Lukas Helminger BSc

The Elliptic Curve Discrete Logarithm Problem

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Mathematics

submitted to

Graz University of Technology

Supervisor

Ao.Univ.-Prof. Dipl.-Ing. Dr. Günter Lettl

Institut für Mathematik und
Wissenschaftliches Rechnen

Ass.-Prof. Mag. Dr.rer.nat Florian Kainrath

Graz, 03/2019

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Acknowledgments

I would like to thank my supervisors Professor Günther Lettel and Assistant Professor Florian Kainrath for their understanding, guidance and support. It also was Assistant Professor Florian Kainrath who gave the lecture “Algebraic Curves and Cryptography” which got me interested in elliptic curves.

I am deeply grateful to my parents, Julia and Franz Helminger. They did not only encouraged me to go to university but also backed me financially, so that I could fully concentrate on my studies.

I also want to thank my fiancee Birgit Ebster-Schwarzenberger for her support and love. She was the one who helped me through the tough periods in the writing process.

At last I want to acknowledge the help of Stefan Golja. His advice about layout and formulation improved the legibility of my thesis.

Abstract

More and more applications are using elliptic curve cryptography to exchange keys for encryption. The complexity-theoretic security relies on the assumption that finding the discrete logarithm is hard. This thesis is a comprehensive analysis of the most prominent algorithms for computing the discrete logarithm in elliptic curve groups including a quantum algorithm. The main emphases are the mathematical concepts behind these algorithms not their implementation.

Table of contents

1. Introduction

The theory of elliptic curves was at first developed with an analytic approach. Motivation for this new theory came from geometric and physics problems. Since the nineteenth century arithmetic and number theoretic questions arose. It could be shown that an elliptic curve can be seen as an abelian group. In the last years elliptic curves got into the focus of mathematicians due to the role they play in the proof of Fermat's Last Theorem. In 1995 Andrew Wiles could show with the help of elliptic curve theory that the theorem holds [33].

A practical purpose of elliptic curves is their usage in modern Public-key cryptography. The idea is the same as in the Diffie–Hellman key exchange protocol but instead of the multiplicative group of a prime field one takes the elliptic curve group as the underlying group structure. This concept was introduced in the middle of the 1980's independently by Victor S. Miller [20] and Neal Koblitz [15].

It turned out that the Elliptic-curve cryptography (ECC) could provide the same level of security afforded by the leading Public-key cryptosystem RSA (Rivest–Shamir–Adleman) with shorter key length [1]. Mostly for this reason we saw a shift from RSA to ECC. As a prominent example I want to mention the use of ECC in the National Security Agency Suite B Cryptography [6]. This is a set of cryptographic algorithms to encrypt classified information up to the level "Top Secret". The wide spread use of ECC in critical applications explains the extensive research undertaking in this field.

The security of ECC depends on the assumption that finding the discrete logarithm of a random elliptic curve element with respect to a public known base point is infeasible. More precise let P and Q be two given points on the same curve. Where Q is in $\langle P\rangle$ the group generated by P. Further let k be a non-zero integer with

$$Q = kP.$$

Then the discrete logarithm of Q with respect to the base element P is the number k.

The original goal of this master thesis was to work out the mathematical details omitted in the algorithms of the discrete logarithm presented in Annette Werner's book *Elliptische Kurven in der Kryptographie* [36]. Upon researching I came across the current progress made in the field of quantum computing. Therefore I decided to include Shor's quantum algorithm for computing the discrete logarithm [28] in my master thesis.

At the beginning we start with an introduction to elliptic curves and their properties. A special interest is taken in maps between curves. In particular the multiplication-by-m map for an integer m plays an important role in the definition of the Weil-Pairing. The computation of the Weil-Pairing is treated intensively in preparation of the MOV-algorithm.

The first algorithms we introduce are applicable to every abelian group. In that section we describe the algorithms only briefly compared with the following ones. Since these algorithms do not need a deep mathematical theory one gets a good understanding of them despite the rather compact description. Actually we look at one more attack against the discrete logarithm problem (DLP) which is not mentioned in Werner's book. The Index-Calculus algorithm is a sub-exponential algorithm, but it only works for the multiplicative group of a finite field.

We need the Index-Calculus as a part of the MOV-algorithm, our first algorithm specially designed to attack the DLP in the elliptic curve group. More precisely, the Weil-pairing helps us to transform our DLP in the elliptic curve group to a DLP in the multiplicative group of a finite field. For this new problem we can apply the Index-Calculus. We will see that in the case of supersingular elliptic curves the finite field and therefore the multiplicative group is small enough so that the whole MOV-algorithm is sub-exponential.

The second attack that exploits the properties of certain elliptic curves is the SSSA-algorithm. For anomalous curves one can find an isomorphism from the elliptic curve group to the additive group of the corresponding finite field. Then we only have to solve the DLP in a trivial setting. In order to be able to construct the isomorphism efficiently we use elliptic curves over the p-adic numbers. In the end this algorithm runs in polynomial time.

Shor's algorithm also runs in polynomial time but on a quantum computer. First we will give a short introduction into the basics of quantum computing. Then we describe the part where the speed up against conventional computers lies.

2. Elliptic Curves

In this chapter let K always be a field, whose characteristic is different from 2 and 3. Let $\bar{K}$ denote the algebraic closure of K and for every ring R we define R^* to be the ring without the zero element.

There are a few ways to define elliptic curves. A rather abstract approach is to say that an elliptic curve is a curve of genus one having a specified base point. Another way is to define elliptic curves as the set of solutions of a so called Weierstrass equation. This approach can be done in the affine plane, where we have to add a special point with certain properties or in the projective plane. We choose the later one, because we will need it in some computations, mainly in the section about the SSSA algorithm. Additionally it is easier to go to the affine case from the projective case as vice versa.

2.1. Projective Coordinates

Let K be a field, then the two dimensional projective space $\mathbb{P}^2(\bar{K})$ over $\bar{K}$ is the set of all one dimensional subspaces of $\bar{K}^3$. More explicitly, let $(x_1, y_1, z_1), (x_2, y_2, z_2) \in \bar{K}^3$, then we define an equivalence relation on $\bar{K}^3$ as follows:

$$(x_1, y_1, z_1) \sim (x_2, y_2, z_2) :\Leftrightarrow \exists \lambda \in \bar{K}^\times : (x_1, y_1, z_1) = (\lambda x_2, \lambda y_2, \lambda z_2).$$

Then $\mathbb{P}^2(\bar{K})$ is given by the equivalence classes (w.r.t $\sim$) of triplets $(x, y, z) \in \bar{K}^3$, where at least one coordinate is not zero, i.e.

$$\mathbb{P}^2(\bar{K}) := \left(\bar{K}^3 \backslash \{0\}\right) / \sim .$$

For a point $(x, y, z) \in \bar{K}^3$, the equivalence class is denoted by $(x : y : z)$. There is a natural way to embed the affine plane $\bar{K}^2$ in the projective space. We have the inclusion $(x, y) \mapsto (x : y : 1)$. Therefore we call every point $(x : y : z) \in \mathbb{P}^2(\bar{K})$, with $z \neq 0$ finite and all others, points at infinity.

Whether a projective point in $\mathbb{P}^2(\bar{K})$ is a root of a given polynomial $F(X, Y, Z) \in K[X, Y, Z]$ does depend on the representative chosen for this particular point. To fix this issue we only look at homogeneous polynomials. A homogeneous polynomial of degree n in $K[X, Y, Z]$ is a polynomial, where each term $cX^jY^kZ^l$, where $c \in K$, satisfies $j + k + l = n$. Let $F \in K[X, Y, Z]$ be a homogeneous polynomial of degree n and let $(x_1, y_1, z_1) = (\lambda x_2, \lambda y_2, \lambda z_2)$ with $\lambda \in \bar{K}^\times$. Then

$$F(x_1, y_1, z_1) = F(\lambda x_2, \lambda y_2, \lambda z_2) = \lambda^n F(x_2, y_2, z_2).$$

In particular if (x_1, y_1, z_1) is a zero of F, $(\lambda x_2, \lambda y_2, \lambda z_2)$ is a zero of F, i.e. whether or not the coordinates of a projective point evaluate to zero in a homogeneous polynomial does not depend on the representatives chosen.

Definiton 2.1. *An elliptic curve E over the field K is the set of solutions of an equation of the form*

$$Y^2Z = X^3 + aXZ^2 + bZ^3 \tag{2.1}$$

where $a, b \in K$, with the discriminant $\Delta. = -16(4a^3 + 27b^3) \neq 0$, i.e.

$$E = \{(x : y : z) \in \mathbb{P}^2(\bar{K}) \mid y^2 z = x^3 + axz^2 + bz^3\}.$$

Equations of the type (2.1) are called Weierstrass equations.

We are now interested, which points on an elliptic curve do not lie in the affine plane, i.e. are points at infinity. Let $(x : y : z) \in \mathbb{P}^2(\bar{K})$ be a projective point. Remember a point at infinity is one with $z = 0$. Then we have $x^3 = 0$, implying $x = 0$. Since not all three coordinates are allowed to be 0, the only condition for y is to be unequal to 0. Hence $(0 : y : 0) = (0 : 1 : 0) = O$ is the only possible point at infinity on an elliptic curve. Since plugging in $(0 : 1 : 0)$ in the equation (2.1) yield $0 = 0$, we see that O is actually a point on the elliptic curve.

One defines a projective line as the set of solutions of a homogeneous polynomial of degree 1 and a arbitrary projective curve C_g as the zero set of a reduced homogeneous polynomial g. It can be shown that different projective lines intersect in exactly one point and that the intersection of a projective line with an elliptic curve always consists of 3 points (counted with multiplicity), see Bezout theorem [7, 5.3].

Often it is not necessary to compute in the projective plane, if this is the case we switch to the affine plane where the corresponding Weierstrass equation has the form

$$E : y^2 = x^3 + ax + b.$$

We just have to remember to add the point at infinity O to the set of solutions of this equation. The rational points of E are all points with coordinates in K together with the point O, i.e.

$$E(K) = \{O\} \cup \{(x, y) \in K \times K \mid y^2 = x^3 + ax + b\}.$$

Below we have two drawings of the real part of two elliptic curves:

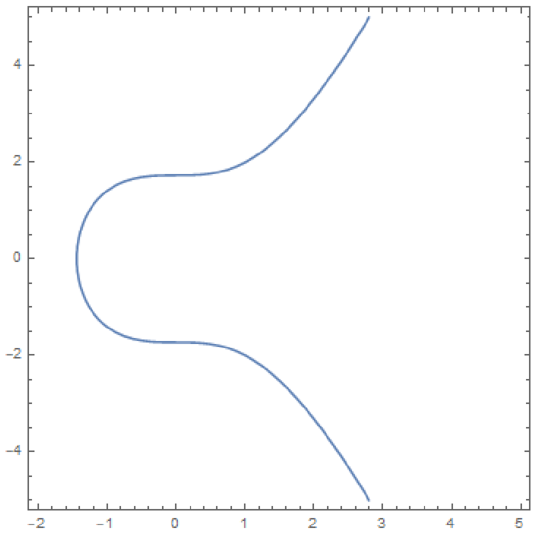

Figure 1: $E : y^2 = x^3 + 3$

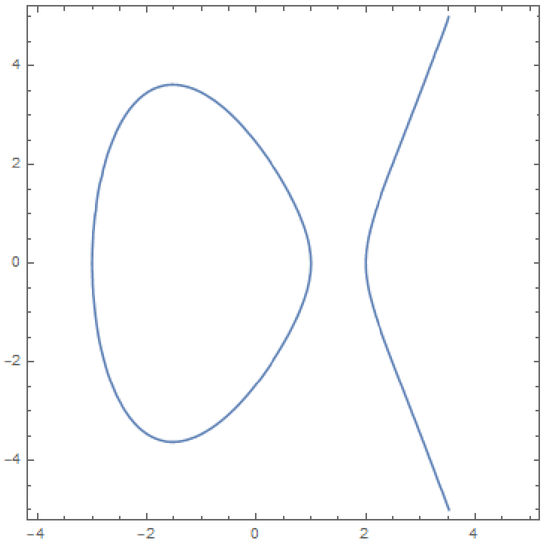

Figure 2: $E : y^2 = x^3 - 7x + 6$

Remark: The condition that the discriminate has to differ from zero is equivalent to the condition that the curve E has no singular points [30, III.1.4.], i.e. for every point $P \in E \backslash \{0\}$

$$\frac{-3x^2 - a}{\partial x}(P) \neq 0 \quad \text{or} \quad \frac{2y}{\partial y}(P) \neq 0.$$

2.2. The Group Law

One reason why elliptic curves are interesting for cryptography is, that they can be equipped with a group operation. Unfortunately the component-wise addition of points is not closed under the set of points of a given elliptic curve, as the next example shows.

Example 2.2. Let $E : y^2 = x^3 - 18x + 8$ be an elliptic curve over $\mathbb{R}$. If we add the two points $P = (4, 0)$ and $Q = (7, 15)$ component-wise, we obtain the point $R = (11, 15)$. Plugging R into the given Weierstrass equation gives us the contradiction $121 = 3113$. So, the point R is not on the elliptic curve E.

As it was the case with the definition of elliptic curves, there are again many ways to describe the addition law on elliptic curves. We choose to give an intuitive geometric definition, because it is sufficient for the algorithms we will present later. In this definition we see the importance of the point O.

Definiton 2.3. *Let P and Q be two points on the elliptic curve E. Further, let L be the projective line connecting P and Q, provided that P and Q are distinct. Otherwise, let L be the tangent line of E at P. The intersection of L and E, taken with multiplicities, consists of exactly three points. This is due to Bezout theorem [7, 5.3]. P and Q are by construction two of them. We take the third point of intersection of L and E to be R. Set L' to be the projective line through R and O. Then L' intersects R, O and a third point. We denote the third point by $P \oplus Q$ (in the (x, y)-plane this is just the reflection of R across the x-axis).*

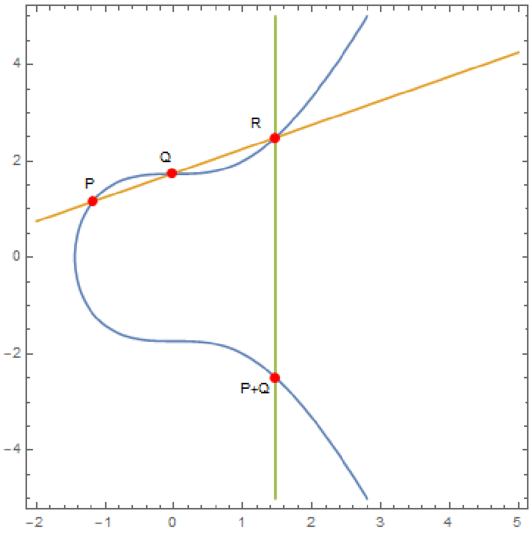

Figure 3: Case: P and Q distinct

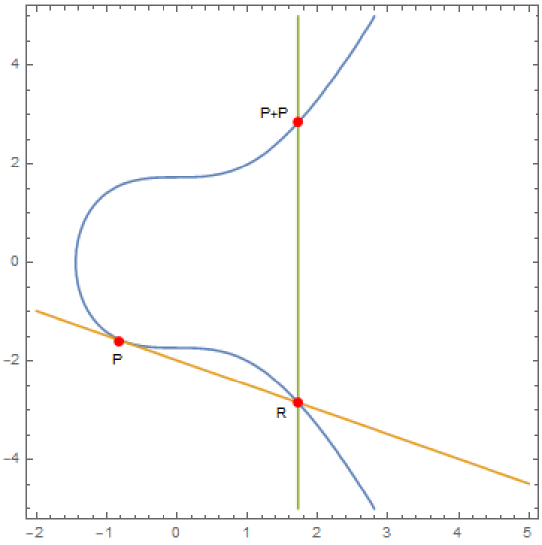

Figure 4: Doubling the point P

Theorem 2.4. *Let E be an elliptic curve over K. Then*

$$(E, \oplus) \text{ is an abelian group.}$$

Proof. We will use the notation of definition 2.3.
First of all, the operation $\oplus$ is commutative, since the geometric construction is symmetric in P and Q.
The neutral element is O. Let $P \in E$. If we try to compute $P \oplus O$, we see that the lines L and L' coincide. Therefore the intersection point of L', besides R and O is again P.
For O it is clear that O itself is the inverse element, because O is the neutral element. The inverse element of an element $P \in E \backslash \{O\}$ is $P' = R$. To see this, we add these two points. The projective line connecting P and R is L, which means the third point is O. We look at the tangent of L' of E at O. Let F be the homogeneous polynomial corresponding to the Weierstrass equation (2.1). Then the partial derivatives of E at O are

$$\frac{\partial F}{\partial X}(0,1,0) = \frac{\partial F}{\partial Y}(0,1,0) = 0 \quad \text{and} \quad \frac{\partial F}{\partial Z}(0,1,0) = 1.$$

Hence the tangent is given through the equation $Z = 0$. Therefore the third intersection point of L' and E is O. This shows $P \oplus P' = O$.
By far the most challenging point to prove is the associativity of $\oplus$. This can be done directly with the formulas below, though one has to make several case distinctions and go through tedious calculations [34, 2.4]. An algebraic proof using the Riemann-Roch theorem can be found in [30, III.3.4.(e)]. □

Since we now know that $\oplus$ is a group operation on E, we write $(E, +)$ instead of $(E, \oplus)$. For adding points of an elliptic curve in the affine plane we can derive explicit formulas, as shown in the next theorem.

Theorem 2.5. *Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over K, and let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2) \in E \backslash \{O\}$ be finite points. Next, let $(x_3, y_3) = P_3 := P_1 + P_2$. Then*

$$P_3 = \begin{cases} O & \text{if } x_1 = x_2 \text{ and } y_1 = -y_2 \\ (\lambda^2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1) & \text{otherwise,} \end{cases}$$

where λ is defined by

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } x_1 \neq x_2 \\ \frac{3x_1^2 + a}{2y_1} & \text{otherwise.} \end{cases}$$

Proof. In the proof of theorem 2.4 we already saw that P_2 is exactly the inverse of P_1 in the case $x_1 = x_2$ and $y_1 = -y_2$.
If $x_1 \neq x_2$, then λ is the slope of the line L through P_1 and P_2. The equation of the line is $L : y = \lambda x + y_1 - \lambda x_1$. Substituting this into the equation for E, gives us

$$(\lambda x + (y_1 - \lambda x_1))^2 = x^3 + ax + b.$$

Therefore,

$$\begin{aligned} 0 &= x^3 - \lambda^2 x^2 + (a - 2\lambda(y_1 - \lambda x_1))x + b + \lambda x_1 - y_1 \\ &= (x - x_1)(x - x_2)(x - x_3). \end{aligned}$$

We see that $-\lambda^2 = -x_1 - x_2 - x_3$, so $x_3 = \lambda^2 - x_1 - x_2$.
To get the y-coordinate of the third intersection point of L and E we plug x_3 into L, and obtain $\lambda x_3 - \lambda x_1 + y_1 = \lambda(x_3 - x_1) + y_1$. By reflecting this point across the x-axis, we have computed $P_3 = (\lambda^2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1)$. The case $x_1 = x_2$ follows along the same lines, one only has to consider that now x_1 is a double root of the cubic polynomial. □

Example 2.6. Let $E : y^2 = x^3 - 7x + 6$ be an elliptic curve over $\mathbb{R}$ and let $P_1 = (3, 2\sqrt{3}), P_2 = (5, 4\sqrt{6}) \in E$ be two points on the curve. We want to compute $P_3 = (x_3, y_3)$. To do this we use the formulas given in the last theorem.

$$\begin{aligned}
x_3 &= \left(\frac{4\sqrt{6} - 2\sqrt{3}}{5 - 3}\right)^2 - 3 - 5 = \frac{16 \cdot 6 - 16\sqrt{18} - 4 \cdot 3}{4} - 8 \\
&= \frac{96 - 48\sqrt{2} + 12}{4} - 8 = 24 - 12\sqrt{2} + 3 - 8 \\
&= 19 - 12\sqrt{2} \\
y_3 &= \frac{4\sqrt{6} - 2\sqrt{3}}{5 - 3}\left(3 - \left(19 - 12\sqrt{2}\right)\right) - 2\sqrt{3} = \left(2\sqrt{6} - \sqrt{3}\right)\left(-16 + 12\sqrt{2}\right) - 2\sqrt{3} \\
&= -32\sqrt{6} + 24\sqrt{12} + 16\sqrt{3} - 12\sqrt{6} - 2\sqrt{3} \\
&= 62\sqrt{3} - 44\sqrt{6}
\end{aligned}$$

So, $(3, 2\sqrt{3}) + (5, 4\sqrt{6}) = (19 - 12\sqrt{2}, 62\sqrt{3} - 44\sqrt{6})$.

2.3. Torsion Points

The interesting thing about torsion point for us is, how the Weil-pairing acts on them and how this can be used to attack the ECDLP, like in the MOV-algorithm.

Definiton 2.7. *(a) Let E be an elliptic curve over K, and let m be an integer. The multiplication-by-m map $[m] : E \to E$ is defined for $P \in E$ as follows*

$$[m]P := \begin{cases} \overbrace{P + \cdots + P}^{m \text{ terms}} & m > 0 \\ O & m = 0 \\ \underbrace{-P - \cdots - P}_{-m \text{ terms}} & m < 0 \end{cases}.$$

(b) Let $n \in \mathbb{N}$. The set of n-torsion points of the group E is denoted by

$$E[n] = \{P \in E : [n]P = O\}.$$

Note that this set is the kernel of the multiplication-by-n map.

Example 2.8. Let again $E : y^2 = x^3 - 7x + 6$ be an elliptic curve over $\mathbb{R}$. We want to determine the set of points of order two $E[2]$.
Since O is the neutral element in E, it is in the set $E[2]$. So, let $P \in E[2] \setminus \{O\}$ be arbitrary. From $[2]P = O$, we know that O lies on the tangent of E at P. Let

$aX + bY + cZ = 0$ be the equation defining the tangent. Since O is on this projective line, we get $b = 0$ and therefore the tangent is is vertical in the affine plane. This implies that the y-coordinate of P must be 0. To get the remaining points in $E[2]$, we now have to solve the cubic equation $0 = x^3 - 7x + 6$. By doing this we obtain

$$E[2] = \{O, (-3, 0), (1, 0), (2, 0)\}.$$

Actually the torsion points form more than just a set. Let $P, Q \in E[n]$ for a given elliptic curve and $n \in \mathbb{N}$. Then the set of n-torsion points is closed under addition

$$[n](P+Q) = (P+Q)+\cdots+(P+Q) = P+\cdots+P+Q+\cdots+Q = [n]P+[n]Q = O+O = O,$$

and also contains the inverse elements

$$[n](-P) = -P - \cdots - P = [-1](P + \cdots + P) = [-1][n]P = [-1]O = -O = O.$$

Therefore $E[n]$ is a subgroup of E, called the n-torsion group. In some cases we know the structure of the group $E[n]$. Obviously the multiplication-by-n map plays an important role in finding the structure of $E[n]$. The crucial fact is the following one.

Theorem 2.9. *Let E be an elliptic curve over K, and let $n \in \mathbb{N}$ such that char(K) $\nmid n$. Then*

$$\#\ker[n] = \#E[n] = n^2.$$

An elementary proof using division polynomials is given in [34, 3.2]. There is also a proof using more theory, in [30, II.6.4.].

Theorem 2.10. *Let E be an elliptic curve over K.*

(a) Let $n \in \mathbb{N}$, and suppose char(K) $= 0$ or char(K) $\nmid n$. Then the torsion group is a product of two cyclic groups

$$E[n] \cong \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}.$$

(b) Let char(K) $= p > 0$, then

$$E[p^e] \cong \{0\},\ \textit{for all } e \geq 1 \quad \textit{or} \quad E[p^e] \cong \mathbb{Z}/p^e\mathbb{Z},\ \textit{for all } e \geq 1.$$

Proof. (a) From theorem 2.9 we obtain

$$\#E[n] = n^2.$$

Further, for every integer d dividing n, we similarly have $\#E[d] = d^2$.
The structure theorem for finite abelian groups tells us that $E[n]$ is isomorphic to

$$\mathbb{Z}/n_1\mathbb{Z} \times \cdots \times \mathbb{Z}/n_k\mathbb{Z} \quad \text{for some integers} \quad n_1, \ldots, n_k \text{ with } n_i \mid n_{i+1} \quad \text{for all } i.$$

Let d be a prime dividing n_1, then d divides all n_i. This means that $E[d] \subset E[n]$ has order d^k. But we know from above $E[d]$ has order d^2, so $k = 2$. Then it is easy to see that the only possibility is $E[n] = \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$.
(b) See [30, III.6.4.(c)]. □

In the case (a), where $E[n] \cong \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ we can view $E[n]$ as a module over the ring $\mathbb{Z}/n\mathbb{Z}$. In particular, when n is prime, then $E[n]$ has a structure as a 2-dimensional vector space over the field $\mathbb{Z}/n\mathbb{Z}$. Even if n is not prime, we always can find a basis existing of two elements; i.e. there exists $P, Q \in E[n]$ such that every point $R \in E[n]$ in the torsion group can be written in the form $R = cP + dQ$ for some integers c and d.

2.4. Divisors & Miller's Algorithm

The at first rather abstract concept of a divisor of a curve turns out to be very practical for the efficient computation of the Weil-pairing, which again is at the heart of the MOV-attack. Before we can define a divisor associated to a function we have to have a look at rational functions on a curve.

2.4.1. Rational Functions

Let E be an elliptic curve over the field K. Then we define $\bar{K}[E] := \{u : E\backslash\{O\} \to \bar{K} : \exists g \in \bar{K}[x, y] : u(P) = g(P)\ \forall P \in E\backslash\{O\}\}$. One can show that $\bar{K}[E]$ is an integral domain. The field of rational functions, denoted by $\bar{K}(E)$, is the fraction field of this integral domain. In [30, II.1.1.] it is shown that for every finite point on the curve $P \in E\backslash\{O\}$ the ring

$$\bar{K}(E)_P = \{f \in \bar{K}(E) \mid \exists g_1, g_2 \in \bar{K}[E] : f = \frac{g_1}{g_2} \wedge g_2(P) \neq 0\}$$

is a discrete valuation ring. If we take the embedding $(x : z) \mapsto (x : 1 : z)$, the point $O = (0 : 1 : 0)$ has the affine coordinate $(0, 0)$. One can show that the field of rational functions in this coordinate system is isomorphic to the one we already defined, via the map $x \mapsto x/y, z \mapsto 1/y$. So every point of the curve has a discrete valuation ring. These are all discrete valuation rings containing $\bar{K}$ [9, 1.6]. Then we know from commutative algebra that there exists an element $u \in \bar{K}(E)_P$ such that for every rational function $f \in \bar{K}(E)^*$ there is an unique integer $s \in \mathbb{Z}$, so that we can write f in the form $f = u^s g$ for some $g \in \bar{K}(E)_P^\times$. We call s the order of f at P and write $\mathrm{ord}_P(f) = s$. If s is negative we say f has a pole at P and when s is positive we say f has a zero at P.

2.4.2. Divisors

Now we have all we need to define divisors of functions.

Definiton 2.11. *Let E be an elliptic curve over K. A divisor D on the curve E is a formal sum*

$$D = \sum_{P \in E} n_P(P), \tag{2.2}$$

with $n_P \in \mathbb{Z}$ and $n_P = 0$, for all but finitely many points $P \in E$. We denote the abelian group of all divisors on E by Div(E). *The degree and the sum of the divisor D (2.2) is defined by*

$$\deg(D) = \sum_{P \in E} n_P \qquad \textit{and} \qquad \mathrm{sum}(D) = \sum_{P \in E} [n_P]P.$$

Let $f \in \bar{K}(E)$ be a rational function, then the associated divisor of f, denoted by $\operatorname{div}(f)$*, is the divisor*

$$\operatorname{div}(f) = \sum_{P \in E} ord_P(f)(P).$$

A divisor D is called principal if there exists a rational function $f \in \bar{K}(E)$ so that the associated divisor of f is D, i.e. $\operatorname{div}(f) = D$.

In order to be a divisor $\operatorname{div}(f)$ has to be a finite sum. This is actually the case and is shown in [7, Ch.8,Prop.1]. When seeing these definitions one could wonder which divisors on a curve are associated divisors of a rational function, or what it says for two functions to have the same associated divisor. The answer of the latter question is given in the next proposition.

Proposition 2.12. *Let E be an elliptic curve over K and $f \in \bar{K}(E)^*$. Then*

(a)

$$\operatorname{div}(f) = 0 \quad \Leftrightarrow \quad f \textit{ is constant, i.e. } f = c, \textit{ for some } c \in \bar{K}^*. \tag{2.3}$$

In particular, if $\operatorname{div}(f) = \operatorname{div}(g)$*, for some $g \in \bar{K}(E)^*$, then f and g only differ by a constant, i.e. $f = gc$, for some $c \in \bar{K}^*$.*

(b) Every principal divisor has degree 0, i.e.

$$\deg(\operatorname{div}(f)) = 0.$$

Proof. See [30, II.3.1.,II.3.7.]. □

To be able to answer the first question we need an immediate result, which will also be helpful in the construction of Miller's algorithm.

Lemma 2.13. *Let $P_1, P_2 \neq O$ be two finite points on an elliptic curve $E : y^2 = x^3 + ax + b$ over K. Further, let l be the equation of the line (in $\bar{K}(E)$) connecting P_1 and P_2, and let v be the vertical equation of the line (in $\bar{K}(E)$) through $P_3 = P_1 + P_2$ (if $P_3 = O$, then set $v = 1$). Then*

(a)

$$\operatorname{div}\left(\frac{l}{v}\right) = (P_1) + (P_2) - (P_3) - (O).$$

(b)

$$\operatorname{sum}\left(\operatorname{div}\left(\frac{l}{v}\right)\right) = O.$$

Proof. (a) Obviously P_1, P_2, and $-P_3$ are zeros of the line. From Bézout we know that the line l intersects E at exactly three points (counted with multiplicity), so there can not be additional zeros. The only pole occurs at the point of infinity. Hence, $\operatorname{div}(l) = (P_1) + (P_2) + (-P_3) - n(O)$, for some $n \in \mathbb{N}$. By 2.12(b) the degree of $\operatorname{div}(l)$ must be 0. Therefore,

$$\operatorname{div}(l) = (P_1) + (P_2) + (-P_3) - 3(O).$$

For the line v one argues in a similar way and obtains $\operatorname{div}(v) = P_3 + (-P_3) - 2(O)$. Note, because of the definition of associated divisors through the order (which is a valuation), the divisor of a quotient of two functions is the difference of the the divisors of the functions. Therefore,

$$\begin{aligned}\operatorname{div}\left(\frac{l}{v}\right) &= \operatorname{div}(l) - \operatorname{div}(v) = (P_1) + (P_2) + (-P_3) - 3(O) - ((P_3) + (-P_3) - 2(O)) \\ &= (P_1) + (P_2) - (P_3) - (O),\end{aligned}$$

where the right-hand side simplifies to $(P_1) + (P_2) - 2(O)$, if $v = 1$.
(b) If $v = 1$ the statement is clear and in the general case the sum is

$$\operatorname{sum}\left(\operatorname{div}\left(\frac{l}{v}\right)\right) = P_1 + P_2 - P_3 = P_1 + P_2 - (P_1 + P_2) = O.$$

□

Theorem 2.14. *Let E be an elliptic curve over K and D be a divisor on E of degree 0. Then*

$$D \text{ is principal} \quad \Leftrightarrow \quad \operatorname{sum}(D) = O.$$

Proof. We can write D in the following form

$$D = (P_1) + \cdots + (P_r) - (Q_1) - \cdots - (Q_s) + m(O),$$

for finite points $P_1, \ldots, P_r, Q_1, \ldots, Q_s, \in E, r, s \in \mathbb{N}$ and $m \in \mathbb{Z}$. If we now inductively use 2.13(a) on the positive terms $P_1, \ldots, P_r$, we obtain

$$D = (P) - (Q_1) - \cdots - (Q_s) + m'(O) + \operatorname{div}(g),$$

for some $P \in E, m' \in \mathbb{Z}$ and $\operatorname{sum}(\operatorname{div}(g)) = O$. That $\operatorname{sum}(\operatorname{div}(g)) = O$ is clear for $r = 2$, by Lemma 2.13(b) and then it is just a induction argument. Basically the principal divisors get added, which means the functions are multiplied. The sum of the product of functions, whose sums are O is again O. We can do a similar thing with $Q_1, \ldots Q_s$ and get D in the simple form

$$D = (P) - (Q) + m''(O) + \operatorname{div}(h),$$

for some $Q \in E, m'' \in \mathbb{Z}$ and $\operatorname{sum}(\operatorname{div}(h)) = O$. From 2.12 we know that $\deg(\operatorname{div}(h)) = 0$. Hence $\deg(D) = 1 - 1 + m''$ and since D has degree 0 the only possibility is that $m'' = 0$.
We have $\operatorname{sum}(\operatorname{div}(h)) = O$. Therefore,

$$\operatorname{sum}(D) = P - Q.$$

Now suppose $\operatorname{sum}(D) = O$, then $O = P - Q$, hence $P = Q$ and therefore $D = \operatorname{div}(h)$ is a principal divisor.
On the other hand let $D = \operatorname{div}(f)$ for some $f \in \bar{K}(E)$ be a principal divisor, then $(P) - (Q) = \operatorname{div}(f/h)$. To show that this can only be the case when $P = Q$ one can use a consequence from the Riemann-Roch theorem, as in [30, III.3.3.]. An elementary but long proof can be found in [34, 11.1]. □

2.4.3. Miller's Algorithm

When we want to efficiently compute the Weil-pairing it all comes down to finding a function f such that

$$\operatorname{div}(f) = n(P+R) - n(R), \tag{2.4}$$

for points $P \in E[n], R \in E$, and then evaluating $f(Q_1)/f(Q_2)$ for two points Q_1 and Q_2. The proofs of the last two statements 2.13 and 2.14 were quite constructive. We can use them to calculate the associated function to a principal divisor, as the next example demonstrates.

Example 2.15. Let $E : y^2 = x^3 + x$ be an elliptic curve over $\mathbb{Z}/7\mathbb{Z}$ and let

$$D = ((0,0)) + ((5,5)) + ((1,4)) + ((3,4)) - 4(O)$$

be a divisor on E. Obviously $\deg(D) = 0$ and a look at the group structure (see example 3.8) shows us that $\operatorname{sum}(D) = O$. So there exists a rational function f such that $\operatorname{div}(f) = D$.

First we note that the line connecting the points $(0,0)$ and $(5,5)$ is given through the equation $0 = y - x$. The third intersection point of the curve with this line is $(3,3)$. By lemma 2.13 we have

$$\begin{aligned}\operatorname{div}\left(\frac{y-x}{x-3}\right) &= ((0,0)) + ((5,5)) - ((0,0)+(5,5)) - (O)\\ &= ((0,0)) + ((5,5)) - ((3,4)) - (O).\end{aligned}$$

Therefore,

$$D = \operatorname{div}\left(\frac{y-x}{x-3}\right) + ((3,4)) + ((1,4)) + ((3,4)) - 3(O). \tag{2.5}$$

The line connecting $(1,4)$ and $(3,4)$ is given through $0 = y - 4$. To apply 2.13(a) we have to compute $(1,4) + (3,4)$. By looking at the tangent to E at $(3,4)$ we see that the line $y = 4$ intersects E at $(3,4)$ with multiplicity 2:

$$\begin{aligned}t(x,y) &= \frac{\partial(x^3+x-y^2)}{\partial x}(3,4)(x-3) + \frac{\partial(x^3+x-y^2)}{\partial y}(3,4)(y-4)\\ &= 28(x-3) - 8(y-4)\\ &= 6(y-4) = 6y - 24\\ &= -y + 4 \qquad\qquad \text{in } \mathbb{Z}/7\mathbb{Z}.\end{aligned}$$

Hence,

$$\begin{aligned}\operatorname{div}\left(\frac{y-4}{x-3}\right) &= ((1,4)) + ((3,4)) - ((1,4)+(3,4)) - (O)\\ &= ((1,4)) + ((3,4)) - ((3,3)) - (O).\end{aligned}$$

Inserting this identity to (2.5) gives us

$$\begin{aligned} D &= \operatorname{div}\left(\frac{y-x}{x-3}\right) + ((3,4)) + \operatorname{div}\left(\frac{y-4}{x-3}\right) + ((3,3)) - 2(O) \\ &= \operatorname{div}\left(\frac{y-x}{x-3}\right) + \operatorname{div}\left(\frac{y-4}{x-3}\right) + \operatorname{div}(x-3) \qquad (2.13(a) \text{ with } P_3 = O) \\ &= \operatorname{div}\left(\frac{(y-x)(y-4)}{x-3}\right). \end{aligned}$$

We can rewrite the numerator by making use of the relation of y and x, given through the equation defining the elliptic curve E, in the following way

$$\begin{aligned} (y-4)(y-x) &= y^2 - xy - 4y + 4x \\ &= x^3 + x - xy - 4y + 4x \\ &= x^3 + 5x - xy - 4y \\ &= (x-3)(x^2 + 3x - y). \end{aligned}$$

Finally we get $D = \operatorname{div}(x^2 + 3y - y)$.

This method works fine as long as the number of points in the divisor is relative small. In our problem (2.4) we would have to do a lot of iterations, if n is a large number. To speed up the computation Miller's algorithm uses successive doubling. We now give a description of this procedure based on [34, 11.4].

```
Miller's Algorithm:
Input:  $n \in \mathbb{N}, P \in E[n], R \in E, Q_1, Q_2 \in E \setminus (\{O, (P+R), R\} \cup \{[j]P \mid 1 \leq j \leq n\})$.
Output: $f(Q_1)/f(Q_2)$ such that $\operatorname{div}(f) = n(P+R) - n(R)$.

(1) Define $v_j := f_j(Q_1)/f_j(Q_2)$,
    where $f_j \in \bar{K}(E)$ such that $\operatorname{div}(f_j) = j(P+R) - j(R) - ([j]P) + (O) := D_j$.

(2) Set $i = n, j = 0, k = 1, f_0 = 1$.
    Compute $f_1$, as in lemma 2.13 or in example 2.15, so that $\operatorname{div}(f_1) = D_1$.

(3) As long as $i \neq 0$, do
    Define $l$ and $v$ such that $\operatorname{div}(l/v) = ([j]P) + ([k]P) - ([j+k]P) - (O)$.
    (This can be done as in lemma 2.13)

    (a) If $i \equiv 0 \pmod 2$, then
        compute $v_{2k} = v_k v_k \frac{l(Q_1)v(Q_2)}{l(Q_2)v(Q_1)}$, set $k \leftarrow 2k$ and save $(v_j, v_k)$.
        $i \leftarrow i/2$

    (b) If $i \equiv 1 \pmod 2$, then
        compute $v_{j+k} = v_j v_k \frac{l(Q_1)v(Q_2)}{l(Q_2)v(Q_1)}$, set $j \leftarrow j + k$ and save $(v_j, v_k)$.
        $i \leftarrow i - 1$

(4) Output $v_k$.
```

If the steps (1) to (3) are performed correctly, then it is clear that the output $v_k = v_n = f_n(Q_1)/f_n(Q_2)$ is the desired result. To check the correctness of the algorithm, we have a look at step (1) and (3). Since

$$\deg(D_j) = j - j - 1 + 1 = 0 \quad \text{and} \quad \operatorname{sum}(D_j) = [j](P+R) - [j]R - [j]P + O = O,$$

D_j is a principal divisor by 2.14 and therefore the function f_j exists for every $j \in \{1, \ldots, n\}$. Until now, it is not clear how we can do step (3) efficiently. Suppose we can compute $f_{j+k}(Q_1)/f_{j+k}(Q_2)$ directly from $f_j(Q_1)/f_j(Q_2)$ and $f_k(Q_1)/f_k(Q_2)$. This would make the steps (3)(a) and (3)(b) efficient, since in case (a) we simply set $j = k$. Assume now we already know the values $f_j(Q_1)/f_j(Q_2)$ and $f_k(Q_1)/f_k(Q_2)$. Then

$$\begin{aligned}
\operatorname{div}(f_{j+k}) &= D_{j+k} = (j+k)(P+R) - (j+k)(R) - ([j+k]P) + (O) \\
&= \underbrace{j(P+R) - j(R) - ([j]P) + (O)}_{=D_j} + \underbrace{k(P+R) - k(R) - ([k]P) + (O)}_{=D_k} \\
&\quad + \underbrace{([j]P) + ([k]P) - ([j+k]P) - (O)}_{=\operatorname{div}(l/v),\text{ with } l \text{ and } v \text{ as in lemma 2.13}} \\
&= D_j + D_k + \operatorname{div}(l/v) \\
&= \operatorname{div}\left(f_j f_k \frac{l}{v}\right).
\end{aligned}$$

Lemma 2.12(a) tells us that the functions f_{j+k} and $f_j f_k(l/v)$ only differ by a unit. Therefore,

$$\frac{f_{j+k}(Q_1)}{f_{j+k}(Q_2)} = \frac{f_j(Q_1)f_k(Q_1)l(Q_1)v(Q_2)}{f_j(Q_2)f_k(Q_2)l(Q_2)v(Q_1)}.$$

This result shows how step (3) is done efficiently.

Example 2.16. Let again $E : y^2 = x^3 + x$ be an elliptic curve over $(\mathbb{Z}/7\mathbb{Z})^\times$. We write down Miller's algorithm for the following input:
Input: $n = 4, P = (1,4) \in E[4], R = (5,5) \in E$
$Q_1 = (3,3), Q_2 = (3,4) \in E \setminus (\{O, (5,2), (5,5)\} \cup \{(1,4), (0,0), (1,3), O\})$.
Output: $f(3,3)/f(3,4)$ such that $\operatorname{div}(f) = 4((5,2)) - 4((5,5))$.

(2) Set $i = 4, j = 0, k = 1$ and $f_0 = 1$, then we compute f_1 such that $\operatorname{div}(f_1) = ((5,2)) - ((5,5)) - ((1,4)) + (O)$ with 2.13(a)
$P_1 = (1,4)$ and $P_2 = (5,5)$ then $P_3 = (5,2)$. The line l is given through the equation $y = 1 \cdot 4^{-1}x + 15 \cdot 4^{-1} = 2x + 2$ and the line v is given through the equation $x - 5$. Then we get

$$\operatorname{div}(f_1) = \operatorname{div}\left(\frac{x-5}{y-2x-2}\right).$$

Therefore

$$v_1 = \frac{3-5}{3-6-2}\bigg/\frac{3-5}{4-6-2} = \frac{5}{2}\bigg/\frac{5}{3} = 6 \cdot 2 = 12 = 5.$$

(3) $\operatorname{div}(l/v)$ is 1 since $([0](1,4)) + (([1](1,4)) - ([0+1](1,4)) - (O)$ is the trivial divisor. We are in case (a) because $4 \equiv 0 \bmod 2$, then $v_2 = v_1 \cdot v_1 = 5 \cdot 5 = 25 = 4$. Set $k = 2$ and $i = 4/2 = 2$. (3) $\operatorname{div}(l/v)$ is 1 since $([0](1,4)) + (([2](1,4)) - ([0+2](1,4)) - (O)$ is the trivial divisor.
We are in case (a) because $2 \equiv 0 \bmod 2$, then $v_4 = v_2 \cdot v_1 = 4 \cdot 4 = 16 = 2$. Set $k = 2$ and $i = 2/2 = 1$. (3) $\operatorname{div}(l/v)$ is 1 since $([0](1,4)) + (([4](1,4)) - ([0+4](1,4)) - (O)$ is

the trivial divisor.
We are in case (b) because $1 \equiv 1 \bmod 2$, then $v_4 = v_4 \cdot v_0 = 2 \cdot 1 = 2$. Set $j = 4$ and $i = 1 - 1 = 0$.
(4) Output 2.

2.5. Weil-Pairing

In the context of the Weil-pairing, let the characteristic of K never divide the integer n. Let f be a rational function and $D = \sum_{P \in E} n_P(P)$ a divisor on an elliptic curve E, with $\operatorname{div}(f)$ and D having disjoint support. If we apply f to D, we actually mean

$$f(D) = f\left(\sum_{P \in E} n_P(P)\right) := \prod_{P \in E} f(P)^{n_P}.$$

Definiton 2.17. *Let E be an elliptic curve over K. For two point $P, Q \in E[n]$ let D_P and D_Q be divisors of degree 0 such that*

$$\operatorname{sum}(D_P) = P \quad \textit{and} \quad \operatorname{sum}(D_Q) = Q$$

and such that D_P and D_Q have disjoint support. Then the divisors nD_P and nD_Q are principal. Hence, there exist functions f_P and f_Q such that $\operatorname{div}(f_P) = nD_P$ and $\operatorname{div}(f_Q) = nD_Q$. The Weil-pairing e_n is defined as follows

$$\begin{aligned} e_n : E[n] \times E[n] &\longrightarrow \bar{K} \\ (P, Q) \quad &\longmapsto e_n(P, Q) = \frac{f_Q(D_P)}{f_P(D_Q)}. \end{aligned}$$

The Weil-pairing has a few nice properties, as we see in the following proposition.

Proposition 2.18. *Let E be an elliptic curve over K and let $P, P_1, P_2, Q, Q_1, Q_2 \in E[n]$ if not otherwise stated. Then the Weil e_n-pairing has the following properties:*

(a) It is bilinear:

$$e_n(P_1 + P_2, Q) = e_n(P_1, Q)\, e_n(P_2, Q),$$
$$e_n(P, Q_1 + Q_2) = e_n(P, Q_1)\, e_n(P, Q_2).$$

(b) It is alternating: $e_n(Q, Q) = 1$, so in particular $e_n(P, Q) = e_n(Q, P)^{-1}$.

(c) It is nondegenerate: If $e_n(P, Q) = 1$ for all $P \in E[n]$, then $Q = O$.

(d) It is Galois invariant: $e_n(P, Q)^\sigma = e_n(P^\sigma, Q^\sigma)$ for all $\sigma \in G_{\bar{K}/K}$.

(e) It is compatible: $e_{nn''}(P, Q) = e_n([n']P, Q)$ for all $P \in E[nn']$ and $Q \in E[n]$.

A proof can be found in every standard book about elliptic curves, as in [30, III.8.1.] or [34, 11.7]. We want to mention that for these proofs one needs an alternative definition

of the Weil-pairing. The image of the Weil-pairing in the alternative definition is $\mu_n = \{\alpha \in \bar{K} \mid \alpha^n = 1\}$ the group of nth roots of unity (important for the corollary below). To show that these two definitions of the Weil-pairing are equal, we would need a lot more theory about abelian varieties. It is proven in [10, Sec.1] with a reference to [17, Sec.6.4].
From the properties of the Weil-pairing we can derive an important result for the MOV-algorithm.

Corollary 2.19. *Let E be an elliptic curve over a perfect field K and let $P \in E$ be a point of order n. Then there exists a point $Q \in E[n]$ such that $e_n(P,Q)$ is a primitive nth root of unity. In particular, if $E[n] \subset E(K)$, then $\mu_n \subset K$.*

Proof. Consider the map

$$\begin{aligned} e_n(P,\cdot) : E[n] &\longrightarrow \mu_n \\ Q &\longmapsto e_n(P,Q). \end{aligned}$$

Suppose to the contrary that this map is not surjective, otherwise we would be done. Since μ_n is a cyclic group, the image of the map must also be a cyclic group μ_d, with $d < n$. But then, for every $Q \in E[n]$:

$$1 = e_n(P,Q)^d \overset{2.18(a)}{=} e_n([d]P,Q).$$

The non-degeneracy of the e_n-paring implies that $[d]P = O$, and it follows that $n = d$, a contradiction to $d < n$.

Finally, if $E[n] \subset E(K)$, then the Galois invariance of the e_n-pairing implies

$$\zeta := e_n(P,Q) = e_n\left(P^\sigma, Q^\sigma\right) = \left(e_n(P,Q)\right)^\sigma = \sigma\left(\zeta\right).$$

Hence $\mu_n \subset K$. □

2.6. Elliptic Curves over Finite Fields

In cryptography one only uses elliptic curves over finite fields. Therefore we have a closer look on these special curves. Let E be an elliptic curve over a finite field $\mathbb{F}_q$. When we are talking about finite fields, let q always be a prime power, unless otherwise mentioned and let $\mathbb{F}_q$ has characteristic greater then 3. We wish to estimate the number of points of $E(\mathbb{F}_q)$, or equivalently, one more than the number of solutions to the equation

$$E : y^2 = x^3 + ax + b \qquad \text{with } (x,y) \in \mathbb{F}_q \times \mathbb{F}_q.$$

Since each value of x yields two values for y at most, a trivial upper bound is $2q+1$. However, since a "randomly chosen" quadratic equation has a 50% chance of being solvable in $\mathbb{F}_q$, we expect that the right order of magnitude should be q.
The next result was conjectured by E. Artin in his thesis and proven by Hasse in the 1930s. It shows that the heuristic reasoning from above is correct.

Theorem 2.20 (Hasse). *Let E be an elliptic curve over the finite field $\mathbb{F}_q$, for some prime power q. Then*

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

Proof. See [30, V.1.1] or [34, 4.2]. □

2.6.1. Supersingular Curves

A special class of elliptic curves over finite fields are supersingular curves. From a computational side of view they have interesting properties. In general we do know much more about their group structure than about other elliptic curves.

Definiton 2.21. *An elliptic curve E over a finite field $\mathbb{F}_q$ is called supersingular, if* $\operatorname{char}(\mathbb{F}_q)$ *divides* $t = q + 1 - \#E(\mathbb{F}_q)$.

In particular, if we assume $q \geq 5$ is a prime, then the condition above is equivalent to saying t must be 0. This is a consequence of Hasse's theorem 2.20. To see this, assume to the contrary that E is supersingular but $t \neq 0$. From the definition of supersingular curves we know that t is a multiple of q. So together we get that $|t| \geq q$. But from Hasse's theorem we get the additional inequality $|t| \leq 2\sqrt{q}$. These two inequalities only hold for $q < 5$.

There is an even better way to characterize supersingular curves, but to see this we have to do a short digression about finite field arithmetic. The generalization of the Legendre symbol for finite fields is the following

$$\left(\frac{x}{\mathbb{F}_q}\right) = \begin{cases} 1 & \text{if } x = a^2 \text{ for some } a \in \mathbb{F}_q^\times \\ 0 & \text{if } x = 0 \\ -1 & \text{otherwise} \end{cases},$$

for every x in the finite field $\mathbb{F}_q$. The values $-1, 0, 1$ are integers. The numbers of solution of a quadratic equation can be very elegantly be represented through the Legendre symbol. In our setting this translates to

$$\#E(\mathbb{F}_q) = q + 1 + \sum_{x \in \mathbb{F}_q} \left(\frac{x^3 + ax + b}{\mathbb{F}_q}\right).$$

The next lemma lists three results about computations in finite fields, which we will need for proving our characterization of supersingular curves.

Lemma 2.22. *Let $\mathbb{F}_q$ be a finite field. Then*

(a)

$$\left(\frac{x}{\mathbb{F}_q}\right) = x^{\frac{q-1}{2}} \qquad \text{in } \mathbb{F}_q.$$

(b) Let $n \in \mathbb{N}$. Then

$$\sum_{x \in \mathbb{F}_q} x^n = \begin{cases} 0 & \text{if } q - 1 \nmid n \\ -1 & \text{otherwise} \end{cases}.$$

(c) Let $f(x) \in \mathbb{F}_q[x]$ be a cubic polynomial (char($\mathbb{F}_q$) $= p \geq 5$) and let a_{p^r} be the coefficient of x^{p^r-1} in $f^{(p^r-1)/2}$, for every $r \in \mathbb{N}$. Then

$$a_{p^r} = a_p^{1+p+\cdots+p^{r-1}}.$$

Proof. The proofs of all three results do not involve advanced methods. Nevertheless we do not show them here and refer for (a) and (b) to [36, 3.4.3] and for (c) to [34, 4.34]. □

This ends our digression about finite fields and we are able to prove the following theorem from [34, 4.32].

Theorem 2.23. *Let $p \geq 5$ be a prime and let E be an elliptic curve given by $y^2 = x(x-1)(x-\lambda)$ with $\lambda \in \overline{\mathbb{F}}_p$. We define the polynomial*

$$H_p(T) = \sum_{i=0}^{(p-1)/2} \binom{(p-1)/2}{i}^2 T^i.$$

Then

$$E \textit{ supersingular } \Leftrightarrow H_p(\lambda) = 0.$$

Proof. By an elementary fact of finite fields ([18]) $\lambda \in \mathbb{F}_q$, where q is a prime power of p. Hence, E is an elliptic curve defined over the finite field $\mathbb{F}_q$. To use our definition of supersingularity we have to compute the number of rational points on the curve $\#E(\mathbb{F}_q)$. We can formalize the heuristic reasoning, which we did before Hasse's theorem. The generalized Legendre symbol can be used to count how many solutions an quadratic equation has in a given finite field. So,

$$\begin{aligned} \#E(\mathbb{F}_q) &= \sum_{x \in \mathbb{F}_q} \left(\frac{x(x-1)(x-\lambda)}{\mathbb{F}_q} \right) + q + 1 \quad \text{in } \mathbb{F}_q \\ &\overset{2.22(a)}{=} \sum_{x \in \mathbb{F}_q} (x(x-1)(x-\lambda))^{\frac{q-1}{2}} + q + 1 \quad \text{in } \mathbb{F}_q. \end{aligned}$$

If we expand the polynomial, we see that the only term x^n with $q-1 \mid n$ is x^{q-1}. Let a_q be the coefficient of this term. By 2.22(b),

$$\#E(\mathbb{F}_q) = -a_q + 1 \quad \text{in } \mathbb{F}_q.$$

Form 2.22(c), we see that a_q is 0 if and only if a_p is 0, and in this case the curve E is supersingular. Now we calculate the coefficient a_p of x^{p-1}. To get a_p we have a look at each linear factor of the polynomial. Since the factor x always has coefficient 1, we look for the coefficient of $x^{(p-1)/2}$ in the rest of the factors:

$$\begin{aligned} (x-1)^{\frac{p-1}{2}} &= \sum_{k=0}^{\frac{p-1}{2}} \binom{\frac{p-1}{2}}{k} x^k (-1)^{\frac{p-1}{2}-k} \\ (x-\lambda)^{\frac{p-1}{2}} &= \sum_{j=0}^{\frac{p-1}{2}} \binom{\frac{p-1}{2}}{j} x^{\frac{p-1}{2}-j} (-\lambda)^j. \end{aligned}$$

Putting together this information, we get

$$\begin{aligned} a_p &= (-1)^{\frac{p-1}{2}} \sum_{n=0}^{\frac{p-1}{2}} \binom{\frac{p-1}{2}}{n}^2 \lambda^n \\ &= (-1)^{\frac{p-1}{2}} H_p(\lambda). \end{aligned}$$

So, E is supersingular $\Leftrightarrow a_p = 0 \Leftrightarrow H_p(\lambda) = 0$. □

Example 2.24. We now check whether our already known elliptic curve $E : y^2 = x^3 + x$ over the finite field $\mathbb{F}_7$ is supersingular. Following the proof technique of 2.23 we compute the coefficient a_7 of x^{7-1} in $(x^3 + x)^{(7-1)/2}$. This is the same as the coefficient of x^3 in $(x^2 + 1)^3$. The term x^3 does not appear in $(x^2 + 1)^3$, because all the exponents are even. Hence, E is supersingular over $\mathbb{F}_7$.

The following theorem tells us that for supersingular curves the quantity $t = q + 1 - \#E(\mathbb{F}_q)$ can only take a few special values. A proof can be found in [35, 4.1].

Theorem 2.25. *Let E be a supersingular curve over $\mathbb{F}_q$, then $t = q + 1 - \#E(\mathbb{F}_q)$ takes one of the following values*

$$0, \pm\sqrt{q}, \pm\sqrt{2q}, \pm\sqrt{3q}, \pm 2\sqrt{q}.$$

Our assumption that the characteristic of $\mathbb{F}_q$ is greater than 3 excludes the cases $\pm\sqrt{2q}, \pm\sqrt{3q}$, since theses values can not be integers. Note that if E is supersingular over $\mathbb{F}_q$, then E is also a supersingular over F_{q^l} [36, 3.4.2].

Proposition 2.26. *Let E be a supersingular elliptic curve over $\mathbb{F}_q$ and $t = q + 1 - \#E(\mathbb{F}_q)$. Then $E[n] \subset E(\mathbb{F}_{q^l})$, if l is chosen according to the table below. The number d to the corresponding l is the exponent of the group $E(\mathbb{F}_{q^l})$, i.e. the smallest natural number d such that $[d]R = O$ for all $R \in E(\mathbb{F}_{q^l})$.*

t	0	$\pm\sqrt{q}$	$\pm\sqrt{2q}$	$\pm\sqrt{3q}$	$\pm 2\sqrt{q}$
l	2	3	4	6	1
d	$q+1$	$\sqrt{q^3} \pm 1$	q^2+1	q^3+1	$\sqrt{q} \mp 1$

Proof. See, [19, Table 1]. □

3. Elliptic Curve Cryptography

3.1. Introduction into Cryptography

Since ancient times there has been a need to encrypt messages. For most of the time only ambassadors and the military tried to hide the content of their communication. The beginning of the information age let the number of users of cryptography rise dramatically. Nowadays nearly everybody, both consciously and unconsciously, encrypts his or her messages. When we use online banking it is clear to us that the information exchanged is somehow protected against fraud. On the other side the normal user is not aware that in messaging applications, like "What's App", the end-to-end encryption is always activated. In fact there is no possibility to turn off the end-to-end encryption. "What's App" is not using some lightweight cryptosystem. For instance the key exchange is done with elliptic curve cryptography (ECC). Before we explain how ECC works, we have to give some terminology, notations and definitions of the field of cryptology, especially its sub field cryptography.

Cryptology is the science of keeping information secret. A part of cryptology is stenography, where one tries to conceal the information. For example the Greek sometimes wrote their messages on the shaved skull of slaves. Then they waited until the hair was grown again and send the slaves to the receiver of the messages. If they would have got caught by the enemy, they would have had no visible messages with them. In this thesis we do not deal with this type of cryptology. Instead we are interested in cryptography and cryptanalysis. The former is the science of encrypting texts so that only the intended receiver is able to decrypt the text. Cryptanalysis is the development and application of methods to break cryptosystems. It is also used as a tool to scrutinize the security of existing systems. Mathematically speaking a cryptosystem is a quintuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ with the following properties:

1. $\mathcal{P}, \mathcal{C}, \mathcal{K}$ are non-empty finite sets, where $\mathcal{P}$ is the plaintext space, $\mathcal{C}$ is the ciphertext space and $\mathcal{K}$ is the key space.

2. $\mathcal{E} = \{e_k : k \in \mathcal{K}\}$ is a family of functions such that each e_k is a injective function from the plaintext space $\mathcal{P}$ to the ciphertext space $\mathcal{C}$. These maps are called encryption functions.

3. $\mathcal{D} = \{d_k : k \in \mathcal{K}\}$ is a family of functions such that there exists a subset $\mathcal{C}' \subset \mathcal{C}$ of the ciphertext space, where every function d_k from $\mathcal{C}'$ to the plaintext space $\mathcal{P}$ is injective. These maps are called decryption functions.

4. The following relation between the encryption functions $\mathcal{E}$ and the decryption functions $\mathcal{D}$ holds:
$$\forall k \in \mathcal{K}\ \exists k' \in \mathcal{K} :\ d_{k'} \circ e_k(p) = p \quad \wedge \quad e_k \circ d_{k'}(c) = c,$$
for all $p \in \mathcal{P}$ and $c \in \mathcal{C}'$.
A cryptosystem is called asymmetric if it is "computationally infeasible" to derive k' from k, otherwise it is called symmetric.

Example 3.1. (Shift cipher)
First we identify the alphabet (only small letters) with the underlying set of the quotient ring $\mathbb{Z}/26\mathbb{Z} = \{\bar{0}, \bar{1}, \ldots, \bar{25}\}$. We take the plaintext space, the ciphertext space and the key space to be equal to $\mathbb{Z}/26\mathbb{Z}$, so $\mathcal{P}, \mathcal{C}, \mathcal{K} = \mathbb{Z}/26\mathbb{Z}$ are non-empty finite sets. For a key $k \in \mathcal{K}$, the encryption function corresponds to a shift of k letters in the alphabet, using modular arithmetic (i.e. the calculation is done in the ring $\mathbb{Z}/26\mathbb{Z}$)

$$e_k(p) = p + k \;\; \forall p \in \mathcal{P}.$$

The decryption function for k corresponds to the negative shift

$$d_k(c) = c - k \;\; \forall c \in \mathcal{C}.$$

Obviously all functions $\mathcal{E} = \{e_k : k \in \mathcal{K}\}$ are injective. For the maps $\mathcal{D} = \{d_k : k \in \mathcal{K}\}$ we can take $\mathcal{C}' = \mathcal{C}$ and all decryption functions are still one-to-one. To show that the required relation between these two families of functions holds, let $k' = k$ and let $c \in \mathcal{C}$ arbitrarily. Then

$$d_{k'} \circ e_k(p) = (p + k) - k' = p \quad \wedge \quad e_k \circ d_{k'}(c) = (c - k') + k = c.$$

So we have shown that the shift cipher is a cryptosystem. Since encryption and decryption use the same key, it is symmetric. For example, Julius Caesar used the shift cipher with the key $k = 3$ for his private correspondence.

There had also been many attempts to keep the cryptosystem itself secret. All such attempts of hiding the system failed, because a cryptosystem is a big secret and every party using it, has to know it. In 1883 Auguste Kerckhoffs wrote about principles for military ciphers. Every modern cryptosystem is based on the Kerckhoffs's principle. The following short reformulation of the original french statement can be found in [23].

Kerckhoffs's principle: *The security of a cryptosystem must not depend on the secrecy of the system used. Rather, the security of a cryptosystem may depend only on the secrecy of the keys used.*

The first goal of cryptography was to provide confidentiality, i.e. that only authorized people are able to read the message. Modern day cryptography has three additional goals:

- authentication,
- data-integrity and
- non-repudiation.

A comprehensive description of these terms can be found in [14]. It is not necessary that one system accomplishes all four tasks. Often a cryptosystem is designed for a single one of them and used with others to cover the desired goals.

Roughly speaking cryptography can be divided into three major epochs. At first the encryption and decryption was done by hand (e.g. shift cipher). In the next epoch one used (electro)-mechanical machines (e.g. Enigma machine) for this task and today we use a computer (e.g. ECC). What the first two epochs have in common is, that all their cryptosystems are symmetric. To understand why there was a need for asymmetric encryption in the computer era, we look at a popular illustration involving padlocks.

When talking about cryptosystems it is a convention to call the communicating entities Alice and Bob and an eavesdropper Eve. Say Alice wants to send a package to Bob, but she is afraid that Eve might intercept the package and opens it. In the symmetric case Alice buys a padlock with the corresponding key. She meets with Bob in advance and hands over the key to him. If she now wants to send Bob a package, she attaches the padlock and nobody except Bob can open it. This concept is secure, but has one drawback - the key delivery. As long as only a limited number of people secured their communication this was fine. At the latest with the upcoming of the internet such a key delivery system was logistically not doable any more.

To solve the key distribution problem asymmetric cryptography, also known as public-key-cryptography, was developed in the 1970's. Back in our setting with a package and a padlock the concept works as follows. First Bob sends Alice an open padlock (public-key) and keeps the key (private-key). Note he does not have to meet Alice, because the padlock itself does not reveal any secret information. Alice then attaches the padlock, which she received form Bob, to the package and sends it to Bob. He can open the padlock with his key. Eve has no chance to open the package, because the key always stayed with Bob.

All prominent examples of public-key-cryptography depend on a certain type of mathematical functions. We give an intuitive definition of one-way functions, a rigorous definition is given in [23].

Definiton 3.2. *Let X and Y be sets and let $f : X \to Y$ be a function. We say f is a one-way function if it has the following properties:*

- *$f(x)$ is easy to compute, for all $x \in X$.*
- *It is computationally infeasible to find any $x \in X$ such that $f(x) = y$, for a y in the image of f.*

Until today it is only conjectured that one-way functions exist (would imply P≠NP). A good candidate is the multiplication of prime numbers. Computing $p \cdot q = n$ for two prime numbers p and q is easy. But the inverse operation, finding p and q, given n, called factorization, is believed to be hard (not computable in polynomial time).
If Alice would apply a one-way function to encrypt her message, nobody could decrypt it in reasonable time. Therefore she must give Bob a hint, how he can invert the function and thus read the message of Alice. This issue is resolved via trapdoor one-way functions.

Definiton 3.3. *Let $f : X \to Y$ be a one-way function. We call f a trapdoor one-way function if it becomes computationally feasible to invert f with some additional information.*

Squaring modulo a positive integer is a candidate for a trapdoor one-way function. For given integers x and n it is easy to compute $y = x^2 \bmod n$. But again, the inverse operation (finding x, given y and n) is believed to be hard. The computation gets feasible if we know the factorization $n = p \cdot q$, for two prime numbers p and q, provided y is co-prime to n. Because now we can compute x mod n by computing x mod p, x mod q and combining the results with the Chinese Reminder Theorem (CRT).

3.2. Discrete Logarithm in Cryptography

We will describe the discrete logarithm for finite cyclic groups and how it is used to secure information. Elliptic curve cryptography (ECC) is then a special case, where we take the cyclic group to be a cyclic subgroup of a given elliptic curve group.

Definiton 3.4. *Let G be a finite cyclic group. We write G in multiplicative notation and let 1 be the identity element. Further let $g \in G$ be a generator of G. Then we define the following function for G and g*

$$\begin{aligned} \exp_{G,g} : \mathbb{Z} &\longrightarrow G \\ n &\longmapsto g^n := \underbrace{g \cdots g}_{n \text{ times}} . \end{aligned}$$

The discrete logarithm in G to the base g is defined as the inverse function of $\exp_{G,g}$.

The function $\exp_{G,g}$ is supposed to be a one-way function. Whitfield Diffie and Martin E. Hellman in 1976 published a scheme for key-agreement, today called the Diffie-Hellman key agreement (DH), in the paper New Directions in Cryptography [3]. The security of this scheme is based on the believed one-way function $\exp_{G,g}$. The original idea was developed for the multiplicative group of a finite field $\mathbb{F}_p$, where p is a prime number. This idea can easily be generalized for finite cyclic groups.

Diffie-Hellman Key Agreement (DH):

1. Alice and Bob agree on a finite cyclic group G and a generator g of G (both are public).

2. Alice chooses a secret integer m, and Bob chooses a secret integer n.

3. Alice computes $a := g^m \in G$. Bob computes $b := g^n \in G$.

4. They exchange their newly computed group elements a and b.

5. Alice computes $k_1 := b^m = (g^n)^m = g^{nm}$.
   Bob computes $k_2 := a^n = (g^m)^n = g^{mn}$.

Since every cyclic group is abelian, we have $k_1 = k_2$ and therefore Alice and Bob have the same key. We demonstrate the DH by an example in the multiplicative group of a finite field.

Example 3.5.

1. Alice and Bob agree on $G = (\mathbb{Z}/11\mathbb{Z}, \cdot)$ and $g = \bar{2}$.
2. Alice chooses $m = 3$ and Bob chooses $n = 7$ as their secret integers.
3. $a = \bar{2}^3 = \bar{8}$, $b = \bar{2}^7 = \bar{7}$.
4. $k = \begin{cases} \bar{7}^3 = \bar{2} \\ \bar{8}^7 = \bar{2} \end{cases}$

The key-agreement is done on an insecure communication channel. So we have to assume that Eve has all the information, which Alice and Bob exchanged. In this case she saw the values a and b and knows G and g. To get the key on which Alice and Bob agreed on, she has to compute g^{nm}, given the values g^m and g^n. The only known way to do this is to solve the following problem.

Discrete Logarithm Problem (DLP):
Given a finite cyclic group, $g \in G$ a generator and $a \in G$ arbitrarily, computing $x \in \mathbb{Z}$ such that $g^x = a$.

Example 3.6. For n a positive integer let $G = (\mathbb{Z}/n\mathbb{Z}, +)$ be the residual classes modulo n together with the addition, and $\bar{g}$ a generator of the group. In this case the DLP has the form $x\bar{g} = \bar{a}$, for some $\bar{a} \in \mathbb{Z}/n\mathbb{Z}$. The $gcd(n, g)$ is 1, because $\bar{g}$ is a generator. Applying Bézout's identity we get

$$1 = x_1 g + yn, \quad \text{, for some } x_1, y \in \mathbb{Z}.$$

Multiplying by x gives us $x = xx_1g + xyn$, which is equivalent to $x - (xx_1)g = xyn$, i.e. $x \equiv ag \bmod n$, when we set $a := xx_1$. Therefore calculating x_1 is enough to solve the DLP in this special group.

Since all cyclic groups of a given order are isomorphic, we see that the hardness of the DLP problem does not depend as much on the structure of the group as it depends on the way we can describe the group (and its operation). In practice certain elliptic curve groups are used. They provide the same security level as other public-key systems (e.g. RSA), but with a smaller key size [1]. Before we describe the DH and the DLP in the context of elliptic curves, we explain a protocol for asymmetric encryption, which is also based on the hardness of the DLP. It was described by Taher Elgamal in 1985 [5].

ElGamal Encryption

1. Bob chooses a secret integer d as his private key. His public key is (G, g, e), where G is a finite cyclic group, g a generator of G and $e = g^d$.
2. Alice chooses a secret integer r and computes $c_1 = g^r$.
3. Alice computes $c_2 = pe^r$, where $p \in G$ is the plaintext.
4. Alice sends (c_1, c_2) to Bob.

5. Bob decrypts the message by computing

$$c_2 c_1^{-d} = pe^r \cdot g^{-rd} = pg^{dr} g^{-rd} = p.$$

Next we do an example of the ElGamal encryption in the multiplicative group of a finite field.

Example 3.7.

1. Bob chooses $d = 5$ as his private key. His public key is $((\mathbb{Z}/11\mathbb{Z}, \cdot), \bar{2}, \bar{2}^5 = \bar{10})$.

2. Alice chooses $r = 9$ and computes $c_1 = \bar{2}^9 = \bar{6}$.

3. Alice computes $c_2 = \bar{4} \cdot \bar{10}^9 = \bar{7}$, where $\bar{4}$ is the plaintext.

4. Alice sends $(\bar{6}, \bar{7})$ to Bob.

5. Bob decrypts the message by computing

$$\bar{7} \cdot \bar{6}^{-5} = \bar{7} \cdot \bar{2}^5 = \bar{4}.$$

3.3. Elliptic Curve Cryptography

The algorithms in this section are only special cases of the ones we saw in the last section.

Elliptic Curve Diffie-Hellman Key Agreement (ECDH):

1. Alice and Bob agree on an elliptic curve E over a finite field $\mathbb{F}_q$ and a point $P \in E(\mathbb{F}_q)$ (both are public).

2. Alice chooses a secret integer m, and Bob chooses a secret integer n.

3. Alice computes $A := [m]P \in \langle P \rangle$. Bob computes $B := [n]P \in \langle P \rangle$.

4. They exchange their newly computed group elements A and B.

5. Alice computes $K_1 := [m]B = [mn]P$.
   Bob computes $K_2 := [n]A = [nm]P$.

Let $E : y^2 = x^3 + x$ be an elliptic curve over $\mathbb{Z}/7\mathbb{Z}$. Just by brute force we get that the elliptic curve group looks as follows

$$E(\mathbb{Z}/7\mathbb{Z}) = \{(0,0), (1,3), (1,4), (3,3), (3,4), (5,2), (5,5), O\}.$$

With a little more calculating effort we get that $P = (3,3)$ is a generator of the group.

n	0	1	2	3	4	5	6	7
nP	O	$(3,3)$	$(1,4)$	$(5,5)$	$(0,0)$	$(5,2)$	$(1,3)$	$(3,4)$

This table will speed up our computation in the next two examples.

Example 3.8. (ECDH)

1. Alice and Bob agree on the elliptic curve defined by the Weierstrass equation $E : y^2 = x^3 + x$ over $\mathbb{Z}/7\mathbb{Z}$, with $P = (3,3)$ as a generator of the elliptic curve group.

2. Alice chooses $m = 5$, and Bob chooses $n = 3$.

3. Alice computes $A := [5](3,3) = (5,2)$. Bob computes $B := [3](3,3) = (5,5)$.

4. They exchange their newly computed group elements A and B.

5. Alice computes $K_1 := [5](5,5) = (3,4)$.
   Bob computes $K_2 := [3](5,2) = (3,4)$.

We now formulate the DLP for elliptic curves.

Elliptic Curve Discrte Logarithm Problem (ECDLP):
Given an elliptic curve E over $\mathbb{F}_q$, $P \in E(\mathbb{F}_q)$ and $Q \in \langle P \rangle$ arbitrarily, computing $k \in \mathbb{Z}$ such that $[k]P = Q$.

As in the general case the asymmetric cryptosystem based on the hardness of the ECDLP is always used together with a symmetric one (e.g. AES). This is necessary, because public-key cryptosystems are in general much slower than symmetric ones. The solution is a hybrid cryptosystem, where the symmetric key is exchanged or encrypted via a asymmetric protocol and the message is encrypted by a symmetric protocol.

Elliptic Curve ElGamal Encryption

1. Bob chooses a secret integer d as his private key. His public key is $(E(\mathbb{F}_q), P, P_e)$, where $E(\mathbb{F}_q)$ is an elliptic curve over $\mathbb{F}_q$, $P \in E(\mathbb{F}_q)$ and $P_e = [d]P$.

2. Alice chooses a secret integer r and computes $C_1 = [r]P$.

3. Alice computes $C_2 = M + [r]P_e$, where $M \in E(\mathbb{F}_q)$ is the message.

4. Alice sends (C_1, C_2) to Bob.

5. Bob decrypts the message by computing

$$C_2 - [d]C_1 = M + [r]P_e - [dr]P = M + [rd]P - [dr]P = M.$$

Example 3.9. (Elliptic Curve ElGamal Encryption)

1. Bob chooses $d = 2$ as his private key. His public key is $(E(\mathbb{Z}/7\mathbb{Z}), (3, 3), [2](3, 3) = (1, 4))$, where the elliptic curve is again defined through the equation $E : y^2 = x^3 + x$.

2. Alice chooses $r = 3$ and computes $C_1 = [3](3, 3) = (5, 5)$.

3. Alice computes $C_2 = (0, 0) + [3](1, 4) = [4](1, 3) + [3]([2](1, 3)) = (1, 4)$, where $(0, 0)$ is the message.

4. Bob decrypts the message by computing

$$(1, 4) - [2](5, 5) = [2](1, 3) - [2 \cdot 3](1, 3) = (0, 0).$$

4. General Attacks

Before we go to algorithms that use the special structure of elliptic curve groups, we explain the most common algorithms to compute the discrete logarithm in finite cyclic groups, except the last one, which only works for multiplicative groups of a finite field. This algorithm will be useful in the MOV-algorithm.
For the next four algorithms we work in the setting of the DLP, with the additional requirement that x should be the smallest positive integer satisfying

$$g^x = a.$$

In this section n will always be the order of the group.

4.1. Babystep-Giantstep

Set $m = \lceil \sqrt{n} \rceil$, with the Euclidean algorithm we can write $x = qm + r$, for some $q \in \mathbb{Z}$ and $r \in \{0, \dots, m-1\}$. This implies $a = g^x = g^{qm+r}$ which is equivalent to

$$ag^{-r} = g^{qm}.$$

Now the idea is to compute all possible values of the left hand side and save the results in a list called the baby steps. In the next step we go through all values on the right hand side (giant steps) until there is a match in the baby steps.
Assuming one can access the list in constant time (e.g. hash table) it is not difficult to show that BGSG has running time and space complexity of $O(\sqrt{n})$ [2].

Example 4.1. Let $G = (\mathbb{Z}/31\mathbb{Z})^\times$, then $m = \lceil \sqrt{30} \rceil = 6$. We want to compute the discrete logarithm of $\overline{14}$ to the base $\overline{3}$. The baby steps are:

r	0	1	2	3	4	5
$\overline{14} \cdot \overline{3}^{-r}$	$\overline{14}$	$\overline{15}$	$\overline{5}$	$\overline{12}$	$\overline{4}$	$\overline{22}$

Now we compute the giant steps and compare them to our list of baby steps.

q	0	1	2	3	4	5
$\overline{3}^{6q}$	$\overline{1}$	$\overline{16}$	$\overline{8}$	$\overline{4}$	$\overline{2}$	$\overline{1}$

We have a match between the 4th giant step and the second last baby step. Therefore the solution to the DLP is $x = 3 \cdot 6 + 4 = 22$.

4.2. Pohlig-Hellman

Let $n = \prod_{i=1}^{m} p_i^{e_i}$ be the prime factorization of n. Instead of computing x modulo n this algorithm computes x modulo each prime power of n. Once we know these values, due to the Chinese Reminder Theorem, we are done.

Given a prime power p^e of n, we want to find $z \in \{0, \ldots, p^e - 1\}$ such that $z \equiv x \bmod p^e$. To solve this congruence Pohlig-Hellman uses the p-adic expansion of z

$$z = z_0 + z_1 p + \cdots + z_{e-1} p^{e-1}$$

with $z_i \in \{0, \ldots, p-1\}$. The algorithm computes the coefficients recursively. Each coefficient z_i is a solution of a DLP in a subgroup of G. More precisely we want to find $b \in G$ of order p and $a_i \in \langle b \rangle$ such that $a_i = b^{z_i}$. The DLP in $\langle b \rangle$ is solved with the BSGS-algorithm (4.1).

The group element $b := g^{n/p}$ is clearly of order p. The initial value for the recursion is $a_0 = a$. From this we get $a^{n/p} = g^{x(n/p)} = b^x = b^z = b^{z_o}$. Assume z_{i-1} has already been computed, then

$$a_i := \left(a g^{-\left(\sum_{j=0}^{i-1} z_j p^j\right)} \right)^{\frac{n}{p^{i+1}}}.$$

Since $z \equiv x \bmod p^e$, we have $a^{n/p^{i+1}} = g^{(nx)/p^{i+1}} = g^{(nz)/p^{i+1}}$, which implies

$$a_i = \left(g^{z - \sum_{j=0}^{i-1} z_j p^j} \right)^{\frac{n}{p^{i+1}}} = g^{\frac{n}{p^{i+1}} \sum_{j=i}^{e-1} z_j p^j} = b^{z_i}.$$

The complexity of this algorithm is $O\left(\sum_{i=1}^{m} \left(e_i \left(\log n + \sqrt{p_i}\right)\right)\right)$, see [2, 11.5.4].

4.3. Pollard's-ρ

To understand why Pollard's-ρ algorithm is correct, one uses a standard result of congruence relations. A proof of this statement can be found in nearly every introductory book about number theory.

Let a and b be integers and let n be a positive integer. We set $d := \gcd(a, n)$. The congruence relation

$$ax \equiv b \pmod{n} \tag{4.1}$$

has a solution, if and only if d divides b. If $s \in \mathbb{Z}$ is a solution of the congruence relation (4.1), then the set of solution is $s + \frac{n}{d}\mathbb{Z}$. We can now begin to describe the Pollard's-ρ algorithm.

Let $f : G \to \{1, \ldots, s\}$ be a function that partitions G into $G_i := f^{-1}(\{i\})$ for $i = 1, \ldots, s$. At the beginning the algorithm randomly chooses integers $a_1, \ldots, a_s, b_1, \ldots, b_s$ and computes elements $c_i := g^{a_i} a^{b_i}$. After that, it constructs a sequence of elements

$$\begin{aligned} h_0 &= g^{x_0} a^{y_0}, \text{ for some } x_0, y_0 \in \mathbb{Z} \\ h_{j+1} &:= h_j c_{f(h_j)} \end{aligned}$$

Note that each h_i is of the form $h_j = g^{x_j} a^{y_j}$ for some $x_j, y_j \in \mathbb{Z}$.

Since G is a finite group we always find two different indexes $j \neq l$ such that $h_j = h_l$, i.e. $g^{x_j - x_l} = h^{y_l - y_j} = g^{x(y_l - y_j)}$, from which we can conclude $x_j - x_l \equiv x\,(y_l - y_j) \pmod{n}$. Therefore a solution of our given DLP is a solution of the congruence relation

$$(y_l - y_j) z \equiv x_j - x_l \pmod{n}. \tag{4.2}$$

Set $d := \gcd(y_l - y_j, n)$. Then the result about congruence relations tells us that the set of solutions is $s + \frac{n}{d}\mathbb{Z}$, where s is the smallest positive integer solution. To find s, we apply the Euclidean algorithm and get

$$a(y_l - y_j) + bn = d, \quad \text{for some } a, b \in \mathbb{Z}. \tag{4.3}$$

Since the congruence relation (4.2) must have a solution, we know that $d \mid x_j - x_l$, i.e. $dr = x_j - x_l$, for some integer r. If we now multiply the equation (4.3) by r, we have

$$ar(y_l - y_j) + rbn = dr = x_j - x_l.$$

So $s = ar$ is a solution of (4.2). Therefore x is in the set $s + \frac{n}{d}\mathbb{Z}$. More precisely, the discrete logarithm x is one of the values $s, s + \frac{n}{d}, \dots, s + (d-1)\frac{n}{d}$, because x is smaller than n. If d is small enough we try all possible values. Otherwise we start with a new random starting point h_0 and repeat the procedure.

If $(h_0, h_1, \dots)$ behaves like a random sequence one can show with probabilistic methods that the first match for large n is expected after $\sqrt{\frac{\pi}{2}}\sqrt{n}$ elements, see [32]. To save space Pollard-ρ uses the algorithm of Floyd. It only stores h_i and h_{2i}. One can easily check that there is also a collision within the sequence of these elements. In total Pollard's-ρ runs in $O(\sqrt{n})$ expected time [14, 3.2.2] and uses less memory than BSGS. In addition it is parallelizable.

Example 4.2. Let $G = (\mathbb{Z}/11\mathbb{Z})^\times$, $g = \overline{2}$ and $a = \overline{7}$. First we define the function $f : G \longrightarrow \{1, 2, 3\}$, which partitions G in the following way

$$\begin{aligned} \overline{1}, \overline{4}, \overline{7}, \overline{10} &\longmapsto 1, \\ \overline{2}, \overline{5}, \overline{8} &\longmapsto 2, \\ \overline{3}, \overline{6}, \overline{9} &\longmapsto 3. \end{aligned}$$

Thus $G_1 = \{\overline{1}, \overline{4}, \overline{7}, \overline{10}\}, G_2 = \{\overline{2}, \overline{5}, \overline{8}\}$ and $G_3 = \{\overline{3}, \overline{6}, \overline{9}\}$. We choose $a_1 = 2, a_2 = 3, a_3 = 5, b_1 = 3, b_2 = 4$ and $b_3 = 6$, and compute

$$c_1 = \overline{2}^2 \cdot \overline{7}^3 = \overline{8}, \;\; c_2 = \overline{2}^3 \cdot \overline{7}^4 = \overline{2} \;\text{ and }\; c_3 = \overline{2}^5 \cdot \overline{7}^6 = \overline{7}.$$

We construct the sequence $(h_0, h_1, \dots)$ until an element is repeated, as starting point we choose $h_0 = \overline{2}^8 \cdot \overline{7}^2 = \overline{4}$.

$$\begin{aligned} h_1 &= \overline{4} \cdot \overline{8} = \overline{10} = \overline{2}^8 \cdot \overline{7}^2 \cdot \overline{2}^2 \cdot \overline{7}^3 = \overline{2}^{10} \cdot \overline{7}^5 \\ h_2 &= \overline{10} \cdot \overline{4} = \overline{7} = \overline{2}^{10} \cdot \overline{7}^5 \cdot \overline{2}^8 \cdot \overline{7}^2 = \overline{2}^{18} \cdot \overline{7}^7 \\ h_3 &= \overline{7} \cdot \overline{4} = \overline{6} = \overline{2}^{18} \cdot \overline{7}^7 \cdot \overline{2}^8 \cdot \overline{7}^2 = \overline{2}^{26} \cdot \overline{7}^9 \\ h_4 &= \overline{6} \cdot \overline{7} = \overline{9} = \dots \\ h_5 &= \overline{9} \cdot \overline{7} = \overline{8} = \dots \\ h_6 &= \overline{8} \cdot \overline{2} = \overline{5} = \dots \\ h_7 &= \overline{5} \cdot \overline{2} = \overline{10} = \overline{2}^{39} \cdot \overline{7}^{25} \cdot \overline{2}^3 \cdot \overline{7}^4 = \overline{2}^{42} \cdot \overline{7}^{29} \end{aligned}$$

We see that $h_1 = h_7$ and thus $\overline{2}^{10-42} = \overline{7}^{29-5} = \overline{2}^{x(29-5)}$. To solve the DLP we compute the set of solutions of the congruence relation

$$24x \equiv -32 \pmod{10},$$

which simplifies to $4x \equiv 8 \bmod 10$. In this form one directly sees that 2 is a solution and since $\gcd(4, 10) = 2$ the set of solutions is $2 + 5\mathbb{Z}$. Thus x is either 2 or 7, but $\overline{2}^2 = \overline{4} \neq \overline{7}$, so $x = 7$.

4.4. Pollard's-λ

This algorithm is similar to Pollard's-ρ-method. As before we define elements $c_1, \dots, c_s$ and a function $f : G \to \{1, \dots, s\}$. Pollard's-$\lambda$ starts with two elements $h_0 = g^{x_0} a^{y_0}$ and $h'_0 = g^{x'_0} a^{y'_0}$, where x_0, x'_0, y_0, y'_0 are integers. Now we also recursively define two sequences

$$h_{j+1} = h_j c_{f(h_j)} \text{ and } h'_{j+1} = h'_j c_{f\left(h'_j\right)}$$

Each of these elements can be written as a product of powers of g and a. A match $h_j = h'_l$ between the sequences implies $g^{x_j - x'_l} = g^{x\left(y'_l - y_j\right)}$. From this point on it is exactly the same as before. Pollard's-λ method is only better than the ρ-method if we already know that the discrete logarithm is in a small enough interval.

4.5. Index-Calculus

This algorithm does not work on all groups. We will show it for $G = (\mathbb{Z}/p\mathbb{Z})^\times$, where p is a prime number. It even works for all multiplicative groups of a finite field. For this algorithm $\overline{g} \in (\mathbb{Z}/p\mathbb{Z})^\times$ is the residual class and $g \in \mathbb{Z}$ is the smallest positive representative of this class. So the discrete logarithm problem in this special case looks like

$$\overline{g}^x = \overline{a}.$$

At first we need a definition. Let $B \in \mathbb{N}$, then

$$F(B) := \{q \text{ prime number} : q \leq B\}$$

is called a factor basis. An integer b is B-smooth if all prime factors are $\leq B$.
The algorithm has two major steps:

1. Compute the discrete logarithm for all elements q in the factor base $F(B)$

$$\overline{g}^{x_q} = \overline{q} \tag{4.4}$$

2. Look for an exponent $y \in \{1, 2, \dots, p-1\}$ such that the integer ag^y modulo p is B-smooth.

Assuming we have completed these two steps, why does this solve our original problem? The second step implies $ag^y \equiv \prod_{q \in F(B)} q^{e_q}$ modulo p, with non-negative integer exponent e_g for $q \in F(B)$.

$$\overset{(4.4)}{\Rightarrow} \overline{a}\overline{g}^y = \prod_{q \in F(B)} \overline{g}^{x_q e_q} = \overline{g}^{\sum_{q \in F(B)} x_q e_q}.$$

If we compare the exponents, we see that $x \equiv \left(\sum_{q \in F(B)} x_q e_q - y\right)$ modulo $p - 1$.

Now we have a closer look on how the Index-Calculus-method solves these two major steps. In order to compute the DLP of the elements in the factor basis, we choose

$z \in \{1, \ldots, p-1\}$ randomly and check whether the integer $b := g^z$ modulo p is B-smooth. If yes, one calculates the factorization of $b \pmod{p} = \prod_{q \in F(B)} q^{f(q,z)}$. Using similar arguments as above we get $z \equiv \sum_{q \in F(B)} x_q f(q,z) \pmod{p-1}$. Each exponent vector $(f(q,z))_{q \in F(B)}$ is called a relation. When we have found $|F(B)|$ relations, the algorithm tries to solve the DLP with Gauss elimination, modulo each prime divisor l of $p-1$. If some power of a prime divisor l divides $p-1$, then it computes x_q modulo this power. After that one gets x_q with the CRT.
For the second step choose $y \in \{1, \ldots, p-1\}$ at random until ag^y mod p is B-smooth. The Index-Calculus has running time $\exp((c+o(1))(\log n)^{1/2}(\log \log n)^{1/2}$, where c is a positive constant depending on the implementation [2, 11.6.4].

Example 4.3. Let $G = (\mathbb{Z}/83\mathbb{Z})^\times, \overline{g} = \overline{2}$ and $\overline{a} = \overline{11}$. We choose $B = 5$, thus the factor basis is $F(5) = \{2, 3, 5\}$. To compute the discrete logarithm for the elements in the factor basis, we check whether $\overline{2}^z$ modulo p is 5-smooth.

$$\begin{aligned}
2^1 &\equiv 2 \\
2^7 &\equiv 45 = 3^2 \cdot 5 \\
2^{15} &\equiv 15 = 3 \cdot 5
\end{aligned}$$

These are the first three linear independent relations. We immediately see that $x_2 = 2$, with simple linear algebra we get $x_3 = 72$ and $x_5 = 27$. For the second step we first choose $y = 6$, then $\overline{11} \cdot \overline{2}^6 = \overline{40}$ and $40 = 2^3 \cdot 5$ is 5-smooth. Thus we can compute x in the following way $x \equiv 3 \cdot 1 + 1 \cdot 27 - 6 = 24$ modulo 82.

5. MOV-Algorithm

This algorithm was developed by Menezes, Okamoto and Vanstone [19]. It uses the Weil-pairing to transfer a ECDLP in the elliptic curve group $E(\mathbb{F}_q)$ to a DLP in the multiplicative group $\mathbb{F}_{q^l}^\times$ for a certain $l \geq 1$. If l is small enough we can solve the DLP in $\mathbb{F}_{q^l}^\times$ and hence the ECDLP in $E(\mathbb{F}_q)$ with the Index-Calculus 4.5.

The description of the algorithm is based on [36, 4.2.1]. In the book the reader is at some points directed to further literature, we try to close these gaps. This is possible mainly because we invested more time to develop a theory about computing the Weil-pairing.

5.1. Algorithm

We are in the setting of the ECDLP. To be able to work with the Weil-pairing we have to assume that n is coprime to $p = \operatorname{char}(\mathbb{F}_q)$. But this assumption is not too restrictive, because if n is not coprime to p, we write

$$n = n'p^a$$

with n' coprime to p and $a \geq 1$. If we set $P_1 = [n']P$ and $P_2 = [p^a]P$, we get two new DLPs. In this case we solve the first one with Pohlig-Hellman combined with Pollard-ρ. On the second one we apply the MOV-algorithm. As in the algorithm of Pohlig-Hellman we then use CRT to get k modulo n.

From now on we can assume n coprime to p. The group $E[n]$ is a finite subgroup of $E(\overline{\mathbb{F}}_q)$. For a finite point $R = (x, y) \in E(\overline{\mathbb{F}}_q)$ both coordinates lie in $\overline{\mathbb{F}}_q$, then as seen before they lie in $\mathbb{F}_{q^l}$ for some $l \in \mathbb{N}$. Hence, every point $R \in E(\overline{\mathbb{F}}_q)$ is in the set $E(\mathbb{F}_{q^l})$ for some $l \in \mathbb{N}$. Since $E[n]$ is finite, we have

$$E[n] \subset E(\mathbb{F}_{q^l})$$

for some sufficiently large l.

MOV-Algorithm:

(1) Determine a number l with $E[n] \subset E(\mathbb{F}_{q^l})$.

(2) Compute a point $R \in E[n]$ such that $a = e_n(P, R)$ is a primitive n-th root of unity, i.e. a has order n in $\mu_n(\overline{\mathbb{F}}_q)$.

(3) Compute $b = e_n(Q, R)$.

(4) Solve the DLP: $b = a^k$ in $\mathbb{F}_{q^l}^\times$.

5.2. Correctness

By assumption the point P has order n. Corollary 2.19 implies that a point R as in (2) exists. Again by corollary 2.19 the values $e_n(P,R)$ and $e_n(Q,R)$ lie in $\mathbb{F}_{q^l}^{\times}$. Then we have

$$b = e_n(Q,R) = e_n([k]P,R) = e_n(P,R)^k = a^k.$$

By solving this DLP in the subgroup $\langle a \rangle$ of $\mathbb{F}_{q^l}^{\times}$ we determine k modulo n. Therefore the MOV-algorithm solves our original ECDLP in the elliptic curve group $E(\mathbb{F}_q)$.

5.3. MOV for Supersingular Curves

In this section we want to show that the MOV-algorithm is very efficient when the elliptic curve is supersingular. This implies that supersingular curves are not a good choice for the ECDH or the Elliptic Curve ElGamal Encryption. Obviously for supersingular curves the first step in the MOV-algorithm is just a lookup in the table of proposition 2.26. The second step can be done by a slight modification of the original algorithm:

(1) Compute $t = q + 1 - \#E(\mathbb{F}_q)$ and determine l such that $E[n] \subset E(\mathbb{F}_{q^l})$ and the corresponding exponent d of the group $E(\mathbb{F}_{q^l})$ with the table above.

(2) Choose an arbitrary point $R' \in E(\mathbb{F}_{q^l})$ and set $R = \left[\frac{d}{n}\right] R'$.

(3) Compute $a = e_n(P,R)$ and $b = e_n(Q,R)$.

(4) Solve the DLP: $b = a^{k'}$ in $\mathbb{F}_{q^l}^{\times}$.

(5) If $[k']P = Q$, then $k' = k$.
Otherwise go again to Step (2).

Since P is a point of order n and $E(\mathbb{F}_q) \subset E(\mathbb{F}_{q^l})$, n must divide the exponent d of $E(\mathbb{F}_{q^l})$. Therefore the point R in (2) is well defined. In addition $R \in E[n]$, because $[n]R = [d]R' = O$, this ensures that we can plug R into the Weil-pairing.
If $a = e_n(P,R)$ is a primitive n-th root of unity, then we have already seen that $k' \equiv k$ modulo n. In the other case we still get

$$b = a^k \in \mathbb{F}_{q^l}^{\times}$$

for our discrete logarithm k. But now we just solve the DL-problem in the subgroup $\langle a \rangle$ of $\mathbb{F}_{q^l}^{\times}$. Let m be the order of the group generated by a. Then we get k modulo m and not n. So it can of course happen that

$$[k']P \neq Q.$$

Then we have to repeat the algorithm with a new R'. In order to have an efficient algorithm we have to make sure that the number of repetitions is small enough. The probability that a is a primitive n-th root of unity is $\phi(n)/n$, where ϕ is the Euler's totient function. This means for the average we need $n/\phi(n)$ repetitions. This number decreases for infinitely many n very fast. A standard estimation gives us [22]

$$\frac{n}{\phi(n)} \leq 6 \log \log n,$$

for infinitely many n.

Example 5.1. Let $E : y^2 = x^3 + x$ be an elliptic curve over $\mathbb{F}_{19}$. By the same argument as in 2.24 we see that E is supersingular over $\mathbb{F}_{19}$. So we can apply the adapted MOV algorithm. The computations in this example were done with the computer algebra system SageMath. Before we perform the actual algorithm note that $\mathbb{F}_{19^2} \cong \mathbb{F}_{11}[X]/(X^2 + 18X + 2)$ and let α be a root of $X^2 + 18X + 2$. We solve the following ECDLP:

$$[k](3, 7) = (5, 4).$$

(1) Compute $t = 19 + 1 - 20 = 0$. A look up in the table of proposition 2.26 gives us $l = 2$ and $d = 19 + 1 = 20$, i.e. $E[20] \subset E(\mathbb{F}_{19^2})$.

(2) Choose $R' = (4\alpha + 1, 14)$ and compute $R = [\frac{20}{20}]R' = R'$.

(3) Compute the values of $a = e_{20}((3, 7), (4\alpha + 1, 14)) = 9\alpha + 1$
and $b = e_{20}((5, 4), (4\alpha + 1, 14)) = \alpha + 11$ with Miller's algorithm, as in the example 2.16.

(4) Solve the DLP: $\alpha + 11 = (9\alpha + 1)^{k'}$ in $\mathbb{F}_{19^2}^{\times}$.
We get $k' = 4$

(5) Since $[4](3, 7) = (5, 4)$ we are done.

6. SSSA-Algorithm

For another type of elliptic curves, the ECDLP can be computed efficiently. We call an elliptic curve $\widetilde{E}$ over the prime field $\mathbb{F}_p$ anomalous if $\#\widetilde{E}(\mathbb{F}_p) = p$. For such curves, Satoh and Araki, Smart as well as Semaev independently developed an effective algorithm for solving the discrete logarithm (see [24], [31] and [26]). The algorithm is named after its developers by the acronym SSSA. The central idea is to lift the curve $\widetilde{E}$ up to a curve E over the p-adic field $\mathbb{Q}_p$. For the definition and some facts about $\mathbb{Q}_p$ look at the Appendix. With the help of this lifting we will see that we can construct an isomorphism from $E(\mathbb{F}_p)$ to $\mathbb{F}_p$. Then we only have to solve the corresponding DLP in the additive group of $\mathbb{F}_p$, which is trivial.

6.1. Reduction map

Let $\widetilde{E} : y^2 = x^2 + \widetilde{a}x + \widetilde{b}$ be an elliptic curve over the the prime field $\mathbb{F}_p$. Since the reduction map

$$\begin{aligned} \pi : \mathbb{Z}_p &\longrightarrow \mathbb{F}_p \\ x = (x_n)_{n\geq 1} &\longmapsto \widetilde{x} = x_1, \end{aligned}$$

is onto, we can choose $a, b \in \mathbb{Z}_p$ such that $\pi(a) = \widetilde{a}$ and $\pi(b) = \widetilde{b}$. If we replace $\widetilde{a}$ and $\widetilde{b}$ with in our Weierstrass equation with a and b and homogenize the equation we get a homogeneous polynomial g which defines a plane projective curve C_g over $\mathbb{Q}_p$.

We want to be able to define the reduction map on an arbitrary point in the projective plane $P = (\alpha : \beta : \gamma) \in \mathbb{P}^2(\mathbb{Q}_p)$. For this reason we have to show that (α, β, γ) can always be chosen such that α, β and γ are in $\mathbb{Z}_p$. Let v_p be the valuation on $\mathbb{Q}_p$ defined in the Appendix. We define $m := \{-v_p(\alpha), -v_p(\beta), -v_p(\gamma)\} \in \mathbb{Z} \cup \{\infty\}$. Since by definition at least one coordinate is always not zero, m is an integer. Then $p^m\alpha, p^m\beta, p^m\gamma \in \mathbb{Z}_p$ and at least one of them is in $\mathbb{Z}_p^\times$. Therefore,

$$\widetilde{P} = (\pi(p^m\alpha) : \pi(p^m\beta) : \pi(p^m\gamma)) \in \mathbb{P}^2(\mathbb{F}_p).$$

The only thing left to show is that $\widetilde{P}$ is well-defined. Let $P = (\alpha : \beta : \gamma) = (\alpha' : \beta' : \gamma')$ with $\alpha, \beta, \gamma, \alpha', \beta', \gamma' \in \mathbb{Z}_p$ and at least one coordinate in each point is a unit in $\mathbb{Z}_p$. Since the two points are equal in the projective plane

$$\alpha = \lambda\alpha', \quad \beta = \lambda\beta', \quad \gamma = \lambda\gamma'$$

for some $\lambda \in \mathbb{Q}_p^\times$. Assume $\alpha \in \mathbb{Z}_p^\times$ and $\beta' \in \mathbb{Z}_p^\times$. Then we have $\lambda^{-1} = \alpha^{-1}\alpha' \in \mathbb{Z}_p$ and $\lambda = \beta\beta'^{-1} \in \mathbb{Z}_p$ and therefore λ is a unit in $\mathbb{Z}_p$. Now we can use the fact that π is a ring homomorphism:

$$\begin{aligned}(\widetilde{\alpha} : \widetilde{\beta} : \widetilde{\gamma}) &= (\pi(\alpha) : \pi(\beta) : \pi(\gamma)) \\ &= (\pi(\lambda\alpha') : \pi(\lambda\beta') : \pi(\lambda\gamma')) \\ &= (\pi(\lambda)\pi(\alpha') : \pi(\lambda)\pi(\beta') : \pi(\lambda)\pi(\gamma')) \\ &= (\widetilde{\alpha'} : \widetilde{\beta'} : \widetilde{\gamma'})\end{aligned}$$

This shows how we can extend the definition of the reduction map π on the projective plane.

The next step is to prove that C_g is an elliptic curve and that $C_g(\mathbb{Q}_p)$ under π reduces to $\widetilde{E}(\mathbb{F}_p)$. The prove of the following lemma is taken from [36, Lem. 4.2.3].

Lemma 6.1. *Let the setting be as above. Then the map*

(i) $\pi : \mathbb{P}^2(\mathbb{Q}_p) \to \mathbb{P}^2(\mathbb{Q}_p)$ *induces a surjective map*

$$\pi : C_g(\mathbb{Q}_p) \to \widetilde{E}(\mathbb{F}_p).$$

(ii) The curve C_g *is non-singular, i.e.* C_g *is an elliptic curve.*

Proof. (i) Let $P = [\alpha : \beta : \gamma] \in C_g(\mathbb{Q}_p)$ with coordinates $\alpha, \beta, \gamma \in \mathbb{Z}_p$ and at least one is a unit in $\mathbb{Z}_p$. We again use that π is a ring homomorphism to get

$$0 = \pi(0) = \pi(g(\alpha, \beta, \gamma)) = \pi(g)(\widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma}).$$

From this follows that the point $(\widetilde{\alpha} : \widetilde{\beta} : \widetilde{\gamma})$ is a zero of the Weierstrass polynomial $\pi(g)$, i.e. $(\widetilde{\alpha} : \widetilde{\beta} : \widetilde{\gamma}) \in \widetilde{E}(\mathbb{F}_p)$.
Left to show is the surjectivity of π. Let $\widetilde{P} = (\widetilde{\alpha} : \widetilde{\beta} : \widetilde{\gamma}) \in \tilde{E}(\mathbb{F}_p)$. Since $\tilde{E}$ is non-singular at least one of the derivatives $\frac{\partial \pi(g)}{\partial X}$, $\frac{\partial \pi(g)}{\partial Y}$ or $\frac{\partial \pi(g)}{\partial Z}$ is not zero at the point $(\widetilde{\alpha} : \widetilde{\beta} : \widetilde{\gamma})$. We assume $\frac{\partial \pi(g)}{\partial X}(\alpha', \beta', \gamma') \neq 0$. Further let β and γ be elements in $\mathbb{Z}_p$ such that $\pi(\beta) = \widetilde{\beta}$ and $\pi(\gamma) = \widetilde{\gamma}$. Consider a new function

$$f(X) = g(X, \beta, \gamma)$$

as a polynomial in one variable. It is obvious that $\pi(f)(X)$ has a zero at $\widetilde{\alpha} \in \mathbb{F}_p$ and that

$$\frac{\partial \pi(f)}{\partial X}(\widetilde{\alpha}) = \frac{\partial \pi(g)}{\partial X}(\widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma}) \neq 0.$$

If we apply Hensel's Lemma A.5, we get the existence of an element $\alpha \in \mathbb{Z}_p$ such that $\pi(\alpha) = \widetilde{\alpha}$ and $f(\alpha) = 0$. As a direct consequence, $g(\alpha, \beta, \gamma) = 0$ and we have a point $P = [\alpha : \beta : \gamma] \in C_g(\mathbb{Q}_p)$ with $\widetilde{P} = (\widetilde{\alpha} : \widetilde{\beta} : \widetilde{\gamma})$.

(ii) Let $P = (\alpha : \beta : \gamma) \in C_q(\mathbb{Q}_p)$ with coordinates $\alpha, \beta, \gamma \in \mathbb{Z}_p$ and at least one is a unit in $\mathbb{Z}_p$. Since $\widetilde{P}$ is a point on the non-singular curve $\tilde{E}$, at least one derivative of $\pi(g)$ is non-zero at the point $(\widetilde{\alpha} : \widetilde{\beta} : \widetilde{\gamma})$, assume as above $\frac{\partial \pi(g)}{\partial X}$. But then

$$\pi\left(\frac{\partial(g)}{\partial X}(\alpha, \beta, \gamma)\right) = \frac{\partial \pi(g)}{\partial X}(\widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma}),$$

and therefore also $\frac{\partial(g)}{\partial X}(\alpha, \beta, \gamma)$ is non-zero. □

From now on we write E for the elliptic curve C_g. Note that the elliptic curve E over $\mathbb{Q}_p$ depends on the preimages a and b we have chosen at the beginning of this section.

Proposition 6.2. *The induced reduction map*

$$\pi : E(\mathbb{Q}_p) \to \tilde{E}(\mathbb{F}_p)$$

is a group homomorphism.

Proof. To prove this fact we use proposition [30, VII.2.1]. Therefore, as in the book of Silverman, we need the additional Lemma 6.3 found in [30, VII.2.1.1].
The group laws on E and $\tilde{E}$ are defined by taking intersections with lines in $\mathbb{P}^2$. For any line L defined over $\mathbb{Q}_p$, we can find an equation for L of the form

$$L : AX + BY + CZ = 0,$$

such that $A, B, C \in \mathbb{Z}_p$ and at least one of A, B, C is in $\mathbb{Z}_p^\times$ (by the same argument as before). Then the reduction of L is given by the equation

$$\tilde{L} : \tilde{A}X + \tilde{B}Y + \tilde{C}Z = 0$$

and it is clear that if $P \in \mathbb{P}^2(\mathbb{Q}_p)$ is a point on the line L, then the reduced point $\widetilde{P}$ is on the reduced line $\widetilde{L}$.

Let $P_1, P_2, P_3 \in E(\mathbb{Q}_p)$ be points satisfying $P_1 + P_2 + P_3 = O$. Thus there is a line L that intersects E at these three points P_1, P_2, P_3, counted with appropriate multiplicities. We are going to prove that $\widetilde{L}$ intersects $\widetilde{E}$ at $\widetilde{P_1}, \widetilde{P_2}, \widetilde{P_3}$ with correct multiplicities, from which follows that $\widetilde{P_1} + \widetilde{P_2} + \widetilde{P_3} = O$. Suppose this is true. Let $R \in E(\mathbb{Q}_p)$ arbitrary. If we set $P_1 = R, P_2 = -R$ and $P_3 = O$ the assumption that the three points sum to O is satisfied. Hence, we have $\pi(R) + \pi(-R) + \pi(O) = O$, i.e. for any point $R \in E(\mathbb{Q}_p)$ the following holds $-\pi(R) = \pi(R)$. For two points $P, Q \in E(\mathbb{Q}_p)$ we choose the third point to be $-(P + Q)$. Since $P + Q + (-(P + Q)) = O$, the assertion tells us

$$\begin{aligned} &\pi(P) + \pi(Q) + \pi(-(P + Q)) = O \\ \Leftrightarrow\ &\pi(P) + \pi(Q) = -\pi(-(P + Q)) = \pi(P + Q). \end{aligned}$$

So we are done once we can show the assertion. We will look at two interesting cases, the others are proven similarly or are direct consequences of the two shown cases together with the theory so far developed about the reduction π.

Let the reduced points $\widetilde{P_1}, \widetilde{P_2}, \widetilde{P_3}$ be distinct. Then $L \cap E = \widetilde{\{P_1, P_2, P_3\}}$, which is our desired result. Now let P_1, P_2, P_3 be distinct points and $\widetilde{P_1} = \widetilde{P_2} \neq \widetilde{P_3}$. We apply Lemma 6.3 with $P = P_1$ and $Q = P_2$. This tells us that $\widetilde{L}$ is tangent to $\widetilde{E}$ at $\widetilde{P_1}$, and we also have $\widetilde{P_3} \in \widetilde{L}$, so we find that $2\widetilde{P_1} + \widetilde{P_3} = O$. Since we assume that $\widetilde{P_1} = \widetilde{P_2}$, we are done. □

Lemma 6.3. *Let $P, Q \in E(\mathbb{Q}_p)$ be distinct points who satisfy $\widetilde{P} = \widetilde{Q}$ and let L be the line through P and Q. Then the line $\widetilde{L}$ is tangent to $\widetilde{E}$ at $\widetilde{P}$.*

Proof. We only show the general case $\widetilde{P} \neq O$. Write

$$P = (\alpha, \beta) \in E(\mathbb{Q}_p) \quad \text{ and } Q = (\alpha + \mu, \beta + \lambda) \in E(\mathbb{Q}_p).$$

The assumption $\widetilde{P} = \widetilde{Q} \neq O$ implies that $\alpha, \beta \in \mathbb{Z}_p$ and $\mu, \lambda \in p\mathbb{Z}_p$. Further $\widetilde{P}$ is a non-singular point of $\tilde{E}$, so either

$$\frac{\partial \pi(g)}{\partial X}\left(\widetilde{P}\right) \neq 0 \qquad \text{or} \qquad \frac{\partial \pi(g)}{\partial Y}\left(\widetilde{P}\right) \neq 0.$$

We do the case $(\partial \pi(g)/\partial Y)(\widetilde{P}) \neq 0$. The fact that $g(P) = g(Q) = 0$ allows us to compute the first few terms of the Taylor expansion of $g(X, Y)$ around Q. Thus

$$\begin{aligned} 0 &= g(\alpha + \mu, \beta + \lambda) \\ &= \underbrace{g(\alpha, \beta)}_{=0} + \frac{\partial g}{\partial X}(\alpha, \beta)\mu + \frac{\partial g}{\partial Y}(\alpha, \beta)\lambda + a\mu^2 + b\mu\lambda + c\lambda^2 \quad \text{for some } a, b, c \in \mathbb{Z}_p \\ &= \frac{\partial g}{\partial X}(\alpha, \beta)\mu + \frac{\partial g}{\partial Y}(\alpha, \beta)\lambda + a\mu^2 + b\mu\lambda + c\lambda^2. \end{aligned} \tag{6.1}$$

Let $v : \mathbb{Q}_p \to \mathbb{Z}$ be the valuation defined in the Appendix. The assumption that $(\partial \pi(g)/\partial Y)(\widetilde{P}) \neq 0$ is equivalent to $(\partial g/\partial Y)(\alpha, \beta) \in \mathbb{Z}_p^\times$, so $v((\partial g/\partial Y)(\alpha, \beta)) = 0$. When using the calculation rules of valuations, we get

$$\begin{aligned} v(\lambda) = v\left(\frac{\partial g}{\partial Y}(\alpha, \beta)\,\lambda\right) &\overset{(6.1)}{=} v\left(\frac{\partial g}{\partial X}(\alpha, \beta)\mu + a\mu^2 + b\mu\lambda + c\lambda^2\right) \\ &\geq \min\left(v(\frac{\partial g}{\partial X}(\alpha, \beta)) + v(\mu), v(a) + 2v(\mu), v(b) + v(\mu) + v(\lambda), v(c) + 2v(\lambda)\right) \\ &\geq v(\mu). \end{aligned}$$

The last inequality holds because $v(\lambda) \geq 2v(\lambda)$ would be a contradiction (since $v(\lambda) \geq 1$ by assumption). Thus $\lambda/\mu \in \mathbb{Z}_p$, so dividing the Taylor expansion by μ and reducing modulo π gives us

$$\pi\left(\frac{\partial g}{\partial X}(\alpha, \beta) + \frac{\partial g}{\partial Y}(\alpha, \beta)\frac{\lambda}{\mu} + \underbrace{a\mu + b\lambda + \frac{c\lambda^2}{\mu}}_{\in p\mathbb{Z}_p, \text{ because } \mu, \lambda \in \mathbb{Z}_p}\right) = \pi\left(\frac{\partial g}{\partial X}(P) + \frac{\partial g}{\partial Y}(P) \cdot \frac{\lambda}{\mu}\right) = 0.$$

This tells us that the slope of the tangent line to $\tilde{E}$ at the point $\widetilde{P}$ is

$$-\frac{(\partial \pi(g)/\partial X)(\widetilde{P})}{(\partial \pi(g)/\partial Y)(\widetilde{P})} = \pi\left(\frac{\lambda}{\mu}\right).$$

The line L through P and Q is given by the equation

$$L : Y - \beta = \frac{\lambda}{\mu}(X - \alpha).$$

We have shown $\lambda/\mu \in \mathbb{Z}_p$, so the reduction of L is the line through $\widetilde{P}$ having slope $\pi(\lambda/\mu)$. This proves that $\widetilde{L}$ is tangent to $\tilde{E}$ at $\widetilde{P}$. □

6.2. From points on the curve to the maximal ideal of the p-adic integers

We want to show that there exist maps λ and ψ such that λ is an isomorphism from $\tilde{E}(\mathbb{F}_p)$ to $\mathbb{F}_p$:

$$\begin{tikzcd}
E(\mathbb{Q}_p) \arrow[r, "{[p]}"] \arrow[d, "\pi"'] & E_1(\mathbb{Q}_p) := \ker \pi(E(\mathbb{Q}_p)) \arrow[d, "\psi"] \\
\tilde{E}(\mathbb{F}_p) \arrow[r, "\lambda"] & \mathbb{F}_p
\end{tikzcd}$$

An important observation is that the points in the domain are of a special form. An element $(\alpha : \beta : \gamma) \in E(\mathbb{Q}_p)$, with $\alpha, \beta, \gamma \in \mathbb{Z}_p$ and at least one of them not in $p\mathbb{Z}_p$ lies in $\tilde{E}_1(\mathbb{Q}_p)$, if and only if $(\widetilde{\alpha} : \widetilde{\beta} : \widetilde{\gamma}) = (0 : 1 : 0)$, i.e. $\widetilde{\beta} \neq 0$ and $\widetilde{\alpha} = \widetilde{\gamma} = 0$.
Therefore,

$$E_1(\mathbb{Q}_p) = \{(x : 1 : z) \mid x, z \in p\mathbb{Z}_p\} \cap E. \tag{6.2}$$

Due to this observation we will look at a different embedding of the affine plane than the normal one. Working with the embedding $(x, z) \mapsto (x : 1 : z)$ makes things a lot easier. We start by defining ψ and showing that it is a homomorphism. Let

$$\begin{aligned} \psi : E_1(\mathbb{Q}_p) &\longrightarrow p\mathbb{Z}_p/p^2\mathbb{Z}_p \\ (\alpha : \beta : \gamma) &\longmapsto \alpha \bmod p^2. \end{aligned}$$

Lemma 6.4. *Let the setting be as above. Then*

$$\psi : E_1(\mathbb{Q}_p) \longrightarrow p\mathbb{Z}_p/p^2\mathbb{Z}_p \cong \mathbb{F}_p$$

is a homomorphism.

Proof. A proof involving formal groups can be found in [30, IV]. We give here an elementary proof for the case that the x-coordinate of two points are different.
So we have to show that for two points $P_1 = (x_1, z_1), P_2 = (x_2, z_2) \in E_1(\mathbb{Q}_p)$ the x-coordinate of the sum of these two points is congruent to $x_1 + x_2 \bmod p^2\mathbb{Z}_p$. Let us assume $x_1 \neq x_2$.

When we set $y = 1$ we get the following equation

$$E : z = x^3 + axz^2 + bz^3. \tag{6.3}$$

Unfortunately this equation has not the form of an Weierstrass equation. Hence, the earlier developed arithmetic formulas do not apply. Instead we have to calculate with the geometric group law. First thing to do is to build the line connecting (x_1, z_1) and (x_2, z_2):

$$z = \lambda x + \nu, \quad \text{where } \lambda = \frac{z_2 - z_1}{x_2 - x_1},\ \nu = z_1 - \lambda x_1.$$

To get the third intersection point of the line with the curve substitute z with the right-hand side of the equation of the line.

$$\begin{aligned} 0 &= (\lambda x + \nu) - x^3 - ax(\lambda x + \nu)^2 - b(\lambda x + \nu)^3 \\ &= -x^3\left(1 + a\lambda^2 + b\lambda^3\right) - (2a\lambda\nu + 3b\lambda^2\nu)x^2 + (\lambda - a\nu^2 - 3b\lambda\nu^2)x + \nu - b\nu^3 \end{aligned}$$

By Vieta, we get

$$\frac{2a\lambda\nu + 3b\lambda^2\nu}{1 + a\lambda^2 + b\lambda^3} = -(x_1 + x_2 + x_3). \tag{6.4}$$

Assertion: $\lambda \in p\mathbb{Z}_p$

Suppose this assertion holds. By (6.2) $x_1, z_1 \in p\mathbb{Z}_p$ and thus $\nu = z_1 - \lambda x_1$ is an element of $p\mathbb{Z}_p$. Since the denominator is a unit the left-hand side of (6.4) is an element of $p^2\mathbb{Z}_p$. Therefore, $x_1 + x_2 + x_3 \in p^2\mathbb{Z}_p$, which implies $x_3 = -x_1 - x_2 + c$, for some $c \in p^2\mathbb{Z}_p$. To get the x-coordinate of $P_1 + P_2$ we have to find the third intersection point of the line connecting O and (x_3, z_3) with the curve. In the x, z-plane O is represented by the point $(0, 0)$. In the equation of the line we get $\nu = 0$. If we set $\nu = 0$ in (6.4) we see that the third root is just the negative of the sum of the first to. This means the x-coordinate of the point $P_1 + P_2$ is $-x_3 = x_1 + x_2 + c$. Therefore $\psi(P_1 + P_2) = \psi(P_1) + \psi(P_2)$.

Proof of assertion:

$$\begin{aligned}
\lambda = \frac{z_2 - z_1}{x_2 - x_1} &= \frac{x_2^3 + ax_2z_2^2 + bz_2^3 - (x_1^3 + ax_1z_1^2 + bz_1^3)}{x_2 - x_1} \\
&= \frac{1}{x_2 - x_1}((x_2^3 - x_1^3 + x_2^2x_1 - x_2^2x_1 + x_2x_1^2 - x_2x_1^2) \\
&\qquad + b(z_2^3 - z_1^3 + z_2^2z_1 - z_2^2z_1 + z_2z_1^2 - z_2z_1^2) \\
&\qquad + a(x_2z_2^2 - x_2z_1^2 + x_2z_1^2 - x_1z_1^2)) \\
&= \frac{1}{x_2 - x_1}((x_2^2 + x_1^2 + x_1x_2)(x_2 - x_1) \\
&\qquad + b(z_2^2 + z_1^2 + z_1z_2)(z_2 - z_1) \\
&\qquad + ax_2(z_2^2 - z_1^2) + az_1^2(x_2 - x_1)) \\
&= x_2^2 + x_1^2 + x_1x_2 + b(z_2^2 + z_1^2 + z_1z_2)\lambda + ax_2(z_2 + z_1)\lambda + az_1^2
\end{aligned}$$

Therefore,

$$\lambda(\underbrace{1 - \underbrace{b(z_2^2 + z_1^2 + z_1z_2) - ax_2(z_2 + z_1)}_{\in p\mathbb{Z}_p}}_{\in\mathbb{Z}_p^\times}) = \underbrace{x_2^2 + x_1^2 + x_1x_2 + az_1^2}_{\in p\mathbb{Z}_p}.$$

Hence, $\lambda \in p\mathbb{Z}_p$. □

6.3. Algorithm

So far, we have not used that $\tilde{E}$ is an anomalous curve. In this case, the multiplication-by-p map on E has a special image. Let $P \in E$, then $\pi([p]P) = [p]\widetilde{P} = 0$ because $\#E(\mathbb{F}_p) = p$. Therefore we have

$$[p] : E(\mathbb{Q}_p) \longrightarrow E_1(\mathbb{Q}_p).$$

We are now able to construct the isomorphism between the elliptic curve group and the prime field. The theorem is due to Satoh and Araki [24, 3.2].

Theorem 6.5. *Let u be any lifting from $\tilde{E}(\mathbb{F}_p)$ to $E(\mathbb{Q}_p)$, i.e. $\pi \circ u = id_{\tilde{E}(\mathbb{F}_p)}$. Let λ_E be the composition of the following maps*

$$\lambda_E : \tilde{E}(\mathbb{F}_p) \xrightarrow{u} E(\mathbb{Q}_p) \xrightarrow{[p]} E_1(\mathbb{Q}_p) \xrightarrow{\psi} p\mathbb{Z}_p/p^2\mathbb{Z}_p \cong \mathbb{F}_p.$$

Then λ_E is a group homomorphism which is independent of choice of u. Moreover , λ_E is the zero map or an isomorphism.

Proof. First we will show that the map is a group homomorphism. Let $\alpha, \beta \in \tilde{E}(\mathbb{F}_p)$ and put $d := u(\alpha) + u(\beta) - u(\alpha + \beta)$. If we apply π on d we get

$$\begin{aligned} \pi(d) &= \pi(u(\alpha) + u(\beta) - u(\alpha + \beta)) \\ &= \pi(u(\alpha)) + \pi(u(\beta)) - \pi(u(\alpha + \beta)) && \pi \text{ homomorphism} \\ &= \alpha + \beta - (\alpha + \beta) = O && u \text{ lifting} \end{aligned}$$

i.e. $d \in E_1(\mathbb{Q}_p)$. Set $F := \psi \circ [p]$. Then $F(d) = \psi([p]d) = p\psi(d) = 0$. By 6.4 ψ is a homomorphism and since π is also homomorphisms, the concatenation of these maps, namely F, is also one. Therefore,

$$\begin{aligned} \lambda_E(\alpha) + \lambda_E(\beta) &= F(u(\alpha)) + F(u(\beta)) = F(u(\alpha) + u(\beta)) \\ &= F(d + u(\alpha + \beta)) = F(d) + F(u(\alpha + \beta)) \\ &= F(u(\alpha + \beta)) = \lambda_E(\alpha + \beta), \end{aligned}$$

i.e. λ_E is a homomorphism.

Let s be another lifting from $\tilde{E}(\mathbb{F}_p)$ to $E(\mathbb{Q}_p)$. Then

$$\pi(u(\alpha) - s(\alpha)) = \pi(u(\alpha)) - \pi(s(\alpha)) = 0 \qquad \text{for any } \alpha \in \tilde{E}(\mathbb{F}_p).$$

Hence we have $u(\alpha) - s(\alpha) \in E_1(\mathbb{Q}_p)$. By the same argument as above we see that $\lambda_{E_u} = F(u(\alpha)) = F(s(\alpha)) = \lambda_{E_s}$ and that λ_E is independent of choice of u.

To show the last statement, simply note that λ_E, as a group homomorphism between two groups of size p, is either the zero map or an isomorphism. □

Since the DLP in the additive group of $\mathbb{F}_p$ can be computed by solving a linear congruence, we can compute this particular DLP in polynomial time with the Extended Euclidean algorithm. If we combine 6.5 with our last observation we can solve the ECDLP (here the notation is $[k]\widetilde{P} = \widetilde{Q}$) in the following way:

SSSA-Algorithm:

(1) Chose preimages $a, b \in \mathbb{Z}_p$ (under the reduction π) of the Weierstrass-coefficients $\widetilde{a}, \widetilde{b}$ of $\tilde{E}(\mathbb{F}_p)$ and with this define $E(\mathbb{Q}_p)$.

(2) Compute $\lambda(\widetilde{P})$ and $\lambda(\widetilde{Q})$.

(3) If $\lambda(\widetilde{P}) \neq 0$, then $k \equiv \frac{\lambda(\widetilde{Q})}{\lambda(\widetilde{P})} \bmod p$.
Otherwise start again with (1).

Here, we reasonably assume that $\widetilde{P} \neq O$. Then $\lambda(\widetilde{P}) \neq 0$ already implies that λ is not the zero map. For our purpose of computing the discrete logarithm, the existence of such an isomorphism is not enough, we also must be able to construct λ_E efficiently.

6.4. Time Analysis

The next theorem [24, 3.5] is the crucial part for analysing the time complexity of the SSSA-algorithm.

Theorem 6.6. *Let $A \in \tilde{E}(\mathbb{F}_p)\backslash\{O\}$ be given. Then there exists a point $P = (x_1, y_1) \in E(\mathbb{Z}_p)$ such that $\widetilde{P} = A$. Further, for every $n \in \mathbb{N}$ such that $[n]P \neq O$, put $(x_n, y_n) := [n]P$. If λ_E is a non-zero map, we have the following:*

(i) $[n]P \in E(\mathbb{Z}_p)\backslash\{O\}$ for $1 \leq n < p$,

(ii) $\widetilde{x}_n \neq \widetilde{x}_m$, for $1 \leq n < m < p$ with $n + m \neq p$,

(iii) $y_{p-1} - y_1 \in \mathbb{Z}_p^\times$, $\frac{x_{p-1}-x_1}{p} \in \mathbb{Z}_p^\times$ and

$$\lambda_E(\widetilde{P}) = \frac{x_{p-1} - x_1}{p(y_{p-1} - y_1)} \text{ mod } p.$$

Proof. First of all, we have to check that we always able to choose the coordinates of P in $\mathbb{Z}_p$. Let $(x : y : 1) \in E(\mathbb{Q}_p)$ such that $(\widetilde{x} : \widetilde{y} : \widetilde{1}) \neq (0 : 1 : 0)$. Set $m = \max\{-v_p(x), -v_p(y), -v_p(1)\}$, then $(x : y : 1) = (p^m x : p^m y : p^m)$. The fact that $\widetilde{p^m} \neq 0$ implies that $m = 0$ and therefore $v_p(x), v_p(y) \geq 0$, i.e. $x, y \in \mathbb{Z}_p$.

(i) Note that $\tilde{E}(\mathbb{F}_p)$ is a cyclic group of order p and $\widetilde{P}$ is a generator. Therefore we have that $[n]\widetilde{P} = O$ if and only if p divides n. From this fact we immediately see that $[n]P = O$ implies $[n]\widetilde{P} = O$, which is a contradiction for $1 \leq n < p$. So we have only to prove $[n]P \in E(\mathbb{Z}_p)$. We use induction on n.

For $n = 1$, this holds by assumption. For $n = 2$, we assume $\widetilde{y_1} = 0$. Then we obtain $[2]\widetilde{P} = [2]\pi(A) = [2](\widetilde{x}_1, 0) = O$, see [30, III.2.3], which contradicts to $\widetilde{P} \neq O$. Therefore, we have

$$y_1 \in \mathbb{Z}_p^\times. \tag{6.5}$$

The addition formula on E for two points with the same x-coordinate yields:

$$x_2 = c_2^2 - 2x_1, \qquad y_2 = -c_2 x_2 - d_2,$$

where

$$c_2 = \frac{3x_1^2 + a_4}{2y_1}, \qquad d_2 = \frac{-x_1^3 + a_4 x_1 + 2a_6}{2y_1}.$$

Since $y_1 \in \mathbb{Z}_p^\times$, we see $x_2, y_2 \in \mathbb{Z}_p$ and (i) hold also for $n = 2$.
For $3 \leq n < p$, suppose $P, [n-1]P \in E(\mathbb{Z}_p)\backslash\{O\}$. Note

$$\widetilde{P} = (\widetilde{x_1}, \widetilde{y_1})$$
$$[n-1]\widetilde{P} = (\widetilde{x}_{n-1}, \widetilde{y}_{n-1}).$$

Assuming $\widetilde{x}_1 = \widetilde{x}_{n-1}$, we obtain $\widetilde{P} = \pm[n-1]\widetilde{P}$ because they have the same x-coordinate and lie on the same elliptic curve; i.e. $[n]\widetilde{P} = O$ or $[n-2]\widetilde{P} = O$ (for that we needed $n = 2$ separately). Again, this implies the contradiction $\widetilde{P} = O$. Therefore we obtain $\widetilde{x}_1 \neq \widetilde{x}_{n-1}$ and obviously $x_1 \neq x_{n-1}$. Then, by the addition formula on E for two points with different x-coordinates, we get

$$x_n = c_n^2 - x_1 - x_{n-1}, \qquad y_n = -c_n^3 + c_n(x_1 + x_{n-1}) - d_n, \tag{6.6}$$

where

$$c_n = \frac{y_{n-1} - y_1}{x_{n-1} - x_1}, \qquad d_n = \frac{y_1 x_{n-1} - y_{n-1} x_1}{x_{n-1} - x_1}. \tag{6.7}$$

By $\widetilde{x}_1 \neq \widetilde{x}_{n-1}$, we see $x_{n-1} - x_1$ is no multiple of p and hence $x_{n-1} - x_1 \in \mathbb{Z}_p^\times$. This implies $c_n, d_n \in \mathbb{Z}_p$ and therefore also $x_n, y_n \in \mathbb{Z}_p$.

(ii) The idea of the proof is similar to the proof above. Assume to the contrary $\widetilde{x}_n = \widetilde{x}_m$. Then $[n]\widetilde{P} = \pm[m]\widetilde{P}$, i.e., $[m \pm n]\widetilde{P} = O$, which is true if and only if $m \pm n = p$ or $m - n = 0$. Since exactly this case is ruled out by assumption, we have $\widetilde{x}_n \neq \widetilde{x}_m$.

(iii) Since, by assumption, λ_E is not the zero map, we see that $\lambda_E(\widetilde{P}) = \psi([p]P) \neq 0$, i.e. $[p]P \neq O$. Note that (6.6) and (6.7) hold for $n = p$, because these are just the addition formulas on E.
Let $(x_p, y_p) := [p]P$. Then $(\widetilde{x}_p : \widetilde{y}_p : 1) = (0 : 1 : 0)$. Set $m = \max\{-v_p(x_p), -v_p(y_p), -v_p(1)\}$. Then multiplying x_p, y_p and 1 by p^m gives us $(x_p : y_p : 1) = (p^m x_p : p^m y : p^m)$. That means $\widetilde{p}^m = 0$, implying $m > 0$. Further from $\widetilde{p^m x_p} = 0$ and $\widetilde{p^m y_p} = 1$ we deduce $v_p(x_p) > v_p(y_p)$ and $v_p(y_p) < 0$. By (ii), we have $P, [p-1]P \in E(\mathbb{Z}_p)$. Let $s := v_p(c_p)$, and assume $s \geq 0$, i.e. $c_p \in \mathbb{Z}_p$. When writing d_p in the following form

$$\begin{aligned} d_p &= \frac{y_1 x_{p-1} - y_{p-1} x_1}{x_{p-1} - x_1} \\ &= \frac{y_1 x_{p-1} - y_1 x_1 - y_{p-1} x_1 + y_1 x_1}{x_{p-1} - x_1} \\ &= \frac{y_1(x_{p-1} - x_1) - x_1(y_{p-1} - y_1)}{x_{p-1} - x_1} \\ &= y_1 - x_1 c_p, \end{aligned} \tag{6.8}$$

it is clear that $d_p \in \mathbb{Z}_p$, and hence $y_p \in \mathbb{Z}_p$, a contradiction. So, s must be negative.

Now we compute the p-adic valuation of some elements and thereby extensively use A.2. Recall, we have $x_1, x_{p-1} \in \mathbb{Z}_p$, so $v_p(-x_1 - x_{p-1}) = v_p(x_1 + x_{p-1}) \geq \min(v_p(x_1), v_p(x_{p-1})) \geq 0$. Then by (6.6), we see

$$v_p(x_p) = v_p(c_p^2 - x_1 - x_{p-1}) = \min(v_p(c_p^2), v_p(-x_1 - x_{p-1})) = 2s \tag{6.9}$$

By (6.8), we also obtain

$$\begin{aligned} v_p(d_p) &\geq \min(v_p(y_1), v_p(x_1) + v_p(c_p)) \\ &\overset{(6.5)}{\geq} \min(0, v_p(c_p)) \overset{v_p(c_p)<0}{=} v_p(c_p) = s. \end{aligned}$$

Moreover $v_p(c_p(x_1 + x_{p-1})) = v_p(c_p) + v_p(x_1 + x_{p-1}) \geq s$, while $v_p(c_p^3) = 3s < s$. Hence

$$\begin{aligned} v_p(y_p) &= v_p(-c_p^3 + c_p(x_1 + x_{p-1}) - d_p) \\ &= \min(v_p(c_p^3), v_p(c_p(x_1 + x_{p-1})), v_p(d_p)) \qquad (6.10) \\ &= 3s. \end{aligned}$$

Therefore, $v_p(\psi([p]P)) = v_p(x_p/y_p) = -s > 0$. We see $v_p(\psi([p]P)) = -s$. By assumption $\lambda_E(\widetilde{P}) \neq 0$. So $v_p(\psi([p]P)) = 1$, because the image of ψ is $p\mathbb{Z}_p$. Summing up, we obtain $s = -1$, $\frac{x_p}{py_p} \in \mathbb{Z}_p^\times$ and $\lambda_E(\widetilde{P}) = \frac{x_p}{py_p} \bmod p$.
By the anomality of $\tilde{E}$, we see $[p-1]\widetilde{P} = -\widetilde{P}$ and hence $\widetilde{y}_{p-1} = -\widetilde{y}_1$. Therefore, $\widetilde{y}_{p-1} - \widetilde{y}_1 = -2\widetilde{y}_1 \neq 0$. So, we have proved $y_{p-1} - y_1 \in \mathbb{Z}_p^\times$.
Since $v_p(c_p) = -1$ (and so $v_p(c_p^{-1}) = 1$), we obtain

$$\frac{y_{p-1} - y_1}{pc_p} = \frac{x_{p-1} - x_1}{p} \in \mathbb{Z}_p^\times. \qquad (6.11)$$

Let $\hat{x} := p^2 x_p$ and $\hat{y} := p^3 y_p$. By $s = -1$, together with (6.9) and (6.10), we see $v_p(\hat{x}) = v_p(p^2) + v_p(x_p) = 2 - 2 = 0$ and similar for $\hat{y}$, i.e. $\hat{x}, \hat{y} \in \mathbb{Z}_p^\times$. Hence $\lambda_E(\widetilde{P}) = \frac{x_p}{py_p} \bmod p = \frac{p^2 x_p}{p^3 y_p} \bmod p = \frac{\hat{x}}{\hat{y}} \bmod p$. Note $pc_p \in \mathbb{Z}_p^\times$ since $v_p(pc_p) = -1 + 1 = 0$. Therefore

$$\hat{x} \bmod p = (p^2 c_p^2 - p^2 \underbrace{(x_1 + x_{p-1})}_{\in \mathbb{Z}_p}) \bmod p = (pc_p)^2 \bmod p$$

and

$$\hat{y} \bmod p = -p^3 c_p^3 + (pc_p) p^2 (x_1 + x_{p-1}) - p^3 d_p \bmod p = -(pc_p)^3 \bmod p.$$

Consequently,

$$\lambda_E(\widetilde{P}) = \frac{(pc_p)^2 \bmod p}{-(pc_p)^3 \bmod p} = \left(-\frac{1}{p} \frac{x_{p-1} - x_1}{y_{p-1} - y_1}\right) \bmod p. \qquad (6.12)$$

This completes the proof. □

The algorithm to compute λ_E and its time analysis can be found in [24, 3.6]. The following procedure computes $\lambda_E(\widetilde{P})$ for $\widetilde{P} = (s, t) \in \tilde{E}(\mathbb{F}_p) \backslash \{0\}$ in $O((\log p)^3)$ time.

(1) Find $P := (X_1, Y_1) \in E$ with $X_1, Y_1 \in \mathbb{Z}/p^2\mathbb{Z}$ satisfying $\widetilde{X_1} = s$ $\widetilde{Y_1} = t$.

(2) Compute $(X_{p-1}, Y_{p-1}) := [p-1]P \in E$ with $X_1, Y_1 \in \mathbb{Z}/p^2\mathbb{Z}$.

(3) If $\widetilde{X}_{p-1} \neq \widetilde{X}_1$, then

$$\lambda_E(\widetilde{P}) = \left(\frac{\widetilde{X_{p-1} - \widetilde{X}_1}}{p}\right)\left(\left(\widetilde{Y}_{p-1} - \widetilde{Y}_1\right)\right)^{-1}.$$

Otherwise $\lambda_E = 0$.

Note that under the same notation as in 6.6, we have only to compute $y_{p-1} - y_1 \bmod p$ and $\frac{1}{p}(x_{p-1} - x_1) \bmod p$. For (1), simply take any $X_1, y \in \mathbb{Z}/p^2\mathbb{Z}$ satisfying $X_1 \bmod p = s$ and $y \bmod p = t$. Then solve the following equation on w:

$$\begin{aligned} & (y + pw)^2 = X_1^3 + aX_1 + b \bmod p^2 \\ \Rightarrow\ & y^2 + 2ypw + \underbrace{p^2w^2}_{\equiv 0 \bmod p^2} = X_1^3 + aX_1 + b \bmod p^2 \\ \Rightarrow\ & 2tw = \frac{X_1^3 + aX_1 + b - y^2}{p} \bmod p. \end{aligned}$$

Note that the right hand side is well defined. Since from (6.5) we get $t \neq 0 \pmod p$ and $p \neq 2$, we obtain $w \in \mathbb{F}_p$. Then put $Y_1 := y + pw$. We can pass from a solution mod p to a true solution (i.e. with coefficients in $\mathbb{Z}_p$), see [27, Chap. 2, §2.2]. Since 6.6(ii) guarantees that the denominator in the recursion formula is not a multiple of p, it is invertible modulo p^2.
Remark: One can show that the computations in the proof of 6.6 can be done only using operations over $\mathbb{Z}/p^2\mathbb{Z}$.
By (6.11), we see $X_{p-1} \neq X_1$ under the condition $\lambda_E \neq 0$. In this case, 6.6(iii) ensures validity of Step (3). Otherwise λ_E must be the zero map. So, $\lambda_E(\widetilde{P}) = 0$.

The number of arithmetic operations over $\mathbb{Z}/p^2\mathbb{Z}$ involved in Steps (1) and (3) are indifferent to p or $\tilde{E}$. Step (2) requires at most $2\log_2 p$ elliptic curve additions [30, XI.1.1]. Summing up, $O((\log p)^3)$ Steps are enough to compute $\lambda_E(\widetilde{P})$.

With some additional technical details one can also show [24, 3.8] that there is no problem getting a non-zero λ_E. In total, this means the SSSA-algorithm runs in $O((\log p)^3)$ time.

7. Shor's Algorithm for the Discrete Logarithm

In the last few years there was a lot of interest in the field of quantum computing, even companies are trying to build their own quantum computer. IBM has currently one operating with about 50 qubits [11]. Also the funding for quantum computing by nations is stepping up. For example the European Union launches the Quantum Flagship program in 2019 with a volume of one billion euros. Through this attention from the industry as well as the academic sector the best known algorithm for quantum computing got new attention. Peter Shor published in 1994 polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer [28]. The great importance for us is that the algorithm can be applied to the group of elliptic curves as we will see. But first we have to introduce the basic theory and notation about quantum computing from the perspective of a mathematician.

7.1. Mathematics of Quantum Computing

A rigorous and axiomatic introduction to the mathematics of quantum computing is the book [25]. For our purpose the following short introduction of the main ideas of quantum information and a solid understanding of linear algebra is enough.

In a quantum computer the data is stored in so called qubits instead of bits. A qubit is a quantum mechanical state and is mathematically expressed as an orthonormal vector of a two dimensional Hilbert space $^{\P}\mathbb{H}$. The elements of a given orthonormal basis of this space are denoted by $|0\rangle$ and $|1\rangle$. Every qubit $|\psi\rangle \in {}^{\P}\mathbb{H}$ can be represented in the following way

$$|\psi\rangle = a|0\rangle + b|1\rangle \quad \text{with } a, b \in \mathbb{C} : |a|^2 + |b|^2 = 1.$$

If we measure the quantum state $|\psi\rangle$ it returns $|0\rangle$ with probability $|a|^2$ and returns $|1\rangle$ with probability $|b|^2$. Once the qubit gets measured the values of a and b are lost. The property that a qubit before it is measured is simultaneously $|0\rangle$ and $|1\rangle$ is called superposition. Besides the superposition there is another property about qubits that is fundamental different from the classical bits. A two qubit register is just the tensor product of two qubits. If $|\psi\rangle = c_{0,0}|0\rangle\otimes|0\rangle + c_{0,1}|0\rangle\otimes|1\rangle + c_{1,0}|1\rangle\otimes|0\rangle + c_{1,1}|1\rangle\otimes|1\rangle \in {}^{\P}\mathbb{H}^{\otimes 2}$ is not of the form $|\psi_1\rangle \otimes |\psi_2\rangle$ we call the state $|\psi\rangle$ entangled. For example

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}}|1\rangle \otimes |1\rangle$$

is entangled. When we measure $|\psi\rangle$ we get with probability $1/2$ the state $|0\rangle \otimes |0\rangle$ and with probability $1/2$ the state $|1\rangle \otimes |1\rangle$. If we just measure the first qubit and get $|0\rangle$ we automatically know that the second qubit is also $|0\rangle$.

To manipulate data in quantum computers we us quantum gates. Mathematically spoken this are unitary operators on $^{\P}\mathbb{H}$. Two simple quantum gates are:

$$\begin{array}{rcc} & \text{IN} & \text{OUT} \\ \text{Not :} & |0\rangle & |1\rangle \\ & |1\rangle & |0\rangle \end{array} \qquad\qquad \begin{array}{rcc} & \text{IN} & \text{OUT} \\ \text{Hardamard :} & |0\rangle & \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ & |1\rangle & \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{array} \tag{7.1}$$

We want to demonstrate the ability of a quantum computer to process data in parallel. Let us have a look at the addition modulo 2 [21]. We work with a 3-qubit-register. Therefore let the D-gate be a quantum gate that works as follows. The third qubit is the sum of the first two qubits if the third qubit starts in the state $|0\rangle$. When the third qubit starts in the state $|1\rangle$ we define the output such that the whole operations is unitary. This can be visualized as below:

INPUT	OUTPUT
$\|0\rangle \otimes \|0\rangle \otimes \|0\rangle$	$\|0\rangle \otimes \|0\rangle \otimes \|0\rangle$
$\|0\rangle \otimes \|0\rangle \otimes \|1\rangle$	$\|0\rangle \otimes \|0\rangle \otimes \|1\rangle$
$\|0\rangle \otimes \|1\rangle \otimes \|0\rangle$	$\|0\rangle \otimes \|1\rangle \otimes \|1\rangle$
$\|0\rangle \otimes \|1\rangle \otimes \|1\rangle$	$\|0\rangle \otimes \|1\rangle \otimes \|0\rangle$
$\|1\rangle \otimes \|0\rangle \otimes \|0\rangle$	$\|1\rangle \otimes \|0\rangle \otimes \|1\rangle$
$\|1\rangle \otimes \|0\rangle \otimes \|1\rangle$	$\|1\rangle \otimes \|0\rangle \otimes \|0\rangle$
$\|1\rangle \otimes \|1\rangle \otimes \|0\rangle$	$\|1\rangle \otimes \|1\rangle \otimes \|0\rangle$
$\|1\rangle \otimes \|1\rangle \otimes \|1\rangle$	$\|1\rangle \otimes \|1\rangle \otimes \|1\rangle$

We start with a 3-qubit-register in the state $|0\rangle \otimes |0\rangle \otimes |0\rangle$ and apply the Hadarmard-gate on the first and second qubit. This puts the first two qubits in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and the register in the state

$$\frac{1}{2}|0\rangle \otimes |0\rangle \otimes |0\rangle + \frac{1}{2}|0\rangle \otimes |1\rangle \otimes |0\rangle + \frac{1}{2}|1\rangle \otimes |0\rangle \otimes |0\rangle + \frac{1}{2}|1\rangle \otimes |1\rangle \otimes |0\rangle.$$

Applying the D-gate only to the third register leads to

$$\frac{1}{2}|0\rangle \otimes |0\rangle \otimes |0\rangle + \frac{1}{2}|0\rangle \otimes |1\rangle \otimes |1\rangle + \frac{1}{2}|1\rangle \otimes |0\rangle \otimes |1\rangle + \frac{1}{2}|1\rangle \otimes |1\rangle \otimes |0\rangle.$$

The result of this operations is that the first and the second qubit are entangled with its sum modulo 2, i.e. a measurement of the quantum state gives us random numbers in the first two registers and their sum modulo 2 in the third register. Since we can not control the sum we measure, this algorithm is not a speed up over the classical computer. Algorithms for quantum computers have to be cleverly designed to get the desired result with high probability.

As in the literature we will from now on omit the tensor product symbol $\otimes$, for reasons of readability.

7.2. The Fourier Transform for Finite Abelian Groups

Shor's algorithm for factoring integers as well as Shor's algorithm for computing the discrete logarithm are based on the Fourier transform. More precisely on the discrete Fourier transform for abelian groups. An early description of this transform with quantum computing methods can be found in [13]. This paper is also the foundation for this subsection.

Let G be a finite abelian group. Then we know that there exists a Hilbert space $\mathbb{H}$ with an orthonormal basis $\{|g\rangle : g \in G\}$. Consider the map (shift)

$$h : |g\rangle \mapsto |hg\rangle, \qquad h, g \in G. \tag{7.2}$$

To be able to define the Fourier transform in the rather general case of finite abelian groups we have to introduce a new basis in our Hilbert space generated by the group elements. The basis elements $|\chi_i\rangle$ should be invariant under the map (7.2), i.e.

$$g|\chi_i\rangle = \phi_i(g)|\chi_i\rangle \qquad \text{for all } g \in G \text{ and } i = 1, \dots, |G|,$$

where $\phi_i : G \to \mathbb{C}$. The existence of such states is due to the fact that all operations commute and are unitary.

Let $\chi : G \to \mathbb{C}^\times$ be a (multiplicative) character on the group G, i.e. χ is a group homomorphism from G to the multiplicative group of the complex numbers. We have the following standard results for characters.

Proposition 7.1. *Let G be a finite abelian group and let $\chi, \rho : G \to \mathbb{C}^\times$ be two characters on G. Then*

(a)

$$\frac{1}{|G|} \sum_{g \in G} \chi(g)\overline{\rho(g)} = \begin{cases} 1 & \textit{if } \chi = \rho \\ 0 & \textit{ohterwise} \end{cases}.$$

(b) There are $|G|$ different characters on G.

Proof. For a proof we refer to [8]. □

Condition (a) ensures that the Fourier transform is unitary. This is very important since every operation on qubits has to be unitary (because of physics).

For a character χ on G consider the state

$$|\chi\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} \overline{\chi(g)}|g\rangle.$$

When we look at 7.1(a) we see that the states $\{|\chi_1\rangle, \dots, |\chi_{|G|}\rangle\}$ form an orthonormal basis of $\mathbb{H}$, called the Fourier basis. Furthermore these basis states are shift-invariant:

$$h|\chi\rangle = \chi(h)|\chi\rangle \;\; h \in G. \tag{7.3}$$

Proof.

$$\begin{aligned}
\chi(h)|\chi\rangle = &\ \chi(h)\left(\frac{1}{\sqrt{|G|}}\sum_{g\in G}\overline{\chi(g)}|g\rangle\right)\\
= &\ \frac{1}{\sqrt{|G|}}\sum_{g\in G}\overline{\chi(g)}\ \overline{\chi(h^{-1})}|g\rangle\\
= &\ \frac{1}{\sqrt{|G|}}\sum_{g\in G}\overline{\chi(gh^{-1})}|g\rangle\\
\overset{\tilde{g}=gh^{-1}}{=} &\ \frac{1}{\sqrt{|G|}}\sum_{g\in G}\overline{\chi(\tilde{g})}|\tilde{g}h\rangle\\
&= h\left(\frac{1}{\sqrt{|G|}}\sum_{g\in G}\overline{\chi(\tilde{g})}|\tilde{g}\rangle\right)\\
&= h|\chi\rangle
\end{aligned}$$

□

If we now choose an ordering $g_1, \dots, g_{|G|}$ of the elements of G and an ordering $\chi_1, \dots, \chi_{|G|}$ of the characters on G we can define the Fourier transform on the abelian group G. It is the unitary transformation which maps $|\chi_i\rangle$ to $|g_i\rangle$, for $1 \le i \le |G|$.

Example 7.2. For Shor's algorithm the situation when $G = \mathbb{Z}/n\mathbb{Z}$ is of special interest. Then the n functions χ_k are defined by

$$\chi_k(1) = e^{2\pi ik/n} \qquad k = 0, \dots, n-1$$

and by the properties of χ_k:

$$\chi_k(\overline{m}) = \chi_k(1)^m = \exp\left(\frac{2\pi ikm}{n}\right) \quad \forall\overline{m} \in \mathbb{Z}/n\mathbb{Z}. \tag{7.4}$$

Definiton 7.3. *Let $n \in \mathbb{N}$ and $\{|0\rangle, \dots, |n-1\rangle\}$ be an orthonormal basis for a Hilbert space $\mathbb{H}$. The Quantum Fourier Transform (QFT) is a map which is defined as follows*

$$\begin{aligned}
&\mathcal{F}_n : \mathbb{H} \longrightarrow \mathbb{H}\\
&\sum_{j=0}^{n-1}|j\rangle \longmapsto \sum_{j=0}^{n-1}\frac{1}{\sqrt{n}}\sum_{k=0}^{n-1}\chi_k(j)|k\rangle.
\end{aligned}$$

Something interesting is happening if look what the QFT does to the input $|0\rangle$:

$$\begin{aligned}
\mathcal{F}_n|0\rangle &= \frac{1}{\sqrt{n}}\sum_{k=0}^{n-1}\chi_k(0)|k\rangle = \frac{1}{\sqrt{n}}\sum_{k=0}^{n-1}\exp\left(\frac{2\pi i0k}{n}\right))|k\rangle\\
&= \frac{1}{\sqrt{n}}\sum_{k=0}^{n-1}|k\rangle.
\end{aligned}$$

In other words we can transform the state $|0\rangle$ in uniform superposition of the states $|0\rangle, \dots, |n-1\rangle$ by applying the QFT $\mathcal{F}_n$.

7.2.1. Efficient Computation of the Quantum Fourier Transform

The classical fast Fourier transform (FFT) algorithm (applicable to certain groups) has a runtime complexity of $O(|G|\log|G|)$ but this, in itself, does not suffice for our quantum algorithm since it is still exponential in $\log|G|$. It may be seen that in a quantum context the implementation of the FFT algorithm combined with extra non-classical properties of entanglement provides an algorithm which runs in $O(\text{poly}(\log|G|))$ time. This feature has been elaborated in [12].

7.3. Shor's Algorithm

Let p be a prime number and let $\bar{g}$ be a generator of the multiplicative group $(\mathbb{Z}/p\mathbb{Z})^\times = \{\overline{1}, \ldots, \overline{p-1}\}$. Given $\overline{x} \in (\mathbb{Z}/p\mathbb{Z})^\times$ we are trying to solve the DLP

$$\overline{g}^r = \overline{x}, \qquad \text{for } 0 \leq r \leq p-2.$$

The following algorithm, discovered by Peter Shor [29], shows how a quantum computer finds discrete logarithms. First we give a high level description of the algorithm and then look at each step more closely.

(1) Choose $q \in \mathbb{N}$ as a power of 2 with $p \leq q \leq 2p$.

(2) Initialize three registers each consisting of q qubits to 0.

(3) Apply the QFT to the first two registers.

(4) Compute $g^a x^{-b}$ modulo p and write the result in the third register, where a, b are the values of the first two registers.

(5) Apply the Fourier-transform to the first two registers.

(6) Measure the quantum state.

(7) If possible compute r from the result of the measurement.
Otherwise start again with Step (2)

ad (1) The reason why we have to choose such a q in the first step is solely a issue of an efficient implementation of the QTF, as it is with implementations on conventional computers. In this description we will assume $q = p$, as found in [37].

ad (2) Let $\{|0\rangle, \ldots, |p-2\rangle\}$ be an orthonormal basis for a Hilbert space $\mathbb{H}_1$ and $\{|1\rangle, \ldots, |p-1\rangle\}$ be an orthonormal basis for a Hilbert space $\mathbb{H}_2$. The three input registers are represented by the following tensor product of Hilbert spaces

$$\mathbb{H}_1 \otimes \mathbb{H}_1 \otimes \mathbb{H}_2.$$

The two first registers are initialized to $|0\rangle$ and the third to $|1\rangle$.

ad (3) In the third step the QFT $\mathcal{F}_{p-1}$ is applied to the first two registers, which leaves them in a uniform superposition of all possible classical inputs $|a\rangle \pmod{(p-1)}$. The third register stays unchanged, i.e. the quantum computer is in the following state.

$$\mathcal{F}_{p-1}|0\rangle \otimes \mathcal{F}_{p-1}|0\rangle \otimes |0\rangle = \frac{1}{\sqrt{p-1}} \sum_{a=0}^{p-2} |a\rangle \otimes \frac{1}{\sqrt{p-1}} \sum_{b=0}^{p-2} |b\rangle \otimes |0\rangle = \frac{1}{p-1} \sum_{a=0}^{p-2} \sum_{b=0}^{p-2} |a\rangle|b\rangle|0\rangle.$$

ad (4) We then create a series of gates that receives the content of the first two registers as input and compute $g^a x^{-b} \bmod p$. The output is written into the third register. That such quantum gates for modular arithmetic exists is shown in detail in [25, 5.4]. The new state is:

$$\frac{1}{p-1} \sum_{a=0}^{p-2} \sum_{b=0}^{p-2} |a\rangle|b\rangle|g^a x^{-b} \bmod p\rangle.$$

ad (5) As in step (3) we apply the QFT $\mathcal{F}_{p-1}$ to the first two register resulting in the entangled state

$$\begin{aligned}
\frac{1}{p-1}\Bigg(\mathcal{F}_{p-1} \sum_{a=0}^{p-2} |a\rangle \otimes \mathcal{F}_{p-1} \sum_{b=0}^{p-2} |b\rangle \otimes |g^a x^{-b} \bmod p\rangle\Bigg) &= \frac{1}{p-1}\Bigg(\sum_{a=0}^{p-2} \frac{1}{\sqrt{p-1}} \sum_{c=0}^{p-2} \exp\left(\frac{2\pi iac}{p-1}\right) |c\rangle \\
&\otimes \sum_{b=0}^{p-2} \frac{1}{\sqrt{p-1}} \sum_{d=0}^{p-2} \exp\left(\frac{2\pi ibd}{p-1}\right) |d\rangle \otimes |g^a x^{-b} \bmod p\rangle\Bigg) \\
&= \frac{1}{(p-1)^2} \sum_{a,b,c,d=0}^{p-2} \exp\left(\frac{2\pi i}{p-1}(ac+bd)\right) |c\rangle|d\rangle|g^a x^{-b} \bmod p\rangle.
\end{aligned}$$

ad (6) We perform the measurement of the current state of our quantum computer. Note that we have the following identity $\overline{g}^a \overline{x}^{-b} = \overline{g}^a (\overline{g}^r)^{-b} = \overline{g}^{a-rb}$. Then the probability that we get a particular state $|c\rangle|d\rangle|g^k \bmod p\rangle$ is

$$\begin{aligned}
\mathbb{P}\Big[|c\rangle|d\rangle|g^k \bmod p\rangle\Big] &= \left| \frac{1}{(p-1)^2} \sum_{\substack{(a,b=0) \\ a-rb\equiv k}}^{p-2} \exp\left(\frac{2\pi i}{p-1}(ac+bd)\right) \right|^2 && (7.5)\\
&= \left| \frac{1}{(p-1)^2} \sum_{b=0}^{p-2} \exp\left(\frac{2\pi i}{p-1}((k+rb)c+bd)\right) \right|^2 && (7.6)\\
&= \left| \frac{1}{(p-1)^2} \sum_{b=0}^{p-2} \exp\left(\frac{2\pi i}{p-1}(kc+rbc+bd)\right) \right|^2. && (7.7)
\end{aligned}$$

ad (7) If $\overline{d} + \overline{rc} = 0$, then we can compute r by solving

$$\overline{r} = -\overline{c}^{-1}\overline{d},$$

using the extended Euclidean algorithm provided that $\gcd(c, p-1) = 1$. This can be done in polynomial time on a classical computer.

In the case that $\overline{d} + \overline{rc} \neq 0$ or $\gcd(c, p-1) \neq 1$ we start again with step (2) an set the registers to their start value.

Analyses of (7):
Let us compute the probability of observing a state $|c\rangle|d\rangle|g^k \bmod p\rangle$ where $\overline{d} + \overline{rc} \neq 0$. We factor out b in the nominator of (7.7)

$$\begin{aligned}\mathbb{P}\left[|c\rangle|d\rangle|g^k \bmod p\rangle \wedge \overline{d} + \overline{rc} \neq 0\right] &= \left|\frac{1}{(p-1)^2}\sum_{b=0}^{p-2}\exp\left(\frac{2\pi i}{p-1}\left(kc + b(rc+d)\right)\right)\right|^2 \\ &= \left|\frac{1}{(p-1)^2}\exp\left(\frac{2\pi ikc}{p-1}\right)\sum_{b=0}^{p-2}\exp\left(\frac{2\pi ib(rc+d)}{p-1}\right)\right|^2 \\ &= \left|\frac{1}{(p-1)^2}\exp\left(\frac{2\pi ikc}{p-1}\right)\cdot 0\right|^2 = 0\end{aligned}$$

So the first condition in (7) does not posse a problem. The only thing left to check is: How high is the probability of getting a c such that $\gcd(1, p-1) = 1$ is? We already answered a similar question in the analysis of the MOV algorithm for supersingular curves. Note that

$$\mathbb{P}[\gcd(c, p-1) = 1] = \frac{\phi(p-1)}{p-1} > \frac{1}{\log p}.$$

for infinitely many p. Hence, we are expecting only polynomial many rounds.

Application to Elliptic Curves
This algorithm does not use many properties of $\mathbb{Z}/p\mathbb{Z}$, so we can use the same algorithm to find the discrete logarithm over other groups. All we need is that we know the order of the generator and that the group operation and taking inverses can be done in polynomial time. An identically algorithm as above in the notation of an elliptic curve group can be found in [4].

References

[1] Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid. Recommendation for key management part 1: General (revision 3). *NIST special publication*, 800(57):1–147, 2012.

[2] Johannes Buchmann. *Einführung in die Kryptographie*, volume 3. Springer, 2001.

[3] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[4] Jodie Eicher and Yaw Opoku. Using the quantum computer to break elliptic curve cryptosystems. 1997.

[5] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[6] NSA Fact Sheet. Suite b cryptography. *NSA, URL: http://www. nsa. gov/ia/industry/crypto_suite_b. cfm*, 2008.

[7] William Fulton. Algebraic curves. *An Introduction to Algebraic Geom*, page 54, 2008.

[8] William Fulton and Joe Harris. *Representation theory: a first course*, volume 129. Springer Science & Business Media, 2013.

[9] Robin Hartshorne. *Algebraic geometry*, volume 52. Springer Science & Business Media, 2013.

[10] Everett W Howe. The weil pairing and the hilbert symbol. *Mathematische Annalen*, 305(1):387–392, 1996.

[11] IBM. *IBM Announces Advances to IBM Quantum Systems & Ecosystem*, besucht am 12.09.2018. https://www-03.ibm.com/press/us/en/pressrelease/53374.wss.

[12] Richard Jozsa. Entanglement and quantum computation. *arXiv preprint quant-ph/9707034*, 1997.

[13] Richard Jozsa. Quantum algorithms and the fourier transform. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 454, pages 323–337. The Royal Society, 1998.

[14] Jonathan Katz, Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography.* CRC press, 1996.

[15] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[16] Serge Lang. *Algebraic number theory*, volume 110. Springer Science & Business Media, 2013.

[17] Serge Lang. *Abelian varieties.* Dover Publications, 2019.

[18] Rudolf Lidl and Harald Niederreiter. *Introduction to finite fields and their applications.* Cambridge university press, 1994.

[19] Alfred J Menezes, Tatsuaki Okamoto, and Scott A Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *iEEE Transactions on information Theory*, 39(5):1639–1646, 1993.

[20] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[21] Tony Phillips. *The Mathematics Behind Quantum Computing: Part II*, besucht am 20.08.2018. `http://www.ams.org/publicoutreach/feature-column/fcarc-quantum-two/`.

[22] J Barkley Rosser, Lowell Schoenfeld, et al. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, 6(1):64–94, 1962.

[23] Jörg Rothe. *Complexity Theory and Cryptology.* Springer-Verlag Berlin Heidelberg, 2005.

[24] Takakazu Satoh and Kiyomichi Araki. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Rikkyo Daigaku sugaku zasshi*, 47(1):81–92, 1998.

[25] Wolfgang Scherer. *Mathematik der Quanteninformatik.* Springer-Verlag, Berlin-Heidleberg, 2016.

[26] Igor Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Mathematics of Computation of the American Mathematical Society*, 67(221):353–356, 1998.

[27] Jean-Pierre Serre. *A course in arithmetic.* Springer Science & Business Media, 1973.

[28] Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. Ieee, 1994.

[29] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[30] Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.

[31] Nigel P Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of cryptology*, 12(3):193–196, 1999.

[32] Paul C Van Oorschot and Michael J Wiener. Parallel collision search with cryptanalytic applications. *Journal of cryptology*, 12(1):1–28, 1999.

[33] Guido Walz. *Lexikon der Mathematik: Band 2: Eig bis Inn.* Springer-Verlag, 2016.

[34] Lawrence C Washington. *Elliptic curves: number theory and cryptography.* Chapman and Hall/CRC, 2003.

[35] William C Waterhouse and JS Milne. *Abelian varieties over finite fields.* PhD thesis, Harvard University, 1968.

[36] Annette Werner. *Elliptische Kurven in der Kryptographie.* Springer, 2002.

[37] Song Y. Yan. *Quantum Attacks on Public-Key Cryptosystems.* Springer US, 2013.

A. Appendix

A.1. Discrete Valuation

Definiton A.1. *Let K be a field. A surjective map*

$$v : K \longrightarrow \mathbb{Z} \cup \{\infty\}$$

is called a discrete valuation on K if the following properties for every element $a, b \in K$ hold:

(a) $v(ab) = v(a) + v(b)$.

(b) $v(a + b) \geq \min(v(a), v(b))$.

(c) $v(a) = \infty \Leftrightarrow a = 0$.

Proposition A.2. *Let K be a field and $v : K \to \mathbb{Z}$ a discrete valuation on K. Then the following statements hold for $a, b \in K$:*

(a) $v(-a) = v(a)$.

(b) $v(a - b) \geq \min(v(a), v(b))$.

(c) $v(a) \neq v(b) \Rightarrow v(a + b) = min(v(a), v(b))$.

(d) $v(1) = 0$.

(e) $v(a^{-1}) = -v(a)$.

A.2. p-adic numbers

Definiton A.3. *Let p be a prime number and for every $n \in \mathbb{N}$ let*

$$\begin{aligned} \pi_n : \mathbb{Z}/p^{n+1}\mathbb{Z} &\longrightarrow \mathbb{Z}/p^n\mathbb{Z} \\ x + p^{n+1}\mathbb{Z} &\longmapsto x + p^n\mathbb{Z}, \end{aligned}$$

where x is an integer. Then the integral domain of p-adic integers is the set

$$\mathbb{Z}_p := \{(x_n)_{n \geq 1} : x_n \in \mathbb{Z}/p^n\mathbb{Z} \mid \pi_n(x_{n+1}) = x_n \quad \forall n \in \mathbb{N}\},$$

together with componentwise addition and multiplication.
If we take the quotient field of this ring, we get the field of the p-adic integers

$$\mathbb{Q}_p = \left\{ \frac{a}{b} \mid a \in \mathbb{Z}_p, b \in \mathbb{Z}_p \backslash \{0\} \right\}.$$

Proposition A.4. *Let p be a prime number. Then*

$$\mathbb{Z}_p^\times = \mathbb{Z}_p \backslash p\mathbb{Z}_p.$$

Proof. See, [36, 6.9.1]. □

From this result we see that we can write every p-adic integer $a \in \mathbb{Z}_p$ as $a = p^n u$, where n is an non-negative integer and $u \in \mathbb{Z}_p^\times$ is a unit. This means we can write every element $x \in \mathbb{Q}_p$ in the form $x = p^m u$, where m is an integer and $u \in \mathbb{Z}_p^\times$. Therefore we have a valuation on $\mathbb{Q}_p$. For $x \in \mathbb{Q}_p$ the valuation of x is $v_p(x)$, the largest integer v such that $x \in p^v \mathbb{Z}_p$.
A simple fact that will often be helpful when we calculate the valuation of an element is the following on

$$a \in \mathbb{Z}_p^\times \Leftrightarrow v_p(a) = 0. \tag{A.1}$$

Theorem A.5 ([36]). *(Hensel's lemma)*
Let $f(x) = a_n x^n + \cdots + a_1 x + a_0 \in \mathbb{Z}_p[x]$ be a polynomial with coefficients in $\mathbb{Z}_p$. Further let $\pi(f)(x) = \pi(a_n)x^n + \cdots + \pi(a_1)x + \pi(a_0) \in \mathbb{F}_p[x]$. If $\pi(f)$ has a zero $\alpha \in \mathbb{F}_p$ with a non-zero derivative at α, then f has a zero $\beta \in \mathbb{Z}_p$ such that $\pi(\beta) = \alpha$.

Proof. A proof (of a more general version) for example can be found in [16, p.46]. □