$$

with $h_k \in \mathbb{Z}_q[Y_0, \ldots, Y_{p-1}]$ homogeneous polynomials of degree p. This implies that $f(x)$ satisfies $f(x) = \sum_{k=0}^{p-1} h_k(f_0(x), \ldots, f_{p-1}(x)) x^k$. The following table lists the first few examples of the polynomials $h_k(Y_0, \ldots, Y_{p-1})$.

$p = 2$	$h_0 = Y_0^2$ $h_1 = -Y_1^2$
$p = 3$	$h_0 = Y_0^3$ $h_1 = Y_1^3 - 3Y_0Y_1Y_2$ $h_2 = Y_2^3$
$p = 5$	$h_0 = Y_0^5$ $h_1 = Y_1^5 + 5(Y_0^2Y_1^2Y_3 - Y_0^3Y_1Y_4 - Y_0^3Y_2Y_3 + Y_0^2Y_1Y_2^2 - Y_0Y_1^3Y_2)$ $h_2 = Y_2^5 + 5(Y_0^2Y_2Y_4^2 + Y_0^2Y_3^2Y_4 + Y_0Y_1^2Y_4^2 - Y_0Y_1Y_2Y_3Y_4 - Y_0Y_1Y_3^3$ $-Y_0Y_2^3Y_4 + Y_0Y_2^2Y_3^2 - Y_1^3Y_3Y_4 + Y_1^2Y_2^2Y_4 + Y_1^2Y_2Y_3^2 - Y_1Y_2^3Y_3)$ $h_3 = Y_3^5 + 5(Y_1Y_3^2Y_4^2 - Y_0Y_3Y_4^3 - Y_1Y_2Y_4^3 + +Y_2^2Y_3Y_4^2 - Y_2Y_3^3Y_4)$ $h_4 = Y_4^5$

Assume we know $f_t(x) \equiv f(x) \bmod p^t$ and let $\delta_t(x) = (f(x) - f_t(x))/p^t$. Substituting $f_t(x) + p^t\delta_t(x)$ in the equation

$$f(x) - \sum_{k=0}^{p-1} h_k(f_0(x), \ldots, f_{p-1}(x)) x^k = 0$$

gives a relation that determines $\delta_t(x)$ modulo p^t. For example, consider the case $p = 2$, which leads to

$$\delta_t(x) - 2(f_{t,0}(x)\delta_{t,0}(x) - x f_{t,1}(x)\delta_{t,1}(x)) + V_t(x) \equiv 0 \bmod 2^t , \tag{3.33}$$

with $V_t(x) \equiv (f_t(x) - f_{t,0}(x)^2 + x f_{t,1}(x)^2)/2^t \bmod 2^t$. Assume we have an algorithm which computes $\delta_t(x) \bmod p^{t'}$ with $t' = \lceil t/2 \rceil$, then we can use the same algorithm to compute $\delta_t(x) \bmod p^t$. Indeed, substituting $\delta_t = \delta_{t'} + p^{t'}\Delta_{t'}$ in equation (3.33) leads to a similar equation modulo $2^{t-t'}$ with δ_t replaced by $\Delta_{t'}$ and V_t replaced by

$$\frac{V_t(x) + \delta_{t'}(x) - 2(f_{t,0}(x)\delta_{t',0}(x) - x f_{t,1}(x)\delta_{t',1}(x))}{2^{t'}} \bmod 2^{t-t'} .$$

Algorithm 3.10.3 (Teichmüller_Field_Pol_Inc_2)

```
INPUT:   Polynomials f_0(x), f_1(x), V(x) ∈ Z_p[x], precision N.
OUTPUT:  Polynomial δ(x) ∈ Z_p[x] which satisfies the equation
         δ(x) − 2(f_0(x)δ_0(x) − x f_1(x)δ_1(x)) + V(x) ≡ 0 mod 2^N.
```

1. If $N = 1$ Then
   - 1.1. $\delta(x) \equiv -V(x) \mod 2$;
2. Else
   - 2.1. $N' = \lceil \frac{N}{2} \rceil$;
   - 2.2. $\delta'(x) :=$ Teichmüller_Field_Pol_Inc(f_0, f_1, V, N');
   - 2.3. $\delta_0(x^2) \equiv (\delta'(x) + \delta'(-x))/2 \mod 2^N$;
   - 2.4. $\delta_1(x^2) \equiv (\delta'(x) - \delta'(-x))/(2x) \mod 2^N$;
   - 2.5. $V'(x) \equiv \frac{V(x)+\delta'(x)-2(f_0(x)\delta_0(x)-xf_1(x)\delta_1(x))}{2^{N'}} \mod 2^{N-N'}$;
   - 2.6. $\Delta(x) :=$ Teichmüller_Field_Pol_Inc ($f_0, f_1, V', N - N'$);
   - 2.7. $\delta(x) \equiv \delta'(x) + 2^{N'}\Delta(x) \mod 2^N$;
3. Return $\delta(x)$;

Since $t - t' \leq t'$ we can use the same algorithm to find $\Delta_{t'}$ and thus δ_t. This immediately leads to Algorithms 3.10.3 and 3.10.4.

The complexity of Algorithms 3.10.3 and 3.10.4 can be determined in a similar fashion to Algorithm 3.10.1 and is also given by $O((nN)^{\mu} \log N)$ bit-operations. Furthermore, it is clear that the space complexity of both algorithms is $O(nN)$ bits.

ALGORITHM 3.10.4 (TEICHMÜLLER_FIELD_POL_2)

```
INPUT:   Monic irreducible polynomial f̄(x) ∈ F_2[x] of degree n,
         precision N.
OUTPUT:  Teichmüller lift f(x) ∈ Z_p[x] up to precision N, i.e.
         f(x)|x^(2^n - 1) mod 2^N and f(x) ≡ f̄(x) mod 2.
```

1. If $N = 1$ Then
   1.1. $f(x) \equiv \overline{f}(x) \bmod 2$;
2. Else
   2.1. $N' = \lceil \frac{N}{2} \rceil$;
   2.2. $f'(x) :=$ Teichmüller_Field_Pol$(\overline{f}, N')$;
   2.3. $f_0(x^2) \equiv (f'(x) + f'(-x))/2 \bmod 2^N$;
   2.4. $f_1(x^2) \equiv (f'(x) - f'(-x))/(2x) \bmod 2^N$;
   2.5. $V(x) \equiv (f'(x) - f_0(x)^2 + x f_1(x)^2)/2^{N'} \bmod 2^{N-N'}$;
   2.6. $\delta(x) :=$ Teichmüller_Field_Pol_Inc $(f_0, f_1, V, N - N')$;
   2.7. $f(x) \equiv f'(x) + 2^{N'} \delta(x) \bmod 2^N$;
3. Return $f(x)$;

3.10.3 Norm Computation III

In the same e-mail [55], Harley suggested an asymptotically fast norm computation algorithm based on a classical formula from number theory which expresses the norm as a resultant. The resultant itself can be computed using an adaptation of Moenck's fast extended GCD algorithm [88].

Let $\mathbb{Z}_q = \mathbb{Z}_p[\theta]$ with θ a root of a monic irreducible polynomial $f(x) \in \mathbb{Z}_p[x]$ of degree n. Then $f(x)$ splits completely over $\mathbb{Z}_q$ as $f(x) = \prod_{i=0}^{n-1}(x - \Sigma^i(\theta))$, with Σ the Frobenius substitution. Since for $\alpha \in \mathbb{Q}_q$ we have

$$\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) = p^{n\,\mathrm{ord}_p(\alpha)} \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha / p^{\mathrm{ord}_p(\alpha)}), \tag{3.34}$$

it suffices to compute the norm of a unit in $\mathbb{Z}_q$. Let $\alpha = \sum_{i=0}^{n-1} a_i \theta^i$ be a unit in $\mathbb{Z}_q$ and define $A(x) = \sum_{i=0}^{n-1} a_i x^i \in \mathbb{Z}_p[x]$. By definition of the norm and the resultant we have

$$\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) = \prod_{i=0}^{n-1} \Sigma^i(\alpha) = \prod_{i=0}^{n-1} A(\Sigma^i(\theta)) = \mathrm{Res}(f(x), A(x))\,. \tag{3.35}$$

The resultant $\mathrm{Res}(f(x), A(x))$ can be computed in softly linear time using a variant of Moenck's fast extended GCD algorithm [88]. We also refer to Chapter 11 of the book by von zur Gathen and Gerhard [134] for more details. The basic idea of Moenck's algorithm is that the first quotients in the Euclidean algorithm only depend on the highest coefficients of the input polynomials.

Let $f = f_n x^n + \cdots + f_0 \in \mathbb{K}[x]$ be a degree n polynomial over a field $\mathbb{K}$, i.e. $f_n \neq 0$. The truncated polynomial $f|k$ is defined as

$$f|k = f_n x^k + \cdots + f_{n-k}\,, \tag{3.36}$$

where $f_i = 0$ for $i < 0$. Furthermore, two pairs of polynomials (f, g) and (f^*, g^*) are said to coincide up to k if

$$\begin{gathered} f|k = f^*|k\,, \\ g|(k - (\deg f - \deg g)) = g^*|(k - (\deg f^* - \deg g^*))\,. \end{gathered} \tag{3.37}$$

Given a pair of monic polynomials r_0 and r_1 with $\deg r_0 > \deg r_1$, the Euclidean algorithm computes the expressions

$$r_{i-1} = q_i r_i + \rho_{i+1} r_{i+1}\,, \tag{3.38}$$

with $q_i, r_i \in \mathbb{K}[x]$, $\rho_{i+1} \in \mathbb{K}$ for $i = 0, \ldots, l$, r_i monic and $r_{l-1} = q_l r_l$. Let $m_i = \deg q_i$ for $1 \leq i \leq l$ and define for $k \in \mathbb{N}$ the number $\eta(k) \in \mathbb{N}$ by

$$\eta(k) = \max\{0 \leq j \leq l \,|\, \sum_{1 \leq i \leq j} m_i \leq k\}\,. \tag{3.39}$$

The basis for Moenck's algorithm is the following lemma, which states that the quotients in the Euclidean algorithm only depend on the highest coefficients of the inputs. A proof can be found in [134, Lemma 11.3].

LEMMA 3.10.1 *Let $k \in \mathbb{N}$ and assume that (f, g) and (f^*, g^*) coincide up to $2k$, then $\eta(k) = \eta^*(k)$, $q_i = q_i^*$ and $\rho_{i+1} = \rho_{i+1}^*$ for $1 \leq i \leq \eta(k)$, with q_i, q_i^* the quotients and ρ_{i+1}, ρ_{i+1}^* the leading coefficients of the remainders computed by the Euclidean algorithm on input (f, g) and (f^*, g^*) respectively.*

This lemma immediately leads to a recursive algorithm to compute the extended GCD of two polynomials. However, to determine the resultant, we have to keep track of the scalars $\rho_i \in \mathbb{K}$ for $i = 0, \ldots, l$. The following lemma can be found in [134, Corollary 11.16].

LEMMA 3.10.2 *Let $f, g \in \mathbb{K}[x]$ of degrees $n = n_0$ and $n_1 \le n_0$ respectively. Let $n_2, \ldots, n_l$ be the degrees of the remainders in the Euclidean algorithm for (f, g) and $\rho_0, \ldots, \rho_l$ their leading coefficients. If $gcd(f, g) = 1$, then*

$$\mathrm{Res}(f, g) = (-1)^\tau \rho_0^{n_1} \prod_{1 \le j \le l} \rho_j^{n_{j-1}}, \tag{3.40}$$

with $\tau = \sum_{1 \le j \le l} n_{j-1} n_j$. This resultant can be calculated using $O(n^\mu \log n)$ operations in $\mathbb{K}$.

ALGORITHM 3.10.5 (XGCD)

INPUT: Monic polynomials $r_0, r_1 \in \mathbb{K}[x]$ with $\deg r_0 > \deg r_1$, $k \in \mathbb{N}$, vectors V and W.

OUTPUT: Integer $h = \eta(k) \in \mathbb{N}$, matrix $R_h \in \mathbb{K}[x]^{2\times 2}$ such that $(r_{h-1}, r_h)' = R_h(r_0, r_1)'$, vector V with $V[i] = \rho_i$, vector W with $W[i] = n_i - n_{i+1}$.

1. If $r_1 = 0$ Or $k \le \deg r_0 - \deg r_1$ Then
   1.1. Return $h = 0$; $R_h = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$; V; W;
2. $d = \lfloor \frac{k}{2} \rfloor$; $(\eta(d), R, V, W) =$ XGCD$(r_0|2d, r_1|(2d - (n_0 - n_1)), d, V, W)$;
3. $j = \eta(d) + 1$; $\begin{pmatrix} r_{j-1} \\ r_j \end{pmatrix} = R \begin{pmatrix} r_0 \\ r_1 \end{pmatrix}$; $n_{j-1} = \deg r_{j-1}$; $n_j = \deg r_j$;
4. If $r_j = 0$ Or $k \le n_0 - n_j$ Then
   4.1. Return $h = \eta(d)$; $R_h = R$; V; W;
5. $r_{j-1} = q_j r_j + \rho_{j+1} r_{j+1}$; /* r_{j+1} monic */
6. If $r_{j+1} = 0$ Then
   6.1. Return $h = j$; $R_h = R$; V; W;
7. $n_{j+1} = \deg r_{j+1}$; $d^* = k - (n_0 - n_j)$;
   Append (V, ρ_{j+1}); Append (W, $n_j - n_{j+1}$);
8. $(\eta(d^*), S, V^*, W^*) =$ XGCD$(r_j|2d^*, r_{j+1}|(2d^* - (n_j - n_{j+1})), d^*, V, W)$;
9. Return $h = \eta(d^*) + j$; $R_h = S \begin{pmatrix} 0 & 1 \\ \rho_{j+1}^{-1} & -q_j \rho_{j+1}^{-1} \end{pmatrix} R$; V; W;

Algorithm 3.10.5 computes the extended GCD of two monic polynomials $r_0, r_1 \in \mathbb{K}[x]$ with $\deg r_0 > \deg r_1$ and also returns vectors V and W such that $V[i] = \rho_i$ for $i = 0, \ldots, l$ and $W[i] = n_i - n_{i+1}$ for $i = 0, \ldots, l-1$. The algorithm given in the book by von zur Gathen and Gerhard [134, Algorithm 11.4] contains several errors; therefore one should use Algorithm 3.10.5 instead. The resultant and thus the norm, follows immediately from equation (3.40). This is formalized in Algorithm 3.10.6.

REMARK 3.10.3 Although Algorithm 3.10.6 has a quadratic time complexity, it is not really useful for cryptographic applications. To recover the correct norm modulo p^N one would have to compute modulo p^{Nc} with c a small constant which depends on the minimum valuation of the coefficients of β/a_d. In practice we therefore use Algorithm 3.6.3.

ALGORITHM 3.10.6 (NORM)

```
INPUT:   Unit α = Σ_{i=0}^{n-1} a_i θ^i ∈ Z_q with θ zero of monic irreducible,
         polynomial f(x) ∈ Z_p[x] of degree n, precision N.
OUTPUT:  Norm N_{Q_q/Q_p}(α) mod p^N.

  1.  β ≡ Σ_{i=0}^{n-1} a_i x^i mod p^N; d = Degree (β);
  2.  V[0] = 1; V[1] = a_d; W[0] = n - d;
  3.  (l, R, V, W) = XGCD(f, β/a_d mod p^N, n, V, W);
  4.  r = 1; s = 1;
  5.  For i = 0 To #W - 1 Do
      5.1.  s ≡ sV[i + 1] mod p^N; r ≡ rs^{W[i]} mod p^N;
  6.  Return r;
```

3.10.4 An $O(n^{2+\varepsilon})$ Time Algorithm

Since $p = 2$ is most important in practice, we only provide the details for this case. The algorithms for $p > 2$ can be devised in a similar way.

Let $\overline{E} : y^2 + xy = x^3 + a$ with $a \in \mathbb{F}_{2^n}^*$ be an elliptic curve with $j(\overline{E}) \notin \mathbb{F}_4$. In Section 3.8.2 we showed that if γ_k satisfies $\Upsilon_2(\gamma_k, \Sigma(\gamma_k)) \equiv 0 \bmod 2^k$, with Υ_2 the modified modular polynomial, then

$$\mathrm{Tr}\overline{F} \equiv t_k + \frac{q}{t_k} \bmod 2^{k+3} \quad \text{with } t_k = \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}\left(\frac{1}{1 + 4\gamma_k}\right).$$

Once the Teichmüller lift of the field polynomial is precomputed, the equation $\Upsilon_2(X, \Sigma(X)) = 0$ can be solved very efficiently using Algorithm 3.10.2. Finally, the norm t_k can be computed using Algorithm 3.10.6. The complete Algorithm 3.10.7 clearly requires $O(n^{2\mu} \log n)$ bit-operations and $O(n^2)$ space since all subroutines are dominated by these complexities.

THEOREM 3.10.1 (HARLEY) *Let $\overline{E}$ be an ordinary elliptic curve over a finite field $\mathbb{F}_q$ with $q = p^n$ and $j(\overline{E}) \notin \mathbb{F}_{p^2}$. Then there exists a deterministic algorithm to compute the number of points $\#\overline{E}(\mathbb{F}_q)$ which requires $O(n^2(\log n)^2 \log\log n)$ bit-operations and $O(n^2)$ memory for fixed p.*

ALGORITHM 3.10.7 (HARLEY)

INPUT: Elliptic curve $\overline{E} : y^2 + xy = x^3 + a$ with $a \in \mathbb{F}_{2^n}^*$.
OUTPUT: The number of points on $\overline{E}(\mathbb{F}_{2^n})$.

1. $N = \lceil \frac{n}{2} \rceil + 2$;
2. $f(x) =$ `Teichmüller_Field_Pol_2` $(\overline{f}, N)$;
3. $x \equiv a \bmod 2$;
4. $x =$ `Gen_Newton_Lift_II` (Υ_2, x, N);
5. $t =$ `Norm` $(\frac{1}{1+4x}, N)$;
6. If $t^2 > 2^{n+2}$ Then $t = t - 2^N$;
7. Return $2^n + 1 - t$;

3.10.5 Implementation and Numerical Results

In this section we present running times of an implementation of Algorithm 3.10.7 in the `C` programming language. The basic operations on integers modulo 2^N for $N \leq 256$ are implemented in assembly language. Elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo the minimal polynomial of a $(q-1)$-th root of unity. Multiplication of two elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ is implemented using Karatsuba's trick [60] in the polynomial dimension and classical multiplication for the coefficients. Since the minimal polynomial is dense we need to use a fast reduction algorithm such as the one described in the book by von zur Gathen and Gerhard [134, Algorithm 9.5].

Table 3.9 contains running times and memory usages of Harley's algorithm for elliptic curves over various finite fields of characteristic 2. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1.

3.11 Overall Comparison

This section presents a final comparison of the running times of all algorithms described in this chapter. Table 3.10 contains running times for the general algorithms, whereas Table 3.11 gives running times for the algorithms that rely on a finite field with Gaussian Normal Basis.

Table 3.9: Running time (s) and memory usage (KB) of Harley's algorithm for elliptic curves over $\mathbb{F}_{2^n}$

n	Lift	Norm	Total	Mem
144	0.042	0.017	0.059	47
168	0.060	0.023	0.083	54
192	0.086	0.038	0.125	81
240	0.147	0.105	0.254	100
288	0.269	0.107	0.378	147
336	0.410	0.185	0.597	203
384	0.620	0.295	0.918	270
480	1.263	0.610	1.873	384

Table 3.10: Running time (s) of algorithms by Fouquet-Gaudry-Harley (FGH), Skjernaa (SKJ), Vercauteren (VER), Mestre (bivariate AGM), Satoh-Skjernaa-Taguchi (SST), Gaudry (MSST), Harley (HAR) for elliptic curves over $\mathbb{F}_{2^n}$

n	FGH	SKJ	VER	AGM	SST	MSST	HAR
144	3.99	2.89	1.91	0.57	0.13	0.06	0.06
168	6.04	4.93	3.22	0.88	0.26	0.08	0.08
192	9.29	6.95	4.62	1.39	0.29	0.13	0.12
240	16.5	17.7	10.9	2.81	0.65	0.26	0.25
288	38.6	27.8	17.6	5.44	0.72	0.39	0.38
336	62.5	46.4	30.2	9.28	1.17	0.64	0.60
384	101.1	77.1	49.3	15.4	1.76	0.97	0.92
480	205.3	157.9	106.5	34.5	3.56	2.03	1.87

Table 3.11: Running time (s) of algorithms by Kim et al. (KIM) and Lercier-Lubicz (LELU) for elliptic curves over $\mathbb{F}_{2^n}$ with GNB of type 1

n	KIM	LELU
148	0.040	0.023
172	0.062	0.029
196	0.081	0.038
268	0.178	0.096
292	0.206	0.111
346	0.341	0.174
388	0.496	0.220
466	0.872	0.374

3.12 Conclusion

In this chapter we have designed a new p-adic algorithm to count the number of points on an elliptic curve over a finite field of small characteristic. Furthermore, we have presented an extensive overview of the more recent developments in p-adic point counting for elliptic curves.

At the end of 1999, Satoh [105] described the first p-adic algorithm that used the canonical lift of an elliptic curve. To count the number of points on an elliptic curve over $\mathbb{F}_{p^n}$, Satoh's algorithm requires $O(n^3 \log n \log\log n)$ bit-operations and $O(n^2)$ space for fixed p.

The main result of this chapter is a memory efficient version of Satoh's algorithm which has a quadratic space complexity and runs faster by a constant. The key idea is to replace the multivariate Newton iteration by an iterated univariate Newton iteration. A priori, it is not clear that this iterated univariate Newton iteration converges at all and in general this will not be the case. Using the Kronecker relation satisfied by the p-th modular polynomial, we were able to prove convergence. To count the number of points on an elliptic curve over $\mathbb{F}_{p^n}$ the algorithm requires $O(n^3 \log n \log\log n)$ bit-operations and $O(n^2)$ space for fixed p. This result has been published in [133].

The second contribution of this chapter is a detailed description of the more recent p-adic point counting algorithms for elliptic curves, culminating in Harley's algorithm which runs in $O(n^2(\log n)^2 \log\log n)$ time and requires $O(n^2)$ space. An abbreviated version of this overview will appear as a contributed chapter in the books [9] and [17].

The final result of all these improvements is that it is now possible to count the number of points on an elliptic curve over a finite field $\mathbb{F}_q$ with $q \simeq 2^{168}$ in less than 0.1 seconds.

Chapter 4

Hyperelliptic Curves

This chapter presents algorithms to compute the zeta function of a hyperelliptic curve over a finite field using Monsky-Washnitzer cohomology. These algorithms are based on a p-adic Lefschetz trace formula and express the zeta function in terms of the characteristic polynomial of a Frobenius operator acting on the cohomology groups. Kedlaya [61] described such an algorithm for hyperelliptic curves over finite fields of odd characteristic. For fixed p, the time complexity for a genus g hyperelliptic curve over $\mathbb{F}_{p^n}$ is $(g^{4+\varepsilon}n^{3+\varepsilon})$ and the space complexity is $O(g^3n^3)$.

In this chapter, we extend Kedlaya's algorithm to hyperelliptic curves defined over finite fields of characteristic two. To ease the exposition, we proceed in two steps: the first algorithm is limited to hyperelliptic curves of rank 0 or 1, whereas the second algorithm can handle arbitrary hyperelliptic curves. For a genus g hyperelliptic curve over $\mathbb{F}_{2^n}$, the average-case time complexity of this algorithm is $O(g^{4+\varepsilon}n^{3+\varepsilon})$ and the average-case space complexity is $O(g^3n^3)$, whereas the worst-case time and space complexities are $O(g^{5+\varepsilon}n^{3+\varepsilon})$ and $O(g^4n^3)$.

A related approach by Lauder and Wan [69] is based on Dwork's proof of the rationality of the zeta function and leads to a polynomial time algorithm to compute the zeta function of an arbitrary variety over a finite field. Despite its polynomial time complexity, the Lauder and Wan algorithm is not practical. Using Dwork cohomology, Lauder and Wan [70, 71] adapted their original algorithm for the special case of Artin-Schreier curves and obtained a time complexity of $O(g^{5+\varepsilon}n^{3+\varepsilon})$. However, this algorithm is somewhat less general than the one given in Section 4.4 and only rigorously proven for $p \geq 5$.

For ordinary hyperellipitc curves in characteristic two, Mestre [84] described a very efficient algorithm to compute the product of the unit roots of the zeta function. To recover the zeta function, a 2-adic version of the LLL algorithm [72] is used. For an ordinary hyperelliptic curve of genus g, the algorithm requires $O(2^g n^{3+\varepsilon})$ bit-operations and is therefore only efficient for small geni. This algorithm has been implemented and optimized by Lercier and Lubicz [76] who obtained a running time of $O(2^g n^{2+\varepsilon})$. Finally, Ritzenthaler [102] extended Mestre's algorithm to all ordinary non-hyperelliptic curves of genus 3.

4.1 Monsky-Washnitzer Cohomology of Curves

In this section we specialize the formalism of Monsky-Washnitzer cohomology, first described in the seminal papers by Monsky and Washnitzer [89, 90, 91], to smooth affine plane curves. Further details can be found in the lectures by Monsky [92] and the survey by van der Put [127].

Let $\overline{C}$ be a smooth affine plane curve over a finite field $\mathbb{F}_q$ with $q = p^n$ elements and let $\mathbb{Q}_q$ be a degree n unramified extension of $\mathbb{Q}_p$ with valuation ring $\mathbb{Z}_q$, such that $\mathbb{Z}_q/(p\mathbb{Z}_q) = \mathbb{F}_q$. The aim of Monsky-Washnitzer cohomology is to express the zeta function of the curve $\overline{C}$ in terms of a Frobenius operator $\mathcal{F}$ acting on p-adic cohomology groups $H^i(\overline{C}/\mathbb{Q}_q)$ associated to $\overline{C}$. These "p-adic cohomology groups" are simply finite dimensional $\mathbb{Q}_q$-vector spaces. Furthermore, the Frobenius operator $\mathcal{F}$ acts linearly on these vector spaces and can therefore be represented by a matrix defined over $\mathbb{Q}_q$. For smooth affine curves the only non-trivial Monsky-Washnitzer cohomology groups are $H^0(\overline{C}/\mathbb{Q}_q)$ and $H^1(\overline{C}/\mathbb{Q}_q)$. Let $M_{\mathcal{F}}$ be the matrix through which the Frobenius operator $\mathcal{F}$ acts on $H^1(\overline{C}/\mathbb{Q}_q)$. Then the crux of the whole construction is that the characteristic polynomial of $M_{\mathcal{F}}$ is equal to $\chi(T)$, i.e. the characteristic polynomial of the Frobenius endomorphism $\overline{F}$.

In the remainder of this section we will give an overview of the construction of the cohomology group $H^1(\overline{C}/\mathbb{Q}_q)$. Suppose the smooth affine plane curve $\overline{C}$ is given by an equation $\overline{g}(x,y) = 0$ and let $\overline{A} = \mathbb{F}_q[x,y]/(\overline{g}(x,y))$ be its coordinate ring. Take any lift $g(x,y) \in \mathbb{Z}_q[x,y]$ of $\overline{g}(x,y)$ and let C be the curve defined by $g(x,y) = 0$ with coordinate ring $\mathbb{Z}_q[x,y]/(g(x,y))$. To compute the zeta function of $\overline{C}$ in terms of a Frobenius operator, one needs to lift the Frobenius endomorphism $\overline{F}$ on $\overline{A}$ to the $\mathbb{Z}_q$-algebra A, but in general this is not possible. Note that in the special case of elliptic curves, Satoh [105] solves this problem by using the Serre-Tate canonical lift, which does admit a lift of the Frobenius endomorphism.

A first attempt to remedy this difficulty is to work with the p-adic completion A^∞ of A, since we can lift $\overline{F}$ to A^∞. But then a new problem arises since the de Rham cohomology of A^∞, which provides the vector spaces we are looking for, is too big. For example, consider the affine line over $\mathbb{F}_p$, then $A = \mathbb{Z}_q[x]$ and A^∞ is the ring of power series $\sum_{k=0}^{\infty} r_k x^k$ with $r_i \in \mathbb{Z}_q$ and $\lim_{k\to\infty} r_k = 0$. We would like to define $H^1(\overline{A}/\mathbb{Q}_q)$ as $(A^\infty \otimes \mathbb{Q}_q)\,dx/\frac{d}{dx}(A^\infty \otimes \mathbb{Q}_q)$, but this turns out to be infinite dimensional. For example, it is clear that each term in the differential form $\sum_{n=0}^{\infty} p^n x^{p^n-1} dx$ is exact but its sum is not, since $\sum_{n=0}^{\infty} x^{p^n}$ is not in A^∞. The fundamental problem is that $\sum_{n=0}^{\infty} p^n x^{p^n-1}$ does not converge fast enough for its integral to converge as well.

Monsky and Washnitzer therefore work with a subalgebra $A^\dagger$ of A^∞, whose elements satisfy growth conditions. This *dagger ring* or *weak completion* $A^\dagger$ is defined as follows: let $A = \mathbb{Z}_q[x,y]/(g(x,y))$, then $A^\dagger := \mathbb{Z}_q\langle x,y\rangle^\dagger/(g(x,y))$, where $\mathbb{Z}_q\langle x,y\rangle^\dagger$ is the ring of overconvergent power series

$$\left\{\sum r_{i,j}x^i y^j \in \mathbb{Z}_q[[x,y]] \mid \exists\, \delta, \varepsilon \in \mathbb{R}, \varepsilon > 0, \forall (i,j) : \mathrm{ord}_p\, r_{i,j} \geq \varepsilon(i+j) + \delta\right\}.$$

The ring $A^\dagger$ satisfies $A^\dagger/(pA^\dagger) = \overline{A}$ and depends up to $\mathbb{Z}_q$-isomorphism only on $\overline{A}$. Furthermore, Monsky and Washnitzer show that if $\overline{\phi}$ is an $\mathbb{F}_q$-endomorphism of $\overline{A}$, then there exists a $\mathbb{Z}_q$-endomorphism ϕ of A lifting $\overline{\phi}$. In particular, we can lift the Frobenius endomorphism $\overline{F}$ on $\overline{A}$ to a $\mathbb{Z}_q$-endomorphism $\mathcal{F}$ on A.

For $A^\dagger$ we define the universal module $D^1(A^\dagger)$ of differentials

$$D^1(A^\dagger) := (A^\dagger\, dx + A^\dagger\, dy)/(A^\dagger(\frac{\partial g}{\partial x}\, dx + \frac{\partial g}{\partial y}\, dy)).$$

Taking the total differential of the equation $g(x, y) = 0$ gives $\frac{\partial g}{\partial x}\, dx + \frac{\partial g}{\partial y}\, dy = 0$, which explains the module $A^\dagger(\frac{\partial g}{\partial x}\, dx + \frac{\partial g}{\partial y}\, dy)$ in the above definition.

The first cohomology group is then defined as $H^1(\overline{A}/\mathbb{Z}_q) := D^1(A^\dagger)/d(A^\dagger)$ and $H^1(\overline{A}/\mathbb{Q}_q) := H^1(\overline{A}/\mathbb{Z}_q) \otimes_{\mathbb{Z}_q} \mathbb{Q}_q$ finally defines the first Monsky-Washnitzer cohomology group. Elements of $d(A^\dagger)$ are called exact differentials. One can prove that $H^1(\overline{A}/\mathbb{Q}_q)$ is well defined and is a finite dimensional vector space over $\mathbb{Q}_q$. Furthermore, for a smooth affine curve of genus g, the dimension of $H^1(\overline{A}/\mathbb{Z}_q)$ is $2g + m - 1$, where m is the number of points needed to complete the smooth affine curve to a smooth projective curve.

Replacing $A^\dagger$ with A in the above construction of the Monsky-Washnitzer cohomology group $H^1(\overline{A}/\mathbb{Q}_q)$ gives rise to the first algebraic de Rham cohomology group $H^1_{DR}(A/\mathbb{Q}_q)$. Unlike the Monsky-Washnitzer cohomology, the algebraic de Rham cohomology essentially depends on the algebra A and in general $H^1(\overline{A}/\mathbb{Q}_q)$ will not be isomorphic to $H^1_{DR}(A/\mathbb{Q}_q)$.

Let $\mathcal{F}$ be a lift of the Frobenius endomorphism $\overline{F}$ to $A^\dagger$, then $\mathcal{F}$ induces an endomorphism $\mathcal{F}_*$ on the cohomology groups. The main theorem of Monsky-Washnitzer cohomology is that the $H^i(\overline{A}/\mathbb{Q}_q)$ satisfy a Lefschetz fixed point formula.

THEOREM 4.1.1 (LEFSCHETZ FIXED POINT FORMULA) *Let $\overline{C}/\mathbb{F}_q$ be a smooth affine curve over $\mathbb{F}_q$, then the number of $\mathbb{F}_q$-rational points on $\overline{C}$ is equal to*

$$\#\overline{C}(\mathbb{F}_q) = \mathrm{Tr}\left(q\mathcal{F}_*^{-1}|H^0(\overline{A}/\mathbb{Q}_q)\right) - \mathrm{Tr}\left(q\mathcal{F}_*^{-1}|H^1(\overline{A}/\mathbb{Q}_q)\right) .$$

4.2 Hyperelliptic Curves over $\mathbb{F}_{p^n}$, p odd

Let $\mathbb{F}_q$ be a finite field with $q = p^n$ elements and fix an algebraic closure $\overline{\mathbb{F}}_q$. Throughout this section we will assume that p is a small odd prime. Let $\overline{f}(x)$ be a monic polynomial of degree $2g + 1$ over $\mathbb{F}_q$ without repeated roots and let $\overline{C}$ be the affine hyperelliptic curve defined by the equation $y^2 = \overline{f}(x)$. Kedlaya does not work with the curve $\overline{C}$ itself, but with the affine curve $\overline{C}'$ which is obtained from $\overline{C}$ by removing the locus of $y = 0$, i.e. the points $(\overline{\xi}_i, 0) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$ where $\overline{\xi}_i$ is a zero of $\overline{f}(x)$. The coordinate ring $\overline{A}$ of $\overline{C}'$ is clearly given by $\mathbb{F}_q[x, y, y^{-1}]/(y^2 - \overline{f}(x))$. It is not really necessary to work with the curve $\overline{C}'$ instead of $\overline{C}$, but in practice it is more efficient to do so (see Remark 4.2.1).

Let $\mathbb{Q}_q$ be a degree n unramified extension of $\mathbb{Q}_p$, with valuation ring $\mathbb{Z}_q$, such that $\mathbb{Z}_q/(p\mathbb{Z}_q) = \mathbb{F}_q$. Take any monic lift $f(x) \in \mathbb{Z}_q[x]$ of $\overline{f}(x)$ and let C be the smooth affine hyperelliptic curve defined by $y^2 = f(x)$. Let C' be the curve obtained from C by removing the locus of $y = 0$. Then the coordinate ring of C' is $A = \mathbb{Z}_q[x, y, y^{-1}]/(y^2 - f(x))$. Let $A^\dagger$ denote the weak completion of A. Since $\overline{F} = \sigma^n$, with σ the p-power Frobenius, it is sufficient to lift σ to an endomorphism Σ of $A^\dagger$. It is natural to define Σ as the Frobenius substitution on $\mathbb{Z}_q$ and to extend it to $A^\dagger$ by mapping x to $\Sigma(x) := x^p$ and y to $\Sigma(y)$ with

$$\begin{aligned}\Sigma(y) :&= y^p \left(1 + \frac{\Sigma(f(x)) - f(x)^p}{f(x)^p}\right)^{1/2} \\ &= y^p \sum_{i=0}^{\infty} \binom{1/2}{i} \frac{(\Sigma(f(x)) - f(x)^p)^i}{y^{2pi}}.\end{aligned} \tag{4.1}$$

An easy calculation shows that $\mathrm{ord}_p \binom{1/2}{i} \geq 0$ which implies that $\Sigma(y)$ is an element of $A^\dagger$, since p divides $\Sigma(f(x)) - f(x)^p$. Note that it is essential that y^{-1} is an element of $A^\dagger$, which explains why we compute with C' instead of C.

REMARK 4.2.1 In Chapter 5 we show that by choosing a different lift of $\overline{F}$ one can avoid working with C' altogether. However, this is less efficient since it results in a more complicated power series which converges slower than (4.1).

Since C' has dimension one, the only non-trivial Monsky-Washnitzer cohomology groups are $H^0(\overline{A}/\mathbb{Q}_q)$ and $H^1(\overline{A}/\mathbb{Q}_q)$. Finding a basis for $H^0(\overline{A}/\mathbb{Q}_q)$ is easy since by definition $H^0(\overline{A}/\mathbb{Q}_q) := \mathrm{Ker}\, d_0$, with d_0 the derivation from $A^\dagger$ into $D^1(A^\dagger)$, which implies that $H^0(\overline{A}/\mathbb{Q}_q)$ is a one dimensional $\mathbb{Q}_q$-vector space. The case $H^1(\overline{A}/\mathbb{Q}_q)$ is more difficult and Kedlaya proceeds in two steps. In the first step, he constructs a basis for the algebraic de Rham cohomology $H^1_{DR}(A/\mathbb{Q}_q)$ of A and devises reduction formulae to express any differential form on this basis. In the second step, he proves that these formulae lead to a convergent process when applied to the de Rham cohomology of $A^\dagger$, i.e. $H^1(\overline{A}/\mathbb{Q}_q)$ and concludes that the basis for the algebraic de Rham cohomology $H^1_{DR}(A/\mathbb{Q}_q)$ also is a basis for $H^1(\overline{A}/\mathbb{Q}_q)$.

The de Rham cohomology of A splits into eigenspaces under the hyperelliptic involution $\imath$: a positive eigenspace generated by $x^i/y^2\, dx$ for $i = 0, \ldots, 2g$ and a negative eigenspace generated by $x^i/y\, dx$ for $i = 0, \ldots, 2g-1$. It is clear that any differential form can be written as $\sum_{k=-B_U}^{B_L} \sum_{i=0}^{2g} a_{i,k} x^i/y^k\, dx$ with $a_{i,k} \in \mathbb{Q}_q$ and $B_U, B_L \in \mathbb{N}$. Indeed, using the equation of the curve we can repeatedly replace $q(x)f(x)$ by $q(x)y^2$. A differential $P(x)/y^s\, dx$ with $P(x) \in \mathbb{Q}_q[x]$ and $s \in \mathbb{N}$ can be reduced as follows. Since $f(x)$ has no repeated roots, we can always write the polynomial $P(x)$ as $P(x) = U(x)f(x) + V(x)f'(x)$. Using the fact that $d(V(x)/y^{s-2})$ is exact, one obtains

$$\frac{P(x)}{y^s}\, dx \equiv \left(U(x) + \frac{2V'(x)}{(s-2)}\right) \frac{dx}{y^{s-2}},$$

where $\equiv$ means equality modulo exact differentials. This congruence can be used to reduce a differential form involving negative powers of y to the case $s = 1$ and $s = 2$. A differential $P(x)/y\,dx$ with $\deg P = m \geq 2g$ can be reduced by repeatedly subtracting suitable multiples of the exact differential $d(x^{i-2g}y)$ for $i = m, \ldots 2g$. Finally, it is clear that the differential $P(x)/y^2\,dx$ is congruent to $(P(x) \bmod f(x))/y^2\,dx$ modulo exact differentials. A differential of the form $P(x)y^s\,dx$ with $P(x) \in \mathbb{Q}_q[x]$ and $s \in \mathbb{N}$ is exact if s is even and equal to $P(x)f(x)^{\lceil s/2 \rceil}/y\,dx$ if s is odd and thus can be reduced using the above reduction formula. This ends the first step in finding a basis for $H^1(\overline{A}/\mathbb{Q}_q)$.

In the second step, Kedlaya proves two lemmata which bound the denominators introduced during the above reduction process. The next lemma summarizes the result and is proven in [61, Lemma 2-3].

LEMMA 4.2.1 (KEDLAYA) *Let $A(x) \in \mathbb{Z}_q[x]$ be a polynomial of degree $\leq 2g$, then for $k \in \mathbb{N}$ the reduction of $A(x)y^{2k+1}\,dx$ becomes integral upon multiplication by $p^{\lfloor \log_p((2g+1)(m+1)-2) \rfloor}$ and the reduction of $A(x)/y^{2k+1}\,dx$ becomes integral upon multiplication by $p^{\lfloor \log_p(2k+1) \rfloor}$.*

This lemma shows that the basis for $H^1_{DR}(A/\mathbb{Q}_q)$ also generates $H^1(\overline{A}/\mathbb{Q}_q)$. Indeed, the overconvergence of the dagger ring implies that the minimum valuation of the coefficients of $A(x)y^{2k+1}$ in Lemma 4.2.1 grows as a linear function of $|k|$, whereas the valuation of the denominators introduced during the reduction process only grows as a logarithmic function of $|k|$. Furthermore, both $H^1(\overline{A}/\mathbb{Q}_q)$ and $H^1_{DR}(A/\mathbb{Q}_q)$ have dimension $4g+1$, so we conclude that the basis for $H^1_{DR}(A/\mathbb{Q}_q)$ is also a basis for $H^1(\overline{A}/\mathbb{Q}_q)$.

The first Monsky-Washnitzer cohomology group $H^1(\overline{A}/\mathbb{Q}_q)$ decomposes as the direct sum of the $\imath$-invariant part $H^1(\overline{A}/\mathbb{Q}_q)^+$ on which $\imath$ acts trivially and the $\imath$-anti-invariant part $H^1(\overline{A}/\mathbb{Q}_q)^-$ on which $\imath$ acts as multiplication by -1. Note that the $\imath$-invariant part of $\overline{A}$ is given by $\mathbb{F}_q[x, y^2, y^{-2}]/(y^2 - f(x))$ which clearly is isomorphic to $\mathbb{F}_q[x, (f(x))^{-1}]$. The latter ring is the coordinate ring of the curve $\mathbb{A}^1 \setminus V_f$ with V_f the set of zeros of $f(x)$. The Lefschetz fixed point formula applied to $\overline{C}'$ and $\mathbb{A}^1 \setminus V_f$ then gives

$$\begin{aligned}
\#\overline{C}'(\mathbb{F}_{q^k}) &= \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^0(\overline{A}/\mathbb{Q}_q)\right) - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)\right) \\
&= q^k - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^+\right) - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^-\right) \\
&= q^k - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\mathbb{A}^1 \setminus V_f/\mathbb{Q}_q)\right) - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^-\right) \\
&= \#(\mathbb{A}^1 \setminus V_f)(\mathbb{F}_{q^k}) - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^-\right) \\
&= q^k - r_k - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^-\right)
\end{aligned} \tag{4.2}$$

with r_k the number of zeros of $\overline{f}(x)$ defined over $\mathbb{F}_{q^k}$. Let $\widetilde{C}$ be the unique smooth projective curve birational to $\overline{C}'$, then

$$\begin{aligned}\#\widetilde{C}(\mathbb{F}_{q^k}) &= \#\overline{C}'(\mathbb{F}_{q^k}) + r_k + 1 \\ &= q^k + 1 - \operatorname{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^-\right) \\ &= q^k + 1 - \sum_{i=1}^{2g} \alpha_i^k,\end{aligned} \tag{4.3}$$

with α_i the eigenvalues of $q\mathcal{F}_*^{-1}$ on $H^1(\overline{A}/\mathbb{Q}_q)^-$. The Weil conjectures [137] imply that $\#\widetilde{C}(\mathbb{F}_{q^k}) = q^k + 1 - \sum_{i=1}^{2g} \beta_i^k$ for all $k > 0$ where the β_i satisfy $\beta_i \beta_{g+i} = q$ for $i = 1, \ldots, 2g$ and the indices are taken modulo $2g$. Therefore, we can label the β_i such that $\alpha_i = \beta_i$ for $i = 1, \ldots, 2g$. Since α_i is an eigenvalue of $q\mathcal{F}_*^{-1}$ on $H^1(\overline{A}/\mathbb{Q}_q)^-$, it is clear that $q/\alpha_i = \alpha_{i+g}$ is an eigenvalue of $\mathcal{F}_*$ on $H^1(\overline{A}/\mathbb{Q}_q)^-$. Let $\chi(t)$ be the characteristic polynomial of $\mathcal{F}_*$ on $H^1(\overline{A}/\mathbb{Q}_q)^-$, then the α_i are the zeros of $\chi(t)$ and we conclude that

$$Z(\widetilde{C}/\mathbb{F}_q; t) = \frac{t^{2g}\chi(1/t)}{(1-t)(1-qt)}.$$

Since $\mathcal{F}_* = \Sigma_*^n$, it suffices to compute the matrix M through which Σ_* acts on the anti-invariant part $H^1(\overline{A}/\mathbb{Q}_q)^-$ of $H^1(\overline{A}/\mathbb{Q}_q)$. Indeed, repeatedly applying Σ_* to both sides of the equation

$$\Sigma_* \begin{pmatrix} \frac{dx}{y} \\ \vdots \\ \frac{x^{2g-1}\,dx}{y} \end{pmatrix} = M \begin{pmatrix} \frac{dx}{y} \\ \vdots \\ \frac{x^{2g-1}\,dx}{y} \end{pmatrix}$$

leads to

$$\mathcal{F}_* \begin{pmatrix} \frac{dx}{y} \\ \vdots \\ \frac{x^{2g-1}\,dx}{y} \end{pmatrix} = \Sigma^{n-1}(M) \cdots \Sigma(M) M \begin{pmatrix} \frac{dx}{y} \\ \vdots \\ \frac{x^{2g-1}\,dx}{y} \end{pmatrix}.$$

The action of Σ_* on the basis of $H^1(\overline{A}/\mathbb{Q}_q)^-$ can be computed as

$$\Sigma_*(\frac{x^i\,dx}{y}) = \frac{x^{p(i+1)-1}\,dx}{\Sigma(y)},$$

for $i = 0, \ldots, 2g-1$. Given a sufficiently precise approximation to $1/\Sigma(y)$, we can use the aforementioned reduction formulae to express $\Sigma_*(x^i/y\,dx)$ on the basis of $H^1(\overline{A}/\mathbb{Q}_q)^-$ and compute the matrix M. The zeta function of $\widetilde{C}$ can then be recovered from the characteristic polynomial of the matrix $\Sigma^{n-1}(M) \cdots \Sigma(M) M$ through which the Frobenius $\mathcal{F}_* = \Sigma_*^n$ acts on $H^1(\overline{A}/\mathbb{Q}_q)^-$.

A detailed complexity analysis of the above algorithm can be found in [61], which proves the following theorem,.

THEOREM 4.2.1 (KEDLAYA) *There exists a deterministic algorithm to compute the zeta function of a smooth hyperelliptic curve of genus g defined over a finite field $\mathbb{F}_{p^n}$ with $p > 2$ using $O(g^{4+\varepsilon}n^{3+\varepsilon})$ bit-operations and $O(g^3n^3)$ space for p fixed.*

4.3 Artin-Schreier Curves over $\mathbb{F}_{2^n}$

4.3.1 Cohomology of Artin-Schreier Curves over $\mathbb{F}_{2^n}$

Let $\mathbb{F}_q$ be a finite field with $q = 2^n$ elements and fix an algebraic closure $\overline{\mathbb{F}}_q$. Let $\mathbb{Q}_q$ be a degree n unramified extension of $\mathbb{Q}_2$ and let $\mathbb{Z}_q$ be its valuation ring with residue field $\mathbb{Z}_q/(2\mathbb{Z}_q) = \mathbb{F}_q$. The hyperelliptic curves we will consider in this section are isomorphic over $\mathbb{F}_q$ to an affine curve given by an equation of the form

$$\overline{C}_{m,\overline{f}} : y^2 - x^m y - \overline{f}(x) = 0,$$

with $0 \leq m \leq g$, $\overline{f} \in \mathbb{F}_q[x]$ monic of degree $2g+1$ and such that $\overline{C}_{m,\overline{f}}$ is non-singular. Let $\mathfrak{p} : \overline{C}_{m,\overline{f}}(\overline{\mathbb{F}}_q) \to \mathbb{A}^1(\overline{\mathbb{F}}_q)$ be the projection map on the x-axis, then the branch locus of $\mathfrak{p}$ is empty if and only if $m = 0$ and consists of the singleton $\{0\}$ if and only if $m > 0$. Without loss of generality we may assume that $\overline{f}(0) = 0$ if $m > 0$, i.e. that $(0,0)$ is the unique ramification point of $\mathfrak{p}$. Indeed, the isomorphism defined by $x \mapsto x$ and $y \mapsto y + \overline{f}(0)^{1/2}$ shows that we can replace $\overline{f}(x)$ with $\overline{f}(x) - \overline{f}(0) + x^m\overline{f}(0)^{1/2}$, which clearly is divisible by x if $m > 0$. Note that since $\overline{C}_{m,\overline{f}}$ is non-singular we have $\overline{f}'(0) \neq 0$.

Let $\overline{H}_m(x)$ be defined as $\overline{H}_m(x) := 1$ for $m = 0$ and $\overline{H}_m(x) := x$ for $m > 0$, i.e. $\overline{H}_m(\theta)$ is zero if and only if the point $(\theta, 0)$ ramifies. Let $\overline{C}'_{m,\overline{f}}$ be the curve obtained from $\overline{C}_{m,\overline{f}}$ by deleting the locus of $\overline{H}_m(x) = 0$. Then the coordinate ring of $\overline{C}'_{m,\overline{f}}$ is $\overline{A}_{m,\overline{f}} := \mathbb{Z}_q[x, y, \overline{H}_m(x)^{-1}]/(y^2 - x^m y - \overline{f}(x))$. Analogous to the odd characteristic case, it is not really necessary to work with the affine curve $\overline{C}'_{m,\overline{f}}$ instead of $\overline{C}_{m,\overline{f}}$, but again it turns out to be more efficient.

Take any lift $f \in \mathbb{Z}_q[x]$ of $\overline{f}$, with the restrictions that f should be monic and of degree $2g+1$ and that $f(0) = 0$ for $m > 0$. Let $H_m(x) := 1$ for $m = 0$ and $H_m(x) := x$ for $m > 0$ and let $C'_{m,f}$ be the curve obtained from

$$C_{m,f} : y^2 - x^m y - f(x) = 0$$

by deleting the locus of $H_m(x) = 0$. Note that for $m > 0$ this effectively removes only the point $(0,0)$, since $f(0) = 0$. If $f(0) \neq 0$, then the locus of $H_m(x) = 0$ would consist of the two points $(0, \pm\sqrt{f(0)})$ and the dimension of the algebraic the Rham cohomology group $H^1_{DR}(A_{m,f}/\mathbb{Q}_q)$ would be bigger than the dimension of the Monsky-Washnitzer cohomology group $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)$. The coordinate ring of $C'_{m,f}$ is $A_{m,f} := \mathbb{Z}_q[x, y, H_m(x)^{-1}]/(y^2 - x^m y - f(x))$ and there exists an involution $\imath$ on $A_{m,f}$ which sends x to x and y to $-y + x^m$.

Let $A^\dagger_{m,f}$ be the dagger ring of $A_{m,f}$. Using the equation of the curve we can always represent elements of $A^\dagger_{m,f}$ as a series $\sum_{l=-\infty}^{+\infty}(a_l + b_l y)x^l$ with $a_l, b_l \in \mathbb{Z}_q$.

If $m = 0$ then all a_l, b_l with $l < 0$ are zero. Furthermore, the growth condition on the dagger ring implies that there exists real numbers δ and $\epsilon > 0$ such that $\mathrm{ord}_2(a_l) \geq \epsilon \cdot |l| + \delta$ and $\mathrm{ord}_2(b_l) \geq \epsilon \cdot |l+1| + \delta$.

Lift the 2-nd power Frobenius σ on $\mathbb{F}_q$ to the Frobenius substitution Σ on $\mathbb{Z}_q$. We extend Σ to an endomorphism of $A^\dagger_{m,f}$ by mapping x to x^2 and y to $\Sigma(y)$, with

$$(\Sigma(y))^2 - x^{2m}\Sigma(y) - \Sigma(f(x)) = 0 \quad \text{and} \quad \Sigma(y) \equiv y^2 \bmod 2.$$

Using Newton iteration we can compute the solution to the above equations as an element of the 2-adic completion of $A_{m,f}$, but it is not immediately clear that there exists a solution in $A^\dagger_{m,f}$. The existence of such a solution follows immediately from a theorem by Bosch [12], but since we need an explicit estimate of the rate of convergence, we prove the following lemma.

LEMMA 4.3.1 *For $k \geq 1$, let $W_k(x,y) := \sum_{l=-L_k}^{A_k} a_l x^l + \sum_{l=-L_k}^{B_k} b_l x^l y \in A_{m,f}$ satisfy*

$$W_k(x,y)^2 - x^{2m}W_k(x,y) - \Sigma(f(x)) \equiv 0 \bmod 2^k \text{ and } W_k(x,y) \equiv y^2 \bmod 2$$

with $a_{A_k} \neq 0$, $b_{B_k} \neq 0$, $a_{-L_k} \neq 0$ or $b_{-L_k} \neq 0$ and such that $a_l = 0$ or $\mathrm{ord}_2(a_l) < k$ for $-L_k \leq l \leq A_k$ and $b_l = 0$ or $\mathrm{ord}_2(b_l) < k$ for $-L_k \leq l \leq B_k$. Then the degrees A_k, B_k and L_k can be bounded for $k \geq 2$ as

$$\begin{aligned} A_k &\leq 2(k-1)(\deg f - 2m) + 2m, \\ B_k &\leq 2(k-2)(\deg f - 2m) + \deg f - m, \\ L_k &\leq 2(k-1)(2m) - 2m. \end{aligned} \tag{4.4}$$

PROOF: An easy calculation shows that $W_1(x,y) \equiv f(x) + x^m y \bmod 2$ and

$$W_2(x,y) \equiv \frac{(f(x)^2 - \Sigma(f(x))) + x^{2m}f(x)}{x^{2m}} + y\frac{2x^m f(x) + x^{3m}}{x^{2m}},$$

so that W_2 indeed satisfies the lemma.

Newton iteration on $Z^2 - x^{2m}Z - \Sigma(f(x)) = 0$ gives

$$W_{k+1} \equiv W_k - \frac{W_k^2 - x^{2m}W_k - \Sigma(f(x))}{2W_k - x^{2m}} \bmod 2^{k+1} \equiv \frac{W_k^2 - \Sigma(f(x))}{x^{2m}} \bmod 2^{k+1}.$$

Let $\alpha_k(x) := \sum_{l=-L_k}^{A_k} a_l x^l$, $\beta_k(x) := \sum_{l=-L_k}^{B_k} b_l x^l$ such that $W_k = \alpha_k + \beta_k y$. Note that $W_k \equiv W_{k-1} \bmod 2^{k-1}$, so we can define

$$\Delta_{\alpha,k}(x) := \frac{\alpha_k(x) - \alpha_{k-1}(x)}{2^{k-1}} \quad \text{and} \quad \Delta_{\beta,k}(x) := \frac{\beta_k(x) - \beta_{k-1}(x)}{2^{k-1}},$$

for $k \geq 1$ with $\Delta_{\alpha,0}(x) := \Delta_{\beta,0}(x) := 0$. It is clear that W_k can be written as

$$W_k = \Delta_{\alpha,1} + 2\Delta_{\alpha,2} + \cdots + 2^{k-1}\Delta_{\alpha,k} + y\left(\Delta_{\beta,1} + 2\Delta_{\beta,2} + \cdots + 2^{k-1}\Delta_{\beta,k}\right).$$

Plugging this into the Newton iteration gives the following equation

$$x^{2m} W_{k+1} \equiv \sum_{\substack{1 \le i < j \\ i+j-1 < k+1}} 2^{i+j-1} \left(\Delta_{\alpha,i} \Delta_{\alpha,j} + (f + x^m y) \Delta_{\beta,i} \Delta_{\beta,j} \right) - \Sigma(f) +$$
$$y \sum_{i+j-1 < k+1} 2^{i+j-1} \Delta_{\alpha,i} \Delta_{\beta,j} + \sum_{2(i-1) < k+1} 2^{2(i-1)} \left(\Delta_{\alpha,i}^2 + (f + x^m y) \Delta_{\beta,i}^2 \right) \bmod 2^{k+1}.$$

Since $\deg \Delta_{\alpha,i} \le A_i$ and $\deg \Delta_{\beta,i} \le B_i$, we get that A_{k+1} is less or equal than

$$\max \Big(\deg \Sigma(f), \max_{i+j<k+2} (A_i + A_j, B_i + B_j + \deg f),$$
$$\max_{2i<k+3} (2A_i, 2B_i + \deg f) \Big) - 2m\,.$$

Using the bounds given in (4.4) for A_i and B_i we see that A_{k+1} also satisfies the bounds (4.4). Note that we have to take into account the values for $\deg \Delta_{\alpha_1} = \deg f$ and $\deg \Delta_{\beta_1} = m$ since these do not satisfy the bounds (4.4), but this does not cause any problems. A similar reasoning for B_{k+1} and L_{k+1} shows that these also satisfy the given bounds. $\square$

The above lemma shows that we can indeed lift the q-power Frobenius $\overline{F}$ to an endomorphism $\mathcal{F}$ on the dagger ring $A_{m,f}^{\dagger}$; it suffices to take $\mathcal{F} := \Sigma^n$. Before we can actually compute the zeta function using the Lefschetz fixed point theorem, we need to determine a basis of the $\mathbb{Q}_q$-vector space $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)$.

We first prove that $x^i y\, dx$ with $i = 0, \ldots, 2g-1$, and $\frac{dx}{x}$ if $m > 0$, form a basis for the algebraic de Rham cohomology $H^1_{DR}(A_{m,f}/\mathbb{Q}_q)$ of $A_{m,f}$. The extra $\frac{dx}{x}$ for $m > 0$ is caused by the fact that we removed the point $(0,0)$ from the curve $C_{m,f}$. Every element of $H^1_{DR}(A_{m,f}/\mathbb{Q}_q)$ can be written as a linear combination of differentials of the form $x^k y^l\, dx$, $x^k y^l\, dy$ with $k \in \mathbb{Z}$, $l \in \mathbb{N}$ and $k \ge 0$ if $m = 0$. Using the equation of the curve, we can reduce to $l = 0$ or 1. Since $d(x^k y)$ and $d(x^k y^2)$ are exact, we conclude that $H^1_{DR}(A_{m,f}/\mathbb{Q}_q)$ is generated by differentials of the form $x^i y\, dx$ with $i \in \mathbb{Z}$ ($i \in \mathbb{N}$ for $m = 0$) and $\frac{dx}{x}$ if $m > 0$.

Rewriting the equation of the curve as $(2y - x^m)^2 = 4f(x) + x^{2m}$ and differentiating gives the equality $(2y - x^m)\, d(2y - x^m) = (2f'(x) + m x^{2m-1})\, dx$. For all $k > 0$ we therefore have

$$\begin{aligned} x^k (2f'(x) + m x^{2m-1})(2y - x^m)\, dx &= x^k (2y - x^m)^2\ d(2y - x^m) \\ &\equiv -\frac{k}{3} x^{k-1} (2y - x^m)^3\, dx \\ &= -\frac{k}{3} x^{k-1} (4f(x) + x^{2m})(2y - x^m)\, dx, \end{aligned} \tag{4.5}$$

where $\equiv$ means equality modulo exact differentials. Thus we conclude that $\left[x^k (2f'(x) + m x^{2m-1}) + \frac{k}{3} x^{k-1} (4f(x) + x^{2m}) \right] y\, dx$ is exact. The polynomial between brackets has degree $2g+k$ and leading coefficient $2(2g+1)+4\frac{k}{3} \neq 0$. A similar argument for $k=0$ shows that $(2f'(x)+mx^{2m-1})y\,dx$ is exact and clearly has degree $2g$ in x. With these formulae we can express $x^{2g+k}y\,dx$ for $k \geq 0$ as a linear combination of $x^i y\,dx$ with $0 \leq i < 2g$.

For $m>0$ and $k<0$ the formulae (4.5) are still valid, but now the conclusion is that $\left[x^k(2f'(x)+mx^{2m-1})+\frac{k}{3}x^{k-1}(4f(x)+x^{2m})\right] y\,dx + \beta\frac{dx}{x}$ is exact for some suitable element $\beta \in \mathbb{Q}_q$. The Laurent polynomial between brackets has valuation at zero k, since $f(0)=0$, but $f'(0) \neq 0$. The term x^k has coefficient $2f'(0)(1+\frac{2k}{3})$ which clearly is different from zero, since $f'(0) \neq 0$. Therefore we can express all differentials of the form $x^k y\,dx$ with $k<0$ as a linear combination of $x^i y\,dx$ for $i=0,\ldots,2g-1$ and $\frac{dx}{x}$ if $m>0$.

A consequence of Lemma 4.3.3 will be that all these differential forms are linear independent and thus form a basis for the algebraic de Rham cohomology $H^1_{DR}(A_{m,f}/\mathbb{Q}_q)$. To show that this also is a basis for the Monsky-Washnitzer cohomology $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)$, we need to bound the denominators introduced during the reduction process. Therefore we prove the following two lemmata.

Lemma 4.3.2 *Let* $A := \mathbb{Z}_q[x,y]/(y^2-x^m y-f(x))$ *and suppose that*

$$x^r y\,dx = \sum_{i=0}^{2g-1} a_i x^i y\,dx + ds, \tag{4.6}$$

with $r \in \mathbb{N}$, $a_i \in \mathbb{Q}_q$ *and* $s \in A \otimes \mathbb{Q}_q$. *Then* $2^c a_i \in \mathbb{Z}_q$, $2^{c'}s-\beta \in A$, *where* $c = 3+\lfloor \log_2(r+g+1)\rfloor$, $c' = 1+c+\lfloor\log_2(2g+m)\rfloor$ *and* β *some suitable element in* $\mathbb{Q}_q$.

Proof: The proof has two distinct parts. The first part is similar to Kedlaya's argument in [61, Lemma 3], and is based on a local analysis around the point at infinity of the curve $C_{m,f}$. Put $t = x^g/y$, then one easily verifies that

$$x = t^{-2}\left(1+\sum_{j=1}^{\infty}\alpha_j t^j\right), \quad y = t^{-2g-1}\left(1+\sum_{j=1}^{\infty}\beta_j t^j\right), \tag{4.7}$$

with $\alpha_j, \beta_j \in \mathbb{Z}_q$. To see this, put $z = 1/x$, rewrite the equation of the curve $C_{m,f}$ as $z - z^{g-m+1} - t^2 z^{2g+1} f(1/z) = 0$ and write z as a power series in t using Newton iteration. The relation (4.6) can be rewritten as

$$2^{c-1}x^r(2y-x^m)\,dx = \sum_{i=0}^{2g-1} 2^{c-1}a_i x^i(2y-x^m)\,dx + dS, \tag{4.8}$$

with $S \in A \otimes \mathbb{Q}_q$. Considering the involution of A which sends x to x and $2y-x^m$ to $-(2y-x^m)$, we see that we can write $S = \sum_{i=0}^{N} A_i x^i(2y-x^m)$, with

N big enough and $A_i \in \mathbb{Q}_q$. This yields

$$2^{c-1}x^r(2y - x^m)\,dx - \sum_{i=0}^{2g-1} 2^{c-1}a_i x^i(2y - x^m)\,dx$$
$$= d\left(\sum_{i=0}^{N} A_i x^i(2y - x^m)\right). \quad (4.9)$$

In the above equation we express x and y in terms of t using equalities (4.7). Since $x^i y = t^{-2i-2g-1} + \cdots$, we get $x^i(2y - x^m)\,dx = (-4t^{-2i-2g-4} + \cdots)\,dt$, which leads to

$$2^{c-1} \sum_{j=-\max(2r+2g+4,6g+2)} \gamma_j t^j\,dt = d\left(\sum_{i=0}^{N} 2A_i(t^{-2i-2g-1} + \cdots) - A_i(t^{-2i-2m} + \cdots)\right),$$

with $\gamma_j \in \mathbb{Q}_q$ for all j and $\gamma_j \in \mathbb{Z}_q$ when $j < -2(2g-1) - 2g - 4 = -6g - 2$. Integrating with respect to t and dividing by 2 gives

$$\sum_{j \geq -\max(2r+2g+3,6g+1)} \gamma'_j t^j = \sum_{i=0}^{N} A_i(t^{-2i-2g-1} + \cdots) - \sum_{i=0}^{N} \frac{A_i}{2}(t^{-2i-2m} + \cdots), \quad (4.10)$$

with $\gamma'_j \in \mathbb{Q}_q$ for all j and $\gamma'_j \in \mathbb{Z}_q$ when $j < -6g - 1$. Indeed the integration process introduces denominators which become integral after multiplication with $2^{\lfloor \log(2r+2g+2) \rfloor} = 2^{c-2}$ if $r \geq 2g - 1$. A first consequence of (4.10) is that $A_i = 0$ for all $i > \max(r + 1, 2g)$. We claim that (4.10) implies that $A_i \in \mathbb{Z}_q$ for all $i > 2g$. Suppose the claim is false. Then let i_0 be the largest integer with $i_0 > 2g$ and $A_{i_0} \notin \mathbb{Z}_q$. Note that $-2i_0 - 2g - 1 < -6g - 1$, since $i_0 > 2g$. Hence the monomials in the left hand side of (4.10) with degree $\leq -2i_0 - 2g - 1$ have coefficients in $\mathbb{Z}_q$. Moreover the monomials of degree $< -2i_0 - 2g - 1$ in the first sum in the right hand side of (4.10) also have coefficients in $\mathbb{Z}_q$, but this is false for the monomial of degree $-2i_0 - 2g - 1$. Hence the second sum in the right hand side of (4.10) contains a monomial of degree $-2i_0 - 2g - 1$ whose coefficient is not in $\mathbb{Z}_q$. That means that there is a maximal i_1 with $A_{i_1}/2 \notin \mathbb{Z}_q$ and $-2i_1 - 2m \leq -2i_0 - 2g - 1$. Because of parity we have that $-2i_1 - 2m < -2i_0 - 2g - 1$. Hence the right hand side of (4.10) contains a monomial of degree $-2i_1 - 2m < -2i_0 - 2g - 1$ whose coefficient is not in $\mathbb{Z}_q$. But this contradicts what we said about the left hand side. This finishes the claim that $A_i \in \mathbb{Z}_q$ for all $i > 2g$.

We now turn to the second part of the proof. Note that $(2y - x^m)^2 = v(x)$ with $v(x) := 4f(x) + x^{2m}$. Moreover, we have $d(2y - x^m) = \frac{w(x)}{2y - x^m}\,dx$, where $w(x) := 2f'(x) + mx^{2m-1}$. We will use these formulae to deduce from (4.9) a relation which does not involve y. Multiplying (4.9) with $\frac{2y - x^m}{dx} = \frac{w(x)}{d(2y - x^m)}$ gives

$$2^{c-1}x^r v(x) - \sum_{i=0}^{2g-1} 2^{c-1}a_i x^i v(x) = \sum_{i=0}^{N} A_i i x^{i-1} v(x) + \sum_{i=0}^{N} A_i x^i w(x).$$

We rewrite this in the form

$$\left(\sum_{i=0}^{2g-1} 2^{c-1}a_i x^i\right) v(x) + \left(\sum_{i=0}^{2g} A_i i x^{i-1}\right) v(x) + \left(\sum_{i=0}^{2g} A_i x^i\right) w(x) = F(x), \quad (4.11)$$

where

$$F(x) := 2^{c-1}x^r v(x) - \sum_{i=2g+1}^{N} A_i i x^{i-1} v(x) - \sum_{i=2g+1}^{N} A_i x^i w(x) \qquad (4.12)$$

is a polynomial over $\mathbb{Z}_q$, since $A_i \in \mathbb{Z}_q$ for all $i > 2g$. To get rid of the disturbing factor 2 in the definition of $w(x)$, we consider

$$u(x) := \frac{1}{2}(xw(x) - mv(x)) = xf'(x) - 2mf(x)\,.$$

We rewrite (4.11) in such a way that $w(x)$ gets replaced by $u(x)$:

$$\left(\sum_{i=0}^{2g} (2^{c-1}a_{i-1} + iA_i + mA_i)x^i\right) v(x) + \left(\sum_{i=0}^{2g} 2A_i x^i\right) u(x) = xF(x), \quad (4.13)$$

with the convention that $a_{-1} = 0$. We consider (4.13) as a linear system of $4g+2$ equations in the unknowns $2^{c-1}a_{i-1} + iA_i + mA_i$ and $2A_i$ for $i = 0, \ldots, 2g$. The determinant of this system is the resultant $\mathrm{Res}(v, u)$ of u and v, because $\deg v(x) = \deg u(x) = 2g + 1$. Since the leading coefficient of u is a unit, we have $\mathrm{Res}(u, v) =$ unit $\cdot \prod_{u(\theta)=0} v(\theta)$, where θ ranges over all roots of u in the algebraic closure of $\mathbb{Q}_q$. All these roots θ have non-negative valuation.

Suppose first that $m = 0$. Then $\mathrm{Res}(u, v)$ is a unit in $\mathbb{Z}_q$ since $v(\theta) = 4f(\theta)+1$ is a unit for each root θ of u. The determinant of the system being a unit, we conclude that $2A_i$ and $2^{c-1}a_{i-1} + iA_i + mA_i$ are in $\mathbb{Z}_q$ for $i = 0, \ldots, 2g$. Hence $2^c a_i \in \mathbb{Z}_q$ and $2S \in A$. So for $m = 0$ the lemma then follows directly from (4.8).

Suppose now that $m \geq 1$. The restrictions on f imply that $f(0) = 0$ and $f'(0) \neq 0 \bmod 2$. Hence 0 is a common zero of u and v and $\mathrm{Res}(u, v) = 0$. From (4.12) it follows that $F(0) = 0$, hence $A_0 = 0$ by (4.11), since $w(0) = 2f'(0) \neq 0$. We now consider (4.13) divided by x^2 as a linear system of $4g$ equations in the unknowns $2^{c-1}a_{i-1} + iA_i + mA_i$ and $2A_i$ for $i = 1, \ldots, 2g$. The determinant of this system is the resultant $\mathrm{Res}(\frac{v}{x}, \frac{u}{x})$. Let θ be a root of $u/x = f'(x) - 2mf(x)/x$, then θ has valuation zero since $f'(0) \neq 0 \bmod 2$. Hence $v(\theta) = 4f(\theta) + \theta^{2m}$ is a unit. Thus $\mathrm{Res}(\frac{v}{x}, \frac{u}{x})$ is a unit and both $2A_i$ and $2^{c-1}a_{i-1} + iA_i + mA_i$ are in $\mathbb{Z}_q$ for $i = 1, \ldots, 2g$. We now continue as in the case $m = 0$. This ends the proof of the lemma. $\square$

Remark 4.3.1 Lemma 4.3.2 remains valid when we replace $\sum_{i=0}^{2g-1}$ by $\sum_{i=\kappa}^{2g-1+\kappa}$ whenever $r \geq \kappa \in \mathbb{N}$. The proof is the same, except that we also have to show that $A_i = 0$ for all $i < \kappa$. This follows from (4.9) by a local analysis at a point on the curve with $x = 0$.

LEMMA 4.3.3 *With the above notation and $m > 0$, suppose that*

$$x^{-r}y\,dx = \sum_{i=0}^{2g-1} a_i x^i y\,dx + b\frac{dx}{x} + ds, \tag{4.14}$$

where $r \in \mathbb{N}$, $a_i, b \in \mathbb{Q}_q$ and $s \in A_{m,f} \otimes \mathbb{Q}_q$. Then $2^c a_i \in \mathbb{Z}_q$, $2^{c'} b \in \mathbb{Z}_q$, $2^{c'} s - \beta \in A_{m,f}$, with $c = 3 + \lfloor \log_2(r+1) \rfloor$, $c' = 1 + c + \lfloor \log_2(2g+m) \rfloor$ and $\beta \in \mathbb{Q}_q$.

REMARK 4.3.2 Actually one can take $c = 3 + \lfloor \log_2(r-2) \rfloor$ when $r \geq 3$ and $c = 0$ when $0 \leq r \leq 2$.

PROOF: The proof again consists of two distinct parts. The first part is similar to Kedlaya's argument in [61, Lemma 2] and is based on a local analysis around the ramification point $(0,0)$ on the curve. In the completion of the local ring of the curve at $(0,0)$ we can write

$$x = \gamma_2 y^2 + \sum_{j \geq 3} \gamma_j y^j, \tag{4.15}$$

with $\gamma_j \in \mathbb{Z}_q$ and γ_2 a unit in $\mathbb{Z}_q$. Indeed, to see this use the equation of the curve and the conditions $f(0) = 0$, $f'(0) \not\equiv 0 \bmod 2$, to express x as a power series in y using Newton iteration.

Considering the involution as in the proof of Lemma 4.3.2, we can transform relation (4.14) to the form

$$2^{c-1}x^{-r}(2y - x^m)\,dx - \sum_{i=0}^{2g-1} 2^{c-1} a_i x^i (2y - x^m)\,dx = d\left(\sum_{i=-N}^{M} A_i x^i (2y - x^m)\right),$$

with N and M large enough integers. Using the expansion at infinity given by the formulae (4.7) in the proof of Lemma 4.3.2 and substituting them in the above equation, one verifies that we can take $M = 2g$.

Expressing x in terms of y using (4.15) and dividing by 2 we obtain

$$2^{c-2} \sum_{j \geq -2r+2} \gamma'_j y^j\,dy = d\Big(\sum_{i=-N}^{2g} A_i(\gamma_2^i y^{2i+1} + \cdots)\Big) - d\Big(\sum_{i=-N}^{2g} \frac{A_i}{2}(\gamma_2^{i+m} y^{2i+2m} + \cdots)\Big),$$

with $\gamma'_j \in \mathbb{Q}_q$ for all j and $\gamma'_j \in \mathbb{Z}_q$ when $j \leq 0$. Integrating the left hand side of this equation with respect to y yields a series whose terms of degree ≤ 1 have coefficients in $\mathbb{Z}_q$. Thus the same argument as in the proof of Lemma 4.3.2 shows that $A_i \in \mathbb{Z}_q$ for all $i \leq 0$. Moreover if $r = 0$, then $A_i = 0$ when $i \leq 0$. This terminates the first part of the proof.

We still have to proof that $2^c a_i \in \mathbb{Z}_q$ for $i = 0, \ldots, 2g-1$ and that $2A_i \in \mathbb{Z}_q$ for $i = 1, \ldots, 2g$. This follows by the same argument as in the second part of the proof of Lemma 4.3.2. However, in the present situation A_0 might not be zero, but we proved already that $A_0 \in \mathbb{Z}_q$. Therefore we bring the terms which contain A_0 to the other side in equation (4.13) of the proof of Lemma 4.3.2. This then ends the proof of Lemma 4.3.3. $\square$

REMARK 4.3.3 Lemma 4.3.3 remains valid when we replace $\sum_{i=0}^{2g-1}$ by $\sum_{i=-\kappa}^{2g-1-\kappa}$ whenever $r \geq \kappa \in \mathbb{N}$. The proof is exactly the same.

REMARK 4.3.4 If $r = 0$, then in the above proof the A_i are zero for all $i \leq 0$, and for $0 \leq i \leq 2g-1$ the a_i are completely determined by (4.13) as we saw by considering resultants. This shows that the $x^i y\,dx$ for $i = 0, \ldots, 2g-1$ and $\frac{dx}{x}$ are linearly independent in $H^1_{DR}(A_{m,f}/\mathbb{Q}_q)$.

Lemma 4.3.2 and 4.3.3 show that the basis for $H^1_{DR}(A_{m,f}/\mathbb{Q}_q)$ is a generating set for $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)$, since the reduction process converges. Indeed, for $a_k x^k y \in A^\dagger_{m,f}$ the valuation of a_k grows as a linear function of $|k|$, while the valuation of the denominators introduced during the reduction of $a_k x^k y\,dx$ are only logarithmic in $|k|$.

The first Monsky-Washnitzer cohomology group $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)$ decomposes as the direct sum of the $\imath$-invariant part $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^+$ on which $\imath$ acts trivially and the $\imath$-anti-invariant part $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^-$ on which $\imath$ acts by multiplication by -1. Note that $\frac{dx}{x}$ is a basis for the invariant part $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^+$ for $m > 0$ and $\mathcal{F}_*$ acts on it by multiplication with q. Hence for $m > 0$ the Lefschetz fixed point theorem yields

$$\begin{aligned}
\#\overline{C}_{m,\overline{f}}(\mathbb{F}_{q^k}) &= 1 + \#\overline{C}'_{m,\overline{f}}(\mathbb{F}_{q^k}) \\
&= 1 + \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^0(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)\right) - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)\right) \\
&= q^k - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^-\right).
\end{aligned}$$

Let $\widetilde{C}_{m,\overline{f}}$ be the unique smooth projective curve birational to $\overline{C}_{m,\overline{f}}$, then

$$\#\widetilde{C}_{m,\overline{f}}(\mathbb{F}_{q^k}) = q^k + 1 - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^-\right) = q^k + 1 - \sum_{i=1}^{2g} \alpha_i^k,$$

with α_i the eigenvalues of $q\mathcal{F}_*^{-1}$ on $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^-$. By the Weil conjectures [137] there exists a polynomial with integer coefficients

$$\chi(t) = t^{2g} + a_1 t^{2g-1} + \cdots + a_{2g},$$

whose roots $\beta_1, \ldots, \beta_{2g}$ satisfy $\#\widetilde{C}_{m,\overline{f}}(\mathbb{F}_{q^k}) = q^k + 1 - \sum_{i=1}^{2g} \beta_i^k$ for all $k > 0$. Furthermore, $\beta_i \beta_{g+i} = q$ and $|\beta_i| = \sqrt{q}$ for $i = 1, \ldots, 2g$ where the indices are taken modulo $2g$. This implies that we can label the β's such that $\alpha_i = \beta_i$ for $i = 1, \ldots, 2g$. Since $\alpha_i \alpha_{g+i} = q$, the α_i are also the eigenvalues of $\mathcal{F}_*$ on $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^-$. The zeta function of $\widetilde{C}_{m,\overline{f}}$ can be written as

$$Z(\widetilde{C}_{m,\overline{f}}/\mathbb{F}_q; t) = \frac{t^{2g}\chi(1/t)}{(1-t)(1-qt)},$$

so it suffices to recover $\chi(t)$ from an approximation of the characteristic polynomial of the matrix $M_{\mathcal{F}} = \Sigma^{n-1}(M)\cdots\Sigma(M)M$ through which the Frobenius $\mathcal{F}_* = \Sigma_*^n$ acts on $H^1(\overline{A}/\mathbb{Q}_q)^-$.

4.3.2 Detailed Algorithm

Using the results of the previous section, we describe an algorithm for computing the characteristic polynomial of Frobenius $\chi(t)$ and the zeta function of a smooth projective Artin-Schreier curve $\widetilde{C}_{m,\overline{f}}$ of genus g over $\mathbb{F}_q$ with $q = 2^n$. We have shown that we can compute $\chi(t) = t^{2g} + a_1 t^{2g-1} + \cdots + a_{2g}$ as the characteristic polynomial of $\mathcal{F}_*$ on $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^-$. The Weil conjectures [137] imply that $q^{g-i}a_i = a_{2g-i}$, so it suffices to compute $a_1, \ldots, a_g$, and for $i = 1, \ldots, g$ the a_i can be bounded by

$$|a_i| \leq \binom{2g}{i} q^{i/2} \leq \binom{2g}{g} q^{g/2} \leq 2^{2g} q^{g/2}.$$

Thus to determine the zeta function, we have to compute the action of $\mathcal{F}_*$ on a basis of $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^-$ modulo 2^B with $B \geq \left\lceil \log_2 \left(2\binom{2g}{g} q^{g/2}\right)\right\rceil$. However, we need to take into account the loss of precision caused by the reduction process.

Suppose we have an approximation of $\Sigma(y)$ to precision N, then Lemma 4.3.1 shows that the degrees of the anti-invariant part $\sum_{l=-L_N}^{B_N} b_l x^l y$ of $\Sigma(y)$ are bounded by

$$B_N \leq 2(N-2)(\deg f - 2m) + \deg f - m \quad \text{and} \quad L_N \leq 2(N-1)(2m) - 2m\,.$$

Since we have to reduce the differential $2x^{4g-1}\sum_{l=-L_N}^{B_N} b_l x^l y\, dx$, Lemma 4.3.2 and 4.3.3 imply that the loss of precision will be bounded by

$$3 + \max\{\lfloor \log_2(B_N + 5g)\rfloor, \lfloor \log_2(L_N)\rfloor\} \leq 3 + \lfloor \log_2(2N \deg f)\rfloor\,.$$

Since the result should be correct modulo 2^B, we have to choose N such that

$$N - \lfloor \log_2(2N \deg f)\rfloor \geq B + 3. \tag{4.16}$$

Remark 4.3.5 Note that since the matrix M is not necessarily defined over $\mathbb{Z}_q$, we could lose an extra cn bits of precision during the computation of $M_{\mathcal{F}} = \Sigma^{n-1}(M)\cdots\Sigma(M)M$, where 2^c is the largest denominator appearing in M. By Lemmata 4.3.1 and 4.3.2, c is bounded by $O(\log g)$ independently of n. In theory we would therefore have to replace the bound B by $B + cn$, which does not change the complexity of the algorithm. In practice however it turns out that the largest denominator appearing in $M_{\mathcal{F}}$ is always the same as the largest denominator appearing in M and therefore it is not necessary to increase B. This phenomenon can be heuristically explained as follows: since the eigenvalues of $\mathcal{F}_* = \Sigma_*^n$ on $H^1(\overline{A}/\mathbb{Q}_q)^-$ have non-negative 2-adic valuation there is a $\mathbb{Z}_q$-submodule of $H^1(\overline{A}/\mathbb{Q}_q)^-$ which is stable under the action of Σ_*.

ALGORITHM 4.3.1 (ARTIN_SCHREIER_ZETA_FUNCTION)

INPUT: Artin-Schreier curve $\overline{C}_{m,\overline{f}}$ over $\mathbb{F}_q$ given by equation $y^2 - x^m y = \overline{f}(x)$.

OUTPUT: The zeta function $Z(\widetilde{C}_{m,\overline{f}}/\mathbb{F}_q;t)$.

1. Compute $N \in \mathbb{N}$ with $N - 3 - \lfloor \log_2(2N \deg f + g) \rfloor \geq B$;
2. $\overline{f} = \overline{f} - \overline{f}(0)+$ sqrt($\overline{f}(0)$)x^m; $f = \mathbb{Z}_q[x] \leftarrow \overline{f} \bmod 2^N$;
3. $\alpha_N(x), \beta_N(x) =$ Lift_p_Frobenius_y(m, f, N);
4. For $i = 0$ To $2g - 1$ Do
   - 4.1. Red$_i(x) =$ Reduce_MW_Cohomology($2x^{2i+1}\beta_N(x), m, f, B$);
   - 4.2. For $j = 0$ To $2g - 1$ Do $M[i][j] =$ Coeff(Red$_i, j$);
5. Norm$_M = \Sigma^{n-1}(M) \cdots \Sigma(M) \cdot M \bmod 2^B$;
6. $\chi(t) =$ Characteristic_Pol(Norm$_M$) $\bmod 2^B$;
7. For $i = 0$ To $i = g$ Do
   - 7.1. If Coeff($\chi, 2g - i$) $> \binom{2g}{i} q^{i/2}$ Then
     Coeff($\chi, 2g - i$) $-= 2^B$;
   - 7.2. Coeff(χ, i) $= q^{g-i}$ Coeff($\chi, 2g - i$);
8. Return $Z(\widetilde{C}_{m,\overline{f}}/\mathbb{F}_q;t) = \dfrac{t^{2g}\chi(1/t)}{(1-t)(1-qt)}$.

ALGORITHM 4.3.2 (LIFT_P_FROBENIUS_ON_Y)

INPUT: Artin-Schreier curve $C_{m,f}$ over $\mathbb{Z}_q$ and precision N.
OUTPUT: $\alpha_N, \beta_N \in \mathbb{Z}_q[x, x^{-1}]$ with $\Sigma(y) \equiv \alpha_N + \beta_N y \mod 2^N$.

1. If $N = 1$ Then $\alpha_N = f(x)$; $\beta_N = x^m$;
2. Else
   2.1. $N' = \lceil \frac{N}{2} \rceil$;
   2.2. $\alpha_{N'}, \beta_{N'} =$ `Lift_p_Frobenius_y`$(m, f(x), N')$;
   2.3. $\gamma_N, \delta_N =$ `Series_Invert`$(1 - \frac{2(\alpha_{N'}(x) + \beta_{N'}(x)y)}{x^{2m}}, m, f(x), N)$;
   2.4. $\mu_N \equiv -\alpha_{N'} + x^{-2m}(\alpha_{N'}^2 + \beta_{N'}^2 f(x) - \Sigma(f(x))) \mod 2^N$;
   2.5. $\nu_N \equiv -\beta_{N'} + x^{-2m}(2\alpha_{N'}\beta_{N'} + \beta_{N'}^2 x^m) \mod 2^N$;
   2.6. $\alpha_N \equiv \alpha_{N'} + \mu_N \gamma_N + \nu_N \delta_N f(x) \mod 2^N$;
   2.7. $\beta_N \equiv \beta_{N'} + \mu_N \delta_N + \nu_N(\gamma_N + \delta_N x^m) \mod 2^N$;
3. Return α_N, β_N.

ALGORITHM 4.3.3 (REDUCE_MW_COHOMOLOGY)

INPUT: Artin-Schreier curve $C_{m,f}$, precision B and element $G \in \mathbb{Z}_q[x, x^{-1}]$.
OUTPUT: $S \in K[x]$, with $\deg S < 2g$ such that $Sy\,dx \sim Gy\,dx \mod 2^B$.

1. Compute $N \in \mathbb{N}$ with $N - 3 - \lfloor \log_2(2N \deg f) \rfloor \geq B$;
2. $D =$ `Degree`(G); $V =$ `Valuation`(G); $T = G$;
3. For $i = D$ To $2g$ By -1
   3.1. $P \equiv x^{i-2g}(2f' + mx^{2m-1}) + \frac{i-2g}{3} x^{i-2g-1}(4f + x^{2m}) \mod 2^N$;
   3.2. $T \equiv T - ($`Coeff`$(T, i) \cdot P)/(2(2g+1) + \frac{4(i-2g)}{3}) \mod 2^N$;
4. For $i = V$ To -1
   4.1. $P \equiv x^i(2f' + mx^{2m-1}) + \frac{i}{3} x^{i-1}(4f + x^{2m}) \mod 2^N$;
   4.2. $T \equiv T - ($`Coeff`$(T, i) \cdot P)/(2(1 + \frac{2i}{3})f'(0)) \mod 2^N$;
5. Return $S \equiv T \mod 2^B$.

For this $\mathbb{Z}_q$-submodule we can take for instance the canonical image of the crystalline cohomology of C over $\mathbb{Z}_q$. Note that the $\mathbb{Z}_q$-submodule generated by $x^i y\, dx$ for $i = 0, \ldots, 2g-1$ is not canonical and in general not stable under Σ_*. Let A_0 be the matrix that expresses $x^i y\, dx$ for $i = 0, \ldots, 2g-1$ in terms of a basis of such a stable $\mathbb{Z}_q$-submodule and let A be A_0 times a power of 2 such that A is a matrix over $\mathbb{Z}_q$ which is not zero modulo 2. Then $M = A^{-1}U\Sigma(A)$ where U is the matrix of Σ_* with respect to the new basis. Note that U is a matrix over $\mathbb{Z}_q$ and that the norm of M equals $A^{-1}U\Sigma(U)\cdots\sigma^{n-1}(U)A$. Thus the loss of precision is no more than $2d$ bits where d is the 2-adic valuation of $\det(A)$. If U and A are generic enough then $|c-d|$ is small. Furthermore, the bound (4.16) turns out to be slightly larger than what is needed and compensates for the loss of $2d$ bits.

The function Artin_Schreier_Zeta_Function given in Algorithm 4.3.1 computes the zeta function of of an Artin-Schreier curve $\overline{C}_{m,\overline{f}}$ over a finite field $\mathbb{F}_q$ of characteristic 2. In step 2 we replace $\overline{C}_{m,\overline{f}}$ by an isomorphic curve such that $\overline{f}(0) = 0$ if $m > 0$ and lift the resulting curve to $\mathbb{Z}_q$, subject to the condition that $f(0) = 0$ if $m > 0$. Step 3 computes a sufficiently precise approximation of $\Sigma(y)$ using the subroutine Lift_p_Frobenius_y given in Algorithm 4.3.2. This essentially is a Newton iteration on the equation $Y^2 - x^{2m}Y - \Sigma(f(x)) = 0$ starting with $Y_0 = y^2$, but in each iteration step we replace y^2 by $x^m y + f(x)$. In step 2.3 of Algorithm 4.3.2 we use the function Series_Invert which takes as input an invertible element S of the dagger ring $A^{\dagger}_{m,f}$ and computes the inverse of S up to precision N, i.e.

$$\texttt{Series\_Invert}(S, m, f, N) = \gamma_N(x, x^{-1}) + \delta_N(x, x^{-1})y \equiv S^{-1} \bmod 2^N.$$

This function is an easy Newton iteration on the equation $SY - 1 = 0$, so we have omitted the pseudo-code. Note that for efficiency reasons one should reuse the intermediate results obtained in the previous call. In step 4 of Algorithm 4.3.1 we compute an approximation M of the matrix through which the p-th power Frobenius Σ_* acts on a basis of $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^-$. Given any differential of the form $Gy\, dx$ with $G \in \mathbb{Z}_q[x, x^{-1}]$ we compute its reduction on the basis of $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^-$ using Reduce_MW_Cohomology given in Algorithm 4.3.2. The result of this function is a polynomial $R \in \mathbb{Q}_q[x]$, with $\deg R < 2g$ such that for a given precision N we have the following equivalence in $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^-$

$$G(x, x^{-1})y\, dx \equiv R(x)y\, dx \bmod 2^N.$$

Once we have found the matrix M, we compute $\texttt{Norm}_M = \Sigma^{n-1}(M)\cdots\Sigma(M)M$ which is an approximation of the action of $\mathcal{F}_*$ on $H^1(\overline{A}_{m,\overline{f}}/\mathbb{Q}_q)^-$. In step 6 we determine its characteristic polynomial modulo 2^B and in step 7 we recover the characteristic polynomial of Frobenius $\chi(t)$ from the first g coefficients. Finally we return the zeta function of $\widetilde{C}_{m,\overline{f}}$ as

$$Z(\widetilde{C}_{m,\overline{f}}/\mathbb{F}_q; t) = \frac{t^{2g}\chi(1/t)}{(1-t)(1-qt)}.$$

4.3.3 Complexity Analysis

In this section we analyse the space and time complexity of Algorithm 4.3.1 for a genus g Artin-Schreier curve $\overline{C}_{m,\overline{f}}$ over a finite field $\mathbb{F}_q$ with $q = 2^n$.

The basic objects we compute with are Laurent polynomials with coefficients in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ for some precision N. Every element of this extension requires $O(nN)$ space, so we can perform multiplication and division in time $O(n^\mu N^\mu)$.

In step 3 we call the subroutine LIFT_P_FROBENIUS_Y which needs the Frobenius substitution of $O(g)$ elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$. Using a polynomial basis to represent $\mathbb{Z}_q$, one such substitution requires $O(n^{\mu+1}N^\mu)$ bit-operations. Since we have to repeat this computation for all coefficients of f we need $O(gn^{\mu+1}N^\mu)$ bit-operations to determine $\Sigma(f)$.

The remaining steps of LIFT_P_FROBENIUS_Y consist of a Newton lifting on the quadratic equation $Y^2 - x^{2m}Y - \Sigma(f) = 0$. Since the necessary precision halves in every recursive call, we see that the total complexity is determined by the last iteration step. Lemma 4.3.1 bounds the size of the Laurent polynomials in this last step by $2N \deg f$. Since the coefficients are in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$, each of these Laurent polynomials requires $O(gnN^2)$ storage space, so we can perform multiplication and division of these objects in $O(g^\mu n^\mu N^{2\mu})$ time.

In step 4 we call the function REDUCE_MW_COHOMOLOGY $2g$ times on Laurent polynomials with $O(gN)$ coefficients over $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$. To reduce each term in this polynomial, we need to compute the multiplication of a degree $2g+1$ polynomial over $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ with an integer modulo 2^N and subtract it from the intermediate result. This results in $O(gnN^\mu)$ bit-operations for each term or $O(g^2nN^{\mu+1})$ for each Laurent polynomial. Since we need to repeat this $2g$ times the total complexity of step 4 amounts to $O(g^3nN^{\mu+1})$ bit-operations.

In step 5 we need to determine the norm of a $2g \times 2g$ matrix M over $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ as $\Sigma^{n-1}(M)\cdots\Sigma(M)M$. This can be done by computing

$$M_{i+1} = \Sigma^{2^i}(M_i)M_i \quad \text{for} \quad i = 0, \ldots, \lfloor \log_2 n \rfloor + 1$$

and combining these to recover the norm of M. Multiplication of two matrices takes $O(g^3)$ ring operations or $O(g^3n^\mu N^\mu)$ bit-operations. To compute

$$\Sigma^{2^i}(M_i) \quad \text{for} \quad i = 0, \ldots, \lfloor \log_2 n \rfloor + 1,$$

we need $O(g^2 \log n)$ applications of powers of Σ. If we use a polynomial basis to represent $\mathbb{Z}_q$, this can be accomplished in $O(g^2n^{\mu+1}N^\mu \log n)$ bit-operations.

In step 6 we need to compute the characteristic polynomial of a $2g \times 2g$ matrix over $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$, which can be done using the classical algorithm based on the Hessenberg form [16, Section 2.2.4]. The complexity of this algorithm is $O(g^3)$ ring operations or $O(g^3n^\mu N^\mu)$ bit-operations.

Since the first g coefficients of the characteristic polynomial can be bounded by $2^{2g}q^{g/2}$, we have $B \leq ng$ and equation (4.16) then shows that N itself also is $O(gn)$. This implies that the overall time complexity of Algorithm 4.3.1 is

$$O(n^{\mu+1}g^{\mu+2}(g^2n + n^\mu \log n))$$

and the overall space complexity amounts to $O(g^3n^3)$, which proves the following theorem.

THEOREM 4.3.1 *There exists a deterministic algorithm to compute the zeta function of a smooth Artin-Schreier curve of genus g defined over a finite field $\mathbb{F}_q$ with $q = 2^n$ elements using $O(g^3n^3(g^2 + (\log n)^2 \log\log n) \log gn \log\log gn)$ bit-operations and $O(g^3n^3)$ space.*

4.3.4 Implementation and Numerical Results

In this section we present running times of an implementation of Algorithm 4.3.1 in the `C` programming language and give some examples of Jacobians of Artin-Schreier curves with almost prime group order.

The basic operations on integers modulo 2^N for $N \leq 256$ are implemented in assembly language. Elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo a degree n irreducible polynomial, which is either a trinomial or a pentanomial. For multiplication of elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$, polynomials over $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ and Laurent series over $\mathbb{Z}_q/(2^N\mathbb{Z}_q)[x]$ we used a combination of Karatsuba [60] and Toom [126] multiplication.

Table 4.1 contains running times and memory usages of our algorithm for genus 2, 3 and 4 Artin-Schreier curves over various finite fields of characteristic 2. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1. Note that the field degrees are chosen such that gn is constant across each row.

Table 4.1: Running time (s) and memory usage (MB) for genus 2, 3 and 4 Artin-Schreier curves over $\mathbb{F}_{2^n}$

	Genus 2 curves		Genus 3 curves		Genus 4 curves	
gn	Time	Mem	Time	Mem	Time	Mem
120	7.1	2.9	9.3	3.1	10.9	3.4
144	11.9	4.1	14.1	4.4	16.1	4.5
168	18.6	7.0	22.0	7.4	28.6	8.1
192	32.6	9.2	38.5	9.7	42.6	10.2
240	82.7	17.9	97.9	18.8	109	19.7
288	148	25.3	177	26.5	217	32.8
336	258	40.8	296	42.9	341	43.9
384	490	61.6	595	64.7	619	66.6
480	1393	99.1	1619	104	1730	107

Artin-Schreier Curve Examples: Low Genus Curves

In this subsection we give some examples of Jacobians of Artin-Schreier curves with almost prime group order. The correctness of these results is easily proved by multiplying a random divisor with the given group order and verifying that the result is principal, i.e. is the zero element in the Jacobian $J_{\overline{C}}(\mathbb{F}_q)$. The given curves are non-supersingular, since the coefficient a_1 of $\chi(t)$ is odd [44]. Furthermore, all curves withstand the MOV-FR attack [42, 81].

Genus 2 Artin-Schreier curve over $\mathbb{F}_{2^{83}}$

Let $\mathbb{F}_{2^{83}}$ be defined as $\mathbb{F}_2[w]/\overline{P}(w)$ with $\overline{P}(w) = w^{83} + w^7 + w^4 + w^2 + 1$ and consider the random Artin-Schreier curve $\overline{C}_2$ of genus 2 defined by

$$y^2 + xy = x^5 + \sum_{i=1}^{4} f_i x^i \,,$$

where

$$\begin{aligned} f_1 &= \texttt{32F5702317481C763A60D} & \quad f_2 &= \texttt{1E55F75EC726E4CE27932} \\ f_3 &= \texttt{4AE0138A0655DC0F8B8D5} & \quad f_4 &= \texttt{46A72EE2E0A435E80A9EE} \end{aligned}$$

The group order of the Jacobian $J_{\widetilde{C}_2}$ of $\overline{C}_2$ over $\mathbb{F}_{2^{83}}$ is

$$\# J_{\widetilde{C}_2} = 2 \cdot 46768052394586940539891778534901849764240032635053 \,,$$

where the last factor is prime. The coefficients a_1 and a_2 of the characteristic polynomial of Frobenius $\chi(t) = t^4 + a_1 t^3 + a_2 t^2 + a_3 t + a_4$ are given by

$$a_1 = -403838389927 \quad \text{and} \quad a_2 = 258141381429417736562278 4 \,.$$

Genus 3 Artin-Schreier curve over $\mathbb{F}_{2^{59}}$

Let $\mathbb{F}_{2^{59}}$ be defined as $\mathbb{F}_2[w]/\overline{P}(w)$ with $\overline{P}(w) = w^{59} + w^7 + w^4 + w^2 + 1$ and consider the random Artin-Schreier curve $\overline{C}_3$ of genus 3 defined by

$$y^2 + xy = x^7 + \sum_{i=1}^{6} f_i x^i \,,$$

where

$$\begin{aligned} f_1 &= \texttt{749FBD6529A6BE6} & \quad f_2 &= \texttt{787C8FD4E3BE0AB} \\ f_3 &= \texttt{364379150158499} & \quad f_4 &= \texttt{723B968312B8E6B} \\ f_5 &= \texttt{408D539886ABACE} & \quad f_6 &= \texttt{7762021B54EE8E7} \end{aligned}$$

The group order of the Jacobian $J_{\widetilde{C}_3}$ of $\overline{C}_3$ over $\mathbb{F}_{2^{59}}$ is

$$\# J_{\widetilde{C}_3} = 2 \cdot 95780971254332681347837855333631196111431141330759 07 \,,$$

where the last factor is prime. The coefficients a_1, a_2 and a_3 of the characteristic polynomial of Frobenius $\chi(t) = t^6 + a_1 t^5 + a_2 t^4 + a_3 t^3 + a_4 t^2 + a_5 t + a_6$ are given by

$$\begin{aligned} a_1 &= 299634811, \\ a_2 &= 388773818941833216, \\ a_3 &= 1874976781627117473491845 12 \,. \end{aligned}$$

Genus 4 Artin-Schreier curve over $\mathbb{F}_{2^{47}}$

Let $\mathbb{F}_{2^{47}}$ be defined as $\mathbb{F}_2[w]/\overline{P}(w)$ with $\overline{P}(w) = w^{47} + w^5 + 1$ and consider the random Artin-Schreier curve $\overline{C}_4$ of genus 4 defined by

$$y^2 + xy = x^9 + \sum_{i=1}^{8} f_i x^i\,,$$

where

$f_1 =$ `0F82877AADD8`	$f_2 =$ `5E109F46E079`	$f_3 =$ `77936D8E5B83`
$f_4 =$ `1C217ED2FD56`	$f_5 =$ `052CE3614BCE`	$f_6 =$ `4DE01C9A582A`
$f_7 =$ `6F2C067A3D9C`	$f_8 =$ `0B9A21EA7868`	

The group order of the Jacobian $J_{\widetilde{C}_4}$ of $\overline{C}_4$ over $\mathbb{F}_{2^{47}}$ is

$$\#J_{\widetilde{C}_4} = 2 \cdot 19615943835209431554861697594003370758946955755501264283\,,$$

where the last factor is prime. The coefficients a_1, a_2, a_3 and a_4 of the characteristic polynomial of Frobenius $\chi(t) = t^8 + a_1t^7 + a_2t^6 + a_3t^5 + a_4t^4 + a_5t^3 + a_6t^2 + a_7t + a_8$ are given by

$$\begin{aligned} a_1 &= 6544181, \\ a_2 &= 351218501550080, \\ a_3 &= 1427631072901840502784, \\ a_4 &= 5844444065073703103619923968 0. \end{aligned}$$

Artin-Schreier Curve Examples: High Genus Curves

The asymptotic behavior of Algorithm 4.3.1 is polynomial in both the degree of the finite field and the genus of the curve. Furthermore, the complexity analysis shows that the reduction of the differentials dominates the running time for large geni. This is illustrated by Table 4.2 which contains running times for Artin-Schreier curves over $\mathbb{F}_2$ for varying genus $20 \leq g \leq 240$.

4.4 Hyperelliptic Curves over $\mathbb{F}_{2^n}$

4.4.1 Cohomology of Hyperelliptic Curves over $\mathbb{F}_{2^n}$

Let $\mathbb{F}_q$ be a finite field with $q = 2^n$ elements and fix an algebraic closure $\overline{\mathbb{F}}_q$. Consider the smooth affine hyperelliptic curve $\overline{C}$ of genus g defined by the equation

$$\overline{C} : y^2 + \overline{h}(x)y = \overline{f}(x)\,,$$

with $\overline{h}(x), \overline{f}(x) \in \mathbb{F}_q[x]$, $\overline{f}(x)$ monic of degree $2g + 1$ and $\deg \overline{h} \leq g$. Write $\overline{h}(x)$ as $\overline{c} \cdot \prod_{i=0}^{s}(x - \overline{\theta}_i)^{m_i}$ with $\overline{\theta}_i \in \overline{\mathbb{F}}_q$, $\overline{c} \in \mathbb{F}_q \setminus \{0\}$ the leading coefficient of $\overline{h}(x)$ and define $\overline{H}(x) = \prod_{i=0}^{s}(x - \overline{\theta}_i) \in \mathbb{F}_q[x]$. If $\overline{h}(x)$ is a constant, we set $\overline{H}(x) = 1$. Without loss of generality we can assume that $\overline{H}(x) \,|\, \overline{f}(x)$. Indeed,

Table 4.2: Running time (s) and memory usage (MB) for Artin-Schreier curves over $\mathbb{F}_2$ and genus $20 \leq g \leq 240$

Genus	Lift $\mathcal{F}$	Reduction	Total	Mem
20	1.06	0.84	1.94	0.32
40	6.97	16.8	24.1	1.37
60	23.4	99.3	124	2.49
80	51.0	311	366	4.39
100	94.4	874	975	7.88
120	158	2086	2256	10.8
160	392	7359	7784	20.9
200	1065	24582	25716	32.5
240	1289	66722	68152	50.8

the isomorphism defined by $x \mapsto x$ and $y \mapsto y + \sum_{i=0}^{s} \overline{b}_i x^i$ transforms the curve in

$$y^2 + \overline{h}(x)y = \overline{f}(x) - \sum_{i=0}^{s} \overline{b}_i^2 x^{2i} - \overline{h}(x) \sum_{i=0}^{s} \overline{b}_i x^i .$$

The polynomial $\overline{H}(x)$ will divide the right hand side of the above equation if and only if $\overline{f}(\overline{\theta}_j) = \sum_{i=0}^{s} \overline{b}_i^2 \cdot \overline{\theta}_j^{2i}$ for $j = 0, \ldots, s$. This is a system of linear equations in the indeterminates $\overline{b}_i^2$ and its determinant is a Vandermonde determinant. Since the $\overline{\theta}_j$ are the zeros of a polynomial defined over $\mathbb{F}_q$, the system of equations is invariant under the q-th power Frobenius $\overline{F}$ and it follows that the $\overline{b}_i^2$ and therefore the $\overline{b}_i$ are elements of $\mathbb{F}_q$. We conclude that we can always assume that $\overline{H}(x) \,|\, \overline{f}(x)$.

Let $\overline{\pi} : \overline{C}(\overline{\mathbb{F}}_q) \to \mathbb{A}^1(\overline{\mathbb{F}}_q)$ be the projection on the x-axis. It is clear that $\overline{\pi}$ ramifies at the points $(\overline{\theta}_i, 0) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$ for $i = 0, \ldots, s$ where $\overline{H}(\overline{\theta}_i) = 0$. Note that the ordinate of these points is zero, since we assumed that $\overline{H}(x) \,|\, \overline{f}(x)$. Let $\overline{C}'$ be the curve obtained from $\overline{C}$ by removing the points $(\overline{\theta}_i, 0)$ for $i = 0, \ldots, s$. Then the coordinate ring $\overline{A}$ of $\overline{C}'$ is

$$\mathbb{F}_q[x, y, \overline{H}(x)^{-1}]/(y^2 + \overline{h}(x)y - \overline{f}(x))$$

Analogous to the odd characteristic case, it is not really necessary to work with the affine curve $\overline{C}'$ instead of $\overline{C}$, but again it turns out to be more efficient. The coordinate ring of $\overline{C}'$ contains the inverse of $\overline{H}(x)$ which will allow us to choose a particular lift of the Frobenius endomorphism $\overline{F}$ of $\overline{A}$.

Let $\mathbb{Q}_q$ be a degree n unramified extension of $\mathbb{Q}_2$ with valuation ring $\mathbb{Z}_q$ and residue field $\mathbb{Z}_q/(2\mathbb{Z}_q) \cong \mathbb{F}_q$. Write $\overline{h}(x) = \overline{c} \cdot \prod_{i=0}^{r} \overline{P}_i(x)^{t_i}$, where the $\overline{P}_i(x)$ are monic and irreducible over $\mathbb{F}_q$. Let $D = \max_i t_i$, then $\overline{h}(x)$ divides $\overline{H}(x)^D$, since we have the identity $\overline{H}(x) = \prod_{i=0}^{r} \overline{P}_i(x)$. Lift $\overline{P}_i(x)$ for $i = 0, \ldots, r$ to any monic polynomial $P_i(x) \in \mathbb{Z}_q[x]$. Define $H(x) = \prod_{i=0}^{r} P_i(x)$ and let

$h(x) = c \cdot \prod_{i=0}^{r} P_i(x)^{t_i}$, with c any lift of $\overline{c}$ to $\mathbb{Z}_q$. Since $\overline{H}(x)$ divides $\overline{f}(x)$ we can define $Q_{\overline{f}}(x) = \overline{f}(x)/\overline{H}(x)$. Let $Q_f(x) \in \mathbb{Z}_q[x]$ be any monic lift of $Q_{\overline{f}}(x)$ and finally set $f(x) = H(x)Q_f(x)$. The result is that we have now constructed a lift C of the curve $\overline{C}$ to $\mathbb{Z}_q$ defined by the equation

$$C : y^2 + h(x)y = f(x).$$

Note that due to the careful construction of C we have the following properties: $H(x) \mid h(x)$, $H(x) \mid f(x)$ and $h(x) \mid H(x)^D$. Let $\mathbb{Q}_q^{\mathrm{ur}}$ be the maximal unramified extension of $\mathbb{Q}_q$ with valuation ring $\mathbb{Z}_q^{\mathrm{ur}}$. For $k = 0, \ldots, s$, let θ_k be the zeros of $H(x)$ and note that these are units in $\mathbb{Z}_q^{\mathrm{ur}}$. Furthermore, let $\pi : C(\overline{\mathbb{Q}}_q) \to \mathbb{A}^1(\overline{\mathbb{Q}}_q)$ be the projection on the x-axis, then the $(\theta_k, 0)$ are ramification points of π.

Consider the curve C' obtained from C by deleting the points $(\theta_k, 0)$ for $k = 0, \ldots, s$, then the coordinate ring A of C' is

$$\mathbb{Z}_q[x, y, H(x)^{-1}]/(y^2 + h(x)y - f(x))$$

and there exists an involution $\imath$ on A which sends x to x and y to $-y - h(x)$. Let $A^\dagger$ denote the weak completion of A. Using the equation of the curve, we can represent any element of $A^\dagger$ as a series $\sum_{i=-\infty}^{+\infty} (U_i(x) + V_i(x)y)S(x)^i$, with the degree of $U_i(x)$ and $V_i(x)$ smaller that the degree of $S(x)$, where $S(x) = H(x)$ if $\deg H > 0$ and $S(x) = x$ if $H(x) = 1$. The growth condition on the dagger ring implies that there exist real numbers δ and $\epsilon > 0$ such that $\mathrm{ord}_2(U_i(x)) \geq \epsilon \cdot |i| + \delta$ and $\mathrm{ord}_2(V_i(x)) \geq \epsilon \cdot |i+1| + \delta$, where $\mathrm{ord}_2(P(x))$ is defined as $\min_j \mathrm{ord}_2(p_j)$ for $P(x) = \sum p_j x^j \in \mathbb{Q}_q[x]$.

Lift the 2-nd power Frobenius σ on $\mathbb{F}_q$ to the Frobenius substitution Σ on $\mathbb{Z}_q$. We extend Σ to an endomorphism of $A^\dagger$ by mapping x to x^2 and y to $\Sigma(y)$, with

$$\Sigma(y)^2 + \Sigma(h(x))\Sigma(y) - \Sigma(f(x)) = 0 \quad \text{and} \quad \Sigma(y) \equiv y^2 \bmod 2.$$

Using Newton lifting we can compute the solution to the above equations as an element of the 2-adic completion of A as

$$W_{k+1} \equiv W_k - \frac{W_k^2 + \Sigma(h(x))\, W_k - \Sigma(f(x))}{2W_k + \Sigma(h(x))} \bmod 2^{k+1}. \tag{4.17}$$

The only remaining difficulty in the above Newton iteration is that we have to invert $2W_k + \Sigma(h(x))$ in the ring A^∞. Since $h(x) \mid H(x)^D$, it makes sense to define $Q_H(x) := H(x)^D/h(x)$ and we clearly have $1/h(x) = Q_H(x)/H(x)^D$. We can now compute the inverse of $2W_k + \Sigma(h(x))$ as

$$\frac{Q_H(x)^2}{H(x)^{2D} \cdot \left(1 + \frac{Q_H(x)^2(2W_k + \Sigma(h(x)) - h(x)^2)}{H(x)^{2D}}\right)}. \tag{4.18}$$

Note that $\Sigma(h(x)) \equiv h(x)^2 \bmod 2$, which implies that the denominator in the above formula is invertible in A^∞. Contrary to the odd characteristic case it is not immediately clear that the solution $W := \lim_{k\to+\infty} W_k$ is an element of $A^\dagger$. The existence of such a solution follows immediately from a theorem by Bosch [12], but since we need an explicit estimate of the rate of convergence, we prove the following lemma.

LEMMA 4.4.1 *For* $k \geq 1$, *let*

$$W_k = \sum_{i=-L_k}^{A_k} U_i(x)S(x)^i + \sum_{i=-L_k}^{B_k} V_i(x)S(x)^i y \in A,$$

with $S(x) = H(x)$ *if* $\deg H > 0$, $S(x) = x$ *if* $H(x) = 1$ *and* $\deg U_i < \deg S$, $\deg V_i < S$ *satisfy*

$$W_k^2 + \Sigma(h(x))\, W_k - \Sigma(f(x)) \equiv 0 \bmod 2^k \quad \textit{and} \quad W_k \equiv y^2 \bmod 2$$

with $U_{A_k} \neq 0$, $V_{B_k} \neq 0$, $U_{-L_k} \neq 0$ *or* $V_{-L_k} \neq 0$ *and such that* $U_i = 0$ *or* $\mathrm{ord}_2(U_i(x)) < k$ *for* $-L_k \leq i \leq A_k$ *and similarly,* $V_i = 0$ *or* $\mathrm{ord}_2(V_i(x)) < k$ *for* $-L_k \leq i \leq B_k$. *Then* A_k, B_k *and* L_k *can be bounded for* $k \geq 2$ *as*

$$\begin{aligned} A_k &\leq 2(k-1)(d_S^f - 2d_S^h) + 2d_S^h, \\ B_k &\leq 2(k-2)(d_S^f - 2d_S^h) + (d_S^f - d_S^h), \\ L_k &\leq 4(k-1)D - 2D, \end{aligned} \tag{4.19}$$

with $d_S^f := \deg f / \deg S$ *and* $d_S^h := \deg h / \deg S$.

PROOF: An easy calculation shows that $W_1 \equiv f(x) - h(x)y \bmod 2$, therefore $A_1 \leq d_S^f$, $B_1 \leq d_S^h$, $L_1 \leq 0$ and

$$W_2 \equiv \frac{\Sigma(f(x)) - f(x)^2 - \Sigma(h(x))f(x)}{h(x)^2} + y\frac{\Sigma(h(x)) + 2f(x)}{h(x)} \bmod 4$$

which implies that W_2 satisfies the lemma. The Newton iteration (4.17) can be rewritten as

$$h(x)^2\, W_{k+1} \equiv -W_k^2 + (h(x)^2 - \Sigma(h(x)))\, W_k + \Sigma(f(x)) \bmod 2^{k+1}.$$

Let $\alpha_k(x) := \sum_{i=-L_k}^{A_k} U_i(x)S(x)^i$ and $\beta_k(x) := \sum_{l=-L_k}^{B_k} V_i(x)S(x)^i$ such that $W_k = \alpha_k(x) + \beta_k(x)y$. Note that $W_k \equiv W_{k-1} \bmod 2^{k-1}$, so we can define

$$\Delta_{\alpha,k}(x) := \frac{\alpha_k(x) - \alpha_{k-1}(x)}{2^{k-1}} \quad \text{and} \quad \Delta_{\beta,k}(x) := \frac{\beta_k(x) - \beta_{k-1}(x)}{2^{k-1}},$$

for $k \geq 1$ and $\Delta_{\alpha,0}(x) := \Delta_{\beta,0}(x) := 0$. It is clear that W_k can be written as

$$W_k = \Delta_{\alpha,1} + 2\Delta_{\alpha,2} + \cdots + 2^{k-1}\Delta_{\alpha,k} + y\left(\Delta_{\beta,1} + 2\Delta_{\beta,2} + \cdots + 2^{k-1}\Delta_{\beta,k}\right).$$

Plugging this into the Newton iteration gives the following equation

$$\begin{aligned} h(x)^2 W_{k+1} \equiv & - \sum_{\substack{1 \le i < j \\ i+j-1 < k+1}} 2^{i+j-1} \left(\Delta_{\alpha,i}\Delta_{\alpha,j} + (f(x) - h(x)y)\,\Delta_{\beta,i}\Delta_{\beta,j}\right) \\ & - y \sum_{i+j-1<k+1} 2^{i+j-1} \Delta_{\alpha,i}\Delta_{\beta,j} - \sum_{2(i-1)<k+1} 2^{2(i-1)} \left(\Delta_{\alpha,i}^2 + (f(x) - h(x)y)\Delta_{\beta,i}^2\right) \\ & + (h(x)^2 - \Sigma(h(x))) \sum_{i<k+1} 2^{i-1}(\Delta_{\alpha,i} + \Delta_{\beta,i} y) + \Sigma(f(x)) \bmod 2^{k+1}. \end{aligned}$$

By definition $Q_H(x)h(x) = H(x)^D$, which implies $1/h(x)^2 = Q_H(x)^2/H(x)^{2D}$ and $\deg Q_H = D \deg H - \deg h$. Since $\deg \Delta_{\alpha,i} \le A_i$ and $\deg \Delta_{\beta,i} \le B_i$, we conclude that A_{k+1} is less than or equal to

$$\max\Big(\max_{i+j<k+2} (A_i + A_j, B_i + B_j + d_S^f), \max_{2i<k+3} (2A_i, 2B_i + d_S^f), \\ \max_{i<k+1} A_i + 2d_S^h, 2d_S^f \Big) - 2d_S^h.$$

Using the bounds given in (4.19) for A_i and B_i and the bounds $A_1 \le d_S^f$, $B_1 \le d_S^h$ and $L_1 \le 0$, we see that A_{k+1} also satisfies the bounds (4.19). An analogous argument can be used to prove that B_{k+1} and L_{k+1} also satisfy the given bounds. □

The previous lemma shows that we can lift the q-th power Frobenius $\overline{F}$ to an endomorphism $\mathcal{F}$ on the dagger ring $A^\dagger$; it suffices to take $\mathcal{F} := \Sigma^n$. Before we can actually compute the zeta function using the Lefschetz fixed point theorem, we need to determine a basis of the $\mathbb{Q}_q$-vector space $H^1(\overline{A}/\mathbb{Q}_q)$.

Analogous to the odd characteristic case, the algebraic de Rham cohomology $H^1_{DR}(A/\mathbb{Q}_q)$ splits into eigenspaces under the hyperelliptic involution: the positive eigenspace $H^1_{DR}(A/\mathbb{Q}_q)^+$ is generated by $x^i/H(x)\,dx$ for $i = 0, \ldots, s$ and the negative eigenspace $H^1_{DR}(A/\mathbb{Q}_q)^-$ is generated by $x^i y\,dx$ for $i = 0, \ldots, 2g-1$. Note that the positive eigenspace corresponds to the deleted ramification points $(\theta_k, 0)$ for $k = 0, \ldots, s$. Every element of $H^1_{DR}(A/\mathbb{Q}_q)$ can be written as a linear combination of differentials of the form $x^k H(x)^m y^l\,dx$, $x^k H(x)^m y^l\,dy$ with $k, l \in \mathbb{N}$ and $m \in \mathbb{Z}$. Using the equation of the curve, we can reduce to the case $l = 0$ or 1. Since $d(x^k H(x)^m y)$ and $d(x^k H(x)^m y^2)$ are exact, we conclude that $H^1_{DR}(A/\mathbb{Q}_q)$ is generated by differentials of the form $x^k H(x)^m\,dx$ and $x^k H(x)^m y\,dx$ with $k \in \mathbb{N}$ and $m \in \mathbb{Z}$.

It is clear that $x^k H(x)^m\,dx$ is exact for $k \in \mathbb{N}$ and $m \ge 0$. If $\deg H > 0$ and $m < 0$ we can assume that $0 \le k < \deg H$ and since $H(x)$ is square-free we can write x^k as $A(x)H(x) + B(x)H'(x)$, which leads to

$$x^k H(x)^m\,dx = A(x)H(x)^{m+1}\,dx + B(x)H'(x)H(x)^m\,dx\,.$$

Since $d(B(x)H(x)^{m+1})$ is exact we can reduce the above differential further for $m < -1$ by using the relation

$$B(x)H'(x)H(x)^m \, dx \equiv -\frac{B'(x)H(x)^{m+1}}{m+1} \, dx \,,$$

where $\equiv$ means equality modulo exact differentials. As a result we can now reduce any differential $x^k H(x)^m \, dx$ to a linear combination of the differentials $x^i / H(x) \, dx$ for $i = 0, \ldots, s$.

For $m > 0$ we can reduce the differential form $x^k H(x)^m y \, dx$ for $k \in \mathbb{N}$ if we know how to reduce the form $x^i y \, dx$ for $i \in \mathbb{N}$. Rewriting the equation of the curve as $(2y + h(x))^2 = 4f(x) + h(x)^2$ and differentiating both sides leads to $(2y + h(x)) \, d(2y + h(x)) = (2f'(x) + h(x)h'(x)) \, dx$. Furthermore, for all $j \geq 1$, we have the following relations

$$\begin{aligned} x^j (2f'(x) + h(x)h'(x))(2y + h(x)) \, dx &= x^j (2y + h(x))^2 \, d(2y + h(x)) \\ &\equiv -\frac{1}{3}(2y + h(x))^3 \, dx^j \\ &= -\frac{j}{3} x^{j-1} (4f(x) + h(x)^2)(2y + h(x)) \, dx \,. \end{aligned}$$

Since $P(x)h(x) \, dx$ is exact for any polynomial $P(x) \in \mathbb{Q}_q[x]$, we finally obtain that

$$\left[x^j (2lfxp + h(x)h'(x)) + \frac{j}{3} x^{j-1} (4f(x) + h(x)^2) \right] y \, dx \equiv 0 \,.$$

The polynomial between brackets has degree $2g + j$ and its leading coefficient is $2(2g+1) + 4j/3 \neq 0$. Note that the formula is also valid for $j = 0$. This means that we can reduce $x^i y \, dx$ for any $i \geq 2g$ by subtracting a suitable multiple of the above differential for $j = i - 2g$.

For $m < 0$ we need an extra trick to reduce the form $x^k H(x)^m y \, dx$ with $k \in \mathbb{N}$. Recall that $Q_f(x) = f(x)/H(x)$ and since the curve is non-singular, we conclude that $\gcd(Q_f(x), H(x)) = 1$. Furthermore, $H(x)$ has no repeated roots which implies $\gcd(H(x), Q_f(x)H'(x)) = 1$. Let $i = -m > 0$, then we can partially reduce $x^k y / H(x)^i \, dx$ by writing x^k as $A(x)H(x) + B(x)Q_f(x)H'(x)$, which leads to

$$\frac{x^k}{H(x)^i} y \, dx = \frac{A(x)}{H(x)^{i-1}} y \, dx + \frac{B(x)Q_f(x)H'(x)}{H(x)^i} y \, dx \,.$$

The latter differential form can be reduced using the following congruence

$$\begin{aligned} \frac{B(x)}{H(x)^i} (2f'(x) + h(x)h'(x))(2y + h(x)) \, dx &= \frac{B(x)}{H(x)^i} (2y + h(x))^2 \, d(2y + h(x)) \\ &\equiv -\frac{1}{3} (2y + h(x))^3 d\left(\frac{B(x)}{H(x)^i} \right) \,. \end{aligned}$$

Substituting the following expressions $h(x) = Q_h(x)H(x)$, $f(x) = Q_f(x)H(x)$ and $(2y + h(x))^2 = 4f(x) + h(x)^2$, we get

$$\frac{B(x)Q_f(x)H'(x)}{H(x)^i} y\,dx \equiv \frac{B(iH'Q_h^2 - 6Q_f' - 3Q_h h') - B'(4Q_f + Q_h h)}{(6-4i)H^{i-1}} y\,dx + \frac{I}{H}\,dx\,,$$

where $I(x)/H(x)\,dx$ is a suitable invariant differential. As a result we can write any form $x^k H(x)^m y\,dx$ for $k \in \mathbb{N}$ and $m \in \mathbb{Z}$ as a linear combination of the differentials $x^i y\,dx$ for $i = 0, \ldots, 2g-1$ and $x^i/H(x)\,dx$ for $i = 0, \ldots, s$.

To show that the Monsky-Washnitzer cohomology $H^1(\overline{A}/\mathbb{Q}_q)$ is generated by the same differential forms as the algebraic de Rham cohomology, we need to bound the denominators introduced during the reduction process. Therefore we prove the following two lemmata.

Lemma 4.4.2 *Let $A := \mathbb{Z}_q[x,y]/(y^2 + h(x)y - f(x))$ and suppose that*

$$x^r y\,dx = \sum_{i=0}^{2g-1} a_i x^i y\,dx + dS, \tag{4.20}$$

with $r \in \mathbb{N}$, $a_i \in \mathbb{Q}_q$ and $S \in A \otimes \mathbb{Q}_q$. Then $2^m a_i \in \mathbb{Z}_q$, $2^{m'}S - \beta \in A$, where $m = 3 + \lfloor \log_2(r+g+1) \rfloor$, $m' = 1 + m + \lfloor \log_2(2g + \deg h) \rfloor$ and β some suitable element in $\mathbb{Q}_q$.

Proof: The proof has two distinct parts. The first part is similar to Kedlaya's argument in [61, Lemma 3], and is based on a local analysis around the point at infinity of the curve C. Put $t = x^g/y$, then one easily verifies that

$$x = t^{-2}\left(1 + \sum_{j=1}^{\infty} \alpha_j t^j\right) \quad \text{and} \quad y = t^{-2g-1}\left(1 + \sum_{j=1}^{\infty} \beta_j t^j\right), \tag{4.21}$$

with $\alpha_j, \beta_j \in \mathbb{Z}_q$. To see this, put $z = 1/x$, rewrite the equation of the curve C as $z + tz^{g+1}h(1/z) - t^2 z^{2g+1} f(1/z) = 0$ and write z as a power series in t using Newton iteration. The relation (4.20) can be rewritten as

$$2^{m-1} x^r (2y + h(x))\,dx = \sum_{i=0}^{2g-1} 2^{m-1} a_i x^i (2y + h(x))\,dx + dT, \tag{4.22}$$

with $T \in A \otimes \mathbb{Q}_q$. Considering the involution $\imath$ of A which sends x to x and $2y + h(x)$ to $-(2y + h(x))$, we see that we can write $T = \sum_{i=0}^{N} A_i x^i (2y + h(x))$, with N big enough and $A_i \in \mathbb{Q}_q$. This yields

$$2^{m-1}x^r(2y+h(x))\,dx - \sum_{i=0}^{2g-1} 2^{m-1}a_i x^i(2y+h(x))\,dx$$
$$= d\left(\sum_{i=0}^{N} A_i x^i(2y+h(x))\right). \quad (4.23)$$

In the above equation we express x and y in terms of t using equalities (4.21). Since $x^i y = t^{-2i-2g-1} + \cdots$, we get $x^i(2y+h(x))\,dx = (-4t^{-2i-2g-4} + \cdots)\,dt$, which leads to

$$2^{m-1} \sum_{j=-\max(2r+2g+4,6g+2)} \gamma_j t^j\,dt$$
$$= d\left(\sum_{i=0}^{N} 2A_i(t^{-2i-2g-1} + \cdots) + A_i(ct^{-2i-2\deg h} + \cdots)\right),$$

with $\gamma_j \in \mathbb{Q}_q$ for all j and $\gamma_j \in \mathbb{Z}_q$ when $j < -2(2g-1) - 2g - 4 = -6g - 2$ and c the leading coefficient of $h(x)$. Note that c is a unit in $\mathbb{Z}_q$. Integrating with respect to t and dividing by 2 gives

$$\sum_{j\geq -\max(2r+2g+3,6g+1)} \gamma'_j t^j = \sum_{i=0}^{N} A_i(t^{-2i-2g-1} + \cdots)$$
$$+ \sum_{i=0}^{N} \frac{A_i}{2}(ct^{-2i-2\deg h} + \cdots), \quad (4.24)$$

with $\gamma'_j \in \mathbb{Q}_q$ for all j and $\gamma'_j \in \mathbb{Z}_q$ when $j < -6g - 1$. Indeed the integration process introduces denominators that become integral upon multiplication with $2^{\lfloor \log(2r+2g+2) \rfloor} = 2^{m-2}$ if $r \geq 2g-1$. A first consequence of (4.24) is that $A_i = 0$ for all $i > \max(r+1, 2g)$. We claim that (4.24) implies that $A_i \in \mathbb{Z}_q$ for all $i > 2g$. Suppose the claim is false. Then let i_0 be the largest integer with $i_0 > 2g$ and $A_{i_0} \notin \mathbb{Z}_q$. Note that $-2i_0 - 2g - 1 < -6g - 1$, since $i_0 > 2g$. Hence the monomials in the left hand side of (4.24) with degree $\leq -2i_0 - 2g - 1$ have coefficients in $\mathbb{Z}_q$. Moreover the monomials of degree $< -2i_0 - 2g - 1$, in the first sum in the right hand side of (4.24) also have coefficients in $\mathbb{Z}_q$, but this is false for the monomial of degree $-2i_0 - 2g - 1$. Hence the second sum in the right hand side of (4.24) contains a monomial of degree $-2i_0 - 2g - 1$ whose coefficient is not in $\mathbb{Z}_q$. That means that there is a maximal i_1 with $A_{i_1}/2 \notin \mathbb{Z}_q$ and $-2i_1 - 2\deg h \leq -2i_0 - 2g - 1$. Because of parity we have that $-2i_1 - 2\deg h < -2i_0 - 2g - 1$. Since c is a unit, the right hand side of (4.24) contains a monomial of degree $-2i_1 - 2\deg h < -2i_0 - 2g - 1$ whose coefficient is not in $\mathbb{Z}_q$. But this contradicts what we said about the left hand side. This finishes the claim that $A_i \in \mathbb{Z}_q$ for all $i > 2g$.

We now turn to the second part of the proof. Note that $(2y+h(x))^2 = v(x)$ with $v(x) := 4f(x)+h(x)^2$. Moreover, we have $d(2y+h(x)) = \frac{w(x)}{2y+h(x)}\,dx$, where $w(x) := 2f'(x)+h(x)h'(x)$. We will use these formulae to deduce from (4.23) a relation which does not involve y. Multiplying (4.23) with $\frac{2y+h(x)}{dx} = \frac{w(x)}{d(2y+h(x))}$ leads to

$$2^{m-1}x^r v(x) - \sum_{i=0}^{2g-1} 2^{m-1}a_i x^i v(x) = \sum_{i=0}^{N} A_i i x^{i-1} v(x) + \sum_{i=0}^{N} A_i x^i w(x).$$

We rewrite this in the form

$$\left(\sum_{i=0}^{2g-1} 2^{m-1}a_i x^i\right) v(x) + \left(\sum_{i=0}^{2g} A_i i x^{i-1}\right) v(x) + \left(\sum_{i=0}^{2g} A_i x^i\right) w(x) = F(x), \quad (4.25)$$

where

$$F(x) := 2^{m-1}x^r v(x) - \sum_{i=2g+1}^{N} A_i i x^{i-1} v(x) - \sum_{i=2g+1}^{N} A_i x^i w(x) \quad (4.26)$$

is a polynomial over $\mathbb{Z}_q$, since $A_i \in \mathbb{Z}_q$ for all $i > 2g$. From equations (4.25) and (4.26) it follows that $\sum_{i=0}^{2g} A_i\theta_k^i$ has valuation ≥ 0 for each root θ_k of $H(x)$, because $v(\theta_k) = 0$ and $w(\theta_k) \neq 0$. To get rid of the disturbing factor 2 in the definition of $w(x)$, we consider the polynomials $q(x) := h'(x)H(x)/h(x) \in \mathbb{Z}_q[x]$ and $u(x) := \frac{1}{2}(w(x) - q(x)v(x)/H(x)) = f'(x) - 2q(x)f(x)/H(x)$. Note that $u(x) \in \mathbb{Z}_q[x]$, $\deg q = \max(0, \deg H - 1)$, $\deg u = 2g$ and that the leading coefficient of $u(x)$ is a unit in $\mathbb{Z}_q$. Rewrite equation (4.25) in such a way that $w(x)$ gets replaced by $u(x)$:

$$\left(\sum_{i=0}^{2g-1} 2^{m-1}a_i x^i + \sum_{i=0}^{2g} A_i i x^{i-1} + \frac{q(x)}{H(x)}\sum_{i=0}^{2g} A_i x^i\right) v(x) + \left(\sum_{i=0}^{2g} 2A_i x^i\right) u(x) = F(x). \quad (4.27)$$

Write $q(x)\sum_{i=0}^{2g} A_i x^i = H(x)\sum_{i=0}^{2g-1} B_i x^i + \mathrm{Rem}(x)$, with $\mathrm{Rem}(x) \in \mathbb{Q}_q[x]$ of degree $< \deg H$. Since $\sum_{i=0}^{2g} A_i\theta_k^i$ has valuation ≥ 0 for each root θ_k of $H(x)$, the same holds for $\mathrm{Rem}(\theta_k)$. Thus $\mathrm{Rem}(x) \in \mathbb{Z}_q[x]$ since the discriminant of $H(x)$ is a unit in $\mathbb{Z}_q$. Hence

$$\left(\sum_{i=0}^{2g-1} \left(2^{m-1}a_i + (i+1)A_{i+1} + B_i\right) x^i\right) v(x) + \left(\sum_{i=0}^{2g} 2A_i x^i\right) u(x) = F(x) - \frac{\mathrm{Rem}(x)v(x)}{H(x)}. \quad (4.28)$$

We consider (4.28) as a system of $4g+1$ linear equations in the unknowns $2^{m-1}a_i + (i+1)A_{i+1} + B_i$ for $i = 0, \dots, 2g-1$ and $2A_i$ for $i = 0, \dots, 2g$. The determinant of this system is the resultant $\mathrm{Res}(v(x), u(x))$ of $v(x)$ and $u(x)$ because $\deg v = 2g+1$ and $\deg u = 2g$. This resultant is a unit in $\mathbb{Z}_q$ because the valuation of $v(\xi)$ is zero for each root ξ of $u(x)$, since the resultant of $f'(x)$ and $h(x)$ is a unit. We conclude that the solutions of the linear system are elements of $\mathbb{Z}_q$, thus $2A_i \in \mathbb{Z}_q$ and $2^{m-1}a_i + (i+1)A_{i+1} + B_i \in \mathbb{Z}_q$. From the definition of the B_i it follows that $2B_i \in \mathbb{Z}_q$ since $2A_i \in \mathbb{Z}_q$ and $\mathrm{Rem}(x) \in \mathbb{Z}_q[x]$. Hence $2^m a_i \in \mathbb{Z}_q$, which concludes the proof of Lemma 4.4.2. □

REMARK 4.4.1 Lemma 4.4.2 remains valid when we replace $\sum_{i=0}^{2g-1}$ by $\sum_{i=\kappa}^{2g-1+\kappa}$ whenever $r \geq \kappa \in \mathbb{N}$. The proof is the same, except that we also have to show that $A_i = 0$ for all $i < \kappa$. This follows from (4.23) using a local analysis at a point on the curve with $x = \theta_k$. Such a local analysis is given in the proof of Lemma 4.4.3 below.

LEMMA 4.4.3 *Let* $A := \mathbb{Z}_q[x, y, H(x)^{-1}]/(y^2 + h(x)y - f(x))$ *with* $\deg h > 0$ *and suppose that*

$$\frac{B(x)}{H(x)^r} y\, dx = \sum_{i=0}^{2g-1} a_i x^i y\, dx + \sum_{i=0}^{s} \frac{b_i x^i}{H(x)}\, dx + \ dS, \tag{4.29}$$

where $r \in \mathbb{N}$, $B(x) \in \mathbb{Z}_q[x]$ *of degree* $< \deg H$, $a_i, b_i \in \mathbb{Q}_q$ *and* $S \in A \otimes \mathbb{Q}_q$. *Then* $2^m a_i \in \mathbb{Z}_q$, $2^{m'} b_i \in \mathbb{Z}_q$, $2^{m'} S - \beta \in A$, *with* $m = 3 + \lfloor \log_2(r+1) \rfloor$, $m' = 1 + m + \lfloor \log_2(2g + \deg h) \rfloor$ *and* β *some suitable element in* $\mathbb{Q}_q$.

PROOF: The proof again consists of two distinct parts. The first part is similar to Kedlaya's argument in [61, Lemma 2] and is based on a local analysis around the ramification points $(\theta_k, 0)$ for $k = 0, \dots, s$. In the completion of the local ring of the curve at $(\theta_k, 0)$ we can write

$$x - \theta_k = \gamma_{k,2} y^2 + \sum_{j \geq 3} \gamma_{k,j} y^j, \tag{4.30}$$

with $\gamma_{k,j} \in \mathbb{Z}_q^{\mathrm{ur}}$ and $\gamma_{k,2}$ a unit in $\mathbb{Z}_q^{\mathrm{ur}}$. Indeed, to see this write $h(x)$ and $f(x)$ as a Taylor expansion around θ_k and use the equation of the curve and the condition $f'(\theta_k) \not\equiv 0 \bmod 2$, to express $x - \theta_k$ as a power series in y using Newton iteration.

Applying the involution $\imath$ to equation (4.29), we see that this relation implies

$$\begin{aligned} 2^{m-1} B(x) H(x)^{-r} (2y + h(x))\, dx - \sum_{i=0}^{2g-1} 2^{m-1} a_i x^i (2y + h(x))\, dx \\ = d\left(\sum_{i=-N}^{M} B_i(x) H(x)^i (2y + h(x)) \right), \end{aligned} \tag{4.31}$$

with N and M large enough integers. Expressing $x-\theta_k$ in terms of y, we get $B_i(x)H(x)^i = u_{k,i}B_i(\theta_k)y^{2i}+\cdots$ with $u_{k,i}$ a unit in $\mathbb{Z}_q^{\mathrm{ur}}$. Substituting this in equation (4.31) and dividing by 2 we obtain

$$\begin{aligned}2^{m-2}\sum_{j\geq -2r+2}\gamma'_{k,j}y^j\,dy = d\left(\sum_{i=-N}^{M}(u_{k,i}B_i(\theta_k)y^{2i+1}+\cdots)\right)\\ + d\left(\frac{u_{k,i}B_i(\theta_k)}{2}\frac{\gamma_{k,2}^{m_k}h^{(m_k)}(\theta_k)}{m_k!}y^{2i+2m_k}+\cdots\right)\end{aligned} \tag{4.32}$$

with $\gamma'_{k,j}\in\mathbb{Q}_q^{\mathrm{ur}}$ for all j and $\gamma'_{k,j}\in\mathbb{Z}_q^{\mathrm{ur}}$ when $j\leq 1$. Integrating the left hand side of this equation with respect to y yields a series whose terms of degree ≤ 2 have coefficients in $\mathbb{Z}_q^{\mathrm{ur}}$. The leading term of the right hand side is $u_{k,-N}B_{-N}(\theta_k)y^{-2N+1}$, which implies that $B_{-N}(\theta_k)$ is integral for $k=0,\ldots,s$. Since the discriminant of $H(x)$ is a unit in $\mathbb{Z}_q$ we conclude that $B_{-N}(x)$ has integral coefficients. Bringing the integral terms to the left hand side and repeating the same argument, shows that $B_i(x)\in\mathbb{Z}_q[x]$ for $i=-N,\ldots,0$. This terminates the first part of the proof.

The second part of the proof proceeds along the same lines as in Lemma 4.4.2. Rewrite the sum $\sum_{i=1}^{M}B_i(x)H(x)^i(2y+h(x))$ as $\sum_{i=0}^{M'}A_ix^i(2y+h(x))$ with $M'\in\mathbb{N}$ and $A_i\in\mathbb{Q}_q$. Using the same formulae as in Lemma 4.4.2 we deduce from (4.31) a relation which does not involve y by multiplying both sides with $\frac{2y+h(x)}{dx}=\frac{w(x)}{d(2y+h(x))}$, which leads to

$$\begin{aligned}2^{m-1}\frac{B(x)}{H(x)^r}v(x)-\sum_{i=0}^{2g-1}2^{m-1}a_ix^iv(x) = \sum_{i=-N}^{0}B_i(x)H(x)^iw(x)+\sum_{i=0}^{M'}A_ix^iw(x)\\ +\sum_{i=-N}^{0}\left(B_i(x)iH(x)^{i-1}H'(x)+B'_i(x)H(x)^i\right)v(x)+\sum_{i=0}^{M'}A_iix^{i-1}v(x)\,.\end{aligned} \tag{4.33}$$

Comparing the valuation at infinity of both sides shows that $A_i=0$ for $i>2g$. We can therefore rewrite the above equation in the form

$$\begin{aligned}\left(\sum_{i=0}^{2g-1}2^{m-1}a_ix^i\right)v(x)+\left(\sum_{i=0}^{2g}A_iix^{i-1}\right)v(x)+\\ \left(\sum_{i=0}^{2g}A_ix^i\right)w(x)=F(x),\end{aligned} \tag{4.34}$$

where

$$F(x) := 2^{m-1}\frac{B(x)}{H(x)^r}v(x) - \sum_{i=-N}^{0} B_i(x)H(x)^i w(x)$$
$$- \sum_{i=-N}^{0} (B_i(x)iH(x)^{i-1}H'(x) + B_i'(x)H(x)^i)v(x)$$

is a polynomial over $\mathbb{Z}_q$ since the $B_i(x) \in \mathbb{Z}_q[x]$ for $i = -N, \ldots, 0$ and the left hand side of (4.34) is a polynomial. From the definition of the A_i it follows that $H(x)$ divides $\sum_{i=0}^{2g} A_i x^i$. It is now easy to see that the rest of the proof is exactly the same as in the proof of Lemma 4.4.2 with $\mathrm{Rem}(x) = 0$, hence $2^m a_i \in \mathbb{Z}_q$ and this concludes the proof of Lemma 4.4.3. □

REMARK 4.4.2 Lemma 4.4.3 remains valid when we replace $\sum_{i=0}^{2g-1} a_i x^i y\, dx$ in equation (4.29) by $\sum_{i=-\kappa}^{\Delta-\kappa} C_i(x)H(x)^i y\, dx$, with $\Delta = \lfloor (2g-1)/\deg H \rfloor$ and $C_i(x) \in \mathbb{Q}_q[x]$ of degree $< \deg H$ whenever $r \geq \kappa \in \mathbb{N}$. The proof is exactly the same.

REMARK 4.4.3 If $r = 0$, then in the above proof the $B_i(x)$ are zero for all $i \leq 0$, and for $0 \leq i \leq 2g-1$ the a_i are completely determined by (4.34) as we saw by considering resultants. This shows that the $x^i y\, dx$ for $i = 0, \ldots, 2g-1$ and the $\frac{x^i}{H(x)}\, dx$ for $i = 0, \ldots, s$ are linearly independent in $H^1_{DR}(A/\mathbb{Q}_q)$.

Lemma 4.4.2 and 4.4.3 show that the basis for $H^1_{DR}(A/\mathbb{Q}_q)$ is a generating set for $H^1(\overline{A}/\mathbb{Q}_q)$, since the reduction process converges. Indeed, for $\sum_{k,l} a_{k,l} x^k S(x)^l y \in A^\dagger$ with $k, l \in \mathbb{Z}$ and $0 \leq k < \deg S$ the valuation of $a_{k,l}$ grows as a linear function of $|l|$, while the valuation of the denominators introduced during the reduction process are only logarithmic in $|l|$.

The Monsky-Washnitzer cohomology $H^1(\overline{A}/\mathbb{Q}_q)$ is the direct sum of the $\imath$-invariant part $H^1(\overline{A}/\mathbb{Q}_q)^+$ on which $\imath$ acts trivially and the $\imath$-anti-invariant part $H^1(\overline{A}/\mathbb{Q}_q)^-$ on which $\imath$ acts as multiplication by -1. Note that the $\imath$-invariant part of $\overline{A}$ is isomorphic to $\mathbb{F}_q[x, \overline{H}(x)^{-1}]$ which is the coordinate ring of the curve $\mathbb{A}^1 \setminus V_{\overline{H}}$, with $V_{\overline{H}}$ the set of zeros of $\overline{H}$. The Lefschetz fixed point formula applied to $\overline{C}'$ and $\mathbb{A}^1 \setminus V_{\overline{H}}$ then gives

$$\begin{aligned}
\#\overline{C}'(\mathbb{F}_{q^k}) &= \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^0(\overline{A}/\mathbb{Q}_q)\right) - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)\right) \\
&= q^k - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^+\right) - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^-\right) \\
&= q^k - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\mathbb{A}^1 \setminus V_{\overline{H}}/\mathbb{Q}_q)\right) - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^-\right) \\
&= \#(\mathbb{A}^1 \setminus V_{\overline{H}})(\mathbb{F}_{q^k}) - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^-\right) \\
&= q^k - r_k - \mathrm{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^-\right)
\end{aligned}$$

with r_k the number of ramification points $(\overline{\theta}, 0)$ defined over $\mathbb{F}_{q^k}$. Let $\widetilde{C}$ be the unique smooth projective curve birational to $\overline{C}$, then

$$\begin{aligned}\#\widetilde{C}(\mathbb{F}_{q^k}) &= \#\overline{C}'(\mathbb{F}_{q^k}) + r_k + 1\\ &= q^k + 1 - \operatorname{Tr}\left(q^k \mathcal{F}_*^{-k} | H^1(\overline{A}/\mathbb{Q}_q)^-\right)\\ &= q^k + 1 - \sum_{i=1}^{2g} \alpha_i^k,\end{aligned}$$

with α_i the eigenvalues of $q\mathcal{F}_*^{-1}$ on $H^1(\overline{A}/\mathbb{Q}_q)^-$. By the Weil conjectures [137] there exists a polynomial $\chi(t) \in \mathbb{Z}[t]$ of the form $t^{2g} + a_1 t^{2g-1} + \cdots + a_{2g}$, whose roots $\beta_1, \ldots, \beta_{2g}$ satisfy $\beta_i \beta_{g+i} = q$ for $i = 1, \ldots, g$, $|\beta_i| = \sqrt{q}$ for $i = 1, \ldots, 2g$ and $\#\widetilde{C}(\mathbb{F}_{q^k}) = q^k + 1 - \sum_{i=1}^{2g} \beta_i^k$ for all $k > 0$. This implies that we can label the β's such that $\alpha_i = \beta_i$ for $i = 1, \ldots, 2g$. Since $\alpha_i \alpha_{g+i} = q$, the α_i are also the eigenvalues of $\mathcal{F}_*$ on $H^1(\overline{A}/\mathbb{Q}_q)^-$. The zeta function of $\widetilde{C}$ can be written as

$$Z(\widetilde{C}/\mathbb{F}_q; t) = \frac{t^{2g}\chi(1/t)}{(1-t)(1-qt)},$$

so it suffices to recover $\chi(t)$ from an approximation of the characteristic polynomial of the matrix $M_{\mathcal{F}} = \Sigma^{n-1}(M) \cdots \Sigma(M) M$ through which the Frobenius $\mathcal{F}_* = \Sigma_*^n$ acts on $H^1(\overline{A}/\mathbb{Q}_q)^-$.

4.4.2 Detailed Algorithm

Using the formulae of the previous section, we describe an algorithm for computing the characteristic polynomial of Frobenius $\chi(t)$ and the zeta function of a smooth projective hyperelliptic curve $\widetilde{C}$ of genus g over $\mathbb{F}_q$ with $q = 2^n$.

We have shown that $\chi(t) = t^{2g} + a_1 t^{2g-1} + \cdots + a_{2g}$ can be computed as the characteristic polynomial of $\mathcal{F}_*$ on $H^1(\overline{A}/\mathbb{Q}_q)^-$. The Weil conjectures imply that $q^{g-i} a_i = a_{2g-i}$, so it suffices to compute $a_1, \ldots, a_g$. Furthermore, for $i = 1, \ldots, g$ the a_i can be bounded by

$$|a_i| \le \binom{2g}{i} q^{i/2} \le \binom{2g}{g} q^{g/2} \le 2^{2g} q^{g/2}.$$

Therefore it suffices to compute the action of $\mathcal{F}_*$ on a basis of $H^1(\overline{A}/\mathbb{Q}_q)^-$ modulo 2^B with

$$B \ge \left\lceil \log_2 \left(2 \binom{2g}{g} q^{g/2} \right) \right\rceil.$$

However, it is not sufficient to compute $\Sigma(y)$ modulo 2^B since we need to take into account the loss of precision caused by the reduction of the differential forms. Let $\Sigma(y) \equiv \alpha_N + \beta_N y \bmod 2^N$ and write β_N as $\beta_N = \sum_{i=-L_N}^{B_N} V_i(x) S(x)^i$, then Lemma 4.4.1 implies that

$$L_N \le 4(N-1)D - 2D \quad \text{and} \quad B_N \le 2(N-2)(d_S^f - 2d_S^h) + (d_S^f - d_S^h), \tag{4.35}$$

with $d_S^f := \deg f / \deg S$ and $d_S^h := \deg h / \deg S$. Since we have to reduce the forms $x^{2i+1}\Sigma(y)\,dx$ for $i = 0, \ldots, 2g-1$, the loss of precision will be determined by the reduction of $x^{4g-1}V_{B_N}(x)S(x)^{B_N}y\,dx$ and $xV_{-L_N}S(x)^{-L_N}y\,dx$. An easy calculation shows that the highest power of x appearing in the former differential form is less than $2N(\deg f - 2\deg h) + 6g$ and by Lemma 4.4.2 the loss of precision is bounded by

$$c_{N,1} := 3 + \lfloor \log_2(2N(\deg f - 2\deg h) + 7g + 1) \rfloor \ .$$

Similarly, Lemma 4.4.3 implies that the loss of precision caused by reducing the latter differential form is bounded by

$$c_{N,2} := 3 + \lfloor \log_2(4ND - 6D + 1) \rfloor \ .$$

As a result, we conclude that it is sufficient to compute modulo 2^N with

$$N - \max(c_{N,1}, c_{N,2}) > B \ . \tag{4.36}$$

REMARK 4.4.4 Note that since the matrix M is not necessarily defined over $\mathbb{Z}_q$, we could lose an extra cn bits of precision during the computation of $M_{\mathcal{F}} = \Sigma^{n-1}(M)\cdots\Sigma(M)M$, where 2^c is the largest denominator appearing in M. By Lemmata 4.4.1 and 4.4.2, c is bounded by $O(\log g)$ independently of n. In theory we would therefore have to replace the bound B by $B + cn$, which does not change the complexity of the algorithm. In practice however it turns out that the largest denominator appearing in $M_{\mathcal{F}}$ is always the same as the largest denominator appearing in M and therefore it is not necessary to increase B. This phenomenon can be heuristically explained as follows: since the eigenvalues of $\mathcal{F}_* = \Sigma_*^n$ on $H^1(\overline{A}/\mathbb{Q}_q)^-$ have non-negative 2-adic valuation there is a $\mathbb{Z}_q$-submodule of $H^1(\overline{A}/\mathbb{Q}_q)^-$ which is stable under the action of Σ_*. For this $\mathbb{Z}_q$-submodule we can take for instance the canonical image of the crystalline cohomology of C over $\mathbb{Z}_q$. Note that the $\mathbb{Z}_q$-submodule generated by $x^i y\,dx$ for $i = 0, \ldots, 2g-1$ is not canonical and in general not stable under Σ_*. Let A_0 be the matrix that expresses $x^i y\,dx$ for $i = 0, \ldots, 2g-1$ in terms of a basis of such a stable $\mathbb{Z}_q$-submodule and let A be A_0 times a power of 2 such that A is a matrix over $\mathbb{Z}_q$ which is not zero modulo 2. Then $M = A^{-1}U\Sigma(A)$ where U is the matrix of Σ_* with respect to the new basis. Note that U is a matrix over $\mathbb{Z}_q$ and that the norm of M equals $A^{-1}U\Sigma(U)\cdots\sigma^{n-1}(U)A$. Thus the loss of precision is no more than $2d$ bits where d is the 2-adic valuation of $\det(A)$. If U and A are generic enough then $|c-d|$ is small. Furthermore, the bound (4.36) turns out to be slightly larger than what is needed and compensates for the loss of $2d$ bits.

The function HYPERELLIPTIC_ZETA_FUNCTION given in Algorithm 4.4.1 computes the zeta function of a smooth projective hyperelliptic curve $\widetilde{C}$ defined over $\mathbb{F}_q$ with $q = 2^n$. Step 1 determines the minimal precision N satisfying inequality (4.36). In step 2 we call the subroutine LIFT_CURVE, which first constructs an isomorphic curve such that $\overline{H}(x) \mid \overline{h}(x)$ and $\overline{H}(x) \mid \overline{f}(x)$ and lifts the

---

Algorithm 4.4.1 (Hyperelliptic_Zeta_Function)

```
INPUT:   Hyperelliptic curve C̄ over F_q given by equation
         y^2 + h̄(x)y = f̄(x).
OUTPUT:  The zeta function Z(C̃/F_q; t).
```

1. $B \geq \left\lceil \log_2 \left(2\binom{2g}{g} q^{g/2} \right) \right\rceil$; $N - \max(c_{N,1}, c_{N,2}) > B$;
2. $(h(x), f(x), H(x), D) =$ `Lift_Curve`$(\overline{h}(x), \overline{f}(x))$;
3. $\alpha_N, \beta_N =$ `Lift_Frobenius_y`(h, f, H, D, N);
4. `For` $i = 0$ `To` $2g - 1$ `Do`
   - 4.1. $R_i(x) =$ `Reduce_MW_Cohomology`$(2x^{2i+1}\beta_N, h, f, H, B)$;
   - 4.2. `For` $j = 0$ `To` $2g - 1$ `Do` $M[i][j] =$ `Coeff`(R_i, j);
5. $M_{\mathcal{F}} = \Sigma^{n-1}(M) \cdots \Sigma(M) M \mod 2^B$;
6. $\chi(t) =$ `Characteristic_Pol` $(M_{\mathcal{F}}) \mod 2^B$;
7. `For` $i = 0$ `To` g `Do`
   - 7.1. `If Coeff`$(\chi, 2g - i) > \binom{2g}{i} q^{i/2}$ `Then`
     `Coeff`$(\chi, 2g - i) \; -= 2^B$;
   - 7.2. `Coeff`$(\chi, i) = q^{g-i}$ `Coeff`$(\chi, 2g - i)$;
8. `Return` $Z(\widetilde{C}/\mathbb{F}_q; t) = \dfrac{t^{2g}\chi(1/t)}{(1-t)(1-qt)}$.

ALGORITHM 4.4.2 (LIFT_FROBENIUS_Y)

`INPUT:`	`Curve` $C : y^2 + h(x)y = f(x)$ `over` $\mathbb{Z}_q$, `polynomial` $H \in \mathbb{Z}_q[x]$ `with` $H \mid h$ `and` $H \mid f$, $D \in \mathbb{N}$ `such that` $h \mid H^D$, `precision` N.
`OUTPUT:`	`Laurent polynomials` α_N, β_N `in` S `with` $S = x$ `if` $H = 1$, $S = H$ `if` $\deg H > 0$ `satisfying` $\Sigma(y) \equiv \alpha_N + \beta_N y \mod 2^N$.

1. $B = \lceil \log_2 N \rceil + 1$; $T = N$; $Q_S := S^D$ `div` h;
2. `For` $i = B$ `Down To 1 Do` $P[i] = T$; $T = \lceil T/2 \rceil$;
3. $\alpha \equiv f \mod 2$; $\beta \equiv -h \mod 2$; $\gamma = 1$; $\delta = 0$;
4. `For` $i = 2$ `To` B `Do`
   - 4.1. $T_A \equiv \big((\alpha + \Sigma(h)) \cdot \alpha + \beta^2 \cdot f - \Sigma(f)\big) \cdot Q_S^2 \cdot S^{-2D} \mod 2^{P[i]}$;
   - 4.2. $T_B \equiv (2\alpha - h \cdot \beta + \Sigma(h)) \cdot \beta \cdot Q_S^2 \cdot S^{-2D} \mod 2^{P[i]}$;
   - 4.3. $D_A \equiv 1 + (\Sigma(h) - h^2 + 2\alpha) \cdot Q_S^2 \cdot S^{-2D} \mod 2^{P[i-1]}$;
   - 4.4. $D_B \equiv 2\beta \cdot Q_S^2 \cdot S^{-2D} \mod 2^{P[i-1]}$;
   - 4.5. $V_A \equiv D_A \cdot \gamma + D_B \cdot \delta \cdot f - 1 \mod 2^{P[i-1]}$;
   - 4.6. $V_B \equiv D_A \cdot \delta + D_B \cdot (\gamma - \delta \cdot h) \mod 2^{P[i-1]}$;
   - 4.7. $\gamma \equiv \gamma - (V_A \cdot \gamma + V_B \cdot \delta \cdot f) \mod 2^{P[i-1]}$;
   - 4.8. $\delta \equiv \delta - (V_A \cdot \delta + V_B \cdot (\gamma - \delta \cdot h)) \mod 2^{P[i-1]}$;
   - 4.9. $\alpha \equiv \alpha - (T_A \cdot \gamma + T_B \cdot \delta \cdot f) \mod 2^{P[i]}$;
   - 4.10. $\beta \equiv \beta - (T_A \cdot \delta + T_B \cdot (\gamma - \delta \cdot h)) \mod 2^{P[i]}$;
5. `Return` $\alpha_N = \alpha$, $\beta_N = \beta$.

ALGORITHM 4.4.3 (REDUCE_MW_COHOMOLOGY)

`INPUT:` Polynomials $h, f, H \in \mathbb{Z}_q[x]$ with $H|h$ and $H|f$, H monic, Laurent polynomial $G = \sum T_i(x)S(x)^i$ with $S = x$ if $H = 1$, $S = H$ if $H \neq 1$, $T_i \in \mathbb{Z}_q[x]$ with $\deg T_i < \deg S$, $B \in \mathbb{N}$.

`OUTPUT:` $\Lambda \in \mathbb{Q}_q[x]$, with $\deg \Lambda < 2g$ such that $\Lambda y\, dx \sim G y\, dx \mod 2^B$.

1. $N - \max(c_{N,1}, c_{N,2}) > B$;
2. $Q_f = f$ `div` S; $Q_h = h$ `div` S; $P = 0$; $V = 0$; $v_G =$ `Val`(G);
3. `For` $i = v_G$ `To` -1
   - 3.1. $V \equiv P +$`Coeff`$(G, i) \mod 2^N$;
   - 3.2. $P \equiv V$ `div` $S \mod 2^N$; $V \equiv V - P \cdot S \mod 2^N$;
   - 3.3. $(1, L_A, L_B) =$ `XGCD`$(S,\ Q_f \cdot S')$;
   - 3.4. $L_A = V \cdot L_A \mod 2^N$; $L_B = V \cdot L_B \mod 2^N$;
   - 3.5. $P \equiv P + L_A + \frac{L_B \cdot (-iQ_h^2 \cdot S' - 3(2Q_f' + Q_h \cdot h')) - L_B' \cdot (4Q_f + Q_h h)}{6+4i} \mod 2^N$;
4. $d_G =$ `Degree`(G); $d_T = (d_G + 1) \cdot$`Degree`(S); $T = 0$;
5. `For` $i = d_G$ `Down To 0 Do`
   - 5.1. $T = T \cdot S +$`Coeff`$(G, i) \mod 2^N$; $T = T + P$;
6. `For` $i = d_T$ `Down To` $2g$
   - 6.1. $P \equiv x^{i-2g}(2f' + h \cdot h') + \frac{i-2g}{3} x^{i-2g-1}(4f + h^2) \mod 2^N$;
   - 6.2. $T \equiv T - ($`Coeff`$(T, i) \cdot P)/$`Coeff`$(P, i) \mod 2^N$;
7. `Return` $\Lambda \equiv T \mod 2^B$.

---

curve using the construction described in Section 4.4.1. The result of this function is a hyperelliptic curve $C : y^2 + h(x)y = f(x)$ over $\mathbb{Z}_q$, a polynomial $H(x)$ and an integer D such that $H(x) \mid h(x)$, $H(x) \mid f(x)$ and $h(x) \mid H(x)^D$. Since this function is rather straightforward, we have omitted the pseudo-code.

In step 3 we compute $\Sigma(y) \bmod 2^N$ using the function LIFT_FROBENIUS_Y described in Algorithm 4.4.2. The parameters α_N, β_N are Laurent polynomials in S with coefficients polynomials over $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ of degree $< \deg S$. This function implements the Newton iteration (4.17), but has quadratic, rather than linear convergence. Note that Algorithm 4.4.2 is in fact a double Newton iteration: $\alpha + \beta y$ converges to $\Sigma(y)$, whereas $\gamma + \delta y$ is an approximation of the inverse of the denominator (4.18) in the Newton iteration.

Once we have determined an approximation of $\Sigma(y)$, we compute the action of Σ_* on the basis of $H^1(\overline{A}/\mathbb{Q}_q)^-$ as $2x^{2i+1}\Sigma(y)\,dx$ for $i = 0, \ldots, 2g-1$. In step 4 we reduce these forms using the function REDUCE_MW_COHOMOLOGY given in Algorithm 4.4.3, which is based on the reduction formulae devised in Section 4.4.1. Given a differential form $Gy\,dx$ with G a Laurent polynomial in S, this function computes a polynomial $\Lambda \in \mathbb{Q}_q[x]$, with $\deg \Lambda < 2g$ such that for a given precision B we have the following equivalence modulo exact and invariant forms $\Lambda y\,dx \sim Gy\,dx \bmod 2^B$, where $\bmod 2^B$ means: modulo elements of $2^B(\mathbb{Z}_q y\,dx + \cdots + \mathbb{Z}_q x^{2g-1} y\,dx)$. In step 3.3 we use the function XGCD which takes as input two polynomials $A(x), B(x) \in \mathbb{Q}_q[x]$ and returns polynomials $C(x), L_A(x), L_B(x) \in \mathbb{Q}_q[x]$ such that $C(x) = \gcd(A(x), B(x))$ and $C(x) = L_A(x)A(x) + L_B(x)B(x)$. Note that Lemmata 4.4.2 and 4.4.3 imply that the result of Algorithm 4.4.3 is correct modulo 2^B since we computed modulo 2^N and N satisfies $N - \max(c_{N,1}, c_{N,2}) > B$.

The result of step 4 of Algorithm 4.4.1 is an approximation modulo 2^B of the matrix M through which Σ_* acts on $H^1(\overline{A}/\mathbb{Q}_q)^-$. In step 5 we compute its norm $M_{\mathcal{F}}$ as $M_{\mathcal{F}} = \Sigma^{n-1}(M) \cdots \Sigma(M)M \bmod 2^B$.

In steps 6 and 7 we recover the characteristic polynomial of Frobenius from the first g coefficients of the characteristic polynomial of $M_{\mathcal{F}}$. Finally, we return the zeta function of the smooth projective hyperelliptic curve $\widetilde{C}$ birational to $\overline{C}$ in step 8.

4.4.3 Complexity Analysis

In this section we analyse the space and time requirements of Algorithm 4.4.1 for a genus g hyperelliptic curve over $\mathbb{F}_{2^n}$. Before proceeding through each step of the algorithm, we analyse the complexity of the basic operations in Algorithm 4.4.1 and the asymptotic behavior of the bounds given in Lemma 4.4.1.

For $q = 2^n$, elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo a sparse irreducible polynomial $P(t)$ of degree n. Since each element of this ring requires $O(nN)$ space, we can perform the basic operations, i.e. multiplication and division, in time $O(n^\mu N^\mu)$.

Lemma 4.4.1 bounds the maximum bit-size of the Laurent series we compute with and therefore determines the complexity of Algorithm 4.4.1. Since these bounds depend on the degree and splitting type of $h(x)$, we make a distinction between average-case and worst-case complexity. To this end we introduce three parameters which allow us to analyse both cases simultaneously.

- Let the asymptotic behavior of $\deg f - 2\deg h$ be $O(g^\lambda)$. Since the degree of $f(x)$ is $2g+1$ and $h(x)$ is a random polynomial of degree $\leq g$, we conclude that $\lambda = 0$ on average and $\lambda = 1$ in the worst case.

- Let the asymptotic behavior of $\deg H$ be $O(g^\eta)$. With very high probability a random polynomial of degree $\leq g$ has $O(g)$ different roots, which implies that $\eta = 1$ on average and $\eta = 0$ in the extreme case.

- Let the asymptotic behavior of D be $O(g^\nu)$, then $\nu = 0$ on average and $\nu = 1$ in the worst case, since with very high probability a random polynomial only has roots with multiplicity $O(1)$.

In step 1 of Algorithm 4.4.1 we determine the precision N satisfying inequality (4.36), which implies that N is $O(gn)$. The function LIFT_FROBENIUS_Y in step 3 is a Newton lifting. Since the precision doubles in every iteration, we conclude that its complexity is determined by the last iteration which consists of $O(1)$ multiplications of Laurent polynomials in S with coefficients polynomials over $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ of degree less than $\deg S$. Lemma 4.4.1 implies that the bit-size of these objects is $O((g^\lambda + g^{\eta+\nu})nN^2)$. Since the cost of the other operations in LIFT_FROBENIUS_Y, e.g. computing the Frobenius substitution of $O(g)$ elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$, is less than the $O(1)$ multiplications, the overall time complexity of step 3 is $O((g^\lambda + g^{\eta+\nu})^\mu n^\mu N^{2\mu})$.

In step 4 of Algorithm 4.4.1 we reduce the $2g$ differential forms $2x^{2i+1}\beta_N y\, dx$ for $i = 0, \ldots, 2g-1$ using the function REDUCE_MW_COHOMOLOGY given in Algorithm 4.4.3. In step 3 the dominant operations are $O(1)$ multiplications of polynomials over $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ of degree $O(g)$ and the XGCD computation of two such polynomials. The former operation clearly takes time $O(g^\mu n^\mu N^\mu)$ and using Moenck's algorithm [88] the latter operation can be performed in time $O(g^\mu n^\mu N^\mu \log g)$. Lemma 4.4.1 implies that these operations have to be repeated $O(g^\nu N)$ times, so the time complexity of step 3 is $O(g^{\nu+\mu} n^\mu N^{\mu+1}\log g)$. Write β_N as $\sum_{i=-L_N}^{B_N} V_i(x)S(x)^i$, then step 5 essentially is Horner's rule to compute $\sum_{i=0}^{B_N} V_i(x)S(x)^i$. Note that in practice we perform this step only once for all of the $2g$ reductions and use a binary tree algorithm which is asymptotically faster than Horner's method. The complexity of step 5 then becomes $O(g^{\lambda\mu} n^\mu N^{2\mu}\log(gN))$. Lemma 4.4.1 implies that steps 6.1 and 6.2 have to be executed $O(g^\lambda N)$ times and since each iteration consists of $O(g)$ multiplications and $O(1)$ divisions in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$, the time complexity of step 6 is $O(g^{1+\lambda} n^\mu N^{\mu+1})$. Since we have to reduce $O(g)$ differential forms, the overall time complexity of step 4 of Algorithm 4.4.1 is $O((g^{2+\lambda} + g^{\nu+\mu+1})n^\mu N^{\mu+1})$.

In step 5 we need to determine the norm of a $2g \times 2g$ matrix M over $\mathbb{Q}_q$ as $\Sigma^{n-1}(M)\cdots\Sigma(M)M$. This can be accomplished by computing

$$M_{i+1} = \Sigma^{2^i}(M_i)M_i\,, \quad \text{for } i = 0, \ldots, \lfloor \log_2 n \rfloor$$

with $M_0 = M$ and combining these to recover the norm of M. This process takes $O(\log n)$ multiplications of $2g \times 2g$ matrices at a cost of $O(g^3 n^\mu N^\mu)$ bit-operations and $O(g^2 \log n)$ applications of powers of Σ which can be computed in time $O(g^2 n^{\mu+1} N^\mu \log n)$ if we precompute $\Sigma^{2^i}(t)$ for $i = 0, \ldots, \lfloor \log_2 n \rfloor$.

Finally, we need to compute the characteristic polynomial of a $2g \times 2g$ matrix over $\mathbb{Q}_q$, which can be done using the classical algorithm based on the Hessenberg form [16, Section 2.2.4]. The complexity of this algorithm is $O(g^3)$ ring operations or $O(g^3 n^\mu N^\mu)$ time.

Since equation (4.36) implies that N is $O(gn)$, the overall space complexity of Algorithm 4.4.1 is $O((g^\lambda + g^{\eta+\nu})g^2 n^3)$. The overall time complexity in the average case is given by

$$O(n^{2\mu} g^\mu (n^\mu g^\mu + ng^2)(g^\mu + \log n)),$$

and in the worst case by

$$O(g^{2\mu} n^{2\mu} (g^3 n + g^\mu n^\mu (\log g + \log n))).$$

This also proves the following theorem.

THEOREM 4.4.1 *There exists a deterministic algorithm to compute the zeta function of a smooth hyperelliptic curve of genus g defined over a finite field $\mathbb{F}_q$ with $q = 2^n$ elements using*

- $O(g^3 n^3 (g + (\log n)^2 \log\log n) \log gn \log\log gn)$ *bit-operations and* $O(g^3 n^3)$ *space in the average case,*
- $O(g^3 n^3 (g^2 + (\log n)^2 \log\log n) \log gn \log\log gn)$ *bit-operations and* $O(g^4 n^3)$ *space in the worst case.*

4.4.4 Implementation and Numerical Results

In this section we present running times of an implementation of Algorithm 4.4.1 in the `C` programming language and give some examples of Jacobians of hyperelliptic curves with almost prime group order.

The basic operations on integers modulo 2^N for $N \leq 256$ were written in assembly language. Elements of $\mathbb{Z}_q/(2^N \mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N \mathbb{Z})$ modulo a degree n irreducible polynomial, which we chose to be either a trinomial or a pentanomial. For multiplication of elements in $\mathbb{Z}_q/(2^N \mathbb{Z}_q)$, polynomials over $\mathbb{Z}_q/(2^N \mathbb{Z}_q)$ and Laurent series over $\mathbb{Z}_q/(2^N \mathbb{Z}_q)[x]$ we used a combination of Karatsuba [60] and Toom [126] multiplication.

Table 4.3 contains running times and memory usages of our algorithm for genus 2, 3 and 4 hyperelliptic curves over various finite fields of characteristic 2. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1. Note that the field degrees are chosen such that gn is constant across each row.

Table 4.3: Running time (s) and memory usage (MB) for genus 2, 3 and 4 hyperelliptic curves over $\mathbb{F}_{2^n}$

	Genus 2 curves		Genus 3 curves		Genus 4 curves	
gn	Time	Mem	Time	Mem	Time	Mem
120	18.1	4.17	25.8	4.42	25.7	4.14
144	36.5	5.93	45.7	6.26	43.8	5.73
168	52.8	7.75	68.6	8.21	76.6	9.85
192	88.8	13.4	113	14.1	113	12.8
240	238	26.2	282	27.6	277	24.9
288	437	36.8	514	38.7	501	34.9
336	757	59.2	936	62.3	882	56.3
384	1229	90.2	1571	95.0	1471	85.2
480	3760	154	3999	161	4034	149

Hyperelliptic Curve Examples: Low Genus Curves

In this subsection we give some examples of Jacobians of hyperelliptic curves with almost prime group order. The correctness of these results is easily proved by multiplying a random divisor with the given group order and verifying that the result is principal, i.e. is the zero element in the Jacobian $J_{\widetilde{C}}(\mathbb{F}_q)$. The given curves are non-supersingular, since the coefficient a_g of $\chi(t)$ is odd [44]. Furthermore, all curves withstand the MOV-FR attack [42, 81].

Genus 2 hyperelliptic curve over $\mathbb{F}_{2^{83}}$

Let $\mathbb{F}_{2^{83}}$ be defined as $\mathbb{F}_2[w]/\overline{P}(w)$ with $\overline{P}(w) = w^{83} + w^7 + w^4 + w^2 + 1$ and consider the random hyperelliptic curve $\overline{C}_2$ of genus 2 defined by

$$y^2 + (\sum_{i=0}^{2} h_i x^i)y = x^5 + \sum_{i=0}^{4} f_i x^i ,$$

where

$h_0 =$ `7FF29B08993336B479CD2` $\qquad h_1 =$ `32C101713C722F8FB5BC9`
$h_2 =$ `553E16B6A3BC6B2432CA8`
$f_0 =$ `7AD44882C02B9743CD58B` $\qquad f_1 =$ `327254FA330B44958262A`
$f_2 =$ `204AB23E12828D061AF04` $\qquad f_3 =$ `1C827250FFDEFF93B43BE`
$f_4 =$ `13D80106C0E5571DFD139`

The group order of the Jacobian $J_{\widetilde{C}_2}$ of $\overline{C}_2$ over $\mathbb{F}_{2^{83}}$ is

$$\#J_{\widetilde{C}_2} = 2 \cdot 46768052394566313810931349196246034325781246483037 ,$$

where the last factor is prime. The coefficients a_1 and a_2 of the characteristic polynomial of Frobenius $\chi(t) = t^4 + a_1 t^3 + a_2 t^2 + a_3 t + a_4$ are given by

$$a_1 = -4669345964042 \quad \text{and} \quad a_2 = 18983903513383986646766787 .$$

Genus 3 hyperelliptic curve over $\mathbb{F}_{2^{59}}$

Let $\mathbb{F}_{2^{59}}$ be defined as $\mathbb{F}_2[w]/\overline{P}(w)$ with $\overline{P}(w) = w^{59} + w^7 + w^4 + w^2 + 1$ and consider the random hyperelliptic curve $\overline{C}_3$ of genus 3 defined by

$$y^2 + (\sum_{i=0}^{3} h_i x^i)y = x^7 + \sum_{i=0}^{6} f_i x^i \,,$$

where

$h_0 =$ `569121E97EB3821`	$h_1 =$ `49F340F25EA38A2`
$h_2 =$ `2DE854D48D56154`	$h_3 =$ `0B6372FF7310443`
$f_0 =$ `1104FDBEB454C58`	$f_1 =$ `0C426890E5C7481`
$f_2 =$ `34967E2EB7D50C3`	$f_3 =$ `1F1728AA28C616C`
$f_4 =$ `1AE177BFE49826A`	$f_5 =$ `3895A0E400F7D18`
$f_6 =$ `6DF634A1E2BFA8E`	

The group order of the Jacobian $J_{\widetilde{C}_3}$ of $\overline{C}_3$ over $\mathbb{F}_{2^{59}}$ is

$$\#J_{\widetilde{C}_3} = 2 \cdot 95780971407243394633762332360123160334059170481903949 \,,$$

where the last factor is prime. The coefficients a_1, a_2 and a_3 of the characteristic polynomial of Frobenius $\chi(t) = t^6 + a_1 t^5 + a_2 t^4 + a_3 t^3 + a_4 t^2 + a_5 t + a_6$ are given by

$$\begin{aligned} a_1 &= 620663068, \\ a_2 &= 848092512078818380, \\ a_3 &= 341008017371409573053936945 \,. \end{aligned}$$

Genus 4 hyperelliptic curve over $\mathbb{F}_{2^{47}}$

Let $\mathbb{F}_{2^{47}}$ be defined as $\mathbb{F}_2[w]/\overline{P}(w)$ with $\overline{P}(w) = w^{47} + w^5 + 1$ and consider the random hyperelliptic curve $\overline{C}_4$ of genus 4 defined by

$$y^2 + (\sum_{i=0}^{4} h_i x^i)y = x^9 + \sum_{i=0}^{8} f_i x^i \,,$$

where

$h_0 =$ `45EDAA69BB7B`	$h_1 =$ `29185CC987F2`	$h_2 =$ `5B56AF467634`
$h_3 =$ `063A420D7308`	$h_4 =$ `3AD67360D2FB`	
$f_0 =$ `116A64DA4E4A`	$f_1 =$ `1267C8BFEDF4`	$f_2 =$ `5DED53867285`
$f_3 =$ `3E2486D3500B`	$f_4 =$ `66718C5D41BD`	$f_5 =$ `5FBD515320F1`
$f_6 =$ `4B960757EC52`	$f_7 =$ `67B0202BA7D5`	$f_8 =$ `545283F149A8`

The group order of the Jacobian $J_{\widetilde{C}_4}$ of $\overline{C}_4$ over $\mathbb{F}_{2^{47}}$ is

$$\#J_{\widetilde{C}_4} = 2 \cdot 19615942964173331615183011742127092423180913572422390278 7 \,,$$

where the last factor is prime. The coefficients a_1, a_2, a_3 and a_4 of the characteristic polynomial of Frobenius $\chi(t) = t^8 + a_1t^7 + a_2t^6 + a_3t^5 + a_4t^4 + a_5t^3 + a_6t^2 + a_7t + a_8$ are given by

$$\begin{aligned} a_1 &= 294806, \\ a_2 &= -5127513198846, \\ a_3 &= 236526738819576049756, \\ a_4 &= 3153492296632744619801811598 5. \end{aligned}$$

Hyperelliptic Curve Examples: High Genus Curves

The asymptotic behaviour of Algorithm 4.4.1 is polynomial in both the degree of the finite field and the genus of the curve. Furthermore, the complexity analysis shows that the reduction of the differentials dominates the running time for large geni. This is illustrated by Table 4.4 which contains running times for hyperelliptic curves over $\mathbb{F}_2$ for varying genus $20 \le g \le 240$.

Table 4.4: Running time (s) and memory usage (MB) for hyperelliptic curves over $\mathbb{F}_2$ and genus $20 \le g \le 240$

Genus	Lift $\mathcal{F}$	Reduction	Total	Mem
20	5.76	7.54	13.3	0.47
40	35.4	80.1	116	1.63
60	87.4	369	458	3.12
80	228	666	897	5.98
100	461	1651	2118	8.77
120	702	3878	4593	11.9
160	2112	9616	11759	22.4
200	3754	29712	33531	32.8
240	5291	33729	39120	52.8

4.5 Conclusion

In this chapter we have presented new algorithms to compute the zeta function of a hyperelliptic curve over a finite field using Monsky-Washnitzer cohomology, i.e. the de Rham cohomology of a ring of overconvergent power series.

Kedlaya [61] described such an algorithm for hyperelliptic curves over finite fields $\mathbb{F}_{p^n}$ of odd characteristic. For fixed p, the time complexity for a genus g hyperelliptic curve over $\mathbb{F}_{p^n}$ is $(g^{4+\varepsilon}n^{3+\varepsilon})$ and the space complexity is $O(g^3n^3)$.

The main result of this chapter is an extension of Kedlaya's algorithm to arbitrary hyperelliptic curves over finite fields of characteristic two. This extension is non-trivial since the algebraic de Rham cohomology of an arbitrary lift of the curve, minus the ramification points, is in general not isomorphic to its

Monsky-Washnitzer cohomology. This problem can be solved by constructing a model with all ramification points on the x-axis. To compute the zeta function of a genus g hyperelliptic curve over a finite field $\mathbb{F}_{2^n}$ the algorithm requires

- $O(g^3 n^3 (g + (\log n)^2 \log\log n) \log gn \log\log gn)$ bit-operations and $O(g^3 n^3)$ space in the average case,
- $O(g^3 n^3 (g^2 + (\log n)^2 \log\log n) \log gn \log\log gn)$ bit-operations and $O(g^4 n^3)$ space in the worst case.

This result has been published in [23, 24, 131].

An implementation of this algorithm in the `C` programming language shows that cryptographic sizes are now feasible for any genus g. For example, the order of a 168-bit Jacobian of a hyperelliptic curve of genus 2, 3 or 4 can be computed in less than 80 seconds. Furthermore, since the time complexity is also polynomial in the genus, we were able to compute the zeta function of a random genus 350 hyperelliptic curve over $\mathbb{F}_2$ in 3.5 days.

Chapter 5

C_{ab} Curves

In this chapter we develop a practical algorithm to compute the zeta function of a C_{ab} curve over any finite field of small characteristic. The algorithm requires $O(g^{5+\varepsilon}n^{3+\varepsilon})$ bit-operations and $O(g^3n^3)$ space for a C_{ab} curve of genus g defined over a finite field $\mathbb{F}_{p^n}$, with p fixed. Again we compute in the Monsky-Washnitzer cohomology of the curve, but the algorithm is much more general than the one described in the previous chapter. Furthermore, the techniques used to lift the Frobenius operator and to prove an explicit bound on the convergence rate of the resulting power series are valid for any smooth affine curve. Only the explicit description of a basis for the first Monsky-Washnitzer cohomology group and the reduction formulae to express any differential on this basis are limited to C_{ab} curves.

There exists several theoretical algorithms that could be used to compute the zeta function of a C_{ab} curve, but none of these is efficient enough to be practical. Pila [99] generalized Schoof's algorithm to abelian varieties, but requires explicit equations for the Jacobian and the group law, which is already very difficult for genus 2. Furthermore, the time complexity of this algorithm is at least doubly exponential in the genus of the curve. Adleman and Huang [3, 4] and later, Huang and Ierardi [59], improved Pila's algorithm for arbitrary curves and obtained a running time which still is exponential in the genus of the curve. Lauder and Wan [69] described a p-adic algorithm based on Dwork's trace formula to compute the zeta function of an arbitrary variety over a finite field of small characteristic. Applying this algorithm to C_{ab} curves results in a time complexity $O(g^{14}n^{13})$ and a space complexity $O(g^9n^8)$.

For special types of C_{ab} curves, two practical algorithms are currently known. The first is an algorithm by Gaudry and Gürel [49] who showed that Kedlaya's algorithm can be easily extended to superelliptic curves. The second is an extension by Ritzenthaler [102] of Mestre's algorithm [84] for ordinary hyperelliptic curves defined over a finite field of characteristic two. However, this extension is limited to ordinary non-hyperelliptic curves of genus 3. Furthermore, since the time complexity of Mestre's algorithm is exponential in the genus, any extension to genus larger than 3 will necessarily become less efficient.

5.1 Monsky-Washnitzer Cohomology

In this section we briefly recall the definition and main properties of Monsky-Washnitzer cohomology. More details can be found in the seminal papers by Monsky and Washnitzer [89, 90, 91], the lectures by Monsky [92] and the survey by van der Put [127].

Let $\overline{X}$ be a smooth affine variety over a finite field $\mathbb{F}_q$ with coordinate ring $\overline{A}$ and let $\mathbb{Q}_q$ be a degree n unramified extension of $\mathbb{Q}_p$ with valuation ring $\mathbb{Z}_q$ and residue field $\mathbb{Z}_q/(p\mathbb{Z}_q) = \mathbb{F}_q$. Elkik [31] showed that there always exists a smooth finitely generated $\mathbb{Z}_q$-algebra A such that $A/(pA) \cong \overline{A}$. To compute the zeta function of $\overline{X}$ we need to lift the Frobenius endomorphism $\overline{F}$ on $\overline{A}$ to the $\mathbb{Z}_q$-algebra A, but in general this is not possible. Note that for elliptic curves, Satoh solves this problem by using the Serre-Tate canonical lift which does admit a lift of the Frobenius endomorphism. To remedy this difficulty one could work with the p-adic completion A^∞ of A. But again we run into difficulties since the de Rham cohomology of A^∞ is larger than that of A. For example, consider the affine line over $\mathbb{F}_p$, so $A = \mathbb{Z}_q[x]$, then each term in $\sum_{n=0}^{\infty} p^n x^{p^n-1} dx$ is an exact differential form, but its sum is not, since $\sum_{n=0}^{\infty} x^{p^n}$ is not in A^∞. The main problem is that the series $\sum_{n=0}^{\infty} p^n x^{p^n-1}$ does not converge fast enough for its integral to converge as well. Monsky and Washnitzer solve this problem by working with a subalgebra $A^\dagger$ of A^∞, whose elements satisfy growth conditions. This *dagger ring* or *weak completion* $A^\dagger$ is defined as follows: write $A := \mathbb{Z}_q[x_1, \ldots, x_n]/(f_1, \ldots, f_m)$, then

$$A^\dagger := \mathbb{Z}_q\langle x_1, \ldots, x_n\rangle^\dagger/(f_1, \ldots, f_m)\,,$$

where $\mathbb{Z}_q\langle x_1, \ldots, x_n\rangle^\dagger$ consists of power series

$$\left\{ \sum a_\alpha x^\alpha \in \mathbb{Z}_q[[x_1, \ldots, x_n]] \mid \exists \gamma, \rho \in \mathbb{R}, \gamma > 0, 0 < \rho < 1, \forall \alpha : |a_\alpha|_p \le \gamma \rho^{|\alpha|} \right\}\,,$$

with $\alpha := (\alpha_1, \ldots, \alpha_n)$, $x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ and $|\alpha| := \sum_{i=1}^n \alpha_i$. Equivalently, $\mathbb{Z}_q\langle x_1, \ldots, x_n\rangle^\dagger$ can be defined as the set of overconvergent power series, i.e. elements of $\mathbb{Z}_q[[x_1, \ldots, x_n]]$ that converge in a polydisc

$$\{(x_1, \ldots, x_n) \in \mathbb{Q}_q^n \mid |x_1|_p \le \rho_1, \ldots, |x_n|_p \le \rho_n\}$$

with all $\rho_i > 1$. A homomorphic image of $\mathbb{Z}_q\langle x_1, \ldots, x_n\rangle^\dagger$ is called a weakly complete finitely generated (w.c.f.g.) algebra over $\mathbb{Z}_q$. The ring $A^\dagger$ clearly is a w.c.f.g. algebra and furthermore satisfies $A^\dagger/(pA^\dagger) = \overline{A}$ and is flat over $\mathbb{Z}_q$. An algebra which satisfies these three properties is called a *lift* of $\overline{A}$. One can show that if $\overline{A}$ is smooth and finitely generated, there always exists a lift $A^\dagger$ of $\overline{A}$ and that every lift of $\overline{A}$ is $\mathbb{Z}_q$-isomorphic to $A^\dagger$. Furthermore, let $\overline{B}/\mathbb{F}_q$ be smooth and finitely generated, with lift $B^\dagger$ and let $\overline{G} : \overline{A} \to \overline{B}$ be a morphism of $\mathbb{F}_q$-algebra's, then there exists a $\mathbb{Z}_q$-homomorphism $G : A^\dagger \to B^\dagger$ lifting $\overline{G}$. This implies that we can lift the q-th power Frobenius $\overline{F}$ from $\overline{A}$ to $A^\dagger$.

For $A^\dagger$ we can define the universal module $D^1(A^\dagger)$ of differentials

$$D^1(A^\dagger) := (A^\dagger\, dx_1 + \cdots + A^\dagger\, dx_n)/(\sum_{i=1}^{m} A^\dagger(\frac{\partial f_i}{\partial x_1}\, dx_1 + \cdots + \frac{\partial f_i}{\partial x_n}\, dx_n))\,.$$

Let $D^i(A^\dagger) := \bigwedge^i D^1(A^\dagger)$ be the i-th exterior product of $D^1(A^\dagger)$ and denote with $d_i : D^i(A^\dagger) \to D^{i+1}(A^\dagger)$ the exterior differentiation. Since $d_{i+1} \circ d_i = 0$ we get the de Rham complex $D(A^\dagger)$

$$0 \longrightarrow D^0(A^\dagger) \xrightarrow{d_0} D^1(A^\dagger) \xrightarrow{d_1} D^2(A^\dagger) \xrightarrow{d_2} D^3(A^\dagger) \cdots$$

The i-th cohomology group of $D(A^\dagger)$ is defined as $H^i(\overline{A}/\mathbb{Z}_q) := \mathrm{Ker}\, d_i/\mathrm{Im}\, d_{i-1}$ and $H^i(\overline{A}/\mathbb{Q}_q) := H^i(\overline{A}/\mathbb{Z}_q) \otimes_{\mathbb{Z}_q} \mathbb{Q}_q$ finally defines the i-th Monsky-Washnitzer cohomology group. For smooth, finitely generated $\mathbb{F}_q$-algebra's $\overline{A}$, van der Put [127] proved that the map $\overline{A} \mapsto H^\bullet(\overline{A}/\mathbb{Q}_q)$ is well defined and functorial, which justifies the notation. Replacing $A^\dagger$ with A in the above construction of the i-th Monsky-Washnitzer cohomology group $H^i(\overline{A}/\mathbb{Q}_q)$ gives rise to the i-th algebraic de Rham cohomology group $H^i_{DR}(A/\mathbb{Q}_q)$. Unlike the Monsky-Washnitzer cohomology, the algebraic de Rham cohomology essentially depends on the algebra A and in general $H^i(\overline{A}/\mathbb{Q}_q)$ will not be isomorphic to $H^i_{DR}(A/\mathbb{Q}_q)$.

Let $\mathcal{F}$ be a lift of the q-th power Frobenius endomorphism of $\overline{A}$ to $A^\dagger$, then $\mathcal{F}$ induces an endomorphism $\mathcal{F}_*$ on the cohomology groups $H^i(\overline{A}/\mathbb{Q}_q)$. The main theorem of Monsky-Washnitzer cohomology is that these groups satisfy a Lefschetz fixed point formula.

Theorem 5.1.1 (Lefschetz fixed point formula) *Let $\overline{X}/\mathbb{F}_q$ be a smooth affine variety of dimension d, then the number of $\mathbb{F}_q$-rational points on $\overline{X}$ equals*

$$\sum_{i=0}^{d}(-1)^i \mathrm{Tr}\left(q^d \mathcal{F}_*^{-1} | H^i(\overline{A}/\mathbb{Q}_q)\right) .$$

5.2 Cohomology of C_{ab} Curves

Let $\mathbb{F}_q$ be a finite field with $q = p^n$ elements and fix an algebraic closure $\overline{\mathbb{F}}_q$. For coprime positive integers $a, b \in \mathbb{N}$, a C_{ab} curve $\overline{C}$ is defined by an equation of the form

$$\overline{C} : y^a + \sum_{i=1}^{a-1} \overline{f}_i(x) y^i + \overline{f}_0(x) = 0 , \tag{5.1}$$

with $\overline{f}_i(x) \in \mathbb{F}_q[x]$ for $i = 0, \ldots, a-1$, $\deg \overline{f}_0 = b$ and $a \deg \overline{f}_i + bi < ab$ for $i = 1, \ldots, a-1$. Furthermore, $\overline{C}$ should be non-singular as an affine curve. Let $\overline{C}(x,y)$ denote the polynomial $y^a + \sum_{i=1}^{a-1} \overline{f}_i(x) y^i + \overline{f}_0(x)$, then the coordinate ring of $\overline{C}$ is $\overline{A} = \mathbb{F}_q[x,y]/(\overline{C}(x,y))$. Matsumoto [79] showed that a C_{ab} curve is always absolutely irreducible and has a unique $\mathbb{F}_q$-rational place $\overline{P}_\infty$ at infinity. Furthermore, the pole divisors of the functions x and y are $a\overline{P}_\infty$ and $b\overline{P}_\infty$ respectively. Since $\overline{C}$ is non-singular as an affine curve, the genus follows easily from the Hurwitz formula [43] and is given by

$$g = \frac{(a-1)(b-1)}{2} .$$

Let $\mathbb{Q}_q$ be a degree n unramified extension of $\mathbb{Q}_p$, with valuation ring $\mathbb{Z}_q$ and residue field $\mathbb{Z}_q/(p\mathbb{Z}_q) = \mathbb{F}_q$. Take arbitrary lifts $f_i(x) \in \mathbb{Z}_q[x]$ of $\overline{f}_i(x)$ such that $\deg f_i = \deg \overline{f}_i$ for $i = 0, \dots, a-1$ and consider the C_{ab} curve C defined by the equation

$$C : y^a + \sum_{i=1}^{a-1} f_i(x) y^i + f_0(x) = 0 \,. \tag{5.2}$$

Let $C(x,y)$ be the polynomial $y^a + \sum_{i=1}^{a-1} f_i(x) y^i + f_0(x)$, then the coordinate ring of C is $A = \mathbb{Z}_q[x,y]/(C(x,y))$.

Denote with $A^\dagger$ the weak completion of A. Using the equation of the curve, we can represent any element of $A^\dagger$ as a power series $\sum_{j=0}^{a-1}\sum_{i=0}^{\infty} a_{i,j} x^i y^j$. The growth condition on the dagger ring implies that there exist real numbers δ and $\epsilon > 0$ such that $\mathrm{ord}_p(a_{i,j}) \geq \epsilon(i+j) + \delta$. Lift the p-th power Frobenius σ on $\mathbb{F}_q$ to the the Frobenius substitution Σ on $\mathbb{Z}_q$. Any extension of Σ to an endomorphism of $A^\dagger$ satisfies

$$\Sigma(x) \equiv x^p \bmod p \,, \qquad \Sigma(y) \equiv y^p \bmod p \,, \qquad \Sigma(C(x,y)) = 0 \,. \tag{5.3}$$

Let $\Sigma(x) := x^p + \delta_x Z$ and $\Sigma(y) := y^p + \delta_y Z$ with $Z \in A^\dagger$ and δ_x, δ_y polynomials over $\mathbb{Z}_q$ which will be determined later, then Z must satisfy

$$G(Z) := \Sigma(C(x,y)) = C^\Sigma(x^p + \delta_x Z, y^p + \delta_y Z) = 0 \,, \tag{5.4}$$

where $C^\Sigma(x,y)$ is obtained by applying Σ to the coefficients of $C(x,y)$. Note that $Z = 0$ is a solution of the above equation modulo p. A zero of $G(Z)$ can be computed using the Newton iteration $Z_{k+1} \leftarrow Z_k - G(Z_k)/G'(Z_k)$ only if the derivative $G'(0)$ is invertible in $A^\dagger$, i.e. if $G'(0) \bmod p$ is a unit in $\mathbb{F}_q$. This leads to the following condition on the polynomials δ_x and δ_y

$$\begin{aligned} G'(0) &\equiv \delta_x \frac{\partial C^\Sigma}{\partial x}(x^p, y^p) + \delta_y \frac{\partial C^\Sigma}{\partial y}(x^p, y^p) \bmod p \\ &\equiv \delta_x \left(\frac{\partial C}{\partial x}(x,y)\right)^p + \delta_y \left(\frac{\partial C}{\partial y}(x,y)\right)^p \bmod p \,. \end{aligned} \tag{5.5}$$

Since the affine curve $\overline{C}$ is non-singular, Hilbert's Nullstellensatz implies that we can find polynomials $\overline{\alpha}, \overline{\beta}, \overline{\gamma} \in \mathbb{F}_q[x,y]$ such that

$$\overline{\alpha}(x,y)\frac{\partial \overline{C}}{\partial x}(x,y) + \overline{\beta}(x,y)\frac{\partial \overline{C}}{\partial y}(x,y) + \overline{\gamma}(x,y)\overline{C}(x,y) = 1 \,. \tag{5.6}$$

Thus $G'(0)$ will be invertible in $A^\dagger$ if δ_x and δ_y are arbitrary lifts of $\overline{\alpha}^p$ and $\overline{\beta}^p$ respectively. Note that the Newton polytopes of the polynomials $\overline{\alpha}$ and $\overline{\beta}$ determine the degrees of the coefficients of $G(Z)$, which in term influence the rate of convergence of the power series that satisfies $G(Z) = 0$. So before we can prove a lower bound of the convergence rate of the power series $\Sigma(x)$ and $\Sigma(y)$, we need to bound the Newton polytopes of the polynomials $\overline{\alpha}$, $\overline{\beta}$ and $\overline{\gamma}$. To this end we prove the following lemma.

LEMMA 5.2.1 *Let C be a C_{ab} curve defined by the polynomial $C(x,y) \in \mathbb{Z}_q[x,y]$ and assume that C is non-singular as an affine curve. Let $\mathcal{N}(C)$ denote the Newton polytope of $C(x,y)$, then there exists polynomials $\alpha, \beta, \gamma \in \mathbb{Z}_q[x,y]$ with Newton polytopes $\mathcal{N}(\alpha) \subset 2\mathcal{N}(C)$, $\mathcal{N}(\beta) \subset 2\mathcal{N}(C)$, $\mathcal{N}(\gamma) \subset 2\mathcal{N}(C)$ such that*

$$\alpha(x,y)\frac{\partial C}{\partial x}(x,y) + \beta(x,y)\frac{\partial C}{\partial y}(x,y) + \gamma(x,y)C(x,y) = 1\,. \tag{5.7}$$

PROOF: Define the polynomials $F_0, F_1, F_2 \in \mathbb{Z}_q[X,Y]$ by

$$F_0 := \frac{\partial C}{\partial x}(X^a, Y^b)\,, \quad F_1 := \frac{\partial C}{\partial y}(X^a, Y^b)\,, \quad F_2 := C(X^a, Y^b)\,.$$

Note that the Newton polytope of F_2 is given by $ab \cdot S_2$ with S_2 the standard simplex in $\mathbb{R}^2$. The transformation $x \leftarrow X^a, y \leftarrow Y^b$ can thus be interpreted as a reduction to the case where the polynomial F_2 is dense. Furthermore, since C is a C_{ab} curve $\deg F_0 = (b-1)a$ and $\deg F_1 = (a-1)b$. Let F_i^h be the homogenization of F_i for $i = 0,1,2$, then the system of homogeneous equations

$$F_0^h(X,Y,Z) = F_1^h(X,Y,Z) = F_2^h(X,Y,Z) = 0$$

has no solutions in $\mathbb{P}^2(\overline{\mathbb{F}}_q)$. Indeed, if $Z \neq 0$ then this follows immediately from the fact that C is non-singular as an affine curve. If $Z = 0$ the system of equations reduces to

$$aY^{(a-1)b} = 0, \quad b\lambda X^{a(b-1)} = 0, \quad Y^{ab} = \lambda X^{ab}\,,$$

with λ the leading coefficient of $C(x,0)$. Since $\gcd(a,b) = 1$ and λ is a unit in $\mathbb{Z}_q$, the only solution is $(0,0)$ which does not correspond to a point in $\mathbb{P}^2(\overline{\mathbb{F}}_q)$. Applying Theorem 5.2.1 to F_0, F_1, F_2 shows that there exists polynomials $G_0, G_1, G_2 \in \mathbb{Z}_q[X,Y]$ with $\deg G_0 \le 2ab - b - 2$, $\deg G_1 \le 2ab - a - 2$, $\deg G_2 \le 2ab - a - b - 2$ and

$$G_0 F_0 + G_1 F_1 + G_2 F_2 = 1\,. \tag{5.8}$$

Without loss of generality we can assume that all monomials in G_0, G_1, G_2 are of the form $X^{ai}Y^{bj}$ for $i, j \in \mathbb{N}$. Indeed, we can simply leave out all terms in G_0, G_1, G_2 that are not of this form and equation (5.8) will still hold. Finally, let $\alpha, \beta, \gamma \in \mathbb{Z}_q[x,y]$ be defined by

$$\alpha(X^a, Y^b) := G_0(X,Y)\,, \quad \beta(X^a, Y^b) := G_1(X,Y)\,, \quad \gamma(X^a, Y^b) := G_2(X,Y)\,,$$

then it is clear that $\mathcal{N}(\alpha) \subset 2\mathcal{N}(C)$, $\mathcal{N}(\beta) \subset 2\mathcal{N}(C)$, and $\mathcal{N}(\gamma) \subset 2\mathcal{N}(C)$, which finishes the proof. □

THEOREM 5.2.1 *Let $\mathbb{K}$ be a field or a discrete valuation ring and let $\mathfrak{m}$ be the maximal ideal of $\mathbb{K}$. Let $f_0, \ldots, f_n \in \mathbb{K}[x_1, \ldots, x_n]$ with $\deg f_i = d_i$ and define*

$$\rho = d_0 + \cdots + d_n - n - 1\,.$$

Denote with f_i^h the homogenization of f_i for $i = 0, \dots, n$. Assume that there are no points in projective n-space over the algebraic closure of $\mathbb{K}/\mathfrak{m}$ that satisfy the system of homogeneous equations

$$f_0^h = f_1^h = \cdots = f_n^h = 0\,.$$

Then there exists polynomials $g_0, \dots, g_n \in \mathbb{K}[x_1, \dots, x_n]$ with $\deg g_i \le \rho + 1 - d_i$ for $i = 0, \dots, n$ such that

$$\sum_{i=0}^{n} g_i f_i = 1\,.$$

Proof: Applying Lemma 5.2.2 to $f_0^h, \dots, f_n^h$ shows that there exists polynomials $G_0, \dots, G_n \in \mathbb{K}[x_0, \dots, x_n]$ such that

$$x_0^{\rho+1} = G_0 f_0^h + \cdots + G_n f_n^h\,,$$

and since $x_0^{\rho+1}$ and $f_0^h, \dots, f_n^h$ are homogeneous, we can assume that the G_i are homogeneous and

$$\deg G_i = \rho + 1 - \deg f_i = \rho + 1 - d_i \quad \text{or} \quad G_i = 0$$

for $i = 0, \dots, n$. Substituting $x_0 = 1$ then finishes the proof. □

Lemma 5.2.2 *Let $\mathbb{K}$ be a field or a discrete valuation ring and let $\mathfrak{m}$ be the maximal ideal of $\mathbb{K}$. Let $h_0, \dots, h_n \in \mathbb{K}[x_0, \dots, x_n]$ be homogeneous polynomials over $\mathbb{K}$ with $\deg h_i = d_i$ for $i = 0, \dots, n$ and let*

$$\rho = d_0 + \cdots + d_n - n - 1\,.$$

Suppose that there are no points in projective n-space over the algebraic closure of $\mathbb{K}/\mathfrak{m}$ that satisfy the system of homogeneous equations

$$h_0 = h_1 = \cdots = h_n = 0\,.$$

Then any monomial of degree $\rho + 1$ over $\mathbb{K}$ belongs to the ideal $(h_0, \dots, h_n)$ of $\mathbb{K}[x_0, \dots, x_n]$ generated by $h_0, \dots, h_n$.

Proof: The proof consists of two parts. In the first part, we assume that $\mathbb{K}$ is a field. Let $h_{\mathrm{aff}}(r)$ be the Hilbert function of the graded ring

$$\mathbb{K}[x_0, \dots, x_n]/(h_0, \dots, h_n)\,,$$

and let $h_{\mathrm{proj}}(r)$ be the Hilbert function of the graded ring

$$\mathbb{K}[x_0, \dots, x_n, t]/(h_0, \dots, h_n)\,.$$

By definition of the Hilbert function we have for all $r \in \mathbb{N}$

$$\begin{aligned} h_{\mathrm{aff}}(r) &:= \dim_{\mathbb{K}}\{f \in \mathbb{K}[x_0, \dots, x_n]/(h_0, \dots, h_n) \mid \deg f = r\}\,, \\ h_{\mathrm{proj}}(r) &:= \dim_{\mathbb{K}}\{f \in \mathbb{K}[x_0, \dots, x_n, t]/(h_0, \dots, h_n) \mid \deg f = r\}\,. \end{aligned}$$

Varying the degree of t from 0 to r implies

$$h_{\mathrm{proj}}(r) = \sum_{j=0}^{r} h_{\mathrm{aff}}(j)\,. \tag{5.9}$$

Note that h_{proj} is the Hilbert function of the projective completion of the affine scheme determined by $h_0 = \cdots h_n = 0$ inside $\mathbb{A}^{n+1}$. But this projective completion has only one geometric point namely the origin $(0,\ldots,0,1)$ with a certain multiplicity. From Hartshorne [57, Theorem I.7.5] it follows that its Hilbert function $h_{\mathrm{proj}}(r)$ is a constant for r big enough. Let $\xi \in \mathbb{N}$ such that $h_{\mathrm{proj}}(r)$ is constant for $r \geq \xi$ then by (5.9) we have

$$h_{\mathrm{aff}}(r) = 0 \quad \text{for all } r \geq \xi + 1\,.$$

This means that any homogeneous polynomial over $\mathbb{K}$ of degree $\geq \xi + 1$ belongs to $(h_0,\ldots,h_n)$. Hence it suffices to prove that $\xi \leq \rho$. The Hilbert function h_{aff} can be calculated using syzygies as in Harris [56]. Since the locus of the system of equations $h_0 = \cdots = h_n = 0$ in $\mathbb{P}^{n+1}$ is a complete intersection, we can use the Koszul complex for the syzygies. As shown in Harris [56, pp. 168-170], we can take $\xi \leq d_0 + \cdots d_n - (n+1)$. This finishes the proof of the first part.

In the second part of the proof, we assume that $\mathbb{K}$ is a discrete valuation ring with maximal ideal $\mathfrak{m}$. Note that the first part of the proof implies that Lemma 5.2.2 holds for $\mathbb{K}/\mathfrak{m}$.

Let $S_r(\mathbb{K})$ be the $\mathbb{K}$-module of homogeneous polynomials over $\mathbb{K}$ of degree r in the variables $x_0,\ldots,x_n$. Consider the $\mathbb{K}$-linear map $W_{\mathbb{K}}$

$$\begin{aligned} W_{\mathbb{K}} \,:\, S_{\rho+1-d_0}(\mathbb{K}) \oplus \cdots \oplus S_{\rho+1-d_n}(\mathbb{K}) &\to S_{\rho+1}(\mathbb{K}) \\ (g_0,\ldots,g_n) &\mapsto g_0 h_0 + \cdots + g_n h_n\,, \end{aligned}$$

then we have to prove that this map is surjective. The first part of the proof shows that the map

$$W_{\mathbb{K}/\mathfrak{m}} \,:\, S_{\rho+1-d_0}(\mathbb{K}/\mathfrak{m}) \oplus \cdots \oplus S_{\rho+1-d_n}(\mathbb{K}/\mathfrak{m}) \to S_{\rho+1}(\mathbb{K}/\mathfrak{m})$$

is surjective. Hence the matrix of the linear map $W_{\mathbb{K}/\mathfrak{m}}$ over the field $\mathbb{K}/\mathfrak{m}$ has a minor of maximal dimension whose determinant is non-zero. Therefore, the matrix of the $\mathbb{K}$-linear map $W_{\mathbb{K}}$ has a minor whose determinant is a unit in $\mathbb{K}$, which proves that $W_{\mathbb{K}}$ is surjective. □

REMARK 5.2.1 The first part of Theorem 5.2.1, i.e. for $\mathbb{K}$ a field, is a classical result by Macaulay [78]. The proof of Macaulay's theorem was communicated to us by Martin Sombra. Furthermore, Theorem 5.2.1 and Lemma 5.2.2 remain valid for any local ring $\mathbb{K}$ and the proofs are exactly the same.

Lemma 5.2.1 implies that $\mathcal{N}(\overline{\alpha}) \subset 2\mathcal{N}(\overline{C})$ and $\mathcal{N}(\overline{\beta}) \subset 2\mathcal{N}(\overline{C})$. Therefore we can choose δ_x and δ_y as arbitrary lifts of $\overline{\alpha}^p$ and $\overline{\beta}^p$ such that

$$\mathcal{N}(\delta_x) \subset 2p\mathcal{N}(C) \quad \text{and} \quad \mathcal{N}(\delta_y) \subset 2p\mathcal{N}(C)\,.$$

REMARK 5.2.2 Note that to compute $\overline{\alpha}, \overline{\beta}$ and $\overline{\gamma}$, we can simply apply linear algebra over $\mathbb{F}_q$, instead of using a variant of Buchberger's algorithm [13].

Using Newton iteration we can compute a solution to equation (5.4) as an element of the p-adic completion of A. Furthermore, a theorem by Bosch [12] implies that there exists a solution in $A^\dagger$. However, this theorem does not provide an explicit lower bound on the rate of convergence, which is needed in the complexity analysis. Therefore, we prove the following lemma.

LEMMA 5.2.3 *Let C be a C_{ab} curve over $\mathbb{Z}_q$ with coordinate ring A and let $G(Z) = \sum_{k=0}^{d} g_k(x,y)Z^k \in A[Z]$ with $g_k(x,y) = \sum_{j=0}^{a-1}\sum_{i=0}^{d_k} a_{i,j,k}x^i y^j$. If $G(0) \equiv 0 \bmod p$ and $G'(0) \equiv 1 \bmod p$, then $G(Z) = 0$ has a unique solution $Z_0 = \sum_{j=0}^{a-1}\sum_{i=0}^{\infty} b_{i,j}x^i y^j$ in $A^\dagger$ with $Z_0 \equiv 0 \bmod p$ and*

$$\mathrm{ord}_p(b_{i,j}) \geq \frac{i}{2(b+m)},$$

where $m = \max_k\{d_k\}$.

PROOF: Let $\overline{\mathbb{Q}}_q$ be an algebraic closure of $\mathbb{Q}_q$ and $\overline{\mathbb{Z}}_q$ the ring of integral elements of $\overline{\mathbb{Q}}_q$. Take $\epsilon_1 = p^{1/k_1}, \epsilon_2 = p^{1/k_2}, \epsilon_3 = p^{1/k_3} \in \overline{\mathbb{Z}}_q$ with $k_1, k_2, k_3 \in \mathbb{Q}_{>0}$. Consider the transformation $x' = \epsilon_1 x$, $y' = \epsilon_2 y$ and $Z' = \epsilon_3^{-1}Z$. Since C is a C_{ab} curve, $(y')^a$ will be an integral linear combination of $(y')^i$ and $(x')^j$ for $i = 0, \ldots, a-1$ and $j = 0, \ldots, b$ if

$$k_1 \geq \frac{b}{a}k_2\,. \tag{5.10}$$

After the change of variables $x = \epsilon_1^{-1}x'$, $y = \epsilon_2^{-1}y'$, $Z = \epsilon_3 Z'$ and multiplication with ϵ_3^{-1}, the equation $G(Z) = 0$ becomes

$$\sum_{k=0}^{d} \epsilon_3^{k-1} g_k(\epsilon_1^{-1}x', \epsilon_2^{-1}y')(Z')^k = 0\,. \tag{5.11}$$

Let $\overline{P}$ be the maximal ideal of $\overline{\mathbb{Z}}_q$ containing p. Since $G'(0) \equiv 1 \bmod p$, we can choose k_1, k_2 such that $g_1(\epsilon_1^{-1}x', \epsilon_2^{-1}y') \in 1 + \overline{P}$. Indeed, it suffices to take k_1, k_2 large enough such that

$$\frac{d_1}{k_1} + \frac{a-1}{k_2} < 1\,. \tag{5.12}$$

The equation $G(0) \equiv 0 \bmod p$ implies that $\epsilon_3^{-1}g_0(\epsilon_1^{-1}x', \epsilon_2^{-1}y') \in \overline{P}$ if k_1, k_2, k_3 are chosen such that

$$\frac{d_0}{k_1} + \frac{a-1}{k_2} + \frac{1}{k_3} < 1\,. \tag{5.13}$$

Finally we can choose k_1, k_2, k_3 such that $\epsilon_3^{k-1}g_k(\epsilon_1^{-1}x', \epsilon_2^{-1}y')$ for $k = 2, \ldots, d$ have integral coefficients; it suffices to take

$$\frac{k-1}{k_3} \geq \frac{d_k}{k_1} + \frac{a-1}{k_2} \quad k = 2, \ldots, d\,. \tag{5.14}$$

Let $m = \max_k\{d_k\}$, then one easily verifies that

$$k_1 \geq 2(b+m)\,, \qquad k_2 = \frac{a}{b}\,k_1\,, \qquad k_3 = 2\,,$$

satisfy inequalities (5.10) and (5.12-5.14).

Hensel's lemma implies that there exists a unique element $Z_0' \in \overline{\mathbb{Z}}_q[[x',y']]$ with $Z_0' \equiv 0 \bmod \overline{P}$ which satisfies equation (5.11). Using the equation of the curve and inequality (5.10), we can write Z_0' as $\sum_{j=0}^{a-1}\sum_{i=0}^{\infty} c_{i,j}(x')^i(y')^j$ with $c_{i,j} \in \overline{\mathbb{Z}}_q$. Substituting $x' = \epsilon_1 x$, $y' = \epsilon_2 y$ in Z_0' and multiplying with ϵ_3 we obtain the unique solution $Z_0 = \sum_{j=0}^{a-1}\sum_{i=0}^{\infty} \epsilon_1^i \epsilon_2^j \epsilon_3 c_{i,j} x^i y^j$ of $G(Z) = 0$ with $Z_0 \equiv 0 \bmod p$. Since $\mathrm{ord}_p(\epsilon_1^i\epsilon_2^j\epsilon_3 c_{i,j}) \geq \mathrm{ord}_p(\epsilon_1^i)$, we conclude that

$$\mathrm{ord}_p(b_{i,j}) \geq \frac{i}{k_1} = \frac{i}{2(b+m)}\,.$$

□

Since $G(Z) = C^{\Sigma}(x^p + \delta_x Z, y^p + \delta_y Z)$ with $C(x,y)$ the equation of a C_{ab} curve, we can obtain a much better bound than the one given in Lemma 5.2.3.

Corollary 5.2.1 *With the notation of Lemma 5.2.3, suppose $\delta_x, \delta_y \in \mathbb{Z}_q[x,y]$ satisfy $\mathcal{N}(\delta_x) \subset 2p\mathcal{N}(C)$, $\mathcal{N}(\delta_y) \subset 2p\mathcal{N}(C)$ and*

$$\delta_x\left(\frac{\partial C}{\partial x}(x,y)\right)^p + \delta_y\left(\frac{\partial C}{\partial y}(x,y)\right)^p \equiv 1 \bmod p$$

in $\mathbb{Z}_q[x,y]/(C(x,y))$. Let $G(Z) = C^{\Sigma}(x^p + \delta_x Z, y^p + \delta_y Z)$, then $G(Z) = 0$ has a unique solution $Z_0 = \sum_{j=0}^{a-1}\sum_{i=0}^{\infty} b_{i,j}x^i y^j$ in $A^\dagger$ with $Z_0 \equiv 0 \bmod p$ and

$$\mathrm{ord}_p(b_{i,j}) \geq \frac{i}{7pb}\,.$$

Proof: Since $G(Z) = C^{\Sigma}(x^p+\delta_x Z, y^p+\delta_y Z)$, the degree in Z is $d := \max\{a,b\}$. Let $G(Z) = \sum_{k=0}^{d} g_k(x,y)Z^k$, then an easy calculation shows that

$$\mathcal{N}(g_k) \subset (2k+1)p\mathcal{N}(C)\,.$$

This implies that the $d_k := \deg_x(g_k)$ in the proof of Lemma 5.2.3 are bounded by $d_k \leq (2k+1)pb$ for $k = 0,\ldots,d$. We can therefore use the values

$$k_1 = 7pb\,, \qquad k_2 = 7pa\,, \qquad k_3 = 14/11\,,$$

which ends the proof of Corollary 5.2.1. □

Lemma 5.2.3 implies that we can lift the Frobenius $\overline{F}$ to an endomorphism $\mathcal{F}$ on the dagger ring $A^\dagger$ by defining $\mathcal{F} := \Sigma^n$. Before we can actually compute the zeta function using the Lefschetz fixed point theorem, we need to determine a basis of the $\mathbb{Q}_q$-vector space $H^1(\overline{A}/\mathbb{Q}_q)$. To this end we first construct a basis for the algebraic de Rham cohomology $H^1_{DR}(A/\mathbb{Q}_q)$ of A.

Every element of $H^1_{DR}(A/\mathbb{Q}_q)$ can be written as a linear combination of differentials of the form $x^i y^j\,dx$ and $x^i y^j\,dy$ with $i,j \in \mathbb{N}$. Using the equation of the curve we can take $0 \leq j < a$. Since $d(x^i y^{j+1})$ is exact, we conclude that $H^1_{DR}(A/\mathbb{Q}_q)$ is generated by differentials of the form $x^i y^j\,dx$ with $i,j \in \mathbb{N}$ and $0 < j < a$. Differentiating the equation of the curve and multiplying with $x^l y^j$ leads to

$$x^l(\sum_{k=1}^{a-1} f_k'(x)y^k + f_0'(x))y^j\,dx = -x^l(ay^{a-1} + \sum_{k=1}^{a-1} f_k(x)ky^{k-1})y^j\,dy\,.$$

Since $d(x^l(\frac{a}{a+j}y^{a+j} + \sum_{k=1}^{a-1}\frac{k}{k+j}f_k(x)y^{k+j}))$ is exact, we conclude that

$$\begin{aligned} &x^l\left(\sum_{k=1}^{a-1}\frac{j}{k+j}f_k'(x)y^k + f_0'(x)\right)y^j\,dx \\ &\qquad - lx^{l-1}\left(\frac{a}{a+j}y^a + \sum_{k=1}^{a-1}\frac{k}{k+j}f_k(x)y^k\right)y^j\,dx \equiv 0\,. \end{aligned} \tag{5.15}$$

The differentials $\omega_1 = x^l f_0'(x)y^j\,dx$ and $\omega_2 = -lx^{l-1}\frac{a}{a+j}y^{a+j}\,dx$ determine the pole order of the above exact differential at P_∞. Let λ be the leading coefficient of f_0, then the pole order of ω_1 is determined by $\lambda b x^{l+b-1}y^j\,dx$. Replacing y^a by $-f_0(x) - \sum_{i=1}^{a-1} f_i(x)y^i$ in ω_2 shows that the pole order of ω_2 is determined by $l\frac{a}{a+j}\lambda x^{l+b-1}y^j\,dx$. Thus the pole order of (5.15) is equal to the pole order of

$$(b + l\frac{a}{a+j})\lambda x^{l+b-1}y^j\,dx\,.$$

Note that the leading coefficient $(b + la/(a+j))\lambda$ is non-zero. Since C is a C_{ab} curve, we have $\mathrm{ord}_{P_\infty}(x^i) = -ia$, $\mathrm{ord}_{P_\infty}(y^j) = -jb$ and $\mathrm{ord}_{P_\infty}(dx) = -(a+1)$. Therefore we conclude that the exact differential (5.15) has a pole at P_∞ of order $a(l+b) + jb + 1$. The differential $x^i y^j\,dx$ with $0 < j < a$ can thus be reduced by subtracting a suitable multiple of the exact differential given in equation (5.15) for $l = i - b + 1$. Note that the polynomial in equation (5.15) is understood to be reduced using the equation of the curve. This implies that the reduction of $x^i y^j\,dx$ is of the form $\sum_{l=1}^{a-1}\sum_{k=0}^{N} a_{k,l}x^k y^l\,dx$ with $a_{k,l} \in \mathbb{Q}_q$ and this latter differential has a strictly smaller pole order than the former. Furthermore, taking $l = 0$ in equation (5.15) shows that also differentials of the form $x^{b-1}y^j\,dx$ with $0 < j < a$ can be reduced. This ends the proof that the algebraic de Rham cohomology $H^1_{DR}(A/\mathbb{Q}_q)$ is generated by the differentials $x^i y^j\,dx$ for $i = 0,\ldots,b-2$ and $j = 1,\ldots,a-1$. Since $\dim H^1_{DR}(A/\mathbb{Q}_q) = 2g$, these differentials also form a basis of $H^1_{DR}(A/\mathbb{Q}_q)$.

To prove that the first Monsky-Washnitzer cohomology group $H^1(\overline{A}/\mathbb{Q}_q)$ is generated by the same differential forms as the algebraic de Rham cohomology, we need to bound the denominators introduced during the reduction process.

LEMMA 5.2.4 *Let C be a C_{ab} curve over $\mathbb{Z}_q$ and let $A = \mathbb{Z}_q[x,y]/(C(x,y))$ denote the coordinate ring of C. Suppose that*

$$x^k y^l \; dx = \sum_{j=1}^{a-1} \sum_{i=0}^{b-2} a_{i,j} x^i y^j \; dx + dS, \tag{5.16}$$

with $k, l \in \mathbb{N}$, $0 \leq l < a$, $a_{i,j} \in \mathbb{Q}_q$ and $S \in A \otimes \mathbb{Q}_q$. Then $p^m a_i \in \mathbb{Z}_q$, $p^m S - \beta \in A$, with $m = \lfloor \log_p((k+1)a + lb) \rfloor + \epsilon$ and β a suitable element in $\mathbb{Q}_q$. Furthermore, ϵ is bounded by $4(a-1)b \lfloor \log_p(2a-1) \rfloor$.

PROOF: The proof has two distinct parts. The first part is similar to Kedlaya's argument in [61, Lemma 3], and is based on a local analysis around the point at infinity P_∞ of the curve C. Since $\gcd(a,b) = 1$ we can find $c, d \in \mathbb{N}$, with $0 < c < b$ and $0 < d < a$ such that

$$ac - bd = -1\,.$$

Let $t = x^c / y^d$, then t is a local parameter at P_∞ and one verifies that

$$x = t^{-a} \left(\frac{1}{(-\lambda)^d} + \sum_{j=1}^{\infty} \alpha_j t^j \right) \quad \text{and} \quad y = t^{-b} \left(\frac{1}{(-\lambda)^c} + \sum_{j=1}^{\infty} \beta_j t^j \right), \tag{5.17}$$

where λ equals the leading coefficient of $C(x,0)$ and $\alpha_j, \beta_j \in \mathbb{Z}_q$. Note that λ is a unit in $\mathbb{Z}_q$, since $\deg C(x,0) = \deg \overline{C}(x,0) = b$. Expressing x, y as functions of t we get $x^i y^j = u_{i,j} t^{-ia-jb} + \cdots$ and $dx = (-uat^{-a-1} + \cdots)dt$ with $u_{i,j}$ and u units in $\mathbb{Z}_q$. Writing $S = \sum_{j=0}^{a-1} \sum_{i=0}^{N} b_{i,j} x^i y^j$ with $b_{i,j} \in \mathbb{Q}_q$, $N \in \mathbb{N}$ and substituting these expressions in equation (5.16) gives

$$\left(\sum_{j \geq -(k+1)a - lb - 1} \gamma_j t^j \right) dt = \left(\sum_{j \geq -2g - ab} \eta_j t^j \right) dt + d \left(\sum_{j \geq -Na - (a-1)b} \nu_j t^j \right),$$

with $\gamma_j \in \mathbb{Z}_q$ and $\eta_j, \nu_j \in \mathbb{Q}_q$ for all j. Integrating with respect to t and multiplying with p^m leads to an equation of the form

$$\sum_{j \geq -(k+1)a - lb} \gamma'_j t^j \quad = \quad \sum_{j \geq -2g - ab + 1} \eta'_j t^j \; + \sum_{j \geq -Na - (a-1)b} p^m \nu_j t^j\,, \tag{5.18}$$

with $\gamma'_j \in \mathbb{Z}_q$ and $\eta'_j \in \mathbb{Q}_q$ for all j. Indeed, the integration process introduces denominators which become integral upon multiplication with p^m.

Equation (5.18) implies that $b_{i,j} = 0$ for

$$ia + jb > \max\{(k+1)a + lb, 2g + ab - 1\}\,,$$

so it suffices to take $N = \max\{k + 1 + b, 2b - 1\}$. Indeed, since a and b are coprime, the pole orders of $x^i y^j$ for $i, j \in \mathbb{N}$ and $0 \leq j < a$ are all different. Furthermore, we claim that equation (5.18) also implies that $p^m b_{i,j}$ is integral for $ia + jb \geq 2g + ab$. To see this, choose $(r, s) \in \mathbb{N} \times \mathbb{N}$ such that $b_{r,s} \neq 0$ and $ra + sb$ maximal. If $ra + sb \geq 2g + ab$, then $p^m b_{r,s}$ will be integral, since $b_{r,s} = \nu_{-ra-sb}/u_{r,s}$ with $u_{r,s}$ a unit in $\mathbb{Z}_q$ and from equation (5.18) it follows that $p^m \nu_j$ is integral for $j \leq -2g - ab$. Bringing the expression for $p^m b_{r,s} x^r y^s$ to the left hand side of equation (5.18) and repeating the same argument shows that $p^m b_{i,j}$ is integral for $ia + jb \geq 2g + ab$.

We now turn to the second part of the proof which determines ϵ. Define the differential

$$\omega = p^m x^k y^l \, dx - p^m d \left(\sum_{0 \leq j < a} \sum_{\substack{0 \leq i \leq N \\ ia+jb \geq 2g+ab}} b_{i,j} x^i y^j \right), \tag{5.19}$$

then we have just shown that ω has integral coefficients. Let C_x and C_y denote the partial derivatives of $C(x, y)$ to x and y respectively. From equation (5.19) it is clear that in $D^1(A \otimes \mathbb{Q}_q)$ we have

$$\begin{aligned} C_x \, \omega &= P_1(x, y) \, dy \,, \\ C_y \, \omega &= P_2(x, y) \, dx \,, \end{aligned}$$

with $P_1, P_2 \in \mathbb{Z}_q[x, y]$. Note that a priori we cannot bound the degrees of P_1 and P_2 independently of k. However, equation (5.16) and the definition of S shows that we can also write

$$\omega = p^m \sum_{j=1}^{a-1} \sum_{i=0}^{b-2} a_{i,j} x^i y^j \, dx + p^m d \left(\sum_{0 \leq j < a} \sum_{\substack{0 \leq i \leq N \\ ia+jb < 2g+ab}} b_{i,j} x^i y^j \right). \tag{5.20}$$

Multiplying this equation with C_x and C_y respectively leads to

$$\begin{aligned} C_x \, \omega &= P_1'(x, y) \, dy \,, \\ C_y \, \omega &= P_2'(x, y) \, dx \,, \end{aligned}$$

with P_1', P_2' polynomials over $\mathbb{Q}_q$, but of bounded degree. To make this bound explicit, we compute the pole orders of $C_x \, \omega$ and $C_y \, \omega$. From equation (5.20) it is clear that $-\mathrm{ord}_{P_\infty} \omega \leq 2g + ab$. Furthermore, an easy calculation shows that $-\mathrm{ord}_{P_\infty} C_x = (b - 1)a$, $-\mathrm{ord}_{P_\infty} C_y = (a - 1)b$, $-\mathrm{ord}_{P_\infty} dx = a + 1$ and $-\mathrm{ord}_{P_\infty} dy = b + 1$. Therefore, we conclude that

$$\begin{aligned} -\mathrm{ord}_{P_\infty} P_1'(x, y) &\leq 3ab - 2a - 2b \,, \\ -\mathrm{ord}_{P_\infty} P_2'(x, y) &\leq 3ab - 2a - 2b \,. \end{aligned} \tag{5.21}$$

Using the equation of the curve, we can assume that the degree in y of P_1, P_2, P_1' and P_2' is smaller than a. Since $C_x \, \omega = P_1(x, y) \, dy = P_1'(x, y) \, dy$, we have $P_1 = P_1'$ and similarly, $P_2 = P_2'$. Thus P_1 and P_2 have integral coefficients and their pole orders are bounded by (5.21).

Since the curve is non-singular, Lemma 5.2.1 implies that there exist polynomials $\alpha, \beta \in \mathbb{Z}_q[x, y]$ such that $\alpha C_x + \beta C_y = 1$ in the ring A. Furthermore, the Newton polytopes of α and β satisfy

$$\mathcal{N}(\alpha) \subset 2\mathcal{N}(C) \quad \text{and} \quad \mathcal{N}(\beta) \subset 2\mathcal{N}(C)\,.$$

Therefore, we can recover ω as

$$\omega = \alpha P_1\, dy + \beta P_2\, dx\,,$$

with $-\mathrm{ord}_{P_\infty} \alpha P_1 \leq 5ab - 2a - 2b$ and $-\mathrm{ord}_{P_\infty} \beta P_2 \leq 5ab - 2a - 2b$. From equation (5.19) also follows that

$$\omega \equiv p^m \sum_{j=1}^{a-1} \sum_{i=0}^{b-2} a_{i,j} x^i y^j\, dx\,,$$

so the $p^m a_{i,j}$ are obtained by reducing $\omega = \alpha P_1\, dy + \beta P_2\, dx$. To bound the denominators introduced during this second reduction stage, we analyse the explicit reduction formula (5.15). In every reduction step, the valuation of the denominators is trivially bounded by $\lfloor \log_p(2a-1) \rfloor$ since $j < a$. The bound on the pole order of βP_2 implies that $\deg_x(\beta P_2) \leq 5b - 3$, so the total number of steps to reduce $\beta P_2\, dx$ is strictly less than $4(a-1)b$. Let $T(x,y) = \int \alpha(x,y) P_1(x,y)\, dy$, then clearly $\alpha P_1\, dy \equiv -\frac{\partial T}{\partial x}(x,y)\, dx$ and $-\mathrm{ord}_{P_\infty} \frac{\partial T}{\partial x}(x,y) \leq 5ab - 3a - b$. This implies that the total number of steps to reduce $\alpha P_1\, dy$ is also strictly less than $4(a-1)b$. Taking into account that $T(x,y)$ becomes integral upon multiplication with $\lfloor \log_p a \rfloor$, we conclude that $\epsilon \leq 4(a-1)b \lfloor \log_p(2a-1) \rfloor$.

□

Remark 5.2.3 In Lemmata 4.3.2 and 4.4.2, we used resultants to prove that ϵ is in fact of order $O(\log g)$ and not $O(g \log a)$ as in the above lemma. Practical experiments indicate that $O(\log g)$ is indeed the correct behavior. Furthermore, it appears likely that the reasoning using resultants can be generalized to C_{ab} curves. Fortunately, the worse bound on ϵ does not influence the theoretical time complexity.

Lemma 5.2.3 implies that the basis for $H^1_{DR}(A/\mathbb{Q}_q)$ also generates $H^1(\overline{A}/\mathbb{Q}_q)$, since the reduction process converges. Indeed, for $a_{k,l} x^k y^l \in A^\dagger$ with $k, l \in \mathbb{N}$ and $0 \leq l < a$, the valuation of $a_{k,l}$ grows as a linear function of k, while the valuation of the denominators introduced during the reduction of $a_{k,l} x^k y^l\, dx$ are only logarithmic in k. Furthermore, since $\dim H^1_{DR}(A/\mathbb{Q}_q) = 2g = \dim H^1(\overline{A}/\mathbb{Q}_q)$, the basis for $H^1_{DR}(A/\mathbb{Q}_q)$ is also a basis for $H^1(\overline{A}/\mathbb{Q}_q)$.

Let $\widetilde{C}$ be the unique smooth projective curve birational to C. Applying the

Lefschetz fixed point theorem to $\overline{C}$ then leads to

$$\begin{aligned}\#\widetilde{C}(\mathbb{F}_{q^k}) &= 1 + \#\overline{C}(\mathbb{F}_{q^k}) \\ &= 1 + \operatorname{Tr}\left(q^k\mathcal{F}_*^{-k}|H^0(\overline{A}/\mathbb{Q}_q)\right) - \operatorname{Tr}\left(q^k\mathcal{F}_*^{-k}|H^1(\overline{A}/\mathbb{Q}_q)\right) \\ &= 1 + q^k - \sum_{i=1}^{2g}\alpha_i^k\,,\end{aligned}$$

with α_i the eigenvalues of $q\mathcal{F}_*^{-1}$ on $H^1(\overline{A}/\mathbb{Q}_q)$. The Weil conjectures [137] imply that there exist algebraic integers $\beta_1,\ldots,\beta_{2g}$ such that for all $k > 0$ we have

$$\#\widetilde{C}(\mathbb{F}_{q^k}) = q^k + 1 - \sum_{i=1}^{2g}\beta_i^k\,.$$

Furthermore, the β_i satisfy $\beta_i\beta_{g+i} = q$ and $|\beta_i| = \sqrt{q}$ for $i = 1,\ldots,2g$ where the indices are taken modulo $2g$. Comparing the zeta function using both formulae for $\#\widetilde{C}(\mathbb{F}_{q^k})$ shows that $\{\alpha_i\} = \{\beta_i\}$. Since $\alpha_i\alpha_{g+i} = q$, the α_i are also the eigenvalues of $\mathcal{F}_*$ on $H^1(\overline{A}/\mathbb{Q}_q)$. Let $\chi(t)$ be the characteristic polynomial of $\mathcal{F}_*$ on $H^1(\overline{A}/\mathbb{Q}_q)$, then we conclude that

$$Z(\widetilde{C}/\mathbb{F}_q;t) = \frac{t^{2g}\chi(1/t)}{(1-t)(1-qt)}\,.$$

Since $\mathcal{F}_* = \Sigma_*^n$, we can recover $\chi(t)$ as the characteristic polynomial of the norm $M_{\mathcal{F}} = \Sigma^{n-1}(M)\cdots\Sigma(M)M$ of the matrix M through which Σ_* acts on $H^1(\overline{A}/\mathbb{Q}_q)$.

5.3 Detailed Algorithm

In this section, we give a detailed description of an algorithm to compute the characteristic polynomial of Frobenius $\chi(t)$ and the zeta function of a smooth projective C_{ab} curve $\widetilde{C}$ of genus g over a finite field $\mathbb{F}_q$ with $q = p^n$.

The Weil conjectures imply that $\chi(t) = t^{2g} + a_1t^{2g-1} + \cdots + a_{2g}$ with $a_i \in \mathbb{Z}$ for $i = 1,\ldots,2g$ and $q^{g-i}a_i = a_{2g-i}$, so it suffices to compute $a_1,\ldots,a_g$. Since the a_i are the sum of $\binom{2g}{i}$ i-fold products of eigenvalues of Frobenius, we have

$$|a_i| \le \binom{2g}{i}q^{i/2} \le \binom{2g}{g}q^{g/2} \le 2^{2g}q^{g/2}\,.$$

Therefore it suffices to compute the action of $\mathcal{F}_*$ on $H^1(\overline{A}/\mathbb{Q}_q)$ modulo p^B with

$$B \ge \left\lceil \log_p\left(2\binom{2g}{g}q^{g/2}\right)\right\rceil\,.$$

However, the reduction of differential forms causes a loss of precision which needs to be taken into account if we are to recover the $a_1,\ldots,a_g$ modulo p^B.

Suppose we have computed

$$\Sigma(x) = \sum_{j=0}^{a-1}\sum_{i=0}^{L} a_{i,j} x^i y^j \bmod p^N \quad \text{and} \quad \Sigma(y) = \sum_{j=0}^{a-1}\sum_{i=0}^{L} b_{i,j} x^i y^j \bmod p^N ,$$

with not all $a_{L,j}$ and $b_{L,j}$ zero, then Corollary 5.2.1 implies $\operatorname{ord}_p(a_{i,j}) \geq \epsilon i + \delta$ and $\operatorname{ord}_p(b_{i,j}) \geq \epsilon i + \delta$ with $\epsilon = (7pb)^{-1}$ and $\delta = -2/7$. The maximum loss of precision will be caused by the reduction of the differential $\Sigma(x)^{b-2}\Sigma(y)^{a-1}d\Sigma(x)$. Since $\Sigma(x)$ and $\Sigma(y)$ are overconvergent, also $\Sigma(x)^k$ and $\Sigma(y)^k$ for $k \in \mathbb{N}$ will be overconvergent with the same rate of convergence, i.e. ϵ does not change. However, an easy calculation shows that δ will change according to $\delta_k = -(2k)/7$ for $k \in \mathbb{N}$. Therefore, the maximum degree in x of $\Sigma(x)^{b-2}\Sigma(y)^{a-1}d\Sigma(x) \bmod p^N$ is bounded by

$$U := 7pbN + 2pb(a+b) .$$

Applying Lemma 5.2.4 to the differential $x^U y^{a-1}\, dx$ shows that N should satisfy

$$N - \lfloor \log_p(abp(7N + 2(a+b) + 1)) \rfloor \geq B + \epsilon . \tag{5.22}$$

Note that this implies that N is $O(ng)$, since $B + \epsilon$ itself is $O(ng)$.

Remark 5.3.1 Since the matrix M is not necessarily defined over $\mathbb{Z}_q$, we could lose an extra cn bits of precision during the computation of $M_{\mathcal{F}}$, where 2^c is the largest denominator appearing in M. By Corollary 5.2.1 and Lemma 5.2.4, c is bounded by $O(g \log a)$ independently of n. In theory we would therefore have to replace the bound B by $B + cn$, which does not change the complexity of the algorithm. In practice however it turns out that the largest denominator appearing in $M_{\mathcal{F}}$ is always the same as the largest denominator appearing in M and therefore it is not necessary to increase B. This phenomenon can be heuristically explained as follows: since the eigenvalues of $\mathcal{F}_* = \Sigma_*^n$ on $H^1(\overline{A}/\mathbb{Q}_q)^-$ have non-negative 2-adic valuation there is a $\mathbb{Z}_q$-submodule of $H^1(\overline{A}/\mathbb{Q}_q)^-$ which is stable under the action of Σ_*. For this $\mathbb{Z}_q$-submodule we can take for instance the canonical image of the crystalline cohomology of C over $\mathbb{Z}_q$. Note that the $\mathbb{Z}_q$-submodule generated by $x^i y\, dx$ for $i = 0, \ldots, 2g-1$ is not canonical and in general not stable under Σ_*. Let A_0 be the matrix that expresses $x^i y\, dx$ for $i = 0, \ldots, 2g-1$ in terms of a basis of such a stable $\mathbb{Z}_q$-submodule and let A be A_0 times a power of 2 such that A is a matrix over $\mathbb{Z}_q$ which is not zero modulo 2. Then $M = A^{-1}U\Sigma(A)$ where U is the matrix of Σ_* with respect to the new basis. Note that U is a matrix over $\mathbb{Z}_q$ and that the norm of M equals $A^{-1}U\Sigma(U)\cdots\sigma^{n-1}(U)A$. Thus the loss of precision is no more than $2d$ bits where d is the 2-adic valuation of $\det(A)$. If U and A are generic enough then $|c - d|$ is small. Furthermore, the bound (5.22) turns out to be slightly larger than what is needed and compensates for the loss of $2d$ bits.

The function Cab_Curve_Zeta_Function given in Algorithm 5.3.1 computes the zeta function of a smooth projective C_{ab} curve $\widetilde{C}$ over $\mathbb{F}_q$ with $q = p^n$. In step 1 we determine the minimal precision N that satisfies (5.22). Given a curve $\overline{C} : y^a + \sum_{i=1}^{a-1} \overline{f}_i(x)y^i + \overline{f}_0(x) = 0$, the function LIFT_CAB_CURVE takes arbitrary lifts $f_i(x) \in \mathbb{Z}_q[x]$ of $\overline{f}_i(x)$ such that $\deg f_i = \deg \overline{f}_i$ and returns the curve $C : y^a + \sum_{i=1}^{a-1} f_i(x)y^i + f_0(x) = 0$. In step 3 we compute $\overline{\alpha}, \overline{\beta}, \overline{\gamma} \in \mathbb{F}_q[x]$ satisfying equation (5.6) using the function NULLSTELLENSATZ. This function can be implemented as a variant of Buchberger's Gröbner basis algorithm [13], but Remark 5.2.2 implies that we can simply use linear algebra over $\mathbb{F}_q$ which is much faster.

ALGORITHM 5.3.1 (CAB_CURVE_ZETA_FUNCTION)

`INPUT:` Non-singular C_{ab} curve $\overline{C}$ over $\mathbb{F}_{p^n}$ given by (5.1).
`OUTPUT:` The zeta function $Z(\widetilde{C}/\mathbb{F}_{p^n}; t)$.

1. $B \geq \left\lceil \log_p \left(2\binom{2g}{g} q^{g/2} \right) \right\rceil$;
   $N - \lfloor \log_p(abp(7N + 2(a+b) + 1)) \rfloor \geq B + \epsilon$;
2. $g = (a-1)(b-1)/2$; $C =$ `Lift_Cab_Curve(`$\overline{C}$`)`;
3. $(\overline{\alpha}, \overline{\beta}, \overline{\gamma}) =$ `Nullstellensatz(`$\frac{\partial \overline{C}}{\partial x}, \frac{\partial \overline{C}}{\partial y}, \overline{C}$`)`;
4. $(\Sigma_x, \Sigma_y) =$ `Lift_Frobenius_x_y` $(\overline{\alpha}, \overline{\beta}, C, N)$;
5. `For` $j = 1$ `To` $a-1$ `Do`
   5.1. `For` $i = 0$ `To` $b-2$ `Do`
      5.1.1. $V_{i,j} =$ `Red_MW_Cohom_Cab(`$\Sigma_x^i \Sigma_y^j, \Sigma_x, C, B$`)`;
      5.1.2. $I = (b-1)(j-1) + i$;
      5.1.3. `For` $k = 0$ `To` $2g-1$ `Do` $M[I][k] = V_{i,j}[k]$;
6. $M_{\mathcal{F}} \equiv \Sigma^{n-1}(M) \cdots \Sigma(M) M \mod 2^B$;
7. $\chi(t) =$ `Characteristic_Pol` $(M_{\mathcal{F}}) \mod 2^B$;
8. `For` $i = 0$ `To` g `Do`
   8.1. `If Coeff(`$\chi, 2g-i$`)` $> \binom{2g}{i} q^{i/2}$ `Then`
      `Coeff(`$\chi, 2g-i$`)` $-= 2^B$;
   8.2. `Coeff(`χ, i`)` $= q^{g-i}$ `Coeff(`$\chi, 2g-i$`)`;
9. `Return` $Z(\widetilde{C}/\mathbb{F}_{p^n}; t) = \dfrac{t^{2g}\chi(1/t)}{(1-t)(1-qt)}$.

In step 4 we compute $\Sigma(x) \bmod p^N$ and $\Sigma(y) \bmod p^N$ using the function LIFT_FROBENIUS_X_Y given in Algorithm 5.3.2. In step 3, this function takes lifts $\delta_x, \delta_y \in A := \mathbb{Z}_q[x,y]/(C)$ with $\delta_x \equiv \overline{\alpha}^p \bmod p$ and $\delta_y \equiv \overline{\beta}^p \bmod p$. The remainder of Algorithm 5.3.2 is a basic Newton iteration to determine the zero Z_s of the polynomial $G(Z) = C^{\Sigma}(x^p + \delta_x Z, y^p + \delta_y Z) \in A[Z]$ with $Z_s \equiv 0 \bmod p$.

ALGORITHM 5.3.2 (LIFT_FROBENIUS_X_Y)

INPUT: Polynomials $\overline{\alpha}, \overline{\beta} \in \mathbb{F}_{p^n}[x,y]$, C_{ab} curve C over $\mathbb{Z}_q$ with $\overline{\alpha}\frac{\partial \overline{C}}{\partial x} + \overline{\beta}\frac{\partial \overline{C}}{\partial y} \equiv 1 \mod p$, precision N.

OUTPUT: Polynomials $\Sigma_x, \Sigma_y \in \mathbb{Z}_q[x,y]/(C)$ with $\Sigma_x \equiv \Sigma(x) \mod p^N$ and $\Sigma_y \equiv \Sigma(y) \mod p^N$.

1. $S = \lceil \log_2 N \rceil + 1$; $T = N$;
2. For $i = S$ Down To 1 Do $P[i] = T$; $T = \lceil T/2 \rceil$;
3. $\delta_x \equiv \overline{\alpha}^p \mod p$; $\delta_y \equiv \overline{\beta}^p \mod p$; $Z_s = 0$;
4. $G(Z) \equiv C^{\Sigma}(x^p + \delta_x Z, y^p + \delta_y Z)$; $H(Z) \equiv \frac{\partial G}{\partial Z}(Z)$;
5. For $i = 2$ To B Do
   - 5.1. $D_i \equiv G(Z_s) \mod p^{P[i]}$;
   - 5.2. $N_i \equiv H(Z_s) \mod p^{P[i-1]}$;
   - 5.3. $I_i =$ Invert_Series_Cab $(N_i, C, P[i-1])$;
   - 5.4. $\Delta_i \equiv (D_i / p^{P[i-1]}) \cdot I_i \mod p^{P[i-1]}$;
   - 5.5. $Z_s \equiv Z_s - p^{P[i-1]}\Delta_i \mod p^{P[i]}$;
6. Return $\Sigma_x \equiv x^p + \delta_x Z_s \mod p^N$, $\Sigma_y \equiv y^p + \delta_y Z_s \mod p^N$.

ALGORITHM 5.3.3 (RED_MW_COHOM_CAB)

INPUT: `Polynomials` $D, \Sigma_x \in A := \mathbb{Z}_q[x,y]/(C)$, C_{ab} `curve` C `given by equation` $y^a + \sum_{i=1}^{a-1} f_i(x)y^i + f_0(x) = 0$, `precision` B.

OUTPUT: `Coefficient vector` V `of polynomial` $R \in \mathbb{Q}_q[x,y]/(C)$ `with` $\deg_x R(x,y) < b-1$ `and` $R(x,y)\,dx \sim D(x,y)d\Sigma_x \mod 2^B$.

1. $N - \lfloor \log_p(abp(7N + 2(a+b) + 1)) \rfloor \geq B + \epsilon$;
2. $E(x,y) \equiv \int D(x,y) \frac{\partial \Sigma_x}{\partial y}\, dy \mod p^N$;
3. $F(x,y) \equiv D(x,y)\frac{\partial \Sigma_x}{\partial x} - \frac{\partial E}{\partial x}(x,y) \mod p^N$;
4. `While` $(\deg_x F(x,y) > b-2)$ `Do`
    - 4.1. $(i,j) =$ `Index_Max_Valuation` $(F(x,y))$;
    - 4.2. $l = i - b + 1$;
    - 4.3. $\Delta_x \equiv x\sum_{k=1}^{a-1} \frac{j}{k+j} f_k'(x)y^k + xf_0'(x) \mod p^N$;
    - 4.4. $\Delta_y \equiv -l\frac{a}{a+j}y^a - l\sum_{k=1}^{a-1} \frac{k}{k+j} f_k(x)y^k \mod p^N$;
    - 4.5. $\Delta \equiv y^j x^{l-1}(\Delta_x + \Delta_y) \mod (p^N, C)$;
    - 4.6. $\gamma =$ `Coeff` (Δ, i, j);
    - 4.7. $\nu =$ `Coeff` $(F(x,y), i, j)$;
    - 4.8. $F(x,y) \equiv F(x,y) - \nu\gamma^{-1}\Delta \mod p^N$;
5. `For` $i = 0$ `To` $b-2$ `Do`
    - 5.1. `For` $j = 1$ `To` $a-1$ `Do`
        - 5.1.1. $V[(b-1)(j-1)+i] \equiv$ `Coeff` $(F(x,y), i, j)$;
6. `Return` $V \mod p^B$;

The function INVERT_SERIES_CAB in step 5.3 takes as input an element $J \in A^\dagger$ with $J \equiv 1 \bmod p$ and computes a series $I \in A^\dagger$ such that $IJ \equiv 1 \bmod p^N$. Since this is an easy Newton iteration on the equation $H(Z) = JZ - 1 = 0$, we have omitted the pseudo-code.

Given $\Sigma_x \equiv \Sigma(x) \bmod p^N$ and $\Sigma_y \equiv \Sigma(y) \bmod p^N$, we compute the action of Σ_* on a basis of $H^1(\overline{A}/\mathbb{Q}_q)$ in step 5. To reduce each of the $2g$ differentials $\Sigma_x^i \Sigma_y^j \, d\Sigma_x$ for $i = 0, \ldots, b-2$ and $j = 1, \ldots, a-1$, we call the function RED_MW_COHOM_CAB described in Algorithm 5.3.3. In steps 2 and 3 we compute a polynomial $F(x,y) \in A$ such that $\Sigma_x^i \Sigma_y^j \, d\Sigma_x \sim F(x,y)\, dx \bmod p^N$, where $\sim$ means equivalence modulo exact differentials. Step 4 computes the reduction of $F(x,y)\, dx$ on the basis of $H^1(\overline{A}/\mathbb{Q}_q)$ by subtracting exact differentials until $\deg_x F(x,y) < b-1$. In step 4.1 we call INDEX_MAX_VALUATION which returns the pair (i,j) such that $i > b-2$ and $x^i y^j \, dx$ is the term in $F(x,y)\, dx$ with maximum valuation at P_∞. Steps 4.2 to 4.8 compute a suitable multiple of the exact differential given in (5.15) and reduce $F(x,y)\, dx$. Finally, in step 5 we return the coefficient vector of $F(x,y)\, dx$ which is correct modulo p^B.

The result of step 5 of Algorithm 5.3.1 is an approximation modulo p^B of the matrix M through which Σ_* acts on $H^1(\overline{A}/\mathbb{Q}_q)$. In step 6 we compute its norm $M_{\mathcal{F}}$ as $M_{\mathcal{F}} \equiv \Sigma^{n-1}(M) \cdots \Sigma(M) M \bmod p^B$. In steps 7 and 8 we recover the characteristic polynomial of Frobenius from the first g coefficients of the characteristic polynomial of $M_{\mathcal{F}}$. Finally, in step 9 we return the zeta function of the smooth projective C_{ab} curve $\widetilde{C}$.

5.4 Complexity Analysis

In this section we analyse the time and space requirements of Algorithm 5.3.1 for a smooth projective C_{ab} curve $\widetilde{C}$ over $\mathbb{F}_q$ with $q = p^n$, assuming p is fixed.

Given a precision N, we represent elements of $\mathbb{Z}_q/(p^N \mathbb{Z}_q)$ as polynomials over $\mathbb{Z}_p/(p^N \mathbb{Z}_p)$ modulo a monic sparse irreducible polynomial $f(t) \in \mathbb{Z}_p[t]$. Each element of this ring requires $O(nN)$ space, so we can perform multiplication and inversion in time $O(n^\mu N^\mu)$. The coordinate ring $A_N = (\mathbb{Z}_q/(p^N \mathbb{Z}_q))[x,y]/(C)$ consists of bivariate polynomials $\sum_{j=0}^{a-1} \sum_{i=0}^{L} a_{i,j} x^i y^j$ with $a_{i,j} \in \mathbb{Z}_q/(p^N \mathbb{Z}_q)$. Corollary 5.2.1 implies that for $\Sigma(x) \bmod p^N$ and $\Sigma(y) \bmod p^N$ the maximum degree L is bounded by $pb(7N+2)$, so each of these objects takes $O(abnN^2)$ space. Furthermore, the valuation of the coefficients $a_{i,j}$ grows linearly with i, which implies that the size of $\Sigma(x)^i \Sigma(y)^j \bmod p^N$ also is $O(abnN^2)$. Packing these objects in a large integer, we can therefore perform multiplication and inversion in time $O((abnN^2)^\mu)$.

In step 1 of Algorithm 5.3.1 we determine N satisfying inequality (5.22). This implies that N is $O(gn)$. The function NULLSTELLENSATZ in step 3 consists of solving a system of linear equations over $\mathbb{F}_q$ of dimension $O(ab)$ and can thus be solved in $O((ab)^3 n^\mu)$ time using Gaussian elimination. In step 4 we compute $\Sigma(x) \bmod p^N$ and $\Sigma(y) \bmod p^N$ using the function LIFT_FROBENIUS_X_Y. The complexity of Algorithm 5.3.2 is determined by step 5 which is a Newton iteration on the polynomial $G(Z) = C^\Sigma(x^p + \delta_x Z, y^p + \delta_y Z)$. Since the precision doubles in every iteration, we conclude that the complexity is determined by the last iteration. Using Horner's rule we need $O(\max\{a,b\})$ multiplications of objects of size $O(abnN^2)$, so the total time complexity of Algorithm 5.3.2 is $O(\max\{a,b\}(abnN^2)^\mu)$ and the space complexity is $O(abnN^2)$.

Step 5 of Algorithm 5.3.1 computes an approximation of the matrix M through which Σ_* acts on $H^1(\overline{A}/\mathbb{Q}_q)$ using RED_MW_COHOM_CAB to reduce the $2g$ differential forms $\Sigma_x^i \Sigma_y^j \, d\Sigma_x$ for $i = 0, \ldots, b-2$ and $j = 1, \ldots, a-1$. Computing these differentials takes $O(g)$ multiplications of objects of size $O(abnN^2)$ and thus requires $O(g(abnN^2)^\mu)$ time. The complexity of Algorithm 5.3.3 which is used to reduce each of these differentials, is determined by step 4. For each of the $O(abN)$ terms in $F(x,y)$ we need to perform $O(g)$ multiplications of elements in $\mathbb{Z}_q/(p^N\mathbb{Z}_q)$, so the overall time complexity of Algorithm 5.3.3 is $O(gabN(nN)^\mu)$. Since we need to reduce $O(g)$ differentials, the complexity of step 5 of Algorithm 5.3.1 is

$$O(g(abnN^2)^\mu + g^2abN(nN)^\mu)\,.$$

In step 6 we need to determine the norm of a $2g \times 2g$ matrix M over $\mathbb{Q}_q$ as $\Sigma^{n-1}(M) \cdots \Sigma(M)M$, using a matrix variant of Algorithm 3.7.1. This algorithm needs $O(\log n)$ multiplications of $2g \times 2g$ matrices at a cost of $O(g^3(nN)^\mu \log n)$ time and $O(g^2 \log n)$ applications of powers of Σ which takes $O(g^2n(nN)^\mu \log n)$ time if we precompute $\Sigma^{2^i}(t)$ for $i = 0, \ldots, \lfloor \log_2 n \rfloor$. The overall time complexity of step 6 thus becomes $O((n+g)g^2(nN)^\mu \log n)$.

Finally, we need to compute the characteristic polynomial of a $2g \times 2g$ matrix over $\mathbb{Q}_q$, which can be done using the classical algorithm based on the Hessenberg form [16, Section 2.2.4]. The complexity of this algorithm is $O(g^3)$ ring operations or $O(g^3(nN)^\mu)$ time.

Finally, note that $O(ab)$ is $O(g)$ since $2g = (a-1)(b-1)$ and that N is $O(gn)$ due to inequality (5.22). This finishes the proof of the following theorem.

THEOREM 5.4.1 *There exists a deterministic algorithm to compute the zeta function of a smooth C_{ab} curve of genus g defined over $\mathbb{F}_{p^n}$ which requires $O(g^3n^3(g^2 + (\log n)^2 \log\log n) \log gn \log\log gn)$ bit-operations and $O(g^3n^3)$ space for p fixed.*

5.5 Implementation and Numerical Results

In this section we present running times of an implementation of Algorithm 5.3.1 in the `C` programming language and give some examples of Jacobians of C_{ab} curves with almost prime group order.

The basic operations on integers modulo 2^N for $N \leq 256$ are implemented in assembly language. Elements of $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ are represented as polynomials over $\mathbb{Z}/(2^N\mathbb{Z})$ modulo a degree n irreducible polynomial, which is either a trinomial or a pentanomial. For multiplication of elements in $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$, polynomials over $\mathbb{Z}_q/(2^N\mathbb{Z}_q)$ and Laurent series over $\mathbb{Z}_q/(2^N\mathbb{Z}_q)[x]$ we used a combination of Karatsuba [60] and Toom [126] multiplication.

Table 5.1 contains running times and memory usages of our algorithm for $C_{3,4}$ and $C_{3,5}$ curves over various finite fields of characteristic 2. These results were obtained on an AMD XP 1700+ processor running Linux Redhat 7.1. Note that the field degrees are chosen such that gn is constant across each row.

Table 5.1: Running time (s) and memory usage (MB) for genus 3 and 4 C_{ab} curves over $\mathbb{F}_{2^n}$

	$C_{3,4}$ curves				$C_{3,5}$ curves			
gn	Lift $\mathcal{F}$	Matrix	Total	Mem	Lift $\mathcal{F}$	Matrix	Total	Mem
120	364	995	1360	51.8	573	1866	2440	62.8
144	727	3345	4073	63.1	1111	3814	4926	111
168	982	3994	4978	129	1543	5549	7094	147
192	1562	5732	7297	163	2645	9445	12093	185
240	3966	11937	15909	307	6038	21035	27078	347
288	6935	23097	30044	526	11157	33618	44786	595

C_{ab} Curve Examples

In this subsection we give some examples of Jacobians of C_{ab} curves with almost prime group order. The correctness of these results is easily proved by multiplying a random divisor with the given group order and verifying that the result is principal, i.e. is the zero element in the Jacobian $J_{\tilde{C}}(\mathbb{F}_q)$. The given curves are non-supersingular, since the coefficient a_g of $\chi(t)$ is odd [44]. Furthermore, all curves withstand the MOV-FR attack [42, 81].

$C_{3,4}$ curve over $\mathbb{F}_{2^{59}}$

Let $\mathbb{F}_{2^{59}}$ be defined as $\mathbb{F}_2[w]/\overline{P}(w)$ with $\overline{P}(w) = w^{59} + w^7 + w^4 + w^2 + 1$ and consider the random $C_{3,4}$ curve $\overline{C}_3$ of genus 3 defined by

$$y^3 + (\sum_{i=0}^{1} f_{2,i}x^i)y^2 + (\sum_{i=0}^{2} f_{1,i}x^i)y + \sum_{i=0}^{4} f_{0,i}x^i = 0\,,$$

where

$$\begin{array}{ll} f_{0,0} = \texttt{705525FAE3700DB} & f_{0,1} = \texttt{26316DA6009FEBC} \\ f_{0,2} = \texttt{76A78C03C5E2ED7} & f_{0,3} = \texttt{1F1085A56194A7D} \\ f_{0,4} = \texttt{3E35CB85DCD4F89} & f_{1,0} = \texttt{3E8B51B8D1E3999} \\ f_{1,1} = \texttt{2DB09FF6EB73572} & f_{1,2} = \texttt{52AFAF833B5373F} \\ f_{2,0} = \texttt{3011A38B4AEFEFF} & f_{2,1} = \texttt{70A9DE3F7BBF6C7} \end{array}$$

The group order of the Jacobian $J_{\tilde{C}_3}$ of $\overline{C}_3$ over $\mathbb{F}_{2^{59}}$ is

$$\#J_{\tilde{C}_3} = 2 \cdot 95780971335018281268409575905502026018832106252788059\,,$$

where the last factor is prime. The coefficients a_1, a_2 and a_3 of the characteristic polynomial of Frobenius $\chi(t) = t^6 + a_1 t^5 + a_2 t^4 + a_3 t^3 + a_4 t^2 + a_5 t + a_6$ are given by

$$\begin{aligned} a_1 &= -1207321380, \\ a_2 &= 1369094291744606282, \\ a_3 &= -1856910044660666788886053489\,. \end{aligned}$$

$C_{3,5}$ curve over $\mathbb{F}_{2^{47}}$

Let $\mathbb{F}_{2^{47}}$ be defined as $\mathbb{F}_2[w]/\overline{P}(w)$ with $\overline{P}(w) = w^{47} + w^5 + 1$ and consider the random $C_{3,5}$ curve $\overline{C}_4$ of genus 4 defined by

$$y^3 + (\sum_{i=0}^{1} f_{2,i} x^i) y^2 + (\sum_{i=0}^{3} f_{1,i} x^i) y + \sum_{i=0}^{5} f_{0,i} x^i = 0\,,$$

where

$f_{0,0} =$ `341D2BD9C6D0`	$f_{0,1} =$ `2EA67315735F`	$f_{0,2} =$ `26011B32E639`
$f_{0,3} =$ `7042D4480690`	$f_{0,4} =$ `3AFFCD251865`	$f_{0,5} =$ `74F966B7C1A5`
$f_{1,0} =$ `552915B9D5BD`	$f_{1,1} =$ `0DAD08369AD4`	$f_{1,2} =$ `674BB4A87953`
$f_{1,3} =$ `232A2568DAF0`	$f_{2,0} =$ `489EC7EDF33C`	$f_{2,1} =$ `1C8AB33409FA`

The group order of the Jacobian $J_{\widetilde{C}_4}$ of $\overline{C}_4$ over $\mathbb{F}_{2^{47}}$ is

$$\#J_{\widetilde{C}_4} = 2 \cdot 19615943183351641514836269640553274047264808138704896754 9\,,$$

where the last factor is prime. The coefficients a_1, a_2, a_3 and a_4 of the characteristic polynomial of Frobenius $\chi(t) = t^8 + a_1 t^7 + a_2 t^6 + a_3 t^5 + a_4 t^4 + a_5 t^3 + a_6 t^2 + a_7 t + a_8$ are given by

$$\begin{aligned} a_1 &= 1867333, \\ a_2 &= 30460424199008, \\ a_3 &= 110750802726788 2794005, \\ a_4 &= 9346402437739386469819573567\,. \end{aligned}$$

5.6 Conclusion

In this chapter we described a new algorithm to compute the zeta function of a C_{ab} curve over a finite field of small characteristic. For a C_{ab} curve of genus g over $\mathbb{F}_{p^n}$, the algorithm requires $O(g^3 n^3 (g^2 + (\log n)^2 \log\log n) \log gn \log\log gn)$ bit-operations and $O(g^3 n^3)$ space for p fixed. This algorithm currently is the only practical algorithm to compute the zeta function of an arbitrary C_{ab} curve.

The key idea is a new method to lift the Frobenius endomorphism to the dagger ring of a non-singular affine curve. Furthermore, the technique used to prove a tight bound on the convergence rate of the resulting power series remains valid for any non-singular affine curve. Only the explicit description of a basis for the first Monsky-Washnitzer cohomology group and the reduction formulae to express any differential on this basis are limited to C_{ab} curves.

A first implementation in the `C` programming language shows that cryptographic sizes are now feasible for any genus g. However, since the algorithm is very general, it runs about $20g$ times slower than the algorithm for hyperelliptic curves. For example, the order of a 168-bit Jacobian of a $C_{3,4}$ curve can be computed in 1.4 hours.

Chapter 6

Conclusions and Open Problems

In this thesis we have presented several new p-adic algorithms to compute the zeta function of elliptic, hyperelliptic and C_{ab} curves over finite fields. The time and space complexity of these algorithms is polynomial in both the genus of the curve and the extension degree of the finite field. An immediate consequence is that the group order of the Jacobian $J_C(\mathbb{F}_q)$ can now be computed efficiently as long as the characteristic of $\mathbb{F}_q$ is small.

Section 6.1 summarizes the main conclusions and original contributions of this thesis. Section 6.2 presents some topics for future research.

6.1 Conclusions

The zeta function of a curve C contains important arithmetic and geometric information about the curve and its Jacobian J_C. Motivated by applications such as cryptography and coding theory, efficient algorithms to compute the zeta function have become increasingly important.

In 1985, Schoof [111] described the first polynomial time algorithm to count the number of points on an elliptic curve over a finite field $\mathbb{F}_q$. The key idea is to compute the characteristic polynomial $\chi(t)$ of the q-th power Frobenius endomorphism $\overline{F}$ modulo various small primes l and to recover the zeta function via the Chinese Remainder Theorem. Schoof's approach is generally referred to as being l-adic, since it computes $\chi(t) \bmod l$ from the action of $\overline{F}$ on the l-torsion subgroups. Although the time complexity of Schoof's algorithm is $O((\log q)^{5+\varepsilon})$, it is too slow for cryptographic applications. Several improvements mainly due to Elkies [30] and Atkin resulted in the SEA algorithm [112] which runs in $O((\log q)^{4+\varepsilon})$ time. The SEA algorithm is fast enough to count the number of points on elliptic curves used in cryptography. Furthermore, since its time complexity depends on $\log q$, the algorithm is efficient in both small and large characteristic.

Schoof's algorithm was extended to abelian varieties by Pila [99] and improved by Adleman, Huang and Ierardi [3, 59] for Jacobians of curves. The time complexity of these algorithms is polynomial in the bit-size of the finite field, but exponential in the genus of the curve. Furthermore, they are extremely cumbersome to implement and too slow to be practical, except possibly for genus 2 hyperelliptic curves [50].

At the end of 1999, Satoh [105] proposed a completely new strategy based on p-adic methods which compute a p-adic lift of the Frobenius endomorphism to characteristic zero. Instead of working modulo various small primes l, these methods compute the zeta function modulo a high power of p. The idea of computing with a p-adic lift of Frobenius is not really new and can be traced back to Dwork's proof [27] of the rationality of the zeta function of a variety over a finite field. The major drawback of these algorithms is that they are only efficient if the characteristic of the finite field is small.

In this thesis, we have designed several new p-adic algorithms to compute the zeta function of all types of curves used in cryptography, i.e. elliptic, hyperelliptic and C_{ab} curves, each type of curve representing one major chapter.

Elliptic Curves

In Chapter 3 we have presented an overview of all existing p-adic point counting algorithms for elliptic curves over finite fields of small characteristic and designed a new memory efficient version of Satoh's algorithm.

At the end of 1999, Satoh [106] described the first efficient algorithm to lift an elliptic curve in a canonical way to an unramified p-adic ring using a multivariate Newton iteration and a separate algorithm to compute the trace of Frobenius by lifting the p-torsion. As a consequence, he obtained a point counting algorithm for $p \geq 5$ which runs in time $O(n^3 \log n \log\log n)$ and requires $O(n^3)$ space. The algorithm was extended to $p = 2$ and $p = 3$ by Fouquet, Gaudry and Harley [41] and independently to $p = 2$ by Skjernaa [118].

The main result in Chapter 3 is an optimized version of Satoh's algorithm which only requires $O(n^2)$ space and runs faster by a constant. The key idea is to replace the multivariate Newton iteration by an iterated univariate Newton lifting. A priori, it is not clear that this iterated univariate Newton lifting converges at all and in general this will not be the case. Using the Kronecker relation satisfied by the p-th modular polynomial, we we able to prove convergence. This strategy results in the following theorem which has been published in [133].

THEOREM 1 *Let $\overline{E}$ be an elliptic curve over a finite field $\mathbb{F}_q$ with $q = p^n$ and $j(\overline{E}) \notin \mathbb{F}_{p^2}$. Then there exists a deterministic algorithm to compute the number of points $\#\overline{E}(\mathbb{F}_q)$ which requires $O(n^3 \log n \log\log n)$ bit-operations and $O(n^2)$ space for fixed p.*

The remainder of Chapter 3 analysed the more recent developments in p-adic point counting for elliptic curves.

Mestre [85] designed a very elegant algorithm for $p = 2$ based on the Arithmetic Geometric Mean. This AGM algorithm also runs in $O(n^3 \log n \log\log n)$ time using $O(n^2)$ space, but is faster by a constant and easier to implement.

Satoh, Skjernaa and Taguchi [109] proposed an improved algorithm which requires $O(n^{2.5} \log n \log\log n)$ bit-operations and $O(n^2)$ space, not including precomputation which only depends on the finite field and takes $O(n^3 \log n \log\log n)$ bit-operations. Their algorithm is based on two key ideas: the first idea is to use a representation of the unramified p-adic ring which allows fast Frobenius substitutions. This can be used to speed up the algorithm of Theorem 1 by also taking into account the partial derivatives. The second idea involves computing the norm $\mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha)$ of an element $\alpha \in \mathbb{Z}_q$ which is sufficiently close to unity as $\exp(\mathrm{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\log \alpha))$. Gaudry [48] combined the Satoh-Skjernaa-Taguchi algorithm with a simplified modular polynomial related to the AGM iteration and obtained a faster algorithm for $p = 2$.

Kim, Park, Cheon, Park, Kim and Hahn [63] proposed to use finite fields with a Gaussian Normal Basis of small type. This allows for efficient computation of arbitrary iterates of the Frobenius substitution and can be used to compute the norm in $O(n^2 \log^2 n \log\log n)$ time. As a result, they obtained a point counting algorithm without precomputation which runs in time $O(n^{2.5} \log n \log\log n)$.

Lercier and Lubicz [75] described an algorithm to solve generalized Artin-Schreier equations over $\mathbb{Q}_q$. The main idea is to use the identity $\Sigma^n(x) = x$ with $x \in \mathbb{Q}_q$ and Σ the Frobenius substitution. For fields with Gaussian Normal Basis of small type, this leads to a point counting algorithm without precomputation which requires $O(n^2 (\log n)^2 \log\log n)$ bit-operations and $O(n^2)$ space.

Finally, in an e-mail [55] to the NMBRTHRY list, Harley announced a point counting algorithm which runs in $O(n^2 (\log n)^2 \log\log n)$ time and uses $O(n^2)$ space. The heart of the algorithm is a recursive divide-and-conquer strategy to solve generalized Artin-Schreier equations and an asymptotically fast norm computation based on resultants.

The final result of all these improvements is that it is now possible to count the number of points on an elliptic curve over a finite field $\mathbb{F}_q$ with $q \simeq 2^{168}$ in less than 0.1 seconds.

Hyperelliptic Curves

In Chapter 4 we have presented algorithms to compute the zeta function of hyperelliptic curves over finite fields using Monsky-Washnitzer cohomology, i.e. the de Rham cohomology of a ring of overconvergent p-adic power series. These algorithms proceed in two phases: firstly, the Frobenius endomorphism is lifted to the dagger ring associated to a suitable lift of the curve and secondly, a basis of the first Monsky-Washnitzer cohomology group is constructed together with an algorithm to express any differential on this basis. The zeta function is then recovered from the characteristic polynomial of the Frobenius endomorphism acting on the first cohomology group.

Kedlaya [61] described such an algorithm for hyperelliptic curves over finite fields $\mathbb{F}_{p^n}$ of small, odd characteristic. For fixed p, the time complexity for a genus g hyperelliptic curve over $\mathbb{F}_{p^n}$ is $(g^{4+\varepsilon}n^{3+\varepsilon})$ bit-operations and the space complexity is $O(g^3n^3)$.

The characteristic two case is more complicated since the algebraic de Rham cohomology of an arbitrary lift of the curve minus its ramification points, is in general not isomorphic to the Monsky-Washnitzer cohomology. We showed that this problem can be solved by constructing a model with all ramification points on the x-axis. The main result of Chapter 4 is summarized in the next theorem, which is joint work with J. Denef and has been published in [23, 24, 131].

THEOREM 2 *There exists a deterministic algorithm to compute the zeta function of a smooth hyperelliptic curve of genus g defined over a finite field $\mathbb{F}_q$ with $q = 2^n$ elements using*

- $O(g^3n^3(g + (\log n)^2 \log\log n) \log gn \log\log gn)$ *bit-operations and* $O(g^3n^3)$ *space in the average case,*
- $O(g^3n^3(g^2 + (\log n)^2 \log\log n) \log gn \log\log gn)$ *bit-operations and* $O(g^4n^3)$ *space in the worst case.*

An implementation of this algorithm in the `C` programming language shows that cryptographic sizes are now feasible for any genus g. For example, the order of a 168-bit Jacobian of a hyperelliptic curve of genus 2, 3 or 4 can be computed in less than 80 seconds. Furthermore, since the time complexity is also polynomial in the genus, we were able to compute the zeta function of a random genus 350 hyperelliptic curve over $\mathbb{F}_2$ in 3.5 days.

C_{ab} Curves

In Chapter 5, we have presented a new algorithm to compute the zeta function of a C_{ab} curve over any finite field of small characteristic. This algorithm is also based on Monsky-Washnitzer cohomology, but is much more general than for hyperelliptic curves. The main idea is a new method to lift the Frobenius endomorphism to the dagger ring of an arbitrary non-singular affine curve. Only the explicit description of a basis for the first Monsky-Washnitzer cohomology group and the reduction formulae to express any differential on this basis are limited to C_{ab} curves. The main result of Chapter 5 is the following theorem, which is joint work with J. Denef and will be submitted for publication [132].

THEOREM 3 *There exists a deterministic algorithm to compute the zeta function of a smooth C_{ab} curve of genus g defined over a finite field $\mathbb{F}_{p^n}$ which requires $O(g^3n^3(g^2+(\log n)^2 \log\log n) \log gn \log\log gn)$ bit-operations and $O(g^3n^3)$ space for p fixed.*

Due to its generality, the above algorithm runs about $20g$ times slower than the algorithm for hyperelliptic curves in the average case. Nevertheless, an implementation in the `C` programming language shows that cryptographic sizes are still feasible for any genus g. For example, the order of a 168-bit Jacobian of a $C_{3,4}$ curve can be computed in 1.4 hours. Furthermore, it appears likely that this algorithm can be extended to arbitrary curves.

6.2 Open Problems

This section presents several open problems that may be of interest for further research on the algorithmic theory of zeta functions.

- Harley's algorithm in Chapter 3 requires $O(n^2(\log n)^2 \log\log n)$ time and $O(n^2)$ space to compute the number of points on an elliptic curve over a finite field $\mathbb{F}_{p^n}$, for p fixed. Is it possible to further lower the time complexity? Clearly, as long as the algorithm proceeds in two stages, i.e. compute an element $\alpha \in \mathbb{Z}_q$ and recover the trace of Frobenius as the norm of α, this is impossible. Although it seems unlikely, one could try to combine both stages and compute the norm directly, for instance by lifting the lower part of the minimal polynomial of α. A related but quite different question is the following: is it possible to compute the number of points on $O(n)$ elliptic curves simultaneously in time $O(n^2(\log n)^2 \log\log n)$? Note that this question is relevant for practical applications in cryptography, since many curves need to be tested before a suitable one is found.

- Chapters 4 and 5 used Monsky-Washnitzer cohomology to recover the zeta function from the characteristic polynomial of a Frobenius operator acting on the first cohomology group. For ordinary curves, exactly g roots of the characteristic polynomial will be units in $\mathbb{Z}_q^{\mathrm{ur}}$ and the others follow easily from the Weil conjectures. Given an ordinary curve, is it possible to work with a g-dimensional vector space and only recover the unit root part?

- The first part of the algorithm described in Chapter 5 is valid for any non-singular affine curve. Is it possible to extend the second part of the algorithm, i.e. constructing a basis for the first Monsky-Washnitzer cohomology group together with an algorithm to express any differential on this basis, to all curves? One possibility would be to proceed in the following way. Without loss of generality, we can assume that the curve is given as a plane curve with all singularities at infinity. If these points give rise to m places on the non-singular model, the dimension of the first Monsky-Washnitzer cohomology group will be $2g + m - 1$. Reducing a given differential then boils down to finding an exact differential which has the correct pole order in one of these places and a lower pole order than the given differential in all other places. An effective version of the Riemann-Roch theorem will be indispensable in constructing such differentials.

- Let $\overline{C}$ be a curve of genus g over a finite field $\mathbb{F}_q$ with $q = p^n$ and let $\chi(t)$ be the characteristic polynomial of Frobenius. All p-adic algorithms compute $\chi(t) \bmod p^N$ for some precision

$$N \geq \left\lceil \log_p \left(2 \binom{2g}{g} q^{g/2} \right) \right\rceil .$$

  The Weil conjectures imply that this determines $\chi(t)$ uniquely, but leave the following question unanswered. Given $\chi(t) \bmod p^M$ for some precision

$M \leq N$, how many isogeny classes of abelian varieties over $\mathbb{F}_q$ have a characteristic polynomial of Frobenius which reduces to $\chi(t) \bmod p^M$? Preferably the answer to this question would give the number of such isogeny classes as a function of p, n, M and g. Note that if there would exist an $M < N$ for which the number of isogeny classes is $O(1)$, then this could possibly lead to a faster algorithm to compute $\chi(t)$. For elliptic curves there does not exists such an M as shown by Waterhouse [135]; nonetheless, for large genus it is likely to be true.

- Currently, all p-adic algorithms are only efficient for curves over finite fields of small characteristic. Furthermore, it seems unlikely that the time complexity of similar algorithms will ever become polynomial in $\log p$. Is it possible to design an algorithm to compute the zeta function of an arbitrary curve in time polynomial both in the genus of the curve and the bit-size of the finite field? Currently, the l-adic approach, i.e. using torsion points and division polynomials, leads to algorithms that are exponential in the genus.

Bibliography

[1] L. Adleman and J. DeMarrais. A subexponential algorithm for discrete logarithms over all finite fields. *Math. Comp.*, 61:1–15, 1993.

[2] L. M. Adleman, J. DeMarrais, and M.-D. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In L. M. Adleman and M.-D. Huang, editors, *Algorithmic Number Theory Symposium, Proceedings ANTS-I*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 28–40. Springer, Berlin, 1994.

[3] L. M. Adleman and M.-D. Huang. Counting rational points on curves and abelian varieties over finite fields. In H. Cohen, editor, *Algorithmic Number Theory Symposium, Proceedings ANTS-II*, volume 1122 of *Lecture Notes in Comput. Sci.*, pages 1–16. Springer, Berlin, 1996.

[4] L. M. Adleman and M.-D. Huang. Counting points on curves and abelian varieties over finite fields. *J. Symbolic Comput.*, 32(3):171–189, 2001.

[5] ANSI X9.62. *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, 1999.

[6] ANSI X9.63. *Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography*, 2001.

[7] S. Arita. Algorithms for computations in Jacobian group of C_{ab} curve and their application to discrete-log-based public key cryptosystem. In *Proceedings of Conference of the Mathematics of Public Key Cryptography, Fields Institute Toronto*, 1999.

[8] M. Bauer. A subexponential algorithm for solving the discrete logarithm problem in the Jacobian of high genus hyperelliptic curves over arbitrary fields. Preprint, 1999.

[9] I. F. Blake, G. Seroussi, and N. P. Smart. *Elliptic curves in cryptography II.* Cambridge University Press, Cambridge, 2003. In preparation.

[10] D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices Amer. Math. Soc.*, 46(2):203–213, 1999.

[11] D. Boneh and R. Venkatesan. Breaking RSA may not be equivalent to factoring (extended abstract). In K. Nyberg, editor, *Advances in Cryptology, Proceedings Eurocrypt 1998*, volume 1403 of *Lecture Notes in Comput. Sci.*, pages 59–71. Springer, Berlin, 1998.

[12] S. Bosch. A rigid analytic version of M. Artin's theorem on analytic equations. *Math. Ann.*, 255(3):395–404, 1981.

[13] B. Buchberger. A theoretical basis for the reduction of polynomials to canonical forms. *ACM SIGSAM Bull.*, 10(3):19–29, 1976.

[14] D. G. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Math. Comp.*, 48(177):95–101, 1987.

[15] R. Carls. Generalized AGM sequences and approximation of canonical lifts. Available at `http://www.math.leidenuniv.nl/~carls/`, September 2003.

[16] H. Cohen. *A course in computational algebraic number theory.* Springer-Verlag, Berlin, 1993.

[17] H. Cohen and G. Frey, editors. *Handbook of elliptic and hyperelliptic curve cryptography.* Springer Verlag, 2003. In preparation.

[18] D. Coppersmith. Fast evaluation of logarithms in fields of characteristic two. *IEEE Transactions on Information Theory*, 30:587–594, 1984.

[19] G. Cornell and J. H. Silverman, editors. *Arithmetic geometry.* Springer-Verlag, New York, 1986. Papers from the conference held at the University of Connecticut, Storrs, Connecticut, July 30–August 10, 1984.

[20] J. Daemen and V. Rijmen. *The design of Rijndael: AES — the Advanced Encryption Standard.* Springer-Verlag, 2002.

[21] V. I. Danilov and V. V. Shokurov. *Algebraic curves, algebraic manifolds and schemes.* Springer-Verlag, Berlin, 1998.

[22] P. Deligne. La conjecture de Weil. I. *Inst. Hautes Études Sci. Publ. Math.*, 43:273–307, 1974.

[23] J. Denef and F. Vercauteren. An extension of Kedlaya's algorithm to Artin-Schreier curves in characteristic 2. In C. Fieker and D. R. Kohel, editors, *Algorithmic Number Theory Symposium, Proceedings ANTS-V*, volume 2369 of *Lecture Notes in Comp. Sci.*, pages 308–323. Springer, Berlin, 2002.

[24] J. Denef and F. Vercauteren. An extension of Kedlaya's algorithm to hyperelliptic curves in characteristic 2. Accepted for publication in J. Cryptology, November 2002.

[25] M. Deuring. Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. *Abh. Math. Sem. Hansischen Univ.*, 14:197–272, 1941.

[26] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, IT-22(6):644–654, 1976.

[27] B. Dwork. On the rationality of the zeta function of an algebraic variety. *Amer. J. Math.*, 82:631–648, 1960.

[28] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology, Proceedings Crypto 1984*, volume 196 of *Lecture Notes in Comput. Sci.*, pages 10–18. Springer, Berlin, 1985.

[29] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, 31(4):469–472, 1985.

[30] N. D. Elkies. Elliptic and modular curves over finite fields and related computational issues. In D. A. Buell and J. T. Teitelbaum, editors, *Computational perspectives on number theory*, volume 7 of *AMS/IP Stud. Adv. Math.*, pages 21–76. Amer. Math. Soc., Providence, RI, 1998.

[31] R. Elkik. Solutions d'équations à coefficients dans un anneau hensélien. *Ann. Sci. École Norm. Sup. (4)*, 6:553–603 (1974), 1973.

[32] A. Enge. *Hyperelliptic Cryptosystems*. PhD thesis, Universität Augsburg, 2000.

[33] A. Enge. How to distinguish hyperelliptic curves in even characteristic. In *Public-key cryptography and computational number theory (Warsaw, 2000)*, pages 49–58. de Gruyter, Berlin, 2001.

[34] A. Enge. Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time. *Math. Comp.*, 71(238):729–742 (electronic), 2002.

[35] FIPS 46. *Data Encryption Standard*. Federal Information Processing Standard (FIPS), National Bureau of Standards, U.S. Department of Commerce, Washington D.C., 1977.

[36] FIPS 46-2. *Data Encryption Standard*. Federal Information Processing Standard (FIPS), National Bureau of Standards, U.S. Department of Commerce, Washington D.C., 1993.

[37] FIPS 186. *Digital Signature Standard*. Federal Information Processing Standard (FIPS), National Institute of Standards and Technology, U.S. Department of Commerce, Washington D.C., 1994.

[38] FIPS 186-2. *Digital Signature Standard.* Federal Information Processing Standard (FIPS), National Institute of Standards and Technology, U.S. Department of Commerce, Washington D.C., 2000.

[39] FIPS 197. *Advanced Encryption Standard.* Federal Information Processing Standard (FIPS), National Institute of Standards and Technology, U.S. Department of Commerce, Washington D.C., 2001.

[40] M. Fouquet. *Anneau d'endomorphismes et cardinalité des courbes elliptiques: aspects algorithmiques.* PhD thesis, École Polytechnique, 2001.

[41] M. Fouquet, P. Gaudry, and R. Harley. An extension of Satoh's algorithm and its implementation. *J. Ramanujan Math. Soc.*, 15(4):281–318, 2000.

[42] G. Frey and H.-G. Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 62(206):865–874, 1994.

[43] W. Fulton. *Algebraic curves. An introduction to algebraic geometry.* W. A. Benjamin, Inc., New York-Amsterdam, 1969.

[44] S. D. Galbraith. Supersingular curves in cryptography. In C. Boyd, editor, *Advances in Cryptology, Proceedings Asiacrypt 2001*, volume 2248 of *Lecture Notes in Comput. Sci.*, pages 495–513. Springer, Berlin, 2001.

[45] S. D. Galbraith. Weil descent on Jacobians. *Discrete Applied Mathematics*, 128:165–180, 2003.

[46] S. D. Galbraith, S. M. Paulus, and N. P. Smart. Arithmetic on superelliptic curves. *Math. Comp.*, 71(237):393–405 (electronic), 2002.

[47] P. Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. In B. Preneel, editor, *Advances in Cryptology, Proceedings Eurocrypt 2000*, volume 1807 of *Lecture Notes in Comput. Sci.*, pages 19–34. Springer, Berlin, 2000.

[48] P. Gaudry. A comparison and a combination of SST and AGM algorithms for counting points of elliptic curves in characteristic 2. In Y. Zheng, editor, *Advances in Cryptology, Proceedings Asiacrypt 2002*, volume 2501 of *Lecture Notes in Comput. Sci.*, pages 311–327. Springer, Berlin, 2002.

[49] P. Gaudry. and N. Gürel An extension of Kedlaya's point-counting algorithm to superelliptic curves. In C. Boyd, editor, *Advances in Cryptology, Proceedings Asiacrypt 2001*, volume 2248 of *Lecture Notes in Comput. Sci.*, pages 480–494. Springer, Berlin, 2001.

[50] P. Gaudry and R. Harley Counting points on hyperelliptic curves over finite fields. In W. Bosma, editor, *Algorithmic Number Theory Symposium, Proceedings ANTS-IV*, volume 1838 of *Lecture Notes in Comput. Sci.*, pages 313–332. Springer, Berlin, 2000.

[51] P. Gaudry, F. Hess, and N. P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *J. Cryptology*, 15(1):19–46, 2002.

[52] D. Gordon. Discrete logarithms in $GF(p)$ using the number field sieve. *SIAM Journal on Discrete Mathematics*, 6:124–138, 1993.

[53] IEEE P1363. *Standard Specifications for Public-Key Cryptography*, 2000. Available at `http://grouper.ieee.org/groups/1363/index.html`.

[54] R. Harasawa and J. Suzuki. Fast Jacobian group arithmetic on C_{ab} curves. In W. Bosma, editor, *Algorithmic Number Theory Symposium, Proceedings ANTS-IV*, volume 1838 of *Lecture Notes in Comput. Sci.*, pages 359–376. Springer, Berlin, 2000.

[55] R. Harley. Asymptotically optimal p-adic point-counting. e-mail to NMBRTHRY list, December 2002.

[56] J. Harris. *Algebraic geometry*, volume 133 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1992. A first course.

[57] R. Hartshorne. *Algebraic geometry.* Springer-Verlag, New York, 1977. Graduate Texts in Mathematics, No. 52.

[58] F. Hess. Computing relations in divisor class groups of algebraic curves over finite fields. Preprint, 2003.

[59] M.-D. Huang and D. Ierardi. Counting points on curves over finite fields. *J. Symbolic Comput.*, 25(1):1–21, 1998.

[60] A. A. Karatsuba and Y. Ofman. Multiplication of multidigit numbers on automata. *Soviet Physics Doklady*, 7:595–596, 1963.

[61] K. S. Kedlaya. Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology. *J. Ramanujan Math. Soc.*, 16(4):323–338, 2001.

[62] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX:5–83 Janvier, 161–191 Février, 1883.

[63] H. Y. Kim, J. Y. Park, J. H. Cheon, J. H. Park, J. H. Kim, and S. G. Hahn. Fast elliptic curve point counting using Gaussian Normal Basis. In C. Fieker and D. R. Kohel, editors, *Algorithmic Number Theory Symposium, Proceedings ANTS-V*, volume 2369 of *Lecture Notes in Comp. Sci.*, pages 292–307. Springer, Berlin, 2002.

[64] N. Koblitz. *p-adic numbers, p-adic analysis, and zeta-functions*, volume 58 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, second edition, 1984.

[65] N. Koblitz. Elliptic curve cryptosystems. *Math. Comp.*, 48(177):203–209, 1987.

[66] N. Koblitz. Hyperelliptic cryptosystems. *J. Cryptology*, 1(3):139–150, 1989.

[67] D. Kohel. The AGM-$X_0(N)$ Heegner point lifting algorithm and elliptic curve point counting. Available at `http://magma.maths.usyd.edu.au/users/kohel/research/index.html`, May 2003.

[68] T. Lange. Efficient arithmetic on genus 2 hyperelliptic curves over finite fields via explicit formulae. Available at `http://www.ruhr-uni-bochum.de/itsc/tanja/preprints.html`, 2002.

[69] A. Lauder and Wan. D. Counting points on varieties over finite fields of small characteristic. In J.P. Buhler and P. Stevenhagen, editors, *Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography*, Mathematical Sciences Research Institute Publications, 2002. To appear.

[70] A. G. B. Lauder and D. Wan. Computing zeta functions of Artin-Schreier curves over finite fields. *LMS J. Comput. Math.*, 5:34–55 (electronic), 2002.

[71] A. G. B. Lauder and D. Wan. Computing zeta functions of Artin-Schreier curves over finite fields II. To appear in a special issue of the Journal of Complexity, 2002.

[72] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

[73] H. W. Lenstra, Jr. Factoring integers with elliptic curves. *Ann. of Math. (2)*, 126(3):649–673, 1987.

[74] R. Lercier. *Algorithmique des courbes elliptiques dans les corps finis*. PhD thesis, Laboratoire d'Informatique de l'École polytechnique (LIX), 1997.

[75] R. Lercier and D. Lubicz. Counting points on elliptic curves over finite fields of small characteristic in quasi quadratic time. In E. Biham, editor, *Advances in Cryptology, Proceedings Eurocrypt 2003*, volume 2656 of *Lecture Notes in Comput. Sci.*, pages 360–373. Springer, Berlin, 2003.

[76] R. Lercier and D. Lubicz. A quasi quadratic time algorithm for hyperelliptic curve point counting. Preprint October 2003.

[77] J. Lubin, J.-P. Serre, and J. Tate. Elliptic curves and formal groups. Lecture notes prepared in connection with the seminars held at the Summer Institute on Algebraic Geometry, Whitney Estate, Woods Hole, Massachusetts, July 6-July 31, 1964, 1964.

[78] F. Macaulay. On some formulas in elimination. *Proc. London Math. Soc.*, 3:3–27, 1902.

[79] R. Matsumoto. The C_{ab} curve. Available at `http://www.rmatsumoto.org/cab.ps`.

[80] A. Menezes, I. Blake, S. Gao, R. Mullin, S. Vanstone, and T. Yaghoobian. *Applications of Finite Fields.* Kluwer Academic Publishers, 1992.

[81] A. J. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Trans. Inform. Theory*, 39(5):1639–1646, 1993.

[82] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of applied cryptography.* CRC Press Series on Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, 1997.

[83] W. Messing. *The crystals associated to Barsotti-Tate groups: with applications to abelian schemes.* Springer-Verlag, Berlin, 1972. Lecture Notes in Mathematics, Vol. 264.

[84] J.-F. Mestre. Algorithmes pour compter des points en petite caractéristique en genre 1 and 2. Available at `http://www.math.univ-rennes1.fr/crypto/2001-02/mestre.ps`.

[85] J.-F. Mestre. Lettre adressée à Gaudry et Harley, December 2000. Available at `http://www.math.jussieu.fr/~mestre/`.

[86] V. S. Miller. Use of elliptic curves in cryptography. In H. C. Williams, editor, *Advances in Cryptology, Proceedings Crypto 1985*, volume 218 of *Lecture Notes in Comput. Sci.*, pages 417–426. Springer, Berlin, 1986.

[87] S. Miura. Algebraic geometric codes on certain plane curves. *Trans. IEICE*, J75-A(11):1735–1745, 1992. In Japanese.

[88] R. T. Moenck. Fast computation of GCDs. In *Fifth Annual ACM Symposium on Theory of Computing (Austin, Tex., 1973)*, pages 142–151. Assoc. Comput. Mach., New York, 1973.

[89] P. Monsky and G. Washnitzer. Formal cohomology. I. *Ann. of Math. (2)*, 88:181–217, 1968.

[90] P. Monsky. Formal cohomology. II. The cohomology sequence of a pair. *Ann. of Math. (2)*, 88:218–238, 1968.

[91] P. Monsky. Formal cohomology. III. Fixed point theorems. *Ann. of Math. (2)*, 93:315–343, 1971.

[92] P. Monsky. *p-adic analysis and zeta functions.* Kinokuniya Book-Store Co. Ltd., Tokyo, 1970.

[93] V. Müller. *Ein Algorithmus zur Bestimmung der Punktanzahl elliptischer kurven über endlichen körpen der charakteristik größer drei.* PhD thesis, Technischen Fakultät de Universität des Saarlandes, 1995.

[94] V. Müller, A. Stein, and C. Thiel. Computing discrete logarithms in real quadratic congruence function fields of large genus. *Math. Comp.*, 68(226):807–822, 1999.

[95] D. Mumford. On the equations defining abelian varieties. I. *Invent. Math.*, 1:287–354, 1966.

[96] D. Mumford. *Tata lectures on theta. II*, volume 43 of *Progress in Mathematics*. Birkhäuser Boston Inc., Boston, MA, 1984.

[97] V. Nechaev. Complexity of a determinate algorithm for the discrete logarithm problem. *Mathematical Notes*, 55:165–172, 1994.

[98] J. Pelzl, T. Wollinger, and C. Guajardo, J.and Paar. Hyperelliptic curves cryptosystems: closing the performance gap to elliptic curves. Available at `http://eprint.iacr.org/2003/026/`, 2003.

[99] J. Pila. Frobenius maps of abelian varieties and finding roots of unity in finite fields. *Math. Comp.*, 55(192):745–763, 1990.

[100] S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. *IEEE Trans. Information Theory*, IT-24(1):106–110, 1978.

[101] J. M. Pollard. Monte Carlo methods for index computation (mod p). *Math. Comp.*, 32(143):918–924, 1978.

[102] C. Ritzenthaler. Méthode A.G.M. pour les courbes ordinaires de genre 3 non hyperelliptiques sur F_{2^N}. Available at `http://arxiv.org/abs/math.NT/0303072`, 2003.

[103] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM*, 21(2):120–126, 1978.

[104] H.-G. Rück. On the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 68(226):805–806, 1999.

[105] T. Satoh. The canonical lift of an ordinary elliptic curve over a finite field and its point counting. *J. Ramanujan Math. Soc.*, 15(4):247–270, 2000.

[106] T. Satoh. Asymptotically fast algorithm for computing the frobenius substitution and norms over unramified extension of p-adic number fields. Available at `http://www.rimath.saitama-u.ac.jp/lab.en/TkkzSatoh/Preprint.html`, 2001.

[107] T. Satoh. On p-adic point counting algorithms for elliptic curves over finite fields. In C. Fieker and D. R. Kohel, editors, *Algorithmic Number Theory Symposium, Proceedings ANTS-V*, volume 2369 of *Lecture Notes in Comp. Sci.*, pages 43–66. Springer, Berlin, 2002.

[108] T. Satoh and K. Araki. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Comment. Math. Univ. St. Paul.*, 47(1):81–92, 1998.

[109] T. Satoh, B. Skjernaa, and Y. Taguchi. Fast computation of canonical lifts of elliptic curves and its application to point counting. *Finite Fields Appl.*, 9(1):89–101, 2003.

[110] A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing (Arch. Elektron. Rechnen)*, 7:281–292, 1971.

[111] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.*, 44(170):483–494, 1985.

[112] R. Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7(1):219–254, 1995. Les Dix-huitièmes Journées Arithmétiques (Bordeaux, 1993).

[113] I. A. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Math. Comp.*, 67(221):353–356, 1998.

[114] J.-P. Serre. *Local fields*, volume 67 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1979. Translated from French.

[115] D. Shanks. Class number, a theory of factorization, and genera. In *1969 Number Theory Institute (Proc. Sympos. Pure Math., Vol. XX, State Univ. New York, Stony Brook, N.Y., 1969)*, pages 415–440. Amer. Math. Soc., 1971.

[116] V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *Advances in Cryptology, Proceedings Eurocrypt 1997*, volume 1233 of *Lecture Notes in Comput. Sci.*, pages 256–266. Springer, Berlin, 1997.

[117] J. H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1986.

[118] B. Skjernaa. Satoh's algorithm in characteristic 2. *Math. Comp.*, 72(241):477–487 (electronic), 2003.

[119] N. P. Smart. The discrete logarithm problem on elliptic curves of trace one. *J. Cryptology*, 12(3):193–196, 1999.

[120] A. Stein. Sharp upper bounds for arithmetic in hyperelliptic function fields. *J. Ramanujan Math. Soc.*, 16(2):119–203, 2001.

[121] H. Stichtenoth. Die Hasse-Witt-Invariante eines Kongruenzfunktionenkörpers. *Arch. Math. (Basel)*, 33(4):357–360, 1979/80.

[122] H. Stichtenoth. *Algebraic function fields and codes.* Universitext. Springer-Verlag, Berlin, 1993.

[123] H. Stichtenoth and C. P. Xing. On the structure of the divisor class group of a class of curves over finite fields. *Arch. Math. (Basel)*, 65(2):141–150, 1995.

[124] J. Tate. Endomorphisms of abelian varieties over finite fields. *Invent. Math.*, 2:134–144, 1966.

[125] N. Thériault. Index calculus attack for hyperelliptic curves of small genus. Preprint, 2003.

[126] A. L. Toom. The complexity of a scheme of functional elements simulating the multiplication of integers. *Dokl. Akad. Nauk SSSR*, 150:496–498, 1963.

[127] M. van der Put. The cohomology of Monsky and Washnitzer. *Mém. Soc. Math. France (N.S.)*, 23:4, 33–59, 1986. Introductions aux cohomologies p-adiques (Luminy, 1984).

[128] P. C. van Oorschot and M. J. Wiener. Parallel collision search with cryptanalytic applications. *J. Cryptology*, 12(1):1–28, 1999.

[129] J. Vélu. Isogénies entre courbes elliptiques. *C. R. Acad. Sci. Paris Sér. A-B*, 273:A238–A241, 1971.

[130] F. Vercauteren. `#EC(GF(2^1999))`. e-mail to NMBRTHRY list, October 1999.

[131] F. Vercauteren. Computing zeta functions of hyperelliptic curves over finite fields of characteristic 2. In M. Yung, editor, *Advances in Cryptology, Proceedings Crypto 2002*, volume 2442 of *Lecture Notes in Comput. Sci.*, pages 369–384. Springer, Berlin, 2002.

[132] F. Vercauteren and J. Denef. Counting points on C_{ab} curves using Monsky-Washnitzer cohomology. Preprint, September 2003.

[133] F. Vercauteren, B. Preneel, and J. Vandewalle. A memory efficient version of Satoh's algorithm. In B. Pfitzmann, editor, *Advances in Cryptology, Proceedings Eurocrypt 2001*, volume 2045 of *Lecture Notes in Comput. Sci.*, pages 1–13. Springer, Berlin, 2001.

[134] J. von zur Gathen and J. Gerhard. *Modern computer algebra.* Cambridge University Press, New York, 1999.

[135] W. C. Waterhouse. Abelian varieties over finite fields. *Ann. Sci. École Norm. Sup. (4)*, 2:521–560, 1969.

[136] A. Weil. *Sur les courbes algébriques et les variétés qui s'en déduisent.* Publ. Inst. Math. Univ. Strasbourg **7** (1945). Hermann et Cie., Paris, 1948.

[137] A. Weil. Numbers of solutions of equations in finite fields. *Bull. Amer. Math. Soc.*, 55:497–508, 1949.

Nederlandse Samenvatting

In dit proefschrift bestuderen en ontwerpen we p-adische algoritmen om de zetafunctie van een kromme over een eindig lichaam te berekenen. De zetafunctie bevat belangrijke aritmetische en meetkundige informatie over de kromme en de Jacobiaan van de kromme. Efficiënte algoritmen om de zetafunctie te berekenen hebben praktische toepassingen in de cryptografie en codetheorie.

Hoofdstuk 1: Inleiding

De huidige informatiemaatschappij wordt in toenemende mate afhankelijk van technieken om digitale informatie te authentiseren en te beschermen tegen afluisteren. Enkele voorbeelden zijn elektronische handel, bankieren via het Internet, draadloze communicatie en elektronisch stemmen.

Cryptografie is de wetenschap die bestudeert hoe informatie beveiligd kan worden. De belangrijkste doelstellingen van cryptografie zijn het beschermen van informatie tegen afluisteren en het garanderen van de integriteit van de informatie. Het standaard voorbeeld van een cryptografisch algoritme is encryptie: de informatie m wordt vercijferd met een algoritme E_e dat afhankelijk is van een sleutel e. Bij een goed algoritme is de cijfertekst $C = E_e(m)$ niet begrijpbaar. De cijfertekst kan enkel ontcijferd worden door gebruik te maken van de inverse transformatie $D_d = E_e^{-1}$ aangezien $D_d(C) = E_e^{-1}(E_e(m)) = m$. Dit proces wordt decryptie genoemd.

Voor 1976 waren alle cryptografische primitieven gebaseerd op symmetrische sleutel- of conventionele algoritmen. In dit geval delen de zender en ontvanger dezelfde sleutel $e = d$ die ze beide geheim houden. Het belangrijkste probleem van conventionele cryptografie is het verdelen van de geheime sleutels. In 1976 introduceerden Diffie en Hellman [26] publieke sleutel cryptografie, waarin elke gebruiker I een publieke sleutel e_I en een private sleutel d_I heeft. Iedereen die naar I een vertrouwelijk bericht m wil sturen kan de cijfertekst $E_{e_I}(m)$ berekenen, maar enkel I kan de klaartekst vinden als $D_{d_I}(E_{e_I}(m)) = m$.

Het bekendste publieke sleutel algoritme is RSA dat in 1978 ontworpen werd door Rivest, Shamir en Adleman [103]. Het is gebaseerd op het feit dat het eenvoudig is om twee grote priemgetallen te vinden en deze te vermenigvuldigen, terwijl het factoriseren van hun product zeer moeilijk is. Het snelste algoritme om een getal te factoriseren heeft immers een subexponentiële uitvoeringstijd.

Het enige veelgebruikte alternatief zijn algoritmen gebaseerd op het discreet-logaritme-probleem in een Abelse groep, zoals de multiplicatieve groep van een eindig lichaam. De bekendste voorbeelden zijn het ElGamal algoritme [28, 29] en het DSA-algoritme [38] (Digital Signature Standard).

In dit proefschrift bestuderen we algoritmen om Abelse groepen te vinden waarvoor het discreet-logaritme-probleem moeilijk op te lossen is. Zij G een cyclische Abelse groep met generator g en zij $h \in G$ een willekeurig element. Het discreet logaritme van h met grondtal g is gedefinieerd als het natuurlijke getal $0 \leq x < |G|$ waarvoor $h = g^x$. Het probleem om het getal x te berekenen gegeven g en h wordt het discreet-logaritme-probleem (DLP) genoemd. Pohlig en Hellman [100] toonden aan dat het DLP in G equivalent is met het DLP in deelgroepen van G met een priem aantal elementen. Vanaf nu zullen we dan ook onderstellen dat $|G| = p$ met p een priemgetal.

Door exhaustief te zoeken kan het DLP worden opgelost in $O(p)$ groepsbewerkingen. Echter, het kleine-stap grote-stap (baby-step giant-step) algoritme van Shanks [115] en de rho- en lambda-algoritmen van Pollard [101] vragen slechts $O(\sqrt{p})$ groepsbewerkingen. Bemerk dat dit nog steeds exponentieel is in de bit-lengte van p. Het wordt algemeen aangenomen dat 2^{80} bewerkingen voor de huidige computers onbereikbaar is; dit impliceert dat p tenminste 2^{160} moet zijn. Aangezien er een subexponentieel algoritme [1] bestaat om het discreet-logaritme-probleem in de multiplicatieve groep van een eindig lichaam $\mathbb{F}_q$ op te lossen, wordt in praktijk q groter dan 2^{1024} gekozen. Bemerk dat dit veel groter is dan de ondergrens 2^{160} en de vraag stelt zich dan ook of er groepen bestaan die deze ondergrens bereiken.

In 1985 stelden Koblitz [65] en Miller [86] voor om de multiplicatieve groep van een eindig lichaam te vervangen door de groep op een elliptische kromme $\overline{E}$ over een eindig lichaam $\mathbb{F}_q$. Het grote voordeel is dat er geen subexponentieel algoritme bestaat om het DLP in $\overline{E}(\mathbb{F}_q)$ op te lossen, behalve voor enkele speciale types krommen zoals supersinguliere krommen [42, 81] en anomale krommen [108, 113, 119]. Het snelste algoritme om voor willekeurige $\overline{E}$ het DLP in $\overline{E}(\mathbb{F}_q)$ op te lossen is Pollard's rho algoritme [101]. Een direct gevolg hiervan is dat cryptografische algoritmen gebaseerd op het DLP in $\overline{E}(\mathbb{F}_q)$ veel kortere sleutels kunnen gebruiken.

Enkele jaren later stelde Koblitz [66] voor om te werken in de Jacobiaan $J_{\overline{C}}(\mathbb{F}_q)$ van een hyperelliptische kromme $\overline{C}(\mathbb{F}_q)$; deze constructie werd nadien verder veralgemeend naar de klasse van C_{ab} krommen door Arita [7]. Net zoals voor elliptische krommen wordt de moeilijkheidsgraad van het DLP in $J_{\overline{C}}(\mathbb{F}_q)$ voornamelijk bepaald door de groepsorde $|J_{\overline{C}}(\mathbb{F}_q)|$.

Het berekenen van deze groepsorde is een moeilijk probleem en was eind 1999 zelfs totaal onmogelijk voor hyperelliptische en C_{ab} krommen, wat hun toepassing in cryptografie in de weg stond. Dit proefschrift beschrijft een aantal praktische algoritmen die het voor het eerst mogelijk maken om $|J_{\overline{C}}(\mathbb{F}_q)|$ te berekenen voor alle types krommen die gebruikt worden in cryptografie, i.e. elliptische, hyperelliptische en C_{ab} krommen. Omwille van hun p-adische aard zijn deze algoritmen enkel efficiënt voor krommen over eindige lichamen van kleine karakteristiek p.

Hoofdstuk 2: Wiskundige Achtergrond

Hoofstuk 2 herhaalt de definities en eigenschappen van de wiskundige objecten die verderop in het proefschrift gebruikt worden. De nadruk ligt op algebraïsche variëteiten, i.e. meetkundige objecten die er lokaal uitzien als de oplossingsverzameling van een stelsel veeltermen in meerdere veranderlijken.

Een kromme is een algebraïsche variëteit van dimensie 1. De belangrijkste invariant van een kromme is het geslacht, een begrip dat nauw samenhangt met de graad van de vergelijking van de kromme. In dit proefschrift besteden we vooral aandacht aan de krommen die praktische toepassingen hebben in cryptografie.

- Een *elliptische kromme* $\overline{E}$ over een eindig lichaam $\mathbb{F}_q$ is een kromme van geslacht 1 met een $\mathbb{F}_q$-rationaal punt $\mathcal{O}$ en kan altijd worden voorgesteld door een Weierstrass vergelijking, i.e.

  $$\overline{E} : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \,, \quad a_i \in \mathbb{F}_q \,.$$

  De belangrijkste eigenschap van een elliptische kromme is dat voor elk eindig lichaam $\mathbb{F}_{q^k}$ de verzameling van $\mathbb{F}_{q^k}$-rationale punten $\overline{E}(\mathbb{F}_{q^k})$ een Abelse groep vormt.

- Een *hyperelliptische kromme* $\overline{C}$ over een eindig lichaam $\mathbb{F}_q$ is een overdekking van de projectieve lijn van graad twee en wordt gedefinieerd door een vergelijking van de vorm

  $$\overline{C} : y^2 + \overline{h}(x)y + \overline{f}(x) = 0 \,, \quad \overline{h}, \overline{f} \in \mathbb{F}_q[x] \,,$$

  met $\deg \overline{h} \leq g$ en $\deg \overline{f} = 2g+1$ en g het geslacht van de kromme. Bemerk dat een elliptische kromme een hyperelliptische kromme is van geslacht 1.

- Een C_{ab} *kromme* $\overline{C}$ is een veralgemening van een hyperelliptsche kromme en wordt gedefinieerd door een vergelijking van de vorm

  $$\overline{C} : y^a + \sum_{i=1}^{a-1} \overline{f}_i(x)y^i + \overline{f}_0(x) = 0 \,, \quad \overline{f}_i \in \mathbb{F}_q[x] \,,$$

  met $\gcd(a,b) = 1$, $\deg \overline{f}_0 = b$ en $a \deg \overline{f}_i + bi < ab$ voor $0 < i < a$. Het geslacht van $\overline{C}$ wordt gegeven door $g = (a-1)(b-1)/2$. Een hyperelliptische kromme is dus een $C_{2,2g+1}$ kromme.

Een elliptische kromme is het eenvoudigste voorbeeld van een Abelse variëteit, i.e. een algebraïsche variëteit met een natuurlijke groepsstructuur. Hyperelliptische en C_{ab} krommen zijn echter geen Abelse variëteiten. Cryptografen rekenen daarom in de Jacobiaan $J_{\overline{C}}$ van de kromme; dit is de kleinst mogelijke Abelse variëteit die de kromme bevat. De moeilijkheidsgraad van het discreet-logaritme-probleem wordt voornamelijk bepaald door de orde van de groep van $\mathbb{F}_q$-rationale punten $J_{\overline{C}}(\mathbb{F}_q)$.

Aangezien een elliptische kromme $\overline{E}/\mathbb{F}_q$ isomorf is met zijn Jacobiaan $J_{\overline{E}}$ kunnen we de groepsorde van $J_{\overline{E}}(\mathbb{F}_q)$ berekenen als het aantal $\mathbb{F}_q$-rationale punten op $\overline{E}$. Voor een meer algemene kromme $\overline{C}/\mathbb{F}_q$ van geslacht g wordt de groepsorde van $J_{\overline{C}}(\mathbb{F}_q)$ bepaald door het aantal punten op $\overline{C}$ met coördinaten in de g uitbreidingslichamen $\mathbb{F}_{q^k}$ voor $k = 1, \ldots, g$. Zij $N_k = |\overline{C}(\mathbb{F}_{q^k})|$ voor $k \in \mathbb{N}$ het aantal $\mathbb{F}_{q^k}$-rationale punten op $\overline{C}$, dan is de zetafunctie van de kromme $\overline{C}$ gedefinieerd als de genererende functie

$$Z(\overline{C}/\mathbb{F}_q; t) = \exp\left(\sum_{k=1}^{\infty} \frac{N_k}{k} t^k\right).$$

In 1948 bewees Weil [136] dat de zetafunctie van een gladde projectieve kromme $\overline{C}$ van geslacht g voldoet aan de volgende eigenschappen:

- Rationaliteit: $Z(\overline{C}/\mathbb{F}_q; t)$ is een rationale functie
$$Z(\overline{C}/\mathbb{F}_q; t) = \frac{L(t)}{(1-t)(1-qt)},$$
met $L(t) = 1 + c_1 t + \cdots + c_{2g-1} t^{2g-1} + q^g t^{2g} \in \mathbb{Z}[t]$.
- Functionaalvergelijking:
$$Z(\overline{C}/\mathbb{F}_q; t) = q^{g-1} t^{2g-2} Z(\overline{C}/\mathbb{F}_q; \frac{1}{qt}).$$
- Riemann hypothese: indien we $L(t)$ schrijven als $L(t) = \prod_{i=1}^{2g}(1 - \alpha_i t)$ dan zijn de α_i algebraïsche gehelen met $|\alpha_i| = \sqrt{q}$.

De belangrijkste eigenschap voor toepassingen in cryptografie is ongetwijfeld

$$|J_{\overline{C}}(\mathbb{F}_q)| = L(1).$$

Het aantal $\mathbb{F}_q$-rationale punten op een kromme $\overline{C}/\mathbb{F}_q$ kan ook bekomen worden als het aantal vaste punten van het Frobenius endomorfisme

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

Inderdaad, zij α een element van de algebraïsche afsluiting $\overline{\mathbb{F}}_q$ van $\mathbb{F}_q$ dan zal α een element zijn van $\mathbb{F}_q$ als en slechts als $\alpha^q = \alpha$. Het aantal $\mathbb{F}_q$-rationale punten op $\overline{E}$ voldoet dus aan

$$\#\overline{E}(\mathbb{F}_q) = \#\mathrm{Ker}(\overline{F} - \mathrm{Id}),$$

met Id de identieke afbeelding. Het is dan ook niet verwonderlijk dat het Frobenius endomorfisme een centrale rol zal spelen in de algoritmen om $Z(\overline{C}/\mathbb{F}_q; t)$ te berekenen.

In dit proefschrift bestuderen en ontwerpen we p-adische algoritmen om de zetafunctie te berekenen voor alle types krommen die toepassingen hebben in cryptografie, nl. elliptische, hyperelliptische en C_{ab} krommen. Deze p-adische algoritmen berekenen $L(t)$ mod p^N voor een zekere precisie N door de kromme en het Frobenius endomorfisme te liften naar een onvertakte p-adische ring $\mathbb{Z}_q$ die voldoet aan $\mathbb{Z}_q/(p\mathbb{Z}_q) \cong \mathbb{F}_q$.

Hoofdstuk 3: Elliptische Krommen

In dit hoofdstuk behandelen we p-adische algoritmen om het aantal punten op een elliptische kromme $\overline{E}$ gedefinieerd over een eindig lichaam $\mathbb{F}_q$ met $q = p^n$ elementen, te berekenen. Het aantal $\mathbb{F}_q$-rationale punten $\overline{E}(\mathbb{F}_q)$ wordt gegeven door $\#\overline{E}(\mathbb{F}_q) = q + 1 - t$, met t het spoor van het Frobenius endomorfisme $\overline{F} : \overline{E} \to \overline{E} : (x, y) \mapsto (x^q, y^q)$. De stelling van Hasse [117, Theorem V.1.1] toont aan dat $|t| \leq 2\sqrt{q}$, dus volstaat het om t mod B met $B > 4\sqrt{q}$ te berekenen. In het vervolg van dit hoofdstuk zullen we de complexiteit van een algoritme uitdrukken in functie van een constante μ. Deze constante is gedefinieerd zodanig dat de vermenigvuldiging van twee M-bit getallen kan berekend worden in tijdscomplexiteit $O(M^\mu)$. Zo is $\mu = 2$ voor de klassieke vermenigvuldiging en $\mu = 1 + \varepsilon$ met $\varepsilon \in \mathbb{R}_{>0}$ voor Schönhage-Strassen vermenigvuldiging [110].

In 1985 beschreef Schoof [111] het eerste polynoomtijd algoritme om het aantal punten op een elliptische kromme over $\mathbb{F}_q$ te berekenen. Het basisidee van Schoofs algoritme is om t mod l te bepalen voor alle priemgetallen $l \leq l_m$, met l_m het kleinste priemgetal waarvoor $\prod_{l \leq l_m} l > 4\sqrt{q}$. Deze aanpak wordt l-adisch genoemd omdat het algoritme werkt met de l-torsiepunten $\overline{E}[l]$, die kunnen beschouwd worden als een benadering van de eerste l-adische homologiegroep. De tijdscomplexiteit van Schoof's algoritme bedraagt $O((\log q)^{3\mu+2})$ en de geheugencomplexiteit is $O((\log q)^3)$. Ondanks de polynomiale tijdscomplexiteit is het algoritme van Schoof te traag voor cryptografische toepassingen. Optimalisaties door Elkies en Atkin [30, 111, 112] leidden tot het SEA-algoritme met tijdscomplexiteit $O((\log q)^{2\mu+2})$ en geheugencomplexiteit $O((\log q)^2)$. Het SEA-algoritme is efficiënt genoeg om het aantal punten te berekenen op elliptische krommen die gebruikt kunnen worden in de cryptografie. Aangezien de tijdscomplexiteit afhangt van $\log q$ is het SEA-algoritme efficiënt in zowel kleine als grote karakteristiek.

Eind 1999 beschreef Satoh [105] het eerste p-adische algoritme om $\#\overline{E}(\mathbb{F}_q)$ te berekenen. Het basisidee van Satoh's algoritme is om de elliptische kromme en het Frobenius endomorfisme op canonieke wijze te liften naar een p-adische ring. Zij $\mathbb{Z}_q$ de valuatiering van een onvertakte uitbreiding $\mathbb{Q}_q$ van de p-adische getallen $\mathbb{Q}_p$ zodat $\mathbb{Z}_q/(p\mathbb{Z}_q) \cong \mathbb{F}_q$. Deuring [25] heeft bewezen dat indien $\overline{E}$ ordinair is, er altijd een elliptische kromme E over $\mathbb{Z}_q$ bestaat met de volgende eigenschappen: de reductie modulo p van E is gelijk aan $\overline{E}$ en de endomorfismering $\mathrm{End}(E)$ is isomorf met $\mathrm{End}(\overline{E})$. De elliptische kromme E wordt de canonieke lift van $\overline{E}$ genoemd en speelt de hoofdrol in het algoritme van Satoh. Omdat $\mathrm{End}(E) \cong \mathrm{End}(\overline{E})$ bestaat er een uniek endomorfisme $\mathcal{F}$ van E dat reduceert naar het Frobenius endomorfisme $\overline{F}$. Bovendien is het spoor van $\mathcal{F}$ gelijk aan het spoor van Frobenius t.

Het algoritme van Satoh werkt in twee fasen: de eerste fase berekent de canonieke lift E modulo p^N via een Newton iteratie in meerdere veranderlijken; de tweede fase lift de p-torsiepunten $\overline{E}[p]$ naar een deelgroep van $E[p]$ en bepaalt $t \mod p^N$. Indien $p^N > 4\sqrt{q}$ kan de correcte waarde van t en dus ook $\#\overline{E}(\mathbb{F}_q)$ worden berekend. De tijdscomplexiteit van Satoh's algoritme bedraagt

$O(n^{2\mu+1})$ en de geheugencomplexiteit is $O(n^3)$ voor p constant. Bemerk dat in tegenstelling tot het SEA-algoritme, de complexiteit van Satoh's algoritme afhangt van n, i.e. de uitbreidingsgraad van het eindig lichaam, en niet van $\log q$. De O-constante groeit echter als $O(p^2)$ met als gevolg dat Satoh's algoritme enkel efficiënt is in kleine karakteristiek p. Om technische redenen, beperkte Satoh zich tot het geval $p > 3$. Voor praktische toepassingen is $p = 2$ echter het belangrijkste geval. In 2000 werd Satoh's algoritme dan ook uitgebreid naar $p = 2$ en $p = 3$ door Fouquet, Gaudry en Harley [41] en onafhankelijk naar $p = 2$ door Skjernaa [118].

Het resultaat van Hoofdstuk 3 is een geoptimaliseerde versie van Satoh's algoritme die in de praktijk niet alleen 1.5 keer sneller is, maar bovendien een kwadratische geheugencomplexiteit heeft. Het basisidee van dit algoritme is om de Newton iteratie in meerdere veranderlijken te vervangen door een geïtereerde Newton iteratie in één veranderlijke. A priori is het echter niet duidelijk dat deze geïtereerde Newton iteratie hoe dan ook convergeert en in het algemeen is dit ook niet het geval. Door gebruik te maken van de specifieke eigenschappen van het stelsel veeltermvergelijkingen in Satoh's algoritme kunnen we convergentie bewijzen. Het volgende theorema werd gepubliceerd in [133].

THEOREMA 1 *Zij $\overline{E}$ een elliptische kromme over een eindig lichaam $\mathbb{F}_q$ met $q = p^n$ en $j(\overline{E}) \notin \mathbb{F}_{p^2}$. Dan bestaat er een deterministisch algoritme om het aantal punten $\#\overline{E}(\mathbb{F}_q)$ te berekenen met tijdcomplexiteit $O(n^{2\mu+1})$ en geheugencomplexiteit $O(n^2)$ voor p constant.*

De meer recente optimalisaties zijn bovendien allemaal gebaseerd op het feit dat de geïtereerde Newton iteratie in één veranderlijke convergeert. Het praktisch belang van dit algoritme is dat het voor het eerst mogelijk was om elliptische krommen voor cryptografische doeleinden te genereren in een omgeving met zeer beperkte mogelijkheden, zoals bijvoorbeeld een geavanceerde chipkaart.

Hoofdstuk 3 bevat verder ook een uniforme beschrijving van de meer recente ontwikkelingen op het gebied van p-adische algoritmen om $\#\overline{E}(\mathbb{F}_q)$ te berekenen:

- Mestre's AGM algoritme. In een brief aan Gaudry en Harley [85] geeft Mestre een zeer elegant algoritme voor $p = 2$ gebaseerd op het rekenkundig-meetkundig gemiddelde (Arithmetic Geometric Mean (AGM)) met tijdscomplexiteit $O(n^{2\mu+1})$ en geheugencomplexiteit $O(n^2)$. Aan de oppervlakte lijkt dit algoritme volledig verschillend van Satoh's algoritme, maar de onderliggende theorie is dezelfde. De voordelen van dit algoritme zijn dat het heel eenvoudig te implementeren is en dat het bovendien zeer efficiënt is. In 2003 werd dit algoritme uitgebreid naar karakteristiek $p > 2$ door Carls [15] en Kohel [67], maar het resultaat is minder elegant dan voor $p = 2$.

- Satoh-Skjernaa-Taguchi algoritme. De onvertakte uitbreiding $\mathbb{Q}_q$ kan op vele verschillende wijzen worden gerepresenteerd. Satoh, Skjernaa en Taguchi [109] construeren één welbepaalde representatie die toelaat om de actie van de Frobenius substitutie efficiënt te berekenen. Deze alternatieve representatie van $\mathbb{Q}_q$ laat toe om de geïtereerde Newton iteratie verder te optimaliseren. Het resultaat is een algoritme om de canonieke lift van een elliptische kromme te benaderen in $O(n^{2\mu+1/(\mu+1)})$ bit-operaties, na een voorberekening die enkel afhangt van het eindig lichaam en $O(n^{2\mu+1})$ bit-operaties vergt. Verder beschrijven Satoh, Skjernaa en Taguchi ook een efficiënt algoritme om de norm van een element in $\mathbb{Q}_q$ te berekenen met tijdscomplexiteit $O(n^{2\mu+0.5})$. Het uiteindelijk resultaat is een algoritme om $\#\overline{E}(\mathbb{F}_q)$ te berekenen in $O(n^{2\mu+0.5})$ bit-operaties en $O(n^2)$ geheugen.

- Kim et al.'s algoritme. Kim, Park, Cheon, Park, Kim en Hahn [63] beschrijven een algoritme dat enkel efficiënt is voor eindige lichamen met een Gaussische Normale Basis. Deze basis kan op triviale wijze gelift worden naar $\mathbb{Q}_q$ en laat eveneens toe om een geïtereerde Frobenius substitutie zeer efficiënt te berekenen. Deze eigenschap leidt tot een algoritme om de norm van een element in $\mathbb{Q}_q$ te berekenen in tijd $O(n^{2\mu}\log n)$. Gecombineerd met het algoritme van Satoh, Skjernaa en Taguchi leidt dit tot een algoritme om $\#\overline{E}(\mathbb{F}_q)$ te berekenen met tijdscomplexiteit $O(n^{2\mu+1/(\mu+1)})$ en geheugencomplexiteit $O(n^2)$, zonder enige voorberekening.

- Gaudry's algoritme. Gaudry [48] combineert Mestre's AGM algoritme en het algoritme van Satoh, Skjernaa en Taguchi. Voor $p = 2$ geeft de AGM iteratie aanleiding tot een eenvoudigere modulaire veelterm, wat resulteert in een efficiënter algoritme dan dat van Satoh, Skjernaa en Taguchi, maar met dezelfde complexiteit.

- Lercier-Lubicz' algoritme. Lercier en Lubicz [75] beschrijven het eerste asymptotisch optimaal algoritme om $\#\overline{E}(\mathbb{F}_q)$ te berekenen voor eindige lichamen met Gaussische Normale Basis. Het basisidee van dit algoritme is om de geïtereerde Newton iteratie die slechts lineair convergeert te vervangen door een dubbel recursieve Newton iteratie die kwadratisch convergeert. Deze Newton iteratie maakt echter gebruik van geïtereerde Frobenius substituties en is dus enkel efficiënt is voor eindige lichamen met Gaussische Normale Basis. De tijdscomplexiteit bedraagt $O(n^{2\mu}\log n)$ en de geheugencomplexiteit is $O(n^2)$.

- Harley's algoritme. In een e-mail [55], gepost op de NMBRTHRY lijst in december 2002, schetst Harley een algoritme om $\#\overline{E}(\mathbb{F}_q)$ te berekenen in $O(n^{2\mu}\log n)$ bit-operaties en $O(n^2)$ geheugen. De e-mail bevat enkel een deel van de code van een programma en vele belangrijke subroutines ontbreken, hoogst waarschijnlijk omwille van een lopende octrooi aanvraag. Het basisidee van dit algoritme is een kruising tussen het algoritme van Satoh, Skjernaa en Taguchi en dat van Lercier en Lubicz: een welgekozen representatie van $\mathbb{Q}_q$ die een efficiënte Frobenius substitutie mogelijk maakt, gecombineerd met een dubbel recursieve Newton iteratie die resulteert in een kwadratisch convergerend algoritme.

Het uiteindelijke resultaat van al deze algoritmen is dat het nu mogelijk is om in minder dan 0.1 seconden $\#\overline{E}(\mathbb{F}_q)$ te berekenen voor $q \simeq 2^{160}$.

Hoofdstuk 4: Hyperelliptische Krommen

Dit hoofdstuk behandelt p-adische algoritmen om de zetafunctie van een hyperelliptische kromme $\overline{C}$ over een eindig lichaam $\mathbb{F}_q$ met $q = p^n$ te berekenen. Het basisidee van deze algoritmen gaat terug naar de p-adische cohomologie voorgesteld door Monsky en Washnitzer [89, 90, 91]. Zij $\mathbb{Q}_p$ het lichaam van de p-adische getallen en $\mathbb{Q}_q$ een onvertakte uitbreiding van $\mathbb{Q}_p$ van graad n. Monsky en Washnitzer hadden tot doel om een expliciete uitdrukking voor de zetafunctie van $\overline{C}$ te vinden in termen van een Frobenius operator $\mathcal{F}$ die lineair werkt op p-adische cohomologie groepen $H^i(\overline{C}/\mathbb{Q}_q)$ geassocieerd aan $\overline{C}$. De Monsky-Washnitzer cohomologie is gedefinieerd als de de Rham cohomologie van een ring van overconvergente machtreeksen. Men kan bewijzen dat de $H^i(\overline{C}/\mathbb{Q}_q)$ eindig dimensionale $\mathbb{Q}_q$-vectorruimten zijn en dat voor gladde affiene krommen alleen $H^0(\overline{C}/\mathbb{Q}_q)$ en $H^1(\overline{C}/\mathbb{Q}_q)$ niet-nul zijn. Het belangrijkste resultaat van Monsky-Washnitzer cohomologie is dat de $H^i(\overline{C}/\mathbb{Q}_q)$ voldoen aan een Lefschetz vaste-punten-stelling, i.e. het aantal $\mathbb{F}_q$-rationale punten $\overline{C}(\mathbb{F}_q)$ voldoet aan

$$\#\overline{C}(\mathbb{F}_q) = \mathrm{Tr}\left(q\mathcal{F}_*^{-1}|H^0(\overline{C}/\mathbb{Q}_q)\right) - \mathrm{Tr}\left(q\mathcal{F}_*^{-1}|H^1(\overline{C}/\mathbb{Q}_q)\right) .$$

Indien we deze formule willen gebruiken als basis voor een algoritme om de zetafunctie van $\overline{C}$ te berekenen, moeten we de volgende twee problemen oplossen. Ten eerste, hoe construeren we een expliciete basis voor $H^0(\overline{C}/\mathbb{Q}_q)$ en $H^1(\overline{C}/\mathbb{Q}_q)$ en ten tweede, hoe berekenen we de actie van de operator $\mathcal{F}$ op deze basis? Voor $H^0(\overline{C}/\mathbb{Q}_q)$ is het antwoord zeer eenvoudig aangezien dit een 1-dimensionale $\mathbb{Q}_q$-vectorruimte is waarop $\mathcal{F}_*$ de identieke afbeelding is.

In 2001 gaf Kedlaya [61] een antwoord op deze vragen voor $\overline{C}$ een hyperelliptische kromme over een eindig lichaam $\mathbb{F}_q$ met $q = p^n$ en $p > 2$. Kedlaya construeert een basis voor $H^1(\overline{C}/\mathbb{Q}_q)$ door $\overline{C}$ te liften naar een kromme $C/\mathbb{Q}_q$ en te bewijzen dat de eerste algebraïsche de Rham cohomologiegroep $H^1_{DR}(C/\mathbb{Q}_q)$ isomorf is met $H^1(\overline{C}/\mathbb{Q}_q)$. De Frobenius operator $\mathcal{F}$ wordt uitgedrukt als een overconvergente machtreeks wat toelaat om de actie van $\mathcal{F}_*$ op de basis van $H^1(\overline{C}/\mathbb{Q}_q)$ te berekenen. De tijdscomplexiteit van Kedlaya's algoritme bedraagt $O(g^{4+\varepsilon}n^{3+\varepsilon})$ en de geheugencomplexiteit is $O(g^3n^3)$ met g het geslacht van $\overline{C}$.

Het resultaat van Hoofdstuk 4 is een uitbreiding van Kedlaya's algoritme naar $p = 2$, wat voor praktische toepassingen in de cryptografie het belangrijkste geval is. Omdat deze uitbreiding niet triviaal is, bestaat dit hoofdstuk uit twee delen. In het eerste deel behandelen we enkel hyperelliptische krommen van de vorm

$$\overline{C}_m : y^2 - x^m y - \overline{f}(x) = 0\,,$$

met $\overline{f} \in \mathbb{F}_{2^n}[x]$ een veelterm van graad $2g+1$ en g het geslacht van $\overline{C}_m$. Deze krommen hebben het voordeel dat de groepswet in $J_{\overline{C}_m}(\mathbb{F}_{2^n})$ efficiënter is dan voor willekeurige hyperelliptische krommen. Het grote verschil met Kedlaya's algoritme is dat de kromme op een heel speciale manier gelift moet worden om ervoor te zorgen dat de basis van $H^1_{DR}(C_m/\mathbb{Q}_q)$ nog steeds een basis is van $H^1(\overline{C}_m/\mathbb{Q}_q)$. Het volgende theorema werd bekomen in samenwerking met J. Denef en werd gepubliceerd in [23].

THEOREMA 2 *Zij $\overline{C}_m$ een gladde hyperelliptische kromme over een eindig lichaam $\mathbb{F}_{2^n}$ gedefinieerd door een vergelijking van de vorm $y^2 - x^m y - \overline{f}(x) = 0$, met $\overline{f} \in \mathbb{F}_{2^n}[x]$ een veelterm van graad $2g+1$ en g het geslacht van $\overline{C}_m$. Dan bestaat er een deterministisch algoritme om de zetafunctie van $\overline{C}_m$ te berekenen met tijdscomplexiteit $O(g^3 n^3 (g^2 + (\log n)^2 \log\log n) \log gn \log\log gn)$ en geheugencomplexiteit $O(g^3 n^3)$.*

In het tweede deel veralgemenen we dit algoritme naar willekeurige hyperelliptische krommen $\overline{C}$ over een eindig lichaam $\mathbb{F}_{2^n}$. Opnieuw moet de kromme op een heel speciale manier gelift worden om te garanderen dat $H^1_{DR}(C/\mathbb{Q}_q)$ isomorf is met $H^1(\overline{C}/\mathbb{Q}_q)$. Het volgende resultaat werd bekomen in samenwerking met J. Denef en werd gepubliceerd in [24] en [131].

THEOREMA 3 *Er bestaat een deterministisch algoritme om de zetafunctie van een gladde hyperelliptische kromme van geslacht g gedefinieerd over een eindig lichaam $\mathbb{F}_{2^n}$ te berekenen in*

- *$O(g^3 n^3 (g + (\log n)^2 \log\log n) \log gn \log\log gn)$ bit-operaties en $O(g^3 n^3)$ geheugen in het gemiddelde geval,*
- *$O(g^3 n^3 (g^2 + (\log n)^2 \log\log n) \log gn \log\log gn)$ bit-operaties en $O(g^4 n^3)$ geheugen in het slechtste geval.*

Het praktisch belang van deze stellingen is dat het nu voor het eerst mogelijk is om hyperelliptische krommen te genereren voor cryptografische doeleinden. Bijvoorbeeld, de groepsorde van een 168-bit Jacobiaan van een hyperelliptische kromme van geslacht 2, 3 of 4 kan berekend worden in minder dan 80 seconden. Aangezien de uitvoeringstijd van deze algoritmen polynomiaal is in het geslacht van de kromme, kunnen we ook het geslacht sterk opdrijven. Zo is het bijvoorbeeld mogelijk om de zetafunctie van een willekeurige hyperelliptische kromme van geslacht 350 over $\mathbb{F}_2$ te berekenen in 3.5 dagen.

Een alternatieve strategie gebaseerd op Dwork cohomologie werd beschreven door Lauder en Wan [70, 71] en leidt tot een algoritme met dezelfde complexiteit als in Theorema 3. In 2002 veralgemeende Mestre [84] de AGM naar ordinaire hyperelliptische krommen over een eindig lichaam van karakteristiek twee. Dit algoritme is zeer efficiënt voor geslacht kleiner dan 4. Het grootste nadeel is echter dat de tijdscomplexiteit exponentieel is in het geslacht van de kromme.

Hoofdstuk 5: C_{ab} Krommen

In dit hoofdstuk ontwerpen we een algoritme om de zetafunctie van een C_{ab} kromme $\overline{C}$ over een eindig lichaam $\mathbb{F}_q$ met $q = p^n$ te berekenen. De theoretische achtergrond is dezelfde als voor hyperelliptische krommen, i.e. we passen de Lefschetz vaste-punten-stelling toe op de Monsky-Washnitzer cohomologiegroepen

$H^i(\overline{C}/\mathbb{Q}_q)$. Het algoritme zelf is echter totaal verschillend en veel algemener dan voor hyperelliptische krommen.

Een eerste resultaat in Hoofdstuk 5 is een nieuwe methode om de Frobenius operator $\mathcal{F}$ te berekenen als een overconvergente machtreeks. Deze methode en het bewijs dat de resulterende machtreeks inderdaad onverconvergent is, werkt voor willekeurige gladde affiene krommen over elk eindig lichaam $\mathbb{F}_q$. In het geval van C_{ab} krommen bekomen we bovendien een zeer scherpe ondergrens voor de convergentiestraal van deze machtreeks. Als onderdeel van het bewijs veralgemenen we bovendien een stelling van Macaulay [78] naar discrete valuatie ringen.

Een tweede resultaat is een expliciete beschrijving van een basis voor de algebraïsche de Rham cohomologie $H^1_{DR}(C/\mathbb{Q}_q)$ en een bewijs dat $H^1_{DR}(C/\mathbb{Q}_q)$ isomorf is met de Monsky-Washnitzer cohomologie $H^1(\overline{C}/\mathbb{Q}_q)$. Bovendien geven we reductieformules die toelaten om de actie van $\mathcal{F}$ op deze basis te berekenen.

De combinatie van deze resultaten leidt dan tot een algoritme om de zetafunctie van een C_{ab} kromme in polynoomtijd te berekenen. De volgende stelling is het resultaat van een samenwerking met J. Denef en zal ter publicatie [132] voorgelegd worden.

THEOREMA 4 *Er bestaat een deterministisch algoritme om de zetafunctie van een gladde C_{ab} kromme van geslacht g gedefinieerd over een eindig lichaam $\mathbb{F}_{p^n}$ te berekenen in $O(g^3n^3(g^2 + (\log n)^2 \log\log n) \log gn \log\log gn)$ bit-operaties en $O(g^3n^3)$ geheugen voor p constant.*

Niettegenstaande dit algoritme zeer algemeen is, toont een eerste implementatie in de `C` programmeertaal aan dat dit algoritme toch praktisch is voor cryptografische toepassingen. Zo kan bijvoorbeeld de groepsorde van een 168-bit Jacobiaan van een $C_{3,4}$ kromme berekend worden in 1.4 uur.

Hoofdstuk 6: Besluit en Open Problemen

Besluit

De zetafunctie $Z(\overline{C}/\mathbb{F}_q; t)$ van een kromme $\overline{C}$ over een eindig lichaam $\mathbb{F}_q$ bevat belangrijke informatie over de kromme en de Jacobiaan $J_{\overline{C}}$ van de kromme. Zo is het bijvoorbeeld mogelijk om het aantal $\mathbb{F}_{q^k}$-rationale punten op $\overline{C}$ af te leiden voor iedere $k \in \mathbb{N}$. Bovendien volgt ook de groepsorde van $J_{\overline{C}}(\mathbb{F}_q)$ heel eenvoudig. Deze laatste eigenschap heeft belangrijke toepassingen in de cryptografie aangezien de moeilijkheidsgraad van het discreet-logaritme-probleem in $J_{\overline{C}}(\mathbb{F}_q)$ voornamelijk bepaald wordt door de groepsorde $|J_{\overline{C}}(\mathbb{F}_q)|$.

In dit proefschrift hebben we efficiënte p-adische algoritmen ontworpen om de zetafunctie te berekenen voor alle types krommen die in de cryptografie gebruikt worden, namelijk elliptische, hyperelliptische en C_{ab}-krommen. Eind 1999 was het zelfs totaal onmogelijk om de zetafunctie van hyperelliptische en C_{ab} krommen voor cryptografische doeleinden te berekenen. De enige beperking van deze algoritmen is dat ze door hun p-adische aard enkel efficiënt zijn voor krommen over eindige lichamen van kleine karakteristiek p.

De algoritmen voor elliptische krommen zijn, in navolging van het werk van Satoh [105], gebaseerd op de theorie van de canonieke lift. Satoh's algoritme heeft een tijds- en geheugencomplexiteit die kubisch is in de uitbreidingsgraad van het eindig lichaam. Het belangrijkste resultaat van Hoofdstuk 3 is een geoptimaliseerde versie van Satoh's algoritme die nog steeds een kubische tijdscomplexiteit heeft, maar een kwadratische geheugencomplexiteit. Dit maakt het mogelijk om elliptische krommen voor toepassingen in de cryptografie te genereren in een omgeving met zeer beperkte mogelijkheden zoals bijvoorbeeld een geavanceerde chipkaart.

Het belangrijkste resultaat van Hoofdstuk 4 is een uitbreiding van Kedlaya's algoritme [61] naar hyperelliptische krommen over een eindig lichaam van karakteristiek 2, wat voor toepassingen in de cryptografie het belangrijkste geval is. Dit algoritme is gebaseerd op Monsky-Washnitzer cohomologie en heeft een tijds- en geheugencomplexiteit die polynomiaal is in zowel het geslacht van de kromme als de uitbreidingsgraad van het eindig lichaam. De belangrijkste implicatie is dat het nu voor het eerst mogelijk is om willekeurige hyperelliptische krommen te genereren voor cryptografische doeleinden.

In Hoofdstuk 5 hebben we een totaal nieuw algoritme ontworpen om de zetafunctie van een C_{ab} kromme over elk eindig lichaam van kleine karakteristiek te berekenen. De tijds- en geheugencomplexiteit is polynomiaal in het geslacht van de kromme en de uitbreidingsgraad van het eindig lichaam. De theoretische achtergrond is eveneens Monsky-Washnitzer cohomologie, maar het algoritme is vrij verschillend van dat voor hyperelliptische krommen. Niettegenstaande dit algoritme zeer algemeen is, toont een implementatie in de `C` programmeertaal aan dat het efficiënt genoeg is voor cryptografische toepassingen.

Open Problemen

Enkele ideeën voor verder onderzoek zijn de volgende:

- In Hoofdstuk 3 toonden we aan dat Harley's algoritme het aantal punten op een elliptische kromme over een eindig lichaam $\mathbb{F}_{p^n}$ berekent in tijdscomplexiteit $O(n^2(\log n)^2 \log\log n)$ en geheugencomplexiteit $O(n^2)$ voor p constant. De vraag stelt zich dan ook of het mogelijk is om de complexiteit verder te verlagen? Echter, zolang het algoritme in twee gescheiden fasen werkt, i.e. bereken eerst een element $\alpha \in \mathbb{Z}_q$ en lees dan het spoor van Frobenius af als de norm van α, is het antwoord op deze vraag negatief. Een mogelijke oplossing zou zijn om de norm rechtstreeks te berekenen door bijvoorbeeld een deel van de minimaalveelterm van α te liften. Een verwante maar toch vrij verschillende vraag is de volgende: is het mogelijk om tegelijkertijd het aantal punten op $O(n)$ elliptische krommen over $\mathbb{F}_{p^n}$ te berekenen in tijd $O(n^2(\log n)^2 \log\log n)$? Deze vraag is revelant voor praktische toepassingen in de cryptografie; om een geschikte elliptische kromme te vinden moeten immers heel wat krommen getest worden.

- De algoritmen in Hoofdstukken 4 en 5 berekenen de zetafunctie van een kromme aan de hand van de karakteristieke veelterm van een Frobenius operator die inwerkt op de eerste Monsky-Washnitzer cohomologiegroep. Indien de kromme ordinair is dan zijn g wortels van deze karakteristieke veelterm eenheden in de maximale onvertakte uitbreiding $\mathbb{Z}_q^{\mathrm{ur}}$. De Weil conjecturen tonen aan dat de overige g wortels hierdoor bepaald zijn. De vraag stelt zich dan ook of het mogelijk is om voor een ordinaire kromme in een g-dimensionale vectorruimte te werken en enkel de g wortels die eenheden zijn in $\mathbb{Z}_q^{\mathrm{ur}}$ te berekenen?

- Het eerste deel van het algoritme in Hoofdstuk 5 werkt voor elke gladde affiene kromme. Kan ook het tweede deel veralgemeend worden naar willekeurige affiene krommen? Dit houdt in dat we een basis voor de eerste Monsky-Washnitzer cohomologiegroep moeten construeren en een reductie-algoritme moeten ontwerpen om ieder element in deze basis uit te drukken. Een mogelijke aanpak is de volgende. Zonder verlies aan algemeenheid kunnen we onderstellen dat de kromme gegeven is als een vlakke kromme met enkel singulariteiten op oneindig. Indien deze punten aanleiding geven tot m plaatsen op het niet-singulier model, dan is de dimensie van de eerste Monsky-Washnitzer cohomologiegroep gelijk aan $2g+m-1$. Het reductie-algoritme is dan equivalent met het vinden van een functie die de correcte poolorde heeft in één van deze plaatsen en een lagere poolorde in alle andere plaatsen. Een effectieve versie van de Riemann-Roch stelling kan gebruikt worden om zulke functies te construeren.

- Zij $\overline{C}$ een kromme van geslacht g gedefineerd over $\mathbb{F}_{p^n}$ en zij $\chi(t)$ de karakteristieke veelterm van Frobenius. Alle p-adische algoritmen berekenen $\chi(t) \bmod p^N$ voor een precisie
$$N \geq \left\lceil \log_p \left(2 \binom{2g}{g} q^{g/2} \right) \right\rceil .$$
De Weil conjecturen tonen aan dat dit $\chi(t)$ eenduidig bepaalt, maar laten de volgende vraag onbeantwoord. Gegeven $\chi(t) \bmod p^M$ voor een zekere precisie $M \leq N$, hoeveel isogenie klassen van Abelse variëteiten $A/\mathbb{F}_{p^n}$ zijn er zodanig dat de karakteristieke veelterm van Frobenius op A reduceert tot $\chi(t) \bmod p^M$. Een volledig antwoord op deze vraag zou dit aantal beschrijven als een functie van de parameters p, n, M, g. Indien er een $M < N$ zou bestaan zodat het aantal isogenie klassen $O(1)$ is, leidt dit onmiddellijk tot een geoptimaliseerd algoritme om $\chi(t)$ te berekenen. Waterhouse [135] toonde aan dat voor elliptische krommen alle mogelijke sporen van Frobenius effectief voorkomen; dit impliceert dat de gezochte M voor geslacht 1 niet bestaat. Voor krommen van hoger geslacht lijkt het echter waarschijnlijk dat M wel bestaat.

- Het grootste nadeel van alle gekende p-adische algoritmen is dat deze enkel efficiënt zijn voor krommen over eindige lichamen $\mathbb{F}_{p^n}$ van kleine karakteristiek p. Bovendien is het onwaarschijnlijk dat de tijdscomplexiteit van gelijkaardige algoritmen polynomiaal kan zijn in $\log p$. Een fundamenteel open probleem is dus om een praktisch algoritme te ontwerpen dat de zetafunctie van een willekeurige kromme berekent met een uitvoeringstijd die polynomiaal is in het geslacht van de kromme en de bit-lengte $n \log p$ van het eindig lichaam. De l-adische aanpak gebaseerd op torsiegroepen en divisieveeltermen leidt tot algoritmen die polynomiaal zijn in $n \log p$, maar exponentieel in het geslacht van de kromme.