university of groningen / faculty of science and engineering / mathematics and applied mathematics

The Discrete Logarithm Problem on Supersingular Elliptic Curves

Bachelor's Project Mathematics

July 2020

Student: R. Smit

First supervisor: Dr. M. Djukanovic

Second assessor: Dr. P. Kilicer

The Discrete Logarithm Problem on Supersingular Elliptic Curves

Roelien Smit

Abstract

The elliptic curve discrete logarithm problem is an essential problem in cryptography. In general it is a very complex problem; the best known solving algorithms all have exponential running time. However, for supersingular elliptic curves there exists a subexponential solving algorithm called the MOV attack. The MOV attack reduces an elliptic curve discrete logarithm to a logarithm over a finite field using the Weil pairing. The discrete logarithm problem in a finite field can be solved efficiently using Index Calculus. This thesis deals with analyzing the MOV attack and generating examples to demonstrate its power.

Contents

1 Introduction

The study of cryptography, the area in mathematics that deals with concealing data, originates from the time people started to use written communication. In the early days there were no advanced systems for exchanging secret messages. Messages were encrypted by using straightforward ciphers; a letter in the message was replaced by another letter or symbol. In the 14'th and 15'th century, symmetric key encryption methods occurred for the first time. Encryption methods based on a secret key work as follows. Two people, say Alice and Bob, want to share secret, confidential data. The idea is that Alice and Bob first meet in person to agree on a common secret. This common secret can then be used to encrypt and decrypt the confidential data that has to be shared. This is a symmetric key encryption method since Alice and Bob both have the same amount of information. The fact that Alice and Bob meet in person seems to be crucial in this secret key encryption. However, in 1976 Diffie and Hellman introduced the notion of public key encryption [3]. It appeared that Alice and Bob meeting in person is not necessary at all, since the secret key that is used to encrypt the data can be constructed by exchanging information via a public channel.

Since the introduction of public-key cryptography by Diffie and Hellman, the discrete logarithm problem has been recognized as an important tool in cryptography. A few years later, Elgamal explained how the discrete logarithm problem can be used in public key encryption systems and digital signature schemes [4]. All in all, the discrete logarithm problem has had, and still has, an enormous impact on cryptographical systems.

1.1 Discrete logarithm problem

The discrete logarithm problem for a certain group G can be described as the following problem. For a group G and an element g in the group, the element $h = g^n$ is given for an unknown integer n. Then, given the elements g and h, the task is to find the positive integer n such that $h = g^n$.

The safety of a lot of cryptographical systems, for example the Diffie and Hellman key exchange, relies mainly on the complexity of the discrete logarithm problem. To illustrate this: the usage and importance of the discrete logarithm problem in a cryptographical system, the Diffie and Hellman key exchange will be described here.

Alice and Bob want to exchange secret data via a public channel. It is assumed that Alice and Bob are only able to communicate via the public channel. In order to safely exchange the data, Alice and Bob want to agree on a secret key. To do so, Alice and Bob publicly select a group G such that the discrete logarithm problem is difficult to solve in G. Furthermore, they publicly select an element $g \in G$. Now Alice chooses a secret integer a and computes g^a. Then Alice sends the element g^a to Bob via the public channel. Bob does the same: he chooses a secret integer b, and computes g^b and sends g^b to Alice. Alice computes $(g^b)^a = g^{ba}$ and Bob computes $(g^a)^b = g^{ab}$. Alice and Bob now both know the secret element g^{ab}, from which they can construct a secret key. Publicly, the only things that are known are the group G, the element g and the elements g^a and g^b. This means that if an eavesdropper, say Eve, is able to solve the discrete logarithm problem in the group G in a reasonable amount of time,

then Eve can find out the value of the integers a and b and use this to derive the secret key g^{ab}. Therefore, the complexity of the discrete logarithm problem in the group G determines the safety of the public key exchange.

In 1985, the mathematicians Koblitz and Miller suggested to use elliptic curves in cryptography [14]. This appeared to be a groundbreaking suggestion, as elliptic curve discrete logarithms have proven to be very interesting and particularly difficult to solve. Nowadays, elliptic curves form the foundation of many cryptographical applications based on the discrete logarithm problem and they also appear in many other fields of cryptography.

1.2 Elliptic Curves

An elliptic curve defined over a field K is an algebraic object that is defined as a planar curve, whose points are the solutions of the so-called Weierstrass equation. A Weierstrass equation is an equation of the form

$$y^2 = x^3 + Ax + B. \tag{1}$$

The variables x, y and constants A, B take on values from the field K. Not all Weierstrass equations define an elliptic curve: elliptic curves are not allowed to be singular. This means that the Weierstrass equation of an elliptic curve is not allowed to have multiple roots, the coefficients always satisfy

$$4A^3 + 27B^2 \neq 0.$$

An elliptic curve always contains a special point 'at infinity'. In some sense this point can be seen as the 'top' and 'bottom' of the y-axis. To give a rigorous explanation of the point 'at infinity', the projective plane will be introduced in Section 2.3.

The K-points of an elliptic curve are defined to be the set

$$E(K) = \{\infty\} \cup \{(x, y) \in K \times K \quad | \quad y^2 = x^3 + Ax + B\}.$$

In fact, the set of K-points of an elliptic curve has an abelian group structure with the point at infinity as identity element. The group law that defines this group structure is defined by the statement that the sum of three points on an elliptic curve E is equal to the identity element if and only if the three points are the intersection of E and a line. The projective plane will allow for the derivation of this definition of the group law. In the Section 2.1, the group law will be explained both geometrically and algebraically.

1.3 Elliptic curve discrete logarithms

The currently best known algorithms for solving the elliptic curve discrete logarithm problem for general elliptic curves are all exponential time algorithms. However, there are exceptions to this. The discrete logarithm problem for the class of supersingular elliptic curves, which are curves of a specific order defined over a finite field, as will be explained in Section 5, can be solved a lot faster. In fact, for supersingular elliptic curves there exists a sub-exponential solving algorithm. This algorithm was introduced in 1996 by Menezes, Okamoto and Vanstone, and it is known as the MOV attack.

The main goal in this thesis is to analyse the MOV attack and to construct examples to show the power of the MOV attack. The MOV attack reduces the elliptic curve discrete logarithm problem to a discrete logarithm problem in a finite field by using the Weil pairing. The discrete logarithm problem in the finite field can then be solved efficiently by means of Index Calculus methods.

Analyzing the MOV attack, by studying the Weil pairing and Index Calculus, requires a more complete understanding of elliptic curves in general, elliptic curves over a finite field, supersingular elliptic curves and the discrete logarithm problem. Finally, in Section 7, the MOV attack will be applied to different examples of elliptic curve discrete logarithm problems for supersingular elliptic curves.

From the theory and the actual performance of the MOV attack, it will become clear at the end of this thesis why supersingular elliptic curves are not suitable for applications in cryptography based on the discrete logarithm problem.

2 General theory of elliptic curves

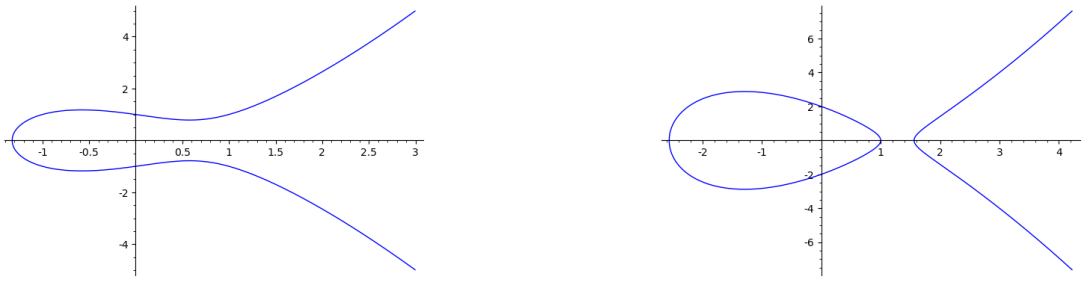
An elliptic curve defined over a field K can be defined in more generality as a planar curve whose points satisfy the general Weierstrass equation,

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

where x, y are variables that take values in the field K and a_1, a_2, a_3, a_4, a_6 are constants from the field K. If the characteristic of the field K is not equal to 2 or 3, a change of variables can always transform the general Weierstrass equation to a Weierstrass equation [26]. This transformation involves dividing by 2 and 3 and completing the square and cube. For convenience, in this paper a finite field is always assumed to have characteristic not equal to 2 or 3. Therefore equation (1) will be the only equation to work with.

An important observation is that an elliptic curve that is given by a Weierstrass equation is always symmetric with respect to the x-axis. This means that if the point (x, y) is on the elliptic curve, then $(x, -y)$ is also a point on the elliptic curve.

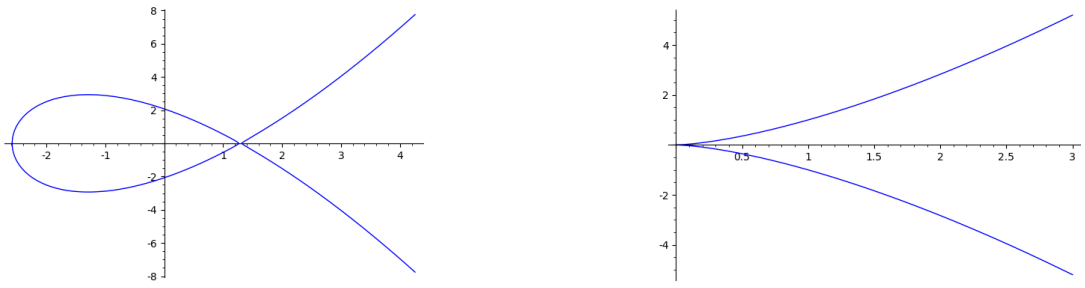
The cubic $x^3 + Ax + B$ is not allowed have multiple roots, i.e. the curve is not allowed to be singular. A curve given by a singular Weierstrass equation is not considered an to be an elliptic curve. An elliptic curve thus never has self-intersections or cusps. The condition $4A^3+27B^2 \neq 0$ implies that the elliptic curve has no singularities since $4A^3 + 27B^2$ is the discriminant of the cubic equation $f(x) = x^3 + Ax + B$.



(a) (b)

Figure 1: Elliptic curves over $\mathbb{Q}$

Figure 1 depicts two elliptic curves over the field $\mathbb{Q}$, the points of these curves satisfy the Weierstrass equation. The curves in Figure 2 also satisfy the Weierstrass equation. However, these curves are not elliptic curves. Figure 2a has a self-intersection and Figure 2b has a cusp. The plots in Figure 1 give a nice intuition on elliptic curves, but it is good to keep in mind that elliptic curves over an arbitrary field do not in general take on this form.



(a) (b)

Figure 2: Singular curves over $\mathbb{Q}$

The point at infinity is essential when doing algebraic operations with points of an elliptic curve. The two main properties of the point at infinity are that any two vertical lines intersect at that point and additionally, the point at infinity is both the 'top' and the 'bottom' of the y-axis and all lines parallel to the y-axis. The projective plane, which will be introduced in Section 2.3 will enable finding specific coordinates for the point at infinity of an elliptic curve.

2.1 The group law of an elliptic curve

The set of points of an elliptic curve E defined over a field K has an abelian group structure with the point at infinity as identity element. The group of K-points of E is denoted by

$$E(K) = \{(x, y) \in K \times K \mid y^2 = x^3 + Ax + B\} \cup \{\infty\}.$$

Here K can be replaced by any suitable field extension, meaning that for every extension $L \supset K$, the L-points of the elliptic curve are denoted by $E(L)$. The procedure that allows for the group structure is sometimes referred to as the "tangent and chord method" [17]. It will shortly become clear why this name is suitable. The procedure, which requires only a few simple operations, yields the possibility of finding more K-points of an elliptic curve when starting with only one or two K-points. The goal of this section is to formally describe the group law, using both a geometric and an algebraic explanation.

2.1.1 Geometric explanation

Let E be an elliptic curve defined over a field K. Then the addition law on E can be described as follows.

Start with two different points, P and Q on E. Then there is a unique line passing through these points. This line will always intersect the elliptic curve in one additional point, R' (see also Section 2.3). Now reflect the point R' in the x-axis to obtain some point R. Note that this is possible since the elliptic curve is symmetric with respect to the x-axis. This reflection in the x-axis sends the point (x, y) on E to the point $(x, -y)$ on E. Define $P \oplus Q = R$ (see Figure 3).

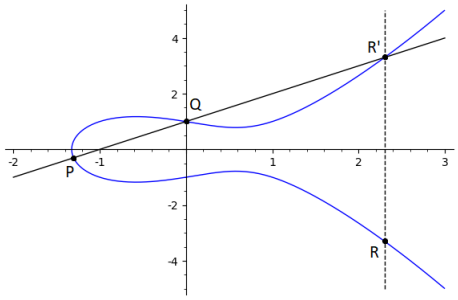


Figure 3: Adding two different points from $E(\mathbb{Q})$

When adding a point to itself the method is a little bit different. Start with a point P on E. There is a unique line that is tangent to E and that passes through the point P. This line will intersect the curve E in one other point, which is denoted by R'. Reflect the point R' with respect to the x-axis. This will give some new point R on E. Define $P \oplus P = 2 \cdot P = R$ (see Figure 4). There is one technical exception here. If the y-coordinate of the point P is equal to 0, then the sum $P \oplus P = 2 \cdot P$ is defined to be ∞.

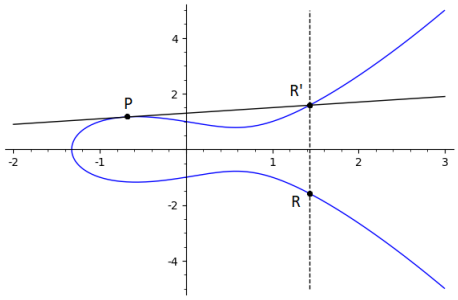


Figure 4: Adding a point from $E(\mathbb{Q})$ to itself

There is another technicality when adding a point to itself. Let P be an 'inflection' point of E. When adding P to itself, the third point of intersection of the tangent line of E and E itself is again the point P; the point P is an intersection point of multiplicity 3. Therefore in this case $P \oplus P = -P$, or $3 \cdot P = \infty$.

There are two other cases that need to be considered. First, let P be a point on the elliptic curve E, and let O be the point at infinity. Now the line through P and O is a vertical line. Because of the symmetry of the curve E, the line intersects E in a point P' that is the reflection of P across the x-axis. Reflecting the point P' across the x-axis therefore results in the point P. Hence in this case it holds that $P \oplus O = P \oplus \infty = P$.

For the second case, let P and Q be two points on the elliptic curve E that have the same x-coordinate. The line through the points P and Q is now a vertical line. This line intersects the elliptic curve in the point at infinity. Reflecting this point ∞ across the x-axis again yields again the point at infinity. This is clear from the definition of the point at infinity of an elliptic

curve; it is both the 'top' and 'bottom' of the y-axis. Therefore in this case, $P \oplus Q = \infty$, i.e. $Q = -P$.

Remark. Note that Figures 3 and 4 are plots of an elliptic curve defined over the field of rational numbers. These plots serve as an intuitive picture for how the group law works. However, it is important to keep in mind that these pictures do not give a general geometric representation of the group law for an elliptic curve defined over an arbitrary field.

2.1.2 Algebraic explanation

The group law can be described by rational functions. This means that adding two points will only involve polynomials and/or fractions of polynomials.

Let $E\colon y^2 = x^3 + Ax + B$ be an elliptic curve and let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points on E. First assume that $\infty \neq P_1 \neq P_2 \neq \infty$ and that $x_1 \neq x_2$. The line L through P_1 and P_2 has slope

$$m = \frac{y_2 - y_1}{x_2 - x_1},$$

and the equation of L is equal to

$$y = m(x - x_1) + y_1.$$

To find the point P_3' on the curve E, which is the third point of intersection (besides P_1 and P_2) of the line L and the curve E, one has to substitute the equation of L into the equation of the elliptic curve E. This yields

$$(m(x - x_1) + y_1)^2 = x^3 + Ax + B.$$

This is equivalent to

$$x^3 - m^2x^2 + (A - 2m^2x_1 - 2my_1) - m^2x_1^2 + 2mx_1y_1 - y_1^2 + B = 0.$$

To find the solution to this equation, the roots of a cubic polynomial have to be found. Two roots of this polynomial are already known: x_1 and x_2. Assume that x_3 is the third solution to this equation. Then

$$x^3 - m^2x^2 + ... = (x - x_1)(x - x_2)(x - x_3) = x^3 - (x_1 + x_2 + x_3)x^2 + \dots,$$

which implies that $m^2 = x_1 + x_2 + x_3$, and therefore

$$x_3 = m^2 - x_1 - x_2. \tag{2}$$

By substituting the point x_3 into the Weierstrass equation and reflecting the point across the x-axis, the corresponding y-coordinate of the point $P_3 = P_1 \oplus P_2$ on the elliptic curve can be found. This gives

$$y_3 = m(x_1 - x_3) - y_1. \tag{3}$$

There is a special case when $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are two points on the elliptic curve E such that $x_1 = x_2$, but $y_1 \neq y_2$. In this case the line through the points P_1 and P_2 is a vertical line. In this case equations (2) and (3) are defined in such a way that $P_1 + P_2 = \infty$.

Now assume that $P_1 = P_2 = (x_1, y_1)$, with $y_1 \neq 0$, is a point on the elliptic curve E. The slope of the tangent line L at this point can be found using implicit differentiation.

$$2y\frac{\mathrm{d}y}{\mathrm{d}x} = 3x^2 + A \quad \Rightarrow \quad m = \frac{\mathrm{d}y}{\mathrm{d}x} = \frac{3x_1^2 + A}{2y_1}.$$

The equation of the tangent line L is given by

$$y = m(x - x_1) + y_1.$$

The point P_3' can be found similarly as before, by substituting the equation for the tangent line into the Weierstrass equation. Again the roots of a cubic polynomial have to be found. This time, one root of multiplicity 2 is already known, namely x_1. This yields

$$x^3 - m^2x^2 + \cdots = (x - x_1)^2(x - x_3) = x^3 - (2x_1 + x_3)x^2 + \ldots$$

This means that the coordinates of P_3' are given by

$$x_3' = m^2 - 2x_1 \quad \text{and} \quad y_3' = m(x_3' - x_1) + y_1.$$

Therefore $P_3 = P_1 \oplus P_2 = 2 \cdot P_1 = (x_3, y_3)$, where

$$x_3 = m^2 - 2x_1 \quad \text{and} \quad y_3 = m(x_1 - x_3) - y_1.$$

The special cases where P_1 and P_2 are two points on the elliptic curve such that either $P_1 = P_2$ and $y_1 = 0$ or $P_2 = \infty$ were already (geometrically) explained in the previous section.

Remark. This derivation is extended to the case where the point at infinity, the identity element of the group, is added to itself. It holds that $\infty \oplus \infty = \infty$.

Finally, the group structure on the group of points of an elliptic curve can be defined properly.

Theorem 2.1. *The points of an elliptic curve form an abelian group under the group law described above, where ∞ is the identity element.*

This means that the points of an elliptic curve E satisfy the group axioms.

1. **Associativity:** $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$ for all points P, Q, R on E.
2. **Identity:** $P \oplus \infty = \infty \oplus P = P$ for all points P on E.
3. **Inverses:** For each point P on E there exists a point P' on E such that $P \oplus P' = \infty$.
4. **Commutativity:** $P \oplus Q = Q \oplus P$ for all points P, Q on E.

Proof. From the geometric and algebraic construction of the group law, the identity element, inverses and commutativity of the group are all clear. The proof of the associativity is quite elaborate and it is not the focus of this thesis. A very nice proof, based on the Riemann-Roch theorem [9], can be found in [21]. □

Remark. If P is a point on an elliptic curve and the goal is to compute $k \cdot P$, it is not very efficient to add P to itself subsequently until $k \cdot P$ is reached. A faster approach for computing

multiples of P is known as *successive doubling* [26]. This works as follows. To compute for example $21 \cdot P$, first compute

$$2 \cdot P, \quad 4 \cdot P = 2 \cdot P + 2 \cdot P, \quad 8 \cdot P = 4 \cdot P + 4 \cdot P, \quad 16 \cdot P = 8 \cdot P + 8 \cdot P,$$

then

$$21 \cdot P = 16 \cdot P + 4 \cdot P + P.$$

This method correctly computes multiples of P because of the associativity of the group law.

2.2 General forms and isomorphisms

Elliptic curves given by a Weierstrass equation are only a very special type of elliptic curves, they are written in a very convenient way. However, it must be said that there are also other forms of elliptic curves and other ways to generate elliptic curves. For example, elliptic curves can be defined in higher dimensions as intersections of surfaces. Some such curves can be reduced to the general Weierstrass equation. This thesis will only focus on elliptic curves given by a Weierstrass equation. For elliptic curves defined by other equations, such as the Legendre equation,

$$y^2 = x(x-1)(x-\lambda), \quad \lambda \in K \setminus \{0, 1\},$$

see [26].

The remaining part of this section will be devoted to transformations of elliptic curves. The transformations that are of most interest are isomorphisms, the transformations that preserve the group structure of an elliptic curve. Formally, two elliptic curves defined over a field K are isomorphic if there exists a bijection between them, given by rational maps, that preserves the group structure. Over an algebraically closed field it will be possible to fully characterize elliptic curves up to isomorphism.

To describe transformations of elliptic curves more precisely, the *j-invariant* of an elliptic curve should be introduced. The j-invariant of an elliptic curve E over K given by the Weierstrass equation is given by

$$j = j(E) = 1728\frac{4A^3}{4A^3 + 27B^2}.$$

Theorem 2.2. *Let E_1 and E_2 be two elliptic curves defined over a field K, given by $E_1 \colon y^2 = x^3 + A_1x + B_1$ and $E_2 \colon y^2 = x^3 + A_2x + B_2$, with j-invariants j_1 and j_2. If $j_1 = j_2$ then E_1 and E_2 are transformations of each other, meaning that there exists a constant $\mu \neq 0$ in the field $\overline{K}$, such that*

$$A_2 = \mu^4 A_1, \quad B_2 = \mu^6 B_1.$$

The transformation is given by $(x', y') = (\mu^2 x, \mu^3 y)$.

Over a field that is not necessarily algebraically closed it is true that two elliptic curves that are isomorphic have the same j-invariant, but the converse statement only holds when the field K is algebraically closed.

Example 2.3. The elliptic curves given by $y^2 = x^3+4x$ and $y^2 = x^3+3x$ defined over the field $\mathbb{Q}$ (which is not algebraically closed) both have j-invariant equal to 1728. The latter curve has infinitely many points with coordinates in $\mathbb{Q}$, namely all the integer multiples of the point $(3, 6)$ and all the integer multiples of the point $(1, 2)$. However, the first curve has only finitely many points with coordinates in $\mathbb{Q}$; the only points with rational coordinates on $y^2 = x^3 + 4x$ are $\infty, (2, 4), (2, -4)$ and $(0, 0)$. This means that there is no rational transformation over the field $\mathbb{Q}$ that transforms the first elliptic curve into the second elliptic curve, even though both curves have the same j-invariant. According to Theorem 2.2, over the field $\mathbb{Q}(\sqrt[4]{12})$ it is possible to find a transformation that transforms one curve into the other. That transformation is $(x, y) \mapsto (\mu^2 x, \mu^3 y)$, where $\mu = \frac{\sqrt[4]{12}}{2}$. ■

Example 2.4. The elliptic curves given by $E_1 \colon y^2 = x^3 + 3$ and $E_2 \colon y^2 = x^3 + 8$ defined over the field $\mathbb{F}_{13}$ (which is not algebraically closed) both have j-invariant equal to 0. However, the groups of points of the elliptic curves have different order, namely $\#E_1(\mathbb{F}_{13}) = 9$ and $\#E_2(\mathbb{F}_{13}) = 16$. This means that the curves E_1 and E_2 cannot be isomorphic over $\mathbb{F}_{13}$. In other words, there exists no rational transformation over the field $\mathbb{F}_{13}$ that transforms E_1 into E_2. As stated in Theorem 2.2, over the field $\mathbb{F}_{13}(\zeta_6)$, where ζ_6 denotes a sixth root of unity, it is possible to find a transformation that transforms one curve into the other. The transformation is given by $(x, y) \mapsto (\mu^2 x, \mu^3 y)$, where $\mu = 4\zeta_6^5 + 8\zeta_6^4 + 2\zeta_6^3 + 2\zeta_6^2 + 10$. ■

Two different elliptic curves defined over the field K that have the same j-invariant are called *twists*. From Theorem 2.2 it is clear that twists are unique up to isomorphism and a twist is isomorphic to the original curve over $\overline{K}$.

If a general curve C has the same group structure as an elliptic curve E that is given by a Weierstrass equation and the curve C can be transformed into the elliptic curve E via a rational transformation over the field K, then the curve C and the elliptic curve E are called isomorphic. For the details of general isomorphisms between curves, see [21]. If a curve C is isomorphic to an elliptic curve, then C is also called an elliptic curve.

2.3 The projective plane

In this section the projective plane will be introduced. The point ∞ of an elliptic curve can be defined more formally and precisely using the notion of the projective plane. The projective plane will also give rise to another, simpler, definition of the group law on an elliptic curve.

The affine plane over a field K is given by

$$\mathbf{A}_K^2 = \{(x, y) \in K \times K\}.$$

The affine plane is just the set of coordinates (x, y), the vector space structure does not play a role in $\mathbf{A}_k^2$. In the affine plane it is known that two lines that are not parallel intersect in one point, but what happens if the lines are parallel? Imagine that you are standing in the middle of a straight road. The sides of the road are parallel lines, that do not seem to intersect at any point. However, when you look at the horizon, the two sides of the road meet. This gives the intuition that two parallel lines do intersect somewhere on the line at the horizon.

Using this explanation one could say that parallel lines meet at a line at infinity, but there is no such line at infinity in the affine plane. This is where the notion of the projective plane has to

be introduced. The projective plane will be the plane where every two lines intersect in exactly one point and in particular parallel lines intersect on the line at infinity. The projective plane will eventually lead to finding coordinates for the points at infinity in general, and specifically coordinates for the point at infinity of an elliptic curve.

2.3.1 An extension of the affine plane

The projective plane $\mathbf{P}^2_K$ is an extension of the affine plane $\mathbf{A}^2_K$, containing some points 'at infinity'.

The projective plane is always introduced over a field K, just like the affine plane. The projective plane $\mathbf{P}^2_K$ over a field K, consists of triples (x, y, z), where x, y, z are elements of the field K that are not all zero. Moreover, in the projective plane two elements (x, y, x) and (ax, ay, az) are considered to be equal for all $a \in K^\times$. This means that elements in the projective plane are triples (x, y, z) modulo the equivalence relation $(x, y, z) \sim (ax, ay, az)$. A triple (x, y, z) in the projective plane is the same as another triple (p, q, r) in the projective plane if and only if one is a scalar multiple of another, i.e.

$$(x, y, z) \sim (p, q, r) \Leftrightarrow (x, y, z) = (\lambda p, \lambda q, \lambda r).$$

A triple (x, y, z) in the projective plane can thus be seen as an equivalence class. The equivalence class of the triple (x, y, z) is denoted by $(x : y : z)$.

The projective plane contains affine points and points at infinity. The affine points in the projective plane are all the points $(x : y : z)$ such that z is not equal to zero. This means that as representative for the affine points in the projective plane the triple $\left(\frac{x}{z}, \frac{y}{z}, 1\right)$ can be chosen and therefore the affine points in the projective plane are all the triples (x, y, z) such that $z = 1$. There is a one-to-one correspondence between the affine points in the projective plane $\mathbf{P}^2_K$ and points in the affine plane $\mathbf{A}^2_K$. The correspondence is given by

$$\mathbf{A}^2_k \hookrightarrow \{\text{affine points of } \mathbf{P}^2_K\}, \quad (x, y) \mapsto (x : y : 1).$$

Therefore, the affine plane is identified with the affine points in the projective plane.

The infinite points or points at infinity in the projective plane are points of the form $(x : y : 0)$. At the end of this section it will be possible to identify the point at infinity of an elliptic curve with one of those points.

The fact that every two lines in the projective plane intersect in one point follows more precisely from Bézout's theorem, which holds in the projective plane.

Theorem 2.5 (Bézout's theorem)**.** *Two distinct curves in $\mathbf{P}^2(\overline{K})$ of degree m and n intersect in mn points, counting multiplicities.*

Since lines are curves of degree 1, Bézout's theorem implies that every two lines in the projective plane intersect in one point [6]. The proof of Bézout's theorem is outside the scope of this thesis. It can be found in [9].

2.3.2 Homogeneous polynomials

To make sure that the affine plane is properly embedded in the projective plane it is also necessary that there is a correspondence between curves in the affine plane and curves in the projective plane. A curve in the affine plane is described as the set of zeros of a polynomial in $K[x, y]$, and a curve in the projective plane is described as the set of zeros of a polynomial in $K[x, y, z]$. To make a proper correspondence, the set of zeros of a polynomial in the affine plane needs to be equal to the set of zeros of a polynomial in the projective plane. This can be guaranteed by introducing homogeneous polynomials.

A *homogeneous polynomial* in three variables is a polynomial in which each summand has the form $ax^iy^jz^k$, where a is an element in K and $n = i + j + k$, where n is the degree of the polynomial.

Let $f(x, y)$ be a polynomial in two variables of degree n. This polynomial can be homogenized in the following way: $F(x, y, z) = z^n f\left(\frac{x}{z}, \frac{y}{z}\right)$. Now $F(x, y, z)$ is a homogeneous polynomial in three variables of degree n. This polynomial has zeros in the projective plane. The set of zeros $(x : y : z)$ of $F(x, y, z)$ is well defined. Let (x, y, z) and $(\lambda x, \lambda y, \lambda z)$ be two different representatives from the equivalence class $(x : y : z)$. Then $F(\lambda x, \lambda y, \lambda z) = \lambda^n F(x, y, z)$, since F is a homogeneous polynomial. This means that $F(x, y, z) = 0$ implies that $F(\lambda x, \lambda y, \lambda z) = 0$. Therefore the $F(x, y, z)$ being zero does not depend on the choice of representative of the equivalence class and hence the set of zeros of $F(x, y, z)$ is well defined.

Now it can be explained with full precision what it means for two parallel lines to meet at infinity. Let $y = mx + b_1$ and $y = mx + b_2$ be two parallel lines in the affine plane. Homogenizing these two equations gives the lines $y = mx + b_1z$ and $y = mx + b_2z$ in the projective plane. These two lines intersect when $z = 0$ (as expected, because points in the projective plane with $z = 0$ are points at infinity) and $y = mx$. Since not all among x, y, z are allowed to be equal to zero, the point of intersection is given by:

$$(x : y : 0) = (x : mx : 0) = (1 : m : 0).$$

Now, consider two vertical lines $x = a$ and $x = b$. Homogenizing these equations gives the lines $x = az$ and $x = bz$. These two lines in the projective plane intersect when $z = 0$ and $x = 0$, so the point of intersection of two vertical lines is given by $(0 : 1 : 0)$.

2.3.3 Point at infinity of an elliptic curve

Using the projective plane, the coordinates of the point at infinity of an elliptic curve can be found. To do so, let E be an elliptic curve in the affine plane given by a Weierstrass equation, so E is given by $y^2 = x^3 + Ax + B$. This curve corresponds to an elliptic curve in the projective plane that is given by a homogeneous Weierstrass equation $y^2z = x^3 + Axz^2 + bz^3$. To find the point at infinity of the elliptic curve, let $z = 0$. Then it follows that $x^3 = 0$. Therefore, the point at infinity of an elliptic curve in the projective plane is the triple $(x : y : z)$ such that $x = z = 0$, since not all x, y, z are allowed to be equal to zero. Therefore $(0 : y : 0) = (0 : 1 : 0)$ is the only point at infinity on an elliptic curve.

As shown before, the point $(0 : 1 : 0)$ is a point on every vertical line. This means that the point at infinity of an elliptic curve has the property that any vertical line intersects the elliptic curve

in this point, as was predicted by Bézout's theorem and already stated in the introduction. Furthermore, it holds that $(0 : 1 : 0) = (0 : -1 : 0)$, which means that indeed the 'top' and 'bottom' of the y-axis are identified.

2.3.4 Restating the group law

The specific form of the point at infinity of an elliptic curve, $(0 : 1 : 0)$, allows for a simpler definition of the group law. Denote the point at infinity of an elliptic curve by $O = (0 : 1 : 0)$.

Let E be an elliptic curve defined over a field K. Consider the elliptic curve in the projective plane, meaning that all the points are the points that satisfy the homogenized Weierstrass equation and the point O. Now consider a line in the projective plane, a line is a curve given by a polynomial of degree 1. Since the elliptic curve is an equation of degree 3, Bézout's theorem implies that the line and the elliptic curve have 3 intersection points in the projective plane. Now the group-law can be described by the following rule.

Let P, Q be two points on the elliptic curve and let L be the line through P and Q (if P is equal to Q, then let L be the tangent line at P). Let R be the third point of intersection of the line L and the elliptic curve E and let L' be the line through O and R. The line L' intersects the elliptic curve E in the points R, O and a third point, R'. Then $P \oplus Q = R'$ [21]. This is equivalent to saying that the sum of three points on the elliptic curve is equal to O in the projective plane if and only if they are the intersection of the elliptic curve E and a line.

In this thesis, elliptic curves will usually be defined in the affine plane. When it is necessary, the projective plane will be recalled.

2.4 Endomorphisms

Previously, isomorphisms of elliptic curves were introduces to deal with the fact that, in some sense, two different equation can describe the same algebraic object. Another important family of maps are the endomorphisms of an elliptic curve. Let E be an elliptic curve defined over a field K. Then an endomorphism of E is a homomorphism of E that is given by rational functions. In other words, an endomorphism of E is a map $\alpha\colon E(\overline{K}) \to E(\overline{K})$ and there exist functions $R(s, y)$ and $Q(s, y)$ that are rational polynomials or quotients of polynomials such that

$$\alpha(x, y) = (R(x, y), Q(x, y)).$$

Furthermore, since α is a homomorphism it holds that $\alpha(P_1 \oplus P_2) = \alpha(P_1) \oplus \alpha(P_2)$ for any two points P_1 and P_2 on E. The fact that α is a homomorphism also implies that α maps the point ∞ of an elliptic curve to itself. This may seem nontrivial, since it does not seem obvious how an endomorphism α is defined at the point at infinity. A more formal explanation of the fact that $\alpha(\infty) = \infty$ can be found in [21], which uses algebraic geometry to deal with the point at infinity.

This section will describe a general form for endomorphisms of an elliptic curve and moreover a few important properties of an endomorphism of an elliptic curve will be stated. Lastly, there is a special endomorphism of an elliptic curve that requires extra attention, namely the Frobenius

endomorphism. This endomorphism will be essential when dealing with elliptic curves defined over a finite field.

2.4.1 General form of an elliptic curve endomorphism

It is useful to find a general form for an endomorphism of an elliptic curve.

Consider an elliptic curve defined over a field K, where the elliptic curve is written in Weierstrass form, $y^2 = x^3 + Ax + B$. Let $\alpha = (R(x,y), Q(x,y))$. Then using the Weierstrass equation, every even power of y in $R(x,y)$ and $Q(x,y)$ can be replaced by an expression that only depends on the variable x. This means that the two rational functions in the endomorphism α can be written in the form:

$$R(x,y) = \frac{p_1(x) + p_2(x)y}{p_3(x) + p_4(x)y}, \quad Q(x,y) = \frac{q_1(x) + q_2(x)y}{q_3(x) + q_4(x)y}, \tag{4}$$

where $p_i, q_i \in K[x]$. Then by multiplying the numerator and denominator of $R(x,y)$ by $p_3(x) - p_4(x)y$ and the numerator and denominator of $Q(x,y)$ by $q_3(x) - q_4(x)y$ and furthermore again replacing any term y^2 according to the Weierstrass equation, $R(x,y)$ and $Q(x,y)$ can be simplified even more. This gives

$$R(x,y) = \frac{p_1(x) + p_2(x)y}{p_3(x)}, \quad Q(x,y) = \frac{q_1(x) + q_2(x)y}{q_3(x)}. \tag{5}$$

Note that the rational functions p_i, q_i in equation (5) are not the same as the functions p_i, q_i in equation (4).

For a point on an elliptic curve it holds that $-(x,y) = (x,-y)$, which means that $\alpha(x,-y) = \alpha(-(x,y))$, and by applying properties of an homomorphism it holds that $\alpha(x,-y) = -\alpha(x,y)$. Furthermore, since

$$\alpha(x,-y) = (R(x,-y), Q(x,-y)),$$

and

$$-\alpha(x,y) = -(R(x,y), Q(x,y)) = (R(x,y), -Q(x,y)),$$

it follows that $R(x,-y) = R(x,y)$ and $Q(x,-y) = -Q(x,y)$.

Consequently, by applying this to the simplified form of $R(x,y)$ and $Q(x,y)$ from equation (5), it holds that $p_2(x) = 0$ and $q_1(x) = 0$.

Therefore in general an endomorphism can be written in the form

$$\alpha(x,y) = (r_1(x), r_2(x)y).$$

2.4.2 Properties of endomorphisms

Endomorphisms of elliptic curves can be characterized in different ways.

Let α be an endomorphism of an elliptic curve E. As was just shown, α can be written as $\alpha(x,y) = (r_1(x), r_2(x)y)$. Write $r_1(x) = \frac{p(x)}{q(x)}$. The *degree* of an endomorphism is defined as

$$\deg(\alpha) = \max\{\deg(p(x)), \deg(q(x))\}.$$

Furthermore, an endomorphism is called *separable* if $r_1'(x)$ is not identically zero. Consider the polynomial given by $P(x) = x^p$ as an example. This polynomial has derivative 0 in the finite field $\mathbb{F}_p$, so an endomorphism given by $\alpha(x, y) = (p(x), p(y))$ is not a separable endomorphism of an elliptic curve over $\mathbb{F}_p$.

It is possible to identify the degree of a separable endomorphism with the number of elements in the kernel of the endomorphism.

Proposition 2.6. *Let E be an elliptic curve defined over a field K and let $\alpha \neq 0$ be a separable endomorphism of E. Then*

$$\deg(\alpha) = \#\ker(\alpha),$$

where $\ker(\alpha)$ *denotes the kernel of the homomorphism* $\alpha\colon E(\overline{K}) \to E(\overline{K})$.

If the endomorphism α is not separable then

$$\deg(\alpha) > \#\ker(\alpha).$$

The idea of the proof is to show that, if α is separable then for a generic point $(a, b) \in E(\overline{K})$, there exist exactly $\deg(\alpha)$ many points $(x, y) \in E(\overline{K})$ such that $\alpha(x, y) = (a, b)$. This then implies that the kernel of α has $\deg(\alpha)$ many elements because α is a group homomorphism. Details of the proof can be found in [26].

A natural question that arises is whether an endomorphism of an elliptic curve is surjective.

Theorem 2.7. *Let E be an elliptic curve defined over a field K and let $\alpha \neq 0$ be an endomorphism of E. Then $\alpha\colon E(\overline{K}) \to E(\overline{K})$ is surjective.*

Let N be a positive integer. The multiplication by N map is a map that will be used to prove a statement about the order of the group of points of an elliptic curve. In fact, the multiplication by N map is an endomorphism.

Proposition 2.8. *Let E be an elliptic curve defined over a field K. Let $N \neq 0$ be a positive integer. Let the multiplication by N on E be given by*

$$N \cdot (x, y) = (R_N(x), yS_N(x)).$$

Then

$$\frac{R_N'(x)}{S_N(x)} = N.$$

This implies that the multiplication by N endomorphism is separable if and only if the characteristic p of the field K does not divide N.

For the proof of Theorem 2.7 and the proof of Proposition 2.8, a proof by induction, see [26].

2.4.3 Frobenius endomorphism

Let E be an elliptic curve E defined over a finite field $\mathbb{F}_q$, where $q = p^n$ for some prime p and positive integer n. In Section 4, elliptic curves defined over a finite field will be explained in detail. The Frobenius map ϕ_q is defined by

$$\phi_q(x, y) = (x^q, y^q).$$

Lemma 2.9. *The Frobenius map ϕ_q is an endomorphism of E of degree q that is not separable.*

Proof. Let E be an elliptic curve defined over a finite field, and let ϕ_q denote the Frobenius map. Since $\phi_q(x, y) = (x^q, y^q)$, the Frobenius map is given by rational functions and the degree of ϕ_q is equal to q. To check that ϕ_q is an homomorphism several different cases need to be considered. First it needs to be checked that $\phi_q((x_1, y_1) + (x_2, y_2)) = \phi_q(x_1, y_1) + \phi_q(x_2, y_2)$ using the addition formulas for adding two different points. To this end, let (x_1, y_1) and (x_2, y_2) be two different points in $E(\mathbb{F}_q)$. Let their sum be equal to (x_3, y_3). Then using the group law for adding two different points from Section 2.1.2, the coordinates of the sum are equal to

$$(x_3, y_3) = (m^2 - x_1 - x_2, m(x_1 - x_3) - y_1), \quad \text{where} \quad m = \frac{y_2 - y_1}{x_2 - x_1}.$$

Then

$$\phi_q((x_1, y_1) + (x_2, y_2)) = \phi_q(x_3, y_3) = (x_3^q, y_3^q) = ((m')^2 - x_1^q - x_2^q,\ m'(x_1^q - x_3^q) - y_1^q),$$

where

$$m' = \frac{y_2^q - y_1^q}{x_2^q - x_1^q}.$$

This can be rewritten to

$$\phi_q((x_1, y_1) + (x_2, y_2)) = (x_1^q, y_1^q) + (x_2^q, y_2^q) = \phi_q(x_1, y_1) + \phi_q(x_2, y_2).$$

Next, it has to be checked that $\phi_q(2(x_1, y_1)) = 2\phi_q(x_1, y_1)$ using the addition formulas for adding a point to itself. Let (x_1, y_1) be a point on the elliptic curve and write (x_3, y_3) as the sum of (x_1, y_1) with itself. Then using the group law for adding a point to itself from Section 2.1.2, the coordinates of (x_3, y_3) are equal to

$$(x_3, y_3) = (m^2 - 2x_1, m(x_1, x_3) - y_1), \quad \text{where} \quad m = \frac{3x_1^2 + A}{2y_1}.$$

By applying the Frobenius endomorphism, it follows that

$$\phi_q(2(x_1, y_1)) = \phi_q(x_3, y_3) = ((m')^2 - 2x_1^q,\ m'(x_1^q - x_3^q) - y_1^q),$$

where

$$m' = \frac{3^q (x_1^q)^2 + A^q}{2^q y_1^q}.$$

Note that in a field $\mathbb{F}_q$, $a^q = a$ for all $a \in \mathbb{F}_q$. Therefore

$$m' = \frac{3(x_1^q)^2 + A}{2y_1^q}.$$

Hence, the equation for $\phi_q(2(x_1, y_1))$ can be rewritten to

$$\phi_q(2(x_1, y_1)) = 2(x_1^q, y_1^q) = 2\phi_q(x_1, y_1).$$

Therefore, it can be concluded that ϕ_q is an endomorphism of E of degree q.

The map ϕ_q is not separable since q is equal to zero in $\mathbb{F}_q$. Hence the Frobenius map ϕ_q is an endomorphism of E of degree q that is not separable. □

2.4.4 Complex multiplication

The set of all endomorphisms of an elliptic curve forms a ring, the endomorphism ring. Usually, the endomorphism ring of an elliptic curve over $\mathbb{Q}$ is isomorphic to $\mathbb{Z}$. However, sometimes the elliptic curve has extra endomorphism, as in the next example.

Example 2.10. Consider the elliptic curve $E\colon y^2 = x^3 + x$ over $\mathbb{Q}$. This elliptic curve has an extra endomorphism. Namely if the point $(x, y) \in E(\overline{\mathbb{Q}})$ is a point on the elliptic curve then also the point $(-x, iy)$, where $i^2 = -1$, is a point on the elliptic curve. The map that maps the point (x, y) to the point $(-x, iy)$ can be seen as the multiplication by i endomorphism. Applying this endomorphism twice to the point (x, y) gives $(x, -y)$ which is the multiplication of (x, y) by -1. In fact, for the elliptic curve E the endomorphism ring is isomorphic to $\mathbb{Z}[i]$. ■

If the endomorphism ring of an elliptic curve is strictly larger than $\mathbb{Z}$ the elliptic curve is said to have complex multiplication. So an elliptic curve with complex multiplication has extra endomorphisms, its endomorphism ring contains integers and algebraic integers. Although complex multiplication is not the main subject in this thesis, some important results will be stated here regarding elliptic curves and complex multiplication.

2.4.5 Elliptic curves over $\mathbb{Q}$ with complex multiplication

Theorem 2.11. *Let E be an elliptic curve over $\mathbb{Q}$. Then the endomorphism ring of E is isomorphic to $\mathbb{Z}$ or to an order in an imaginary quadratic field.*

An imaginary quadratic field is a field that is given by

$$K = \mathbb{Q}(\sqrt{-d}) = \{a + b\sqrt{-d} \mid a, b \in \mathbb{Q}\}.$$

An order in an imaginary quadratic field is a subring of K of the form $\mathbb{Z}[\alpha]$, where α is an algebraic integer, i.e. it is a root of a monic polynomial with integer coefficients. The ring of integers O_k, is called maximal order of K as it contains all elements of K that are roots of polynomials with integer coefficients, it is the largest subring of K that is a finitely generated abelian group. This means that in fact an order is a subring of the ring of integers in K. For the proof of Theorem 2.11, see [26].

It is possible to find all the j-invariants of elliptic curves over $\mathbb{Q}$ such that the elliptic curve has complex multiplication (see Table 1, [22]). In Sections 4 and 5, the theory of complex multiplication will be applied to elliptic curves defined over a finite field and in particular to supersingular elliptic curves.

-262537412640768000	0
-147197952000	1728
-884736000	8000
-12288000	54000
-884736	287496
-32768	16581375
-3375	

Table 1: CM j-invariants (over $\mathbb{Q}$)

The list of j-invariants in Table 1 is related to the class number one problem. A number field has class number one if and only if its ring of integers is a principal ideal domain. The class number one problem states that all imaginary quadratic fields $\mathbb{Q}(\sqrt{-d})$, where $d > 0$ and d is square-free, having class number 1 are given by $\{\mathbb{Q}(\sqrt{-d}) \mid d = 1, 2, 3, 7, 11, 19, 43, 67, 163\}$ [23]. The number fields with class number one correspond to endomorphism rings of elliptic curves with complex multiplication. The class number problems are formulated and conjectured by Gauss. Heegner, Baker and Stark proved the statement about number fields of class number one. For the details of the (proof of the) class number one problem, see [23].

As a final remark regarding this topic, note that there are 13 j-invariants in Table 1, while there are only 9 imaginary quadratic fields having class number one. This has to do with the fact that some of the elliptic curves with j-invariant from Table 1 have endomorphism ring that is not the ring of integers, but a subring of the ring of integers; for more details see [22].

3 Torsion points

To get a better understanding of the group structure of an elliptic curve, the so-called torsion points will be studied. The torsion points of an elliptic curve defined over a field K are all the points of the elliptic curve with coordinates in the algebraic closure of K such that the order is finite. The torsion points will be especially useful later on in this thesis when studying elliptic curves over finite fields and the corresponding discrete logarithm problem. The torsion points of an elliptic curve will also be essential to construct the Weil pairing in Section 3.2.

Let E be an elliptic curve defined over a field K and let N be a positive integer. An N-torsion point of an elliptic curve is a point on the elliptic curve with coordinates in the algebraic closure of K that has order dividing N. The group of N-torsion points of an elliptic curve is denoted by $E[N]$, that is

$$E[N] = \{P \in E(\overline{K}) \mid N \cdot P = \infty\}.$$

3.1 Group structure

The group of N-torsion points of an elliptic curve has a very convenient group structure. In fact, up to isomorphism, there are only two types of groups of N-torsion points. To give some intuition on the group structure, first the group $E[2]$ will be determined.

Example 3.1. Let E be an elliptic curve defined over a finite field (that has characteristic not equal to 2). Then the Weierstrass equation can be written in the form

$$y^2 = \text{cubic equation in } x = (x - \lambda_1)(x - \lambda_2)(x - \lambda_3),$$

with $\lambda_1, \lambda_2, \lambda_3$ element from $\overline{K}$. A point P on this elliptic curve has order 2 if and only if $2 \cdot P = \infty$. From the group law, this is only possible when the tangent line at P is a vertical line. This implies that the y-coordinate of P is equal to 0. Therefore

$$E[2] = \{\infty, (\lambda_1, 0), (\lambda_2, 0), (\lambda_3, 0)\}.$$

This group has order 4 and elements of orders 1 and 2. This implies that $E[2]$ is isomorphic to the Klein four-group:

$$E[2] \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z}.$$

■

Proposition 3.2. *Let E be an elliptic curve over a field K and let N be a positive integer. If the characteristic of K does not divide N, or is 0, then*

$$E[N] \cong \mathbb{Z}/N\mathbb{Z} \oplus \mathbb{Z}/N\mathbb{Z}.$$

If the characteristic of K is $p > 0$ and $p|N$, write $N = p^r N'$ with $p \nmid N'$. Then

$$E[N] \cong \mathbb{Z}/N'\mathbb{Z} \oplus \mathbb{Z}/N'\mathbb{Z} \quad \text{or} \quad \mathbb{Z}/N\mathbb{Z} \oplus \mathbb{Z}/N'\mathbb{Z}.$$

Proof. A sketch of the proof will be given. The group $E[N]$ is the kernel of the multiplication by N endomorphism. Using division polynomials (Appendix A.3), specific formulas for the

multiplication by N endomorphism can be found. Recall that an endomorphism can be written as $\alpha(x, y) = (R(x), yS(x))$. For the multiplication by N endomorphism, it holds that

$$R(x) = \frac{x^{N^2} + \dots}{N^2 x^{N^2-1} + \dots}.$$

Then it follows that if the characteristic of K does not divide N, the multiplication by N endomorphism is separable. From Proposition 2.6 and Corollary 3.3, which is stated below, it follows that the kernel of the multiplication by N endomorphism has order N^2. Now using the Structure Theorem for finitely generated abelian groups (Appendix A.1) and a group structure related argument it follows that

$$E[N] \cong \mathbb{Z}/N\mathbb{Z} \oplus \mathbb{Z}/N\mathbb{Z}.$$

If the characteristic of K does divide N, then multiplication by N is not separable. Let p denote the characteristic of K, so that $N = p^r N'$. The p^r-torsion on the elliptic curve E is isomorphic to $\{\infty\}$ or $\mathbb{Z}/p^r\mathbb{Z}$, which can be deduced via an induction argument. Hence it follows that

$$E[N] \cong \mathbb{Z}/N'\mathbb{Z} \oplus \mathbb{Z}/N'\mathbb{Z} \quad \text{or} \quad \mathbb{Z}/N\mathbb{Z} \oplus \mathbb{Z}/N'\mathbb{Z}.$$

□

Corollary 3.3. *Let E be an elliptic curve defined over a field K and let N be a positive integer. The multiplication by N endomorphism of E has degree N^2.*

The proof of this corollary is also based on the division polynomials. Appendix A.3 offers a short introduction to division polynomials and it states the most important results. More theory on division polynomials, and also the proof of this corollary can be found in [26].

3.2 The Weil pairing

Bilinear pairings are maps that are important in different fields in mathematics. The dot product from linear algebra is an example of a bilinear pairing on the vector space $\mathbb{R}^n$ [9]. It takes as input two vectors from the vector space $\mathbb{R}^n$, and outputs a real number. For the bilinearity of the dot product observe that

$$(av_1 + bv_2) \cdot w = a(v_1 \cdot w) + b(v_2 \cdot w)$$
$$v \cdot (aw_1 + bw_2) = a(v \cdot w_1) + b(v \cdot w_2).$$

Another example of a bilinear pairing is the determinant pairing on the vector space $\mathbb{R}^2$:

$$\det \begin{bmatrix} v_1 & v_2 \\ w_1 & w_2 \end{bmatrix} = v_1 w_2 - v_2 w_1.$$

It maps two vectors in $\mathbb{R}^2$ to a real number. For elliptic curves there also exists a bilinear pairing, a map that takes as input two points on the elliptic curve and outputs an element of the field K. This bilinear pairing, which is called the Weil pairing, the main ingredient of the MOV attack, will be introduced in this section.

The N'th roots of unity of a field K are important to define the Weil pairing on an elliptic curve. An N'th root of unity is an element ζ_N in $\overline{K}$ such that $(\zeta_N)^N = 1$. These elements form a cyclic group; $\mu_N = \{x \in \overline{K} | x^N = 1\}$ is a cyclic group of order N. A primitive N'th root of unity is a generator ζ of the group μ_N. This means that $h = \zeta^k$ for all $h \in \mu_N$ and some integer k. The element ζ is a primitive N'th root of unity if and only if $\zeta^N = 1$ and $\zeta^k \neq 1$ for all $1 \leq k < N$.

Based on the N-torsion points of an elliptic curve and the N'th roots of unity the Weil pairing can be defined. Besides the fact that the Weil pairing is an important ingredient to attack the elliptic curve discrete logarithm problem, it is also essential in the proof of Hasse's theorem. The Weil pairing is defined as a special bilinear map; it associates an N'th root of unity with a given pair of N-torsion points.

Theorem 3.4 (Weil Pairing)**.** *Let E be an elliptic curve defined over a field K and let N be a positive integer. Assume that the characteristic of K does not divide N. Then there is a pairing*

$$e_N : E[N] \times E[N] \to \mu_N,$$

that is called the Weil pairing. It satisfies the following properties:

1. *e_N is bilinear in each variable:*

$$e_N(P_1 + P_2, Q) = e_N(P_1, Q)e_N(P_2, Q)$$

$$e_N(P, Q_1 + Q_2) = e_N(P, Q_1)e_N(P, Q_2)$$

2. *e_N is non-degenerate in each variable:*

$$\text{if} \quad e_N(P, Q) = 1 \quad \forall\, Q \in E[N], \quad \text{then} \quad P = \infty$$

$$\text{if} \quad e_N(P, Q) = 1 \quad \forall\, P \in E[N], \quad \text{then} \quad Q = \infty$$

3. *$e_N(Q, Q) = 1$ for all $Q \in E[N]$*
4. *$e_N(Q, P) = e_N(P, Q)^{-1}$ for all $P, Q \in E[N]$*
5. *$e_N(\sigma P, \sigma Q) = \sigma(e_N(P, Q))$ for all automorphisms σ of $\overline{K}$ s.t. σ is the identity on K.*
6. *$e_N(\alpha(P), \alpha(Q)) = e_N(P, Q)^{\deg(\alpha)}$ for all separable endomorphisms α of E.*

This theorem will be proven in Section 3.2.2. First, a few important corollaries will be stated, and some more theory and notation needs to be introduced.

Corollary 3.5. *Let $\{T_1, T_2\}$ be a basis for $E[N]$. This implies that $e_N(T_1, T_2)$ is a primitive N'th root of unity.*

Note that such a basis $\{T_1, T_2\}$ can be selected because of the group structure of the group of N-torsion points (see Theorem 3.2).

Proof. Recall that ζ is a primitive N'th root of unity if and only if $\zeta^N = 1$ and $\zeta^k \neq 1$ for all $1 \leq k < N$. Equivalently, ζ is a primitive N'th root of unity if and only if $\zeta^d = 1$ implies $N|d$. By definition of the Weil pairing, $e_N(T_1, T_2)$ is an N'th root of unity. Suppose that

$e_N(T_1, T_2) = \zeta$, with $\zeta^d = 1$. Because of the bilinearity and property (3) of the Weil pairing (see Theorem 3.4), the following two equalities hold:

$$\begin{aligned} e_N(T_1, d \cdot T_2) &= e_N(T_1, T_2)^d = \zeta^d = 1 \\ e_N(T_2, d \cdot T_2) &= e_N(T_2, T_2)^d = \zeta^d = 1. \end{aligned} \tag{6}$$

Let S be any point of order dividing N. Then $S \in E[N]$, so S can be written in the form $S = a \cdot T_1 + b \cdot T_2$, since T_1, T_2 form a basis for $E[N]$. Then

$$\begin{aligned} e_N(S, d \cdot T_2) &= e_N(a \cdot T_1 + b \cdot T_2, d \cdot T_2) \\ &= e_N(a \cdot T_1, d \cdot T_2) e_N(b \cdot T_2, d \cdot T_2), \\ &= e_N(T_1, d \cdot T_2)^a e_N(T_2, d \cdot T_2)^b, \\ &= 1 \end{aligned}$$

Here the second equality follows from the bilinearity property of the Weil pairing and the third and fourth equalities follow from equation (6). Since this holds for any point S of order dividing N, by property (2) of the Weil pairing it must be the case that $d \cdot T_2 = \infty$. Recall that $E[N] \cong \mathbb{Z}/N\mathbb{Z} \oplus \mathbb{Z}/N\mathbb{Z}$ and $\{T_1, T_2\}$ forms a basis for $E[N]$. This implies that $d \cdot T_2 = \infty$ if and only if d is a multiple of N. Therefore N divides d, and hence ζ is a primitive N'th root of unity. □

From the properties of the Weil pairing, in particular property 6, it is possible to connect the degree of an endomorphism to the determinant of a related endomorphism.

Remark. Let E be an elliptic curve defined over the field K and let $\alpha : E(\overline{K}) \to E(\overline{K})$ be an endomorphism. Then α maps group of N-torsion points of E to itself. The restriction endomorphism $\alpha_N : E[N] \to E[N]$ can be represented by a matrix. Pick a basis $\{\beta_1, \beta_2\}$ for the N-torsion points of an elliptic curve. Such a basis exists since $E[N] \cong \mathbb{Z}/N\mathbb{Z} \oplus \mathbb{Z}/N\mathbb{Z}$ (see Theorem 3.2). This means that there exist elements $a, b, c, d \in \mathbb{Z}/N\mathbb{Z}$ such that

$$\begin{aligned} \alpha(\beta_1) &= a\beta_1 + c\beta_2 \\ \alpha(\beta_2) &= b\beta_1 + d\beta_2. \end{aligned}$$

Hence the action of α on the n-torsion points of the elliptic curve E is represented by the matrix

$$\alpha_N = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Proposition 3.6. *Let E be an elliptic curve defined over a field K and let α be an endomorphism of E. Let N be a positive integer such that the characteristic of K does not divide N. Then*

$$\det(\alpha_N) \equiv \det(\alpha) \ \textit{mod}\ N.$$

Proof. The element $\zeta = e_N(T_1, T_2)$ is a primitive N'th root of unity by Corollary 3.5. From

properties of the Weil pairing, Theorem 3.4 and the observation, it now follows that

$$\begin{aligned}
\zeta^{\deg(\alpha)} &= e_N(T_1,T_2)^{\deg(\alpha)} \\
&= e_N(\alpha(T_1),\alpha(T_2)) \\
&= e_N(a\cdot T_1 + c\cdot T_2, b\cdot T_1 + d\cdot T_2) \\
&= e_N(T_1,T_1)^{ab} e_N(T_1,T_2)^{ad} e_N(T_2,T_1)^{cb} e_N(T_2,T_2)^{cd} \\
&= e_N(T_1,T_2)^{ad} e_N(T_2,T_1)^{cb} \\
&= e_N(T_1,T_2)^{ad} e_N(T_1,T_2)^{-cb} \\
&= e_N(T_1,T_2)^{ad-cb} \\
&= \zeta^{ad-cb}.
\end{aligned}$$

For the element ζ it holds that $\zeta^x = \zeta^y$ if and only if $x \equiv y \bmod N$, since ζ is a primitive N'th root of unity. Therefore $\deg(\alpha) \equiv ad - cb = \det(\alpha_N) \bmod N$. □

Let α and β be two endomorphisms of the elliptic curve E. Then $m\alpha + n\beta$ is an endomorphism of E that is defined by $(m\alpha + n\beta)(P) = m\alpha(P) + n\beta(P)$ for all points P of the elliptic curve E.

Proposition 3.7. $\deg(m\alpha + n\beta) = m^2\deg(\alpha) + n^2\deg(\beta) + mn(\deg(\alpha+\beta) - \deg(\alpha) - \deg(\beta))$

The proof of this proposition can be found in [26], it is a straightforward computation.

From the properties of the Weil pairing, it follows that the Weil pairing defines a surjective map as can be seen from Theorem 3.8 (see [21]).

Theorem 3.8. *There exist N-torsion points S and T such that $e_N(S,T)$ defines a primitive N'th root of unity. In particular, if $E[N] \subset E(K)$, then $\mu_N \subset K^\times$.*

Proof. Let σ be an automorphism of $\overline{K}$ such that σ is the identity map on K and let T_1, T_2 denote a basis for $E[N]$. Denote $\zeta = e_N(T_1,T_2)$. The points T_1 and T_2 both are N-torsion points of E, so they have coordinates in K. Therefore $\sigma T_1 = T_1$ and $\sigma T_2 = T_2$. From property (5) of the Weil pairing it now follows that

$$\zeta = e_N(T_1,T_2) = e_N(\sigma T_1, \sigma T_2) = \sigma(e_N(T_1,T_2)) = \sigma(\zeta).$$

The fundamental theorem of Galois theory implies that $\zeta \in K$. According to Corollary 3.5, ζ is a primitive N'th root of unity, so ζ is the generator of μ_N. Since the generator of μ_N is in K it follows that $\mu_N \in K$, and this concludes the proof of the theorem. □

3.2.1 Divisors

To construct the Weil pairing and prove the properties of the Weil pairing, the divisors of an elliptic curve need to be introduced. After defining the notion of a divisor of an elliptic curve, some important properties of divisors will be stated.

Let E be an elliptic curve defined over a field K. For each point $P \in E(\overline{K})$, a formal symbol $[P]$ can be defined. A divisor D of the elliptic curve E is defined as a formal sum

$$D = \sum_j a_j[P_j], \quad a_j \in \mathbb{Z}.$$

The coefficients a_j in the sum satisfy the property that $a_j = 0$ for all but finitely many values of j, therefore D is a finite formal sum. The set of all divisors of an elliptic curve forms a group, denoted $\mathrm{Div}(E)$. This group contains formal sums of points in $E(\overline{K})$, counting multiplicities, so the group is freely generated by the formal symbols $[P]$. A divisor has two main characterizations, its *degree* and its *sum*:

$$\deg(\sum_j a_j[P_j]) = \sum_j a_j,$$

$$\mathrm{sum}(\sum_j a_j[P_j]) = \sum a_j P_j.$$

The degree of a divisor is an integer and the sum of a divisor is an element of $E(\overline{K})$. An important subgroup of the group of divisors is the subgroup that contains all divisors of degree zero, denoted by $\mathrm{Div}^0(E)$. An important subgroup of the degree zero divisors is the group of *principal divisors*, which is introduced below.

A function on an elliptic curve is defined as a rational function $f(x,y) = \frac{g(x,y)}{h(x,y)}$ that is defined for at least one point of $E(\overline{K})$. The function takes values in $\overline{K} \cup \{\infty\}$. A function on an elliptic curves can have zeros and poles. A zero of a function is a point P such that the function takes the value 0 at P and a pole of a function is a point Q such that the function takes the value ∞ at the point Q. Now the order of a function f at a point P can be defined. Write $f(x,y) = u(x,y)^r \cdot g(x,y)$, where $u(P) = 0$ and $g(P) \neq 0, \infty$. Then the order of f at P is defined as $\mathrm{ord}_p(f) = r$.

A divisor of a function is defined in the following way

$$\mathrm{div}(f) = \sum_{P \in E(\overline{K})} \mathrm{ord}_P(f)[P].$$

A divisor is called a *principal divisor* if and only if it is a divisor of a function. A principal divisor describes the zeros and poles of a function (counting multiplicities). The principal divisors of an elliptic curve form a subgroup of the group of degree 0 divisors, denoted by $\mathrm{Princ}(E)$.

Theorem 3.9. *Let E be an elliptic curve defined over a field K. Then $D = \sum_j a_j[P]$ is a divisor of a function if and only if* $\deg(D) = 0$ *and* $\mathrm{sum}(D) = \infty$.

In other words, this theorem says that a divisor D of degree zero is describing zeros and poles of a function if and only if $\mathrm{sum}(D) = \infty$.

Corollary 3.10. *Let E be an elliptic curve defined over the field K. Then the following map describes a group isomorphism:*

$$\phi\colon E(\overline{K}) \to \mathrm{Div}^0(E)/\mathrm{Princ}(E), \quad P \mapsto [P] - [\infty].$$

The proof of Theorem 3.9 and Corollary 3.10 can be found in [21].

3.2.2 Explicit description of the Weil pairing

Proof of Theorem 3.4. In the first place Theorem 3.4 claims the existence of a bilinear pairing on an elliptic curve. The existence of this pairing will now be proven. The proof here follows the same strategy as in [18] (and the alternative description of the Weil pairing from [21]).

Let E denote an elliptic curve defined over a finite field $\mathbb{F}_q$ and let P and Q be two N-torsion points. Let D_P and D_Q be two degree 0 divisors on E such that the sum of D_P is equal to P and the sum of D_Q is equal to Q. In other words, D_P and D_Q can be written as

$$D_P = [P] - [\infty] \quad \text{and} \quad D_Q = [Q] - [\infty].$$

Then according to Theorem 3.9 there exist rational functions f_P and f_Q such that

$$\text{div}(f_P) = N \cdot D_p \quad \text{and} \quad \text{div}(f_Q) = N \cdot D_Q.$$

The Weil pairing can be now be defined as

$$e_N(P, Q) = \frac{f_P(D_Q)}{f_Q(D_p)}. \tag{7}$$

To show that the Weil pairing indeed maps N-torsion points to N'th roots of unity, a result from [18] is used. This result states that if f and g are two rational functions on an elliptic curve, then $f(\text{div}(g)) = g(\text{div}(f))$. Then $e_N(P, Q)$ is an N'th root of unity since

$$\begin{aligned} e_N(P, Q)^N &= \Big(\frac{f_P(D_Q)}{f_Q(D_P)}\Big)^N = \frac{f_P(N \cdot D_Q)}{f_Q(N \cdot D_P)} \\ &= \frac{f_P(\text{div}(f_Q))}{f_Q(\text{div}(f_P))} = \frac{f_P(\text{div}(f_Q))}{f_P(\text{div}(f_Q))} = 1. \end{aligned}$$

This means that the Weil pairing in equation (7) defines a pairing that associates an N'th root of unity with a given pair of N-torsion points. According to [18] and [10], the Weil pairing can be defined equivalently as the quantity

$$e_N(P, Q) = \frac{f_S(Q + R)}{f_S(R)} \Big/ \frac{f_T(P - R)}{f_T(-R)}, \tag{8}$$

where R is any point in $E(\overline{K})$ satisfying $R \notin \{\infty, P, -Q, P - Q\}$. Note that this condition on R implies that the quantity $e_N(P, Q)$ is well-defined. In [10] it is shown that the quantity does not depend on the choice of the functions f_P, f_Q or the choice of the point R. This definition of the Weil pairing is often useful when the Weil pairing actually needs to be computed, as in Example 3.11.

Besides the existence of the Weil pairing, the properties of the Weil pairing were also stated in Theorem 3.4. Here, properties 1, 3 and 4 will be proven explicitly. For the details of the proofs of the other properties, [21], [18] or [10] can be consulted. The proofs here closely follow the proof as given in [18].

Property 1: To be shown: e_N is bilinear in each variable, i.e.

$$\begin{aligned} e_N(P_1 + P_2, Q) &= e_N(P_1, Q)e_N(P_2, Q) \quad \text{and} \\ e_N(P, Q_1 + Q_2) &= e_N(P, Q_1)e_N(P, Q_2). \end{aligned}$$

To prove the bilinearity in the first variable, let $P_3 = P_1 + P_2$. Then according to equation (7), $e_N(P_1 + P_2, Q)$ can be written as

$$e_N(P_1 + P_2, Q) = e_N(P_3, Q) = \frac{f_{P_3}(D_Q)}{f_Q(D_{P_3})}. \tag{9}$$

The function f_{P_3} can be written as the composition of the function f_{P_1} and f_{P_2}. Theorem 3.9 implies that there exists a rational function h that satisfies

$$\mathrm{div}(h) = [P_3] - [P_1] - [P_2] + [\infty].$$

Then

$$\mathrm{div}\Big(\frac{f_{P_3}}{f_{P_1} f_{P_2}}\Big) = N[T_3] - N[\infty] - N[T_1] + N[\infty] - N[T_2] + N[\infty] = N\mathrm{div}(h) = \mathrm{div}(h^N),$$

since the functions f_{P_i} for $i = 1, 2, 3$ are defined such that $\mathrm{div}(f_{P_i}) = N[P_i] - N[\infty]$. Therefore

$$f_{P_3} = c f_{P_1} f_{P_2} h^N,$$

for some constant c in K. Substituting this into equation (9) yields

$$\begin{aligned} e_N(P_1 + P_2, Q) &= \frac{f_{P_1}(D_Q) f_{P_2}(D_Q)}{f_Q(D_{P_3})} = \frac{f_{P_1}(D_Q) f_{P_2}(D_Q)}{f_Q(D_{P_1+P_2})} \\ &= \frac{f_{P_1}(D_Q) f_{P_2}(D_Q)}{f_Q(D_{P_1}) f_Q(D_{P_2})} = e_N(P_1, Q) e_N(P_2, Q). \end{aligned}$$

Bilinearity in the second variable can be proven by using exactly the same reasoning.

Property 3. To show: $e_N(P, P) = 1$ for all $P \in E[N]$.
By definition of the Weil pairing from equation (8) it holds that

$$e_N(P, P) = \frac{f_P(P + R)}{f_P(R)} \Big/ \frac{f_P(P - R)}{f_P(-R)} = \frac{f_P(-R)}{f_P(R)} \frac{f_P(P + R)}{f_P(P - R)}.$$

If S is a point of order 2 in $E(\overline{K})$, then $S = -S$, which implies that $e_N(P, P) = 1$. Such a point S of order 2 always exists because of the following reasoning. If the Weil pairing is constructed for N-torsion points, for an odd integer N, then a point S of order 2 in $E(\overline{K})$ can always be selected in such a way that $S \notin \{\infty, P, -P\}$. If the Weil pairing is constructed for even N-torsion points, then the characteristic of the field K is odd. For a field of odd characteristic, there are 3 nontrivial points of order 2 in $E(\overline{K})$ (see Example 3.1), so one of these points must be not equal to P or $-P$. This point can be selected as the point S. Thus, a required point S of order 2 always exists, and therefore $e_N(P, P) = 1$.

Property 4: To show: $e_N(Q, P) = e_N(P, Q)^{-1}$ for all $P, Q \in E[N]$.
From the third property of the Weil pairing it follows that $e_N(P + Q, P + Q) = 1$. By using the bilinearity of the Weil pairing, this can be rewritten as

$$\begin{aligned} e_N(P + Q, P + Q) &= e_N(P, P + Q) e_N(Q, P + Q) \\ &= e_N(P, P) e_N(P, Q) e_N(Q, P) e_N(Q, Q) \\ &= e_N(P, Q) e_N(Q, P). \end{aligned}$$

In the last equality, property 3 of the Weil pairing was again used. This implies that $e_N(Q, P) = e_N(P, Q)^{-1}$ for all $P, Q \in E[N]$.

This concludes the proof of the properties 1,3 and 4 of the Weil pairing. □

3.2.3 Computing the Weil pairing

From Theorem 3.4 and the definitions of the Weil pairing in equations (7) and (8) it is not immediately clear if and how the Weil pairing can actually be computed. However, Miller [18] constructed a method, which is known as Miller's algorithm, to efficiently calculate the Weil pairing. Example 3.12 presents the idea of this algorithm. First, by using equation (8) directly, the Weil pairing will be derived explicitly for the 2-torsion points of an elliptic curve given by a Weierstrass equation.

Example 3.11. Let $E\colon y^2 = x^3 + Ax + B$ be an elliptic curve over a field K. The equation for the curve can be rewritten in the following form

$$y^2 = (x - \lambda_1)(x - \lambda_2)(x - \lambda_3),$$

where $\lambda_1, \lambda_2, \lambda_3$ are elements of $\overline{K}$. Observe that $\lambda_1 + \lambda_2 + \lambda_3 = 0$ since the left hand side of the equation does not contain the term x^2. In Example 3.1 it was derived that the set of 2-torsion points of E is given by

$$E[2] = \{\infty, (\lambda_1, 0), (\lambda_2, 0), (\lambda_3, 0)\}.$$

Let $P_i = (\lambda_i, 0)$ denote an arbitrary nontrivial 2-torsion point. Of course, for any point P_i, it holds that $e_2(P_i, P_i) = 1$. In [10] it is proven that the function $f_{P_i} = x - \lambda_i$ satisfies

$$\operatorname{div}(f_{P_i}) = \operatorname{div}(X - \lambda_i) = 2[P] - 2[\infty],$$

so this function can be used to construct the Weil pairing. To compute the Weil pairing $e_2(P_1, P_2)$, let $R = (a, b)$ be an arbitrary point on E. Then

$$\begin{aligned} e_2(P_1, P_2) &= \frac{f_{P_1}(P_2 + R)}{f_{P_1}(R)} \Big/ \frac{f_{P_2}(P_1 - R)}{f_{P_2}(-R)} \\ &= \frac{x(P_2 + R) - \lambda_1}{x(R) - \lambda_1} \Big/ \frac{x(P_1 - R) - \lambda_2}{x(-R) - \lambda_2}. \end{aligned}$$

Using the formulas from Section 2.1.2, the x-coordinate of the point $P_2 + R$ is given by

$$\begin{aligned} x(P_2 + R) &= \Big(\frac{b}{a - \lambda_2}\Big)^2 - a - \lambda_2 \\ &= \frac{b^2 - (a - \lambda_1)^2(a + \lambda_1)}{(a - \lambda_1)^2} \\ &= \frac{(a - \lambda_1)(a - \lambda_2)(a - \lambda_3) - (a - \lambda_1)^2(a + \lambda_1)}{(a - \lambda_1)^2} \\ &= \frac{\lambda_1 a + \lambda_2\lambda_3 + \lambda_1^2}{a - \lambda_1}, \end{aligned}$$

and similarly, the x-coordinate of the point $P_1 - R$ is given by

$$x(P_2 - R) = \frac{\lambda_2 a + \lambda_1\lambda_3 + \lambda_2^2}{a - \lambda_2}.$$

Now the Weil pairing can actually be computed for two points P_1, P_2 on E such that $P_1 \neq P_2$:

$$\begin{aligned}
e_2(P_1, P_2) &= \frac{x(P_2 + R) - \lambda_2}{x(R) - \lambda_1} \Big/ \frac{x(P_1 - R) - \lambda_2}{x(-R) - \lambda_2} \\
&= \frac{\frac{\lambda_1 a + \lambda_2 \lambda_3 + \lambda_1^2}{a - \lambda_1} - \lambda_1}{a - \lambda_1} \Big/ \frac{\frac{\lambda_2 a + \lambda_1 \lambda_3 + \lambda_2^2}{a - \lambda_2} - \lambda_2}{a - \lambda_2} \\
&= \frac{(\lambda_2 - \lambda_1)a + \lambda_1\lambda_3 + \lambda_2^2 + \lambda_1\lambda_2}{(\lambda_1 - \lambda_2)a + \lambda_2\lambda_3 + \lambda_1^2 + \lambda_1\lambda_2} \\
&= \frac{(\lambda_2 - \lambda_1)a + \lambda_2^2 + \lambda_1(\lambda_3 + \lambda_2)}{(\lambda_1 - \lambda_2)a + \lambda_1^2 + \lambda_2(\lambda_3 + \lambda_1)} \\
&= \frac{(\lambda_2 - \lambda_1)a + \lambda_2^2 - \lambda_1^2}{(\lambda_1 - \lambda_2)a + \lambda_1^2 - \lambda_2^2} \\
&= -1.
\end{aligned}$$

■

This example shows that for the 2-torsion points of an elliptic curve over an arbitrary field K it is possible to construct the Weil pairing.

In the next example, the Weil pairing will be computed for the 3-torsion points of an elliptic curve defined over a finite field.

Example 3.12. In general, third roots of unity are given by the roots of the polynomial $f(x) = x^3 - 1$. In every field $\mathbb{F}_p$, 1 is of course a third root of unity. Since

$$f(x) = x^3 - 1 = (x - 1)(x^2 + x + 1),$$

the other two third roots of unity are given by the roots of the polynomial $h(x) = x^2 + x + 1$.

Consider the finite field $\mathbb{F}_{31}$. By using the reasoning above, the third roots of unity of $\mathbb{F}_{31}$ can be obtained:

$$\mu_3 = \{1, 5, 25\}.$$

Let E denote the elliptic curve given by the Weierstrass equation $y^2 = x^3 + 1$ over the field $\mathbb{F}_{31}$. The group structure of the 3-torsion points of E is given by (Theorem 3.2)

$$E[3] \cong \mathbb{Z}/3\mathbb{Z} \oplus \mathbb{Z}/3\mathbb{Z}.$$

The set of all 3-torsion points of E is given by

$$E[3] = \{\infty, (0,1), (0,30), (3,11), (3,20), (13,11), (13,20), (15,11), (15,20)\}.$$

This set can be found manually, by finding all points of E and computing their orders, or by using software like SageMath to automatically compute the set of 3-torsion points.

Now let us compute the Weil pairing for the two 3-torsion points $P_1 = (0,1)$ and $P_2 = (15,20)$ of E. In order to compute the Weil pairing, two rational functions $f_{(0,1)}$ and $f_{(15,20)}$ that satisfy

$$\operatorname{div}(f_{(0,1)}) = 3[(0,1)] - 3[\infty] \quad \text{and} \quad \operatorname{div}(f_{(15,20)}) = 3[(15,20)] - 3[\infty]$$

need to be found. These rational functions will be derived by following the strategy from [15], which follows the approach of Miller's algorithm [18]. In short, let D_1 and D_2 denote two degree 0 divisors which are given by

$$D_1 = [P_1] - [\infty] + \mathrm{div}(f_1) \quad \text{and}$$
$$D_2 = [P_2] - [\infty] + \mathrm{div}(f_2).$$

Let $P_3 = P_1 + P_2$ and let l be the polynomial that defines the line through the points P_1 and P_2. Let v polynomial that defines the vertical line through P_3. If $P_1 = P_2$, then l is the tangent line of E at the point P_2. If $P_3 = \infty$, then put $v = 1$, i.e. a vertical line at 1. The sum of the divisors D_1 and D_2 is now given by

$$D_1 + D_2 = [P_3] - [\infty] + \mathrm{div}(f_1 f_2 f_3),$$

where $f_3 = \frac{l}{v}$. See [15] for a detailed explanation. Note that the point P_3 and the (tangent) line l can be found by using the formulas from Section 2.1.2. By using exactly this reasoning, the functions $f_{(0,1)}$ and $f_{(15,20)}$ can be found.

For the point P_1 it holds that

$$\begin{aligned}
[P_1] - [\infty] &= [P_1] - [\infty] + \mathrm{div}(1),\\
2[P_1] - 2[\infty] &= ([P_1] - [\infty]) + ([P_1] - [\infty])\\
&= [2P_1] - [\infty] + \mathrm{div}\Big(\frac{y-1}{x}\Big)\\
&= [(0,30)] - [\infty] + \mathrm{div}\Big(\frac{y-1}{x}\Big),\\
3[P_1] - 3[\infty] &= ([P_1] - [\infty]) + (2[P_2] - 2[\infty])\\
&= [P_1 + 2P_1] - [\infty] + \mathrm{div}\Big(\frac{y-1}{x}\frac{x}{1}\Big)\\
&= [\infty] - [\infty] + \mathrm{div}(y-1)\\
&= \mathrm{div}(y-1).
\end{aligned}$$

This implies that $f_{(0,1)}(x, y) = y - 1$. For the point P_2 the derivation is a little bit more involved,

$$\begin{aligned}
[P_2] - [\infty] &= [P_2] - [\infty] + \mathrm{div}(1),\\
2[P_2] - 2[\infty] &= ([P_2] - [\infty]) + ([P_2] - [\infty])\\
&= [2P_2] - [\infty] + \mathrm{div}\Big(\frac{13x - y + 11}{x + 16}\Big)\\
&= [(15,11)] - [\infty] + \mathrm{div}\Big(\frac{13x - y + 11}{x + 16}\Big),\\
3[P_2] - 3[\infty] &= ([P_2] - [\infty]) + (2[P_2] - 2[\infty])\\
&= [P_2 + 2P_2] - [\infty] + \mathrm{div}\Big(\frac{13x - y + 11}{x + 16}\frac{x - 15}{1}\Big)\\
&= [\infty] - [\infty] + \mathrm{div}\Big(\frac{(13x - y + 11)(x - 15)}{x + 16}\Big)\\
&= \mathrm{div}\Big(\frac{(13x - y + 11)(x - 15)}{x + 16}\Big).
\end{aligned}$$

Therefore the function $f_{(15,20)}$ is given by

$$f_{(15,20)}(x,y) = \frac{(13x - y + 11)(x - 15)}{x + 16}.$$

Now the Weil pairing can be computed

$$e_3((0,1),(15,20)) = \frac{f_{(0,1)}(15,20)}{f_{(15,20)}(0,1)} = \frac{19}{10} = 5,$$

which is indeed a third root of unity. ■

4 Elliptic curves over finite fields

In Section 6 it will be shown that the discrete logarithm problem for an elliptic curve defined over a finite field is a lot more complex than the same problem for an elliptic curve defined over a field of infinite order. Hence, for cryptographic purposes, the elliptic curves that are most interesting are the elliptic curves defined over a finite field.

Let $\mathbb{F}_q$ be a finite field, where $q = p^n$ for some prime p and a positive integer n. Let E be an elliptic curve defined over this finite field. A first observation is that the number of points in a finite field is finite, which implies that the number of points of E will also be finite. More specifically, the claim is that the number of points in $E(\mathbb{F}_q)$ is at most $2q + 1$. The finite field $\mathbb{F}_q$ contains at most q elements. For each $x \in \mathbb{F}_q$, there exist at most two points on the elliptic curve, (x, y) and $(x, -y)$. Including the point at infinity yields the claim.

In the rest of this paper, $\mathbb{F}_q$ will always denote a finite field as described previously. As mentioned before, for convenience, the finite field $\mathbb{F}_q$ will be assumed to have characteristic not equal to 2 or 3.

The points in $E(\mathbb{F}_q)$ do not lie on a 'nice curve' as was the case when considering an elliptic curve over the field $\mathbb{Q}$ (Section 2). Elliptic curves defined over a finite field are thus less intuitive than elliptic curves over $\mathbb{R}$ or $\mathbb{Q}$. However, it is possible to find, or rather to count the points of an elliptic curve over a finite field. The order of the group $E(\mathbb{F}_q)$ is information that will turn out to be crucial for cryptographic applications.

When the number of elements in the finite field is small it is possible to list all to points of an elliptic curve easily.

Example 4.1. Let E be the elliptic curve given by $y^2 = x^3 + x + 1$ over the finite field $\mathbb{F}_7$. The points of E can be counted using a table that contains all possible values of x, all values of $x^3 + x + 1$ and the corresponding square roots y. This means that $E(\mathbb{F}_7)$ has order 5.

x	$x^3 + x + 1$	y	Points
0	1	$\pm$ 1	$(0, 1), (0, 6)$
1	3	-	-
2	4	$\pm$ 2	$(2, 2), (2, 5)$
3	3	-	-
4	6	-	-
5	5	-	-
6	6	-	-
∞			∞

Table 2: The points of $E(\mathbb{F}_7)$

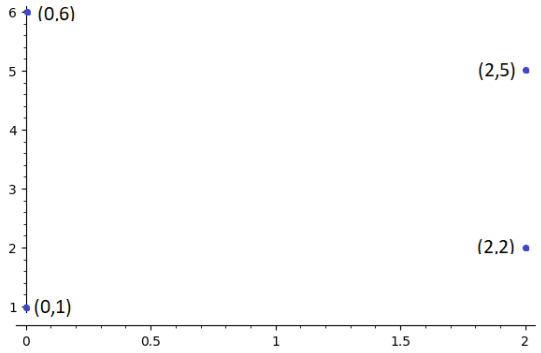


Figure 5: Elliptic curve $E\colon y^2 = x^3 + x + 1$ over $\mathbb{F}_7$

Figure 5 shows a plot of the elliptic curve E. Using the formulas for the group law from Section 2.1.2, it can be derived that the point $(0, 1)$ is the generator of the group. Hence the elliptic curve is cyclic of order 5, $E(\mathbb{F}_5) \cong \mathbb{Z}/5\mathbb{Z}$. ■

The strategy of trying to find all the points of the elliptic curve as in Example 4.1 is not a good strategy when the finite field is very large. The goal of the remaining part of this section is to explain and prove certain properties and tools concerning the order of the group $E(\mathbb{F}_q)$.

4.1 Estimate for the order of the group

The number of points on an elliptic curve over a finite field $\mathbb{F}_q$ is at most $2q+1$. However, this upper bound on the order of $E(\mathbb{F}_q)$ is not really optimal. Hasse's theorem gives a more useful bound on the order of $E(\mathbb{F}_q)$.

Theorem 4.2 (Hasse's theorem)**.** *Let E be an elliptic curve defined over the finite field $\mathbb{F}_q$. Then the order of $E(\mathbb{F}_q)$ satisfies*

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

The quantity $a = q + 1 - \#E(\mathbb{F}_q)$ is called the Frobenius trace of the elliptic curve E. The proof of Hasse's theorem is not very complicated, but it does require some theory about the Frobenius endomorphism that will be explained first.

4.1.1 Frobenius homomorphism

In Section 2.4.3 the Frobenius endomorphism was defined as the map ϕ_q that acts on the points of an elliptic curve in the following way

$$\phi_q(x, y) = (x^q, y^q), \quad \phi_q(\infty) = \infty.$$

From this same section it is known that ϕ_q is an endomorphism of degree q that is not separable. The Frobenius endomorphism will allow for an identification of the group of points of an elliptic curve and the kernel of an endomorphism. This will be essential in the proof of Hasse's theorem.

Proposition 4.3. *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ and let n be a positive integer. Let $\phi_q^n = \phi_q \circ \phi_q \circ \cdots \circ \phi_q$. Then ϕ_q^n is a separable endomorphism and* $\ker(\phi_q^n - 1) = E(\mathbb{F}_{q^n})$.

The proof here closely follows the proof from [26], although the proof in this thesis will elaborate more on some preliminary theory, also from [26], that was not discussed before.

Proof. First note that the composition of endomorphisms is an endomorphism. Therefore ϕ_q^n is an endomorphism for all $n \geq 1$. Also multiplication by the integer -1 is an endomorphism, and the sum of two endomorphisms is an endomorphism. Therefore $\phi_q^n - 1$ is an endomorphism. The claim that $\ker(\phi_q^n - 1) = E(\mathbb{F}_{q^n})$ is equivalent to saying that

$$(x, y) \in \ker(\phi_q^n - 1) \Leftrightarrow (x, y) \in E(\mathbb{F}_{q^n}).$$

To show the latter, note that $x \in \mathbb{F}_{q^n}$ if and only if $\phi_q^n(x) = x$. This implies that

$$\begin{aligned}(x, y) \in E(\mathbb{F}_{q^n}) &\Leftrightarrow x, y \in \mathbb{F}_{q^n} \\ &\Leftrightarrow \phi_q^n(x) = x \quad \text{and} \quad \phi_x^n(y) = y \\ &\Leftrightarrow \phi_q^n(x, y) = (x, y) \\ &\Leftrightarrow \phi_q^n(x, y) - (x, y) = 0 \\ &\Leftrightarrow (x, y) \in \ker(\phi_q^n - 1).\end{aligned}$$

Since an endomorphism $r\phi_q + s$ is separable if and only if $p \nmid s$ (Proposition 2.29 in [26]), it follows that $\phi_q^n - 1$ is a separable endomorphism, which concludes the proof. □

There is a relation between the number of points of an elliptic curve and the Frobenius endomorphism.

Theorem 4.4. *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ and write $a = q + 1 - \#E(\mathbb{F}_q)$. Then a is the unique integer for which it holds that $\phi_q^2 - a\phi_q + q = 0$.*

The proof of this theorem, which makes use of Cayley-Hamilton, can be found in [26]. The polynomial $X^2 - aX + a = 0$ is called the characteristic polynomial of Frobenius and as mentioned before the integer a is called the Frobenius trace of an elliptic curve.

4.1.2 Proof of Hasse's theorem

One last lemma that is needed in the proof of Hasse's theorem is a lemma that follows from property 6 of the Weil pairing.

Lemma 4.5. *If r and s are integers with* $\gcd(r, s) = 1$, *then* $\deg(r\phi_q - s) = r^2 q + s^2 - rsa$.

Proof. From Proposition 3.7 if follows that

$$\deg(r\phi_q - s) = r^2 \deg(\phi_q) + s^2 \deg(-1) + rs(\deg(\phi_q - 1) - \deg(\phi_q) - \deg(-1)).$$

Proposition 2.9 implies that ϕ_q is an endomorphism of degree q and the multiplication by -1 endomorphism is an endomorphism of degree 1 (it is therefore an automorphism). Furthermore, the endomorphism $\phi_q - 1$ is separable by Proposition 4.3, so it follows from Proposition 2.6 and Proposition 4.3 that $\deg(\phi_q - 1) = \#E(\mathbb{F}_q)$. Write $a = q + 1 - \#E(\mathbb{F}_q)$. It follows that

$$\begin{aligned}\deg(r\phi_q - s) &= r^2q + s^2 + rs(\#E(\mathbb{F}_q) - q - 1)\\ &= r^2q + s^2 + rs(-a + q + 1 - q - 1)\\ &= r^2q + s^2 - rsa.\end{aligned}$$

□

Now Hasse's theorem can be proven.

Proof of Theorem 4.2. Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. Let $a = q + 1 - \#E(\mathbb{F}_q)$. According to Proposition 4.3 this can be rewritten as $a = q + 1 - \deg(\phi_q - 1)$. The goal is to show that $|a| \leq 2\sqrt{q}$. Since $\deg(r\phi_q - s) \geq 0$ it follows from Lemma 4.5 that

$$r^2q + s^2 - rsq \geq 0,$$

or equivalently

$$q\Big(\frac{r}{s}\Big)^2 - a\Big(\frac{r}{s}\Big) + 1 \geq 0.$$

If the rational numbers of the form $\frac{r}{s}$ such that $\gcd(s, q) = 1$ are dense in the real numbers, then this inequality can be rewritten to

$$qx^2 - ax + 1 \geq 0, \quad \forall x \in \mathbb{R}.$$

In order for this inequality to hold, the discriminant of the quadratic equation, $D = q^2 - 4q$, needs to have a negative value

$$D = a^2 - 4q \leq 0 \quad \Leftrightarrow \quad |a| \leq 2\sqrt{q}.$$

The only thing that is left to show in the proof is showing that the rational numbers $\frac{r}{s}$ such that $\gcd(s, q) = 1$ are dense in $\mathbb{R}$. A subset $S \subset \mathbb{R}$ is called dense in $\mathbb{R}$ if for every $x \in \mathbb{R}$ there is a sequence s_n of numbers in S such that s_n converges to x [1].

Let S be the set of rational numbers $\frac{r}{s}$ such that $\gcd(s, q) = 1$. Write $s = 2^m$ or $s = 3^m$, then for each integer s at least one of those has gcd 1 with q. Pick an arbitrary real number x and construct the sequence s_n as follows

$$x - \frac{1}{m} < \frac{r}{2^m} < x + \frac{1}{m} \quad \text{or} \quad x - \frac{1}{m} < \frac{r}{3^m} < x + \frac{1}{m}.$$

According to the squeeze theorem [1], the sequence s_n converges to x, which means that S is dense in $\mathbb{R}$. This completes the proof of Hasse's theorem. □

4.2 Computing the order of the group

Hasse's theorem provides a bound for the order of an elliptic curve over a finite field. However, in many cryptographic contexts the actual value of the order of the group of points of an elliptic curve is needed. There are methods that precisely compute the order of the group of points of an elliptic curve.

4.2.1 The order of twists

If the elliptic curve is defined over a finite field then there exists a relation between the order of the twist and the order of the original elliptic curve [2]. Let E be an an elliptic curve over the field $\mathbb{F}_q$ and let $\tilde{E}$ denote a twist of E. Then

$$\#E(K) + \#\tilde{E}(K) = 2q + 2. \tag{10}$$

This equation will be very useful in Section 7, where examples are generated. Equation (10) can be easily verified.

To this end, let $g(x) = x^3 + Ax + B$, so the equation of E is given by $E\colon y^2 = g(x)$. Moreover, write $g_\mu(x) = \mu^3 g(x/\mu)$, so the equation of $\tilde{E}$ is given by $\tilde{E}\colon y^2 = g_\mu(x)$. If $g_\mu(x)$ is a square in $\mathbb{F}_q$ for $x \in \mathbb{F}_q \setminus \{0\}$, then this gives two points on the elliptic curve $\tilde{E}$. On the other hand, $g(x/\mu) = g_\mu(x)/\mu^3$ is not a square and hence for this value of x, E has no points. This means that for every $x \in \mathbb{F}_q \setminus \{0\}$, either E has two points and $\tilde{E}$ has none, or $\tilde{E}$ has two points and E has none. If $g_\mu(x) = 0$ then also $g(x/\mu) = 0$, giving one points on $\tilde{E}$ and one point on E. Therefore, for each of the q possible values $x \in \mathbb{F}_q$, there are two point in the union of $E(\mathbb{F}_q)$ and $\tilde{E}(\mathbb{F}_q)$. Both curves also contain the point at infinity, which yields equation (10).

4.2.2 Elliptic curves over subfields

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. The field $\mathbb{F}_{q^n}$ is an extension of $\mathbb{F}_q$. The goal in this section is to find the number of elements in $E(\mathbb{F}_{q^n})$ given that the order of $E(\mathbb{F}_q)$ is known [8].

Theorem 4.6. *Let $\#E(\mathbb{F}_q) = q + 1 - a$, and write $X^2 - aX + q = (X - \alpha)(X - \beta)$. Then $\#E(F_{q^n}) = q^n + 1 - (\alpha^n + \beta^n)$.*

This theorem only makes sense if $\alpha^n + \beta^n$ is an integer.

Lemma 4.7. *Let $s_n = \alpha^n + \beta^n$, then $s_0 = 2$, $x_1 = a$ and $s_{n+1} = as_n - qs_{n-1}$ for all $n \geq 1$.*

Proof. By the equality $X^2 - aX + q = (X - \alpha)(X - \beta)$, it follows that α and β are roots of the polynomial $X^2 - aX + q$, so $\alpha^2 - a\alpha + q = 0$ and $\beta^2 - a\beta + q = 0$. By multiplying the first equality by α^{n-1} and the second equality by β^{n-1} it follows that

$$\alpha^{n+1} = a\alpha^n - q\alpha^{n-1} \quad \text{and} \quad \beta^{n+1} = a\beta^n - q\beta^{n-1}.$$

When these two equations are added the required equality is obtained:

$$\alpha^{n+1} + \beta^{n+1} = a(\alpha^n + \beta^n) - q(\alpha^{n-1} + \beta^{n-1}) = as_n + qs_{n-1}.$$

□

From Lemma 4.7, the element $\alpha^n + \beta^n$ can be written recursively, as a linear combination of integers, which implies that $\alpha^n + \beta^n$ is an integer. Now Theorem 4.6 can be proven.

Proof of Theorem 4.6. Let E be an elliptic curve defined over the finite field $\mathbb{F}_q$. Then E is also an elliptic curve over the finite field $\mathbb{F}_{q^n}$. Let $a = q^n + 1 - \#E(\mathbb{F}_{q^n})$. From Theorem

4.4 it is known that a is the unique value such that $\phi_{q^n}^2 - a\phi_{q^n} + q = 0$. This means that showing that $\phi_{q^n}^2 - (\alpha^n + \beta^n)\phi_{q^n} + q^n = 0$ implies that $a = \alpha^n + \beta^n$. Now it will be shown that $\phi_{q^n}^2 - (\alpha^n + \beta^n)\phi_{q^n} + q^n = 0$.

Let

$$f(X) = (X^n - \alpha^n)(X^n - \beta^n) = X^{2n} - (\alpha^n + \beta^n)X^n + q^n,$$

then the characteristic polynomial of Frobenius, $g(X) = X^2 - aX + q$ divides $f(X)$. The characteristic polynomial of Frobenius has the property that $g(\phi_q) = 0$, by Theorem 4.4 This implies that

$$f(\phi_q) = (\phi_q^n)^2 - (\alpha^n + \beta^n)\phi_q^n + q^n = Q(\phi_q)(\phi_q^2 - a\phi_q + q) = 0.$$

Here $Q(X)$ denotes the quotient of the polynomial $f(x)$ and the characteristic polynomial of Frobenius. This means that

$$(\phi_q^n)^2 - (\alpha^n + \beta^n)\phi_q^n + q^n = (\phi_{q^n})^2 - (\alpha^n + \beta^n)\phi_{q^n} + q^n = 0.$$

According to the argument at the beginning of the proof this implies that $a = \alpha^n + \beta^n$. Therefore the order of the elliptic curve E over the finite field $\mathbb{F}_{q^n}$ is given by

$$\#E(\mathbb{F}_{q^n}) = q^n + 1 - (\alpha^n + \beta^n),$$

which concludes the proof. □

4.2.3 Schoof's algorithm

The most commonly used algorithms for computing the order of a finite group, such as the baby step, giant step method [26], are not very efficient when applied to elliptic curves over large finite fields. In 1985, Schoof [20] introduced an algorithm for computing the order of the group $E(\mathbb{F}_q)$ that was a lot more efficient in comparison to existing algorithms. To be precise, Schoof's algorithm computes the order of $E(\mathbb{F}_q)$ in polynomial time [21]. The basic idea for Schoof's algorithm will be sketched below.

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ and let $a = q + 1 - \#E(\mathbb{F}_q)$. From Hasse's theorem, it is known that $|a| \leq 2\sqrt{q}$. The idea is to compute

$$a \bmod l$$

for sufficiently many primes l and use the Chinese Remainder Theorem to compute

$$a \bmod \prod l,$$

which uniquely determines a, using Hasse's bound.

It suffices to run the algorithm for the smallest set of all primes l such that $\prod l > 4\sqrt{q}$ [21]. The values $a \bmod l$ are computed using the Frobenius endomorphism in a special ring R, which is defined in terms of the division polynomials.

For a more detailed explanation of Schoof's algorithm and a proof of the running time, see [21] and [20].

4.3 Complex multiplication and elliptic curves over $\mathbb{F}_q$

An elliptic curve over $\mathbb{Q}$ does not always have complex multiplication, as was shown in Section 2.4.4. To be precise, only a finite number of elliptic curves over $\mathbb{Q}$ does have complex multiplication. However, an elliptic curve E over a finite field $\mathbb{F}_q$ always has complex multiplication. The Frobenius endomorphism is an "extra" endomorphism. From Theorem 4.4, the Frobenius endomorphism is a root of the characteristic polynomial

$$X^2 - aX + q = 0.$$

Hasse's theorem states that $|a| \leq 2\sqrt{q}$. If the Frobenius trace of E strictly satisfies this bound, so $|a| < 2\sqrt{q}$, then the discriminant of the characteristic polynomial of Frobenius is less than 0. This means that if $|a| < 2\sqrt{q}$, then the characteristic polynomial of Frobenius only has complex roots, so in this case the Frobenius endomorphism provides complex multiplication. In fact, according to [26], if $a = \pm 2\sqrt{q}$, the elliptic curve also has complex multiplication. If $a = \pm 2\sqrt{q}$, then the characteristic polynomial can be factored as

$$X^2 \mp 2\sqrt{q}X + q = (X \mp \sqrt{q})^2.$$

In fact, this is the characteristic polynomial for a supersingular elliptic curve as will be explained in Section 5.

For an elliptic curve over a finite field, the endomorphism ring is an order in an imaginary quadratic field or an order in a quaternion algebra. Washington [26] presents a complete explanation regarding elliptic curves defined over a finite field and complex multiplication. In Section 5 it will be explained that elliptic curves with endomorphism ring that is a quaternion algebra are the special type of elliptic curves called supersingular elliptic curves.

5 Supersingular elliptic curves

In general, the order of an elliptic curve not straightforward to compute. Algorithms like Schoof's algorithm and the baby step, giant step method can be applied to compute the order of the group of points of an elliptic curve, but it still remains a nontrivial task. However, there are exceptions to this: there exist different families of elliptic curves for which the order of the elliptic curve is immediately clear from the definition. A particularly interesting family of elliptic curves is the family of supersingular elliptic curves. These curves will be the main interest in the remaining part of this paper.

5.1 Definitions and characterizations

Let E be an elliptic curve defined over a finite field of characteristic p. The elliptic curve is called a supersingular elliptic curve if E contains no points of order p with coordinates in the algebraic closure of the field K. In other words, for supersingular elliptic curves, the point at infinity is the only element in the group of p-torsion points: $E[p] = \{\infty\}$. This definition gives an important characterization for supersingular elliptic curves, namely a characterization based on the number of points of order p on an elliptic curve. There are also other ways to identify a supersingular elliptic curve.

Proposition 5.1. *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$, where q is a power of a prime p. Then E is a supersingular elliptic curve if and only if $a \equiv 0$ mod p which holds if and only if $\#E(\mathbb{F}_q) \equiv 1$ mod p.*

Proof. Let E be an elliptic curve defined over the finite field $\mathbb{F}_q$ and write $a = q + 1 - \#E(\mathbb{F}_q)$ for the Frobenius trace of the elliptic curve E.

The equivalence $a \equiv 0 \bmod p \Leftrightarrow \#E(\mathbb{F}_q) \equiv 1 \bmod p$ is straightforward from the definition of the Frobenius trace of the elliptic curve.

It remains to show that $a \equiv 0 \bmod p$ if and only if E is a supersingular elliptic curve. The same notation as in Section 4.2.2 will be used. So write $X^2 - aX + q = (X - \alpha)(X - \beta)$ and $s_n = \alpha^n + \beta^n$. From Lemma 4.7, s_n is an integer as it can be written recursively as

$$\begin{aligned} s_0 &= 0, \\ s_1 &= 1, \\ s_{n+1} &= as_n - qs_{n-1}, \end{aligned}$$

for all $n \geq 1$. Assume that $a \equiv 0 \bmod p$. This implies that $s_{n+1} \equiv 0 \bmod p$ for all values of n, since $q \equiv 0 \bmod p$. Therefore, by Theorem 4.6, the order of the elliptic curve E over $\mathbb{F}_{q^n}$ satisfies

$$\#E(\mathbb{F}_{q^n}) = q^n + q - s_n \equiv 1 \bmod p.$$

The order of an element always divides the order of the group [19]. This means that there is no point of order p in $E(\mathbb{F}_{q^n})$ for any field extension $\mathbb{F}_{q^n}$ of $\mathbb{F}_q$. Since $\overline{\mathbb{F}}_q = \bigcup_n \mathbb{F}_{q^n}$ it follows that there is no point of order P in $E(\overline{\mathbb{F}}_q)$. Therefore E is supersingular.

The other direction will be proven using contraposition. Assume that $a \not\equiv 0 \bmod p$ so that $s_{n+1} \equiv as_n \bmod p$. Since $s_0 = a$ this implies that $s_n \equiv a^n \bmod p$ for all values of n. So the number of points on the subfield curve can be written as $\#E(\mathbb{F}_{q^n}) = q^n + 1 - s_n = 1 - a^n$. Fermat's little theorem states that $a^{p-1} \equiv 1 \bmod p$ (see Appendix A.2). This implies that the group of points E over a particular subfield, namely $E(\mathbb{F}_{q^{p-1}})$, has order that is divisible by the prime p, hence $E(\mathbb{F}_{q^{p-1}})$ contains a point of order p. Using again the argument that $\mathbb{F}_q$ can be written as a union of its subfields, this implies that $E(\mathbb{F}_q)$ contains a point of order p, so E is not supersingular. □

When there is a certain constraint on the size of the prime p, supersingular curves can be characterized even more conveniently.

Corollary 5.2. *Let $p \geq 5$ be a prime and let E be an elliptic curve over a finite field $\mathbb{F}_p$. Then the following are equivalent:*

- *E is a supersingular curve*
- $a = 0$
- $\#E(\mathbb{F}_p) = p + 1$.

Proof. Let E be an elliptic curve defined over a finite filed $\mathbb{F}_p$, where $p \geq 5$. If $a = 0$, then Proposition 5.1 can be applied to conclude that the elliptic curve E is supersingular. The remaining part of the corollary will again be shown using contraposition. Assume that E is a supersingular elliptic curve, but $a \neq 0$. Then by proposition 5.1, $a \equiv 0 \bmod p$, which means that $|a| \geq p$. Now Hasse's theorem says that $|a| \leq 2\sqrt{p}$, so $p \leq 2\sqrt{p}$, and therefore $p \leq 4$. □

Example 5.3. The elliptic curve $E\colon y^2 = x^3 - x + 1$ defined over the field $\mathbb{F}_{97}$ is a supersingular elliptic curve. This can be verified using for example Schoof's algorithm [26] [20], which is implemented in SageMath, to check that the curve has 98 points. It is also feasible to check all possible $x, y \in \mathbb{F}_{97}$ and see how many satisfy the equation, as in Example 4.1, not forgetting the point at infinity (see Figure 6). ■

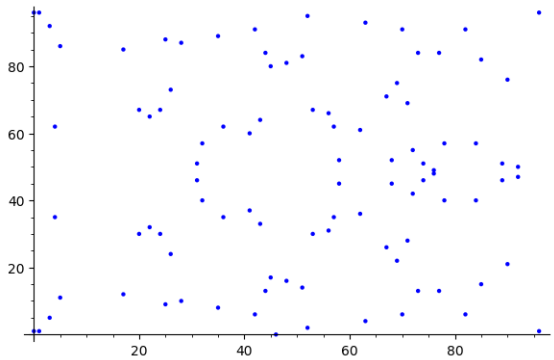


Figure 6: Elliptic curve $y^2 = x^3 - x + 1$ over $\mathbb{F}_{97}$.

5.2 Computing multiples of a point

One of the reasons why supersingular elliptic curves are such interesting curves is that for these curves, there exist a fast method for computing multiples of a point. Normally a multiple of a point can be computed relatively fast using the successive doubling method (Section 2.1). In this subsection an algorithm to compute a multiple of a point, for points on a supersingular elliptic curve, will be described [26].

Let E be a supersingular curve over a finite field $\mathbb{F}_p$, and let $P = (x, y)$ be a point in $E(\mathbb{F}_{q^n})$ for some integer $n \geq 1$. Let k be a positive integer. The goal is to compute the point $k \cdot P$.

Let $a = q + 1 - \#E(F_q)$. Then according to Theorem 4.4, the characteristic polynomial of Frobenius can be written in the following form

$$\phi_q^2 - a\phi_q + q = 0.$$

Assume that $a = 0$, then this equation reduces to

$$\phi_q^2 + q = 0$$

This can be written equivalently as

$$q \cdot (x, y) = -\phi_q^2(x, y) = (x^{q^2}, -y^{q^2}).$$

Expanding the integer k in the base q will now yield the faster method of computing a multiple of a point on this elliptic curve. The algorithm is as follows:

Computing multiples of a point algorithm

1. Expand the integer k in the base q. This gives $k = k_0 + k_1 q + k_2 q^2 + \cdots + k_r q^r$.
2. Compute $k_i \cdot P = (x_i, y_i)$ for each value of i.
3. Compute $q^i(k_i P) = (x_i^{q^{2i}}, (-1)^i y_i^{q^{2i}})$ for each value of i.
4. Sum the elements $q^i(k_i \cdot P)$ for $0 \leq i \leq r$.

This algorithm of course makes sense because $k \cdot P = k_0 \cdot P + q(k_1 \cdot P) + q^2(k_2 \cdot P) + \cdots + q^r(k_r \cdot P)$, when k is expanded in the base q.

The method described above yields a faster algorithm for multiples of a point for points on a supersingular elliptic curve than successive doubling since in the third step of the algorithm all computations are done in a finite field instead of on the elliptic curve. In general computations in a finite field are faster than elliptic curve computations.

5.3 Constructing supersingular elliptic curves

Since supersingular elliptic curves are such special elliptic curves, especially in cryptographic applications, it would be nice to have a method that immediately constructs supersingular elliptic curves.

Proposition 5.4. *Suppose p is an odd prime and $p \equiv 2 \bmod 3$. Let $B \in \mathbb{F}_p^\times$. Then the elliptic curve that is given by $y^2 = x^3 + B$ is supersingular.*

Proof. First note the assumption p is odd and $p \equiv 2 \bmod 3$ implies that Corollary 5.2 can be applied. Let E be the elliptic curve given by the equation $y^2 = x^3 + B$ over the finite field $\mathbb{F}_p$. Corollary 5.2 says that showing that the elliptic curve E is supersingular is equivalent to showing that $\#E(\mathbb{F}_p) = p+1$, therefore the number of points of $E(\mathbb{F}_p)$ has to be counted. This can be done using the homomorphism $\psi\colon \mathbb{F}_p^\times \to \mathbb{F}_p^\times$ that is defined by $\psi(x) = x^3$. In fact the map ψ is an isomorphism.

First the injectivity will be shown. The order of the multiplicative group $\mathbb{F}_p^\times$ is equal to $p-1$. By the assumption that $p \equiv 2 \bmod 3$, it follows that the order of $\mathbb{F}_p^\times$ is not a multiple of 3, so there are no elements of order 3 in the group $\mathbb{F}_p^\times$. This means that the only element that is mapped to zero by the map ψ is the identity element, so the kernel of ψ is trivial, which means that the map ψ is injective. The map ψ is surjective since it maps a finite group to itself. Now, using the isomorphism ψ, it is clear that every element of $\mathbb{F}_p^\times$ has a cube root in $\mathbb{F}_p^\times$, so also every element of $\mathbb{F}_p$ has a cube root in $\mathbb{F}_p$. For this reason, for every $y \in \mathbb{F}_p$ there is exactly one $x \in \mathbb{F}_p$ such that (x, y) lies on the elliptic curve. In particular, for $y \in \mathbb{F}_p$, the point $(\sqrt[3]{y^2 - B}, y)$ lies on the elliptic curve. There are precisely p values of y in the field $\mathbb{F}_p$, so there are p points (x, y) on the elliptic curve E obtained using this reasoning. Including the point at infinity implies that there are $p + 1$ points on the elliptic curve. Therefore the curve given by the equation $y^2 = x^3 + B$ is supersingular. □

Example 5.5. The elliptic curve $E\colon y^2 = x^3 + 1$ defined over the field $\mathbb{F}_{101}$ is a supersingular elliptic curve. Since $101 \equiv 2 \bmod 3$, this is immediately clear from Proposition 5.4. It can also be verified by using Schoof's algorithm as in the previous example (see Figure 7). ■

For Weierstrass equations of the form $y^2 = x^3 + Ax$ there is a statement similar to Proposition 5.4.

Proposition 5.6. *Let $p \geq 5$ be a prime. Let $A \in \mathbb{F}_p^\times$, then the elliptic curve that is given by $y^2 = x^3 + Ax$ is supersingular if and only if $p \equiv 3$ mod 4.*

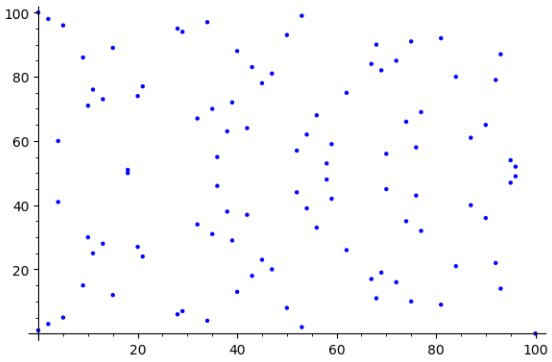


Figure 7: Supersingular elliptic curve $y^2 = x^3 + 1$ over $\mathbb{F}_{97}$

The proof of Theorem 5.6 makes use of the Hasse invariant, which will not be introduced in this thesis. For the details, see [26] and [21].

By means of Proposition 5.4 and 5.6, supersingular elliptic curves can be constructed over a finite field $\mathbb{F}_p$ for an enormous prime p. The only thing that is needed is to check whether the prime satisfies $p \equiv 2 \bmod 3$ or $p \equiv 3 \bmod 4$ (or neither).

Example 5.7. Consider $p = 2^{82589933} - 1$, which is the largest prime number largest prime number known today. Over $\mathbb{F}_p$, a supersingular elliptic curve can easily be constructed. Since $p \equiv 3 \bmod 4$, the elliptic curve given by

$$E_1 \colon y^2 = x^3 + ax,$$

where $a \in \mathbb{F}_p^\times$, is supersingular. ■

5.4 Supersingular curves and complex multiplication

From Section 4.3, elliptic curves over a finite field always have complex multiplication, and in some cases, the endomorphism ring is an order in a quaternion algebra. In fact, this happens exactly when the elliptic curve is supersingular [26].

The next two examples will show that for elliptic curves that have complex multiplication over $\mathbb{Q}$, reducing the elliptic curve modulo p will yield a supersingular elliptic curve quite often.

Example 5.8. Consider the elliptic curve $E \colon y^2 = x^3 + x$ over $\mathbb{Q}$. The j-invariant of E is given by $j(E) = 1728$. This elliptic curve has complex multiplication. The extra endomorphism in this case is the multiplication-by-i endomorphism. Now reduce the elliptic curve modulo a prime p, and determine what happens to the endomorphism ring.

First consider the specific case when $p = 11$. Then E has 12 points over $\mathbb{F}_p$,

$$E(\mathbb{F}_p) = \{\infty, (0,0), (5,3), (5,8), (7,3), (7,8), (8,5), (8,6), (9,1), (9,10), (10,3), (10,8)\},$$

so E is a supersingular elliptic curve. The Frobenius endomorphism satisfies $\phi \circ \phi(P) = -11 \cdot P$, i.e. $\phi^2 = -11$, and there is another algebraic integer e that satisfies $e^2 = -1$. The algebraic integer e gives rise to another endomorphism $\alpha \colon (x, y) \to (-x, ey)$. Since $\phi \circ \alpha = -\alpha \circ \phi$, the ring of endomorphisms is a quaternion algebra.

It turns out that only for the elements p that are still primes in $\mathbb{Z}[i]$ (or irreducible, since $\mathbb{Z}[i]$ is a unique factorization domain), the endomorphism ring of E is larger than just $\mathbb{Z}[\alpha]$. This means that for these primes a supersingular elliptic curve is obtained after reducing modulo p. Equivalently, only for primes that can not be written as the sum of two squares, the elliptic curve E modulo p is supersingular. These are exactly the primes such that $p \equiv 3 \bmod 4$. Thus, E is a supersingular elliptic curve over $\mathbb{F}_p$ if and only if $p \equiv 3 \bmod 4$. ■

Example 5.9. Consider the elliptic curve $E \colon y^2 = x^3 + 1$. The j-invariant of E is given by $j(E) = 0$. This curve has complex multiplication and the extra endomorphism is given by the multiplication by a third root of unity ζ_3. So the endomorphism ring of this elliptic curve is isomorphic to $\mathbb{Z}[\zeta_3]$. A supersingular elliptic curve is obtained when reducing the elliptic curve E modulo p if and only if $p \equiv 2 \bmod 3$. The reasoning behind this is similar to the previous example. This implies that E over the finite field $\mathbb{F}_p$ is supersingular if $p \equiv 2 \bmod 3$. ■

Remark. Note that the claims in these two examples were already stated in Proposition 5.4 and Proposition 5.6.

Approximately half of the primes satisfy $p \equiv 3 \bmod 4$ and also approximately half of the primes satisfy $p \equiv 2 \bmod 3$, asymptotically. The two examples suggest that if an elliptic curve over $\mathbb{Q}$ has complex multiplication, then it is supersingular modulo p for approximately 50% of the primes. In fact, this is true, as was claimed in [21] and proven by Deuring [26]. For an elliptic curve E that has complex multiplication over $\mathbb{Q}$, the general statement states the following. If E (that has complex multiplication) is given by a Weierstrass equation with integer coefficients, then reducing the elliptic curve modulo a prime p (to obtain an elliptic curve over $\mathbb{F}_p$) will yield a supersingular elliptic curve for approximately half of the primes.

In Table 1 all j-invariants are given for elliptic curves over $\mathbb{Q}$ that have complex multiplication. Reducing the elliptic curve with these j-invariants modulo any prime p will give a supersingular elliptic curve for approximately half of the primes.

If the elliptic curve over $\mathbb{Q}$ does not have complex multiplication, then supersingularity of the elliptic curve modulo p is a lot more rare. This set of supersingular curves is even so small that it has density 0. This is an important result that is due to Elkies [5]. Elkies showed that despite the set of this supersingular elliptic curves having density 0, there are still infinitely many primes for which the reduction modulo p of a given elliptic curve E, defined over $\mathbb{Q}$, is supersingular.

In Section 7, where supersingular curves will be generated for demonstrational purposes, it will be important to take the possibility of larger endomorphism rings and elliptic curves with complex multiplication over $\mathbb{Q}$ into account in order to construct novel supersingular elliptic curves.

6 The Discrete logarithm problem

Recall from the introduction that for a group G, the discrete logarithm problem is defined as the following problem. Given group G and a given element g in the group, someone computes $h = g^n$ for a secret integer n. Then, given the elements g and h, the goal is to find the positive integer n such that $h = g^n$. The complexity of the discrete logarithm problem is often defined in terms of the time it takes the currently best known algorithm to solve the problem.

For the safety of a cryptographic application such as the public key cryptography, it is essential that the discrete logarithm problem is difficult to solve. This means that the currently known algorithms should not be able to solve the problem in a reasonable amount of time, say 10 years (in reality this requirement is a lot more strict).

For a group like the real numbers, the discrete logarithm problem is not that complicated. To compute the range of the integer n, one can look at the size of the element h. Here the size of an element means the numbers of digits of the element before the decimal point. In fact, over $\mathbb{R}$ the power series of the logarithm function can be used to find the exact solution to the discrete logarithm problem very efficiently. For an arbitrary group the discrete logarithm problem is a lot more complicated. An example of a discrete logarithm problem in an arbitrary group is the discrete logarithm problem for the integers modulo a prime p. In this case the discrete logarithm problem is defined as follows: given integers a and b, find an integer k such that $a^k \equiv b \bmod p$. A particularly important and difficult discrete logarithm problem is the discrete logarithm problem for the group of points of an elliptic curve over a finite field.

There are different approaches for solving the discrete logarithm problem. First of all there are methods for solving the discrete logarithm problem in an arbitrary group, such as brute force searching, the baby step, giant step algorithm and Pollard's algorithms. There are also algorithms that work in arbitrary groups with a certain condition on the order of the group, details on such algorithms, for example Pohlig-Hellman algorithm, can be found in [26] or [16]; they will not be discussed here. Furthermore there are algorithms that can only solve the discrete logarithm problem in certain groups, such as the Index Calculus methods [16].

6.1 Discrete logarithm problem in an arbitrary group

For an arbitrary finite group, for example the group of points of an elliptic curve defined over a finite field, the discrete logarithm problem is a very complex problem.

Brute-force or exhaustive search is the most obvious approach that can be used to solve this problem. Given a group G and the element g, the elements $g^1, g^2, g^3, \ldots$ can be computed consecutively until the element h is found. This method is not very efficient if the order of the group is large: the number of multiplications that are needed is $\mathcal{O}(N)$, where N denotes the order of the group. In other words, the number of multiplications needed is bounded by a constant times the group order.

Two different, more efficient, methods for solving the discrete logarithm problem in an arbitrary finite group will be described. Both of these methods are collision algorithms that run in

exponential time. The methods need $\mathcal{O}(\sqrt{N})$ group multiplications, where N denotes the order of the group.

6.1.1 The "baby step, giant step" method

The baby step, giant step method is a deterministic algorithm for solving the discrete logarithm problem, described by Shanks in 1971 [26]. The algorithm is deterministic since it is guaranteed to find the solution to the discrete logarithm problem in a certain amount steps. The baby step, giant step method requires approximately $\sqrt{N}$ steps, where N is the order of the group. Let G be a group of order N and let P, Q be two given elements in the group. The goal is to find an integer n such that $n \cdot P = Q$.

Baby step, giant step algorithm

1. Pick an integer $b \geq \sqrt{N}$.
2. The baby steps: compute $j \cdot P$ for $0 \leq j < b$.
3. The giant steps: compute $Q - kb \cdot P$ for $k = 0, 1, 2...$, until one of the giant steps is equal to one of the baby steps.
4. If $j \cdot P = Q - kb \cdot P$, it follows that have that $Q = n \cdot P$, with $n \equiv j + kb \bmod N$.

It is not very hard to see why there always is a collision between a baby step and a giant step of the algorithm. The integer n can be written as $n = n_0 + bn_1$, with $n_0 \equiv n \bmod b$, and let $n_1 = (n - n_0)/b$. When $j = n_0$ and $k = n_1$ in the algorithm, there is a match:

$$Q - n_1 b \cdot P = n \cdot P - n_1 b \cdot P = n_0 \cdot P.$$

The careful reader may have noticed that the baby step, giant step algorithm only requires an upper bound for the order of the group G. This means that it in not necessary to use an algorithm like Schoof's algorithm to find the exact order of G. For an elliptic curve E defined over a finite field $\mathbb{F}_q$ such an upper bound for the order of $E(\mathbb{F}_q)$ can be found easily using Hasse's theorem.

6.1.2 One of Pollard's methods

The mathematician Pollard described two related probabilistic algorithms for solving the discrete logarithm problem in an arbitrary finite group G of order N. The advantage of both of these methods is that they use a relatively small amount of storage compared to the baby step, giant step method. The algorithms described by Pollard are probabilistic algorithms since they will find the solution with high probability within a certain amount of steps. However, it is not guaranteed that the algorithms will have success in the predicted time. A sketch of the approach for Pollard's λ method will be given in this section. The details can be found in [26].

Pollard's λ method mainly relies on finding elements in the group G by iterating a random function on G. Let $f\colon G \to G$ be a random function on the elements of G. Start with an arbitrary element P_0 in G and find other elements in G using the function f iteratively

$$P_{i+1} = f(P_i).$$

Since G is a finite group the following scenario will occur: for some indices $m < n$, it holds that $P_m = P_n$. The iterative property of the elements P_i implies that $P_{m+l} = P_{n+l}$ for all $l \geq 0$. The sequence of elements P_i that is obtained by applying the function f is thus a periodic sequence. Finding a match, i.e indices m, n such that $P_m = P_n$ will take approximately a multiple of $\sqrt{N}$ steps. Using the periodicity of the sequence of elements P_i and by clever storing of only a current pair of elements, this method takes a lot less storage than the baby step, giant step method.

6.2 Finite field discrete logarithm problem

For the discrete logarithm problem in a finite field there exists a sub-exponential solving algorithm, Index Calculus. Index Calculus methods first appeared in 1968, even before the public key cryptography was discovered [10].

Let p be a prime and let g and h be two elements in $\mathbb{F}_p$. The problem is to find an integer k such that $g^k \equiv h \bmod p$. Let $\log(h) = k$ denote the discrete logarithm of h with respect to the element g and the prime p. One important observation is that $\log(h)$ changes multiplication into addition, similar to the normal logarithm function:

$$g^{\log(h_1 h_2)} \equiv h_1 h_1 \equiv g^{\log(h_1)+\log(h_2)} \bmod p \quad \Rightarrow \quad \log(h_1 h_2) \equiv \log(h_1) + \log(h_2) \bmod p - 1.$$

The group $\mathbb{F}_p^\times$ is cyclic, meaning that there is some generator g such that $h = g^k$ for all $h \in \mathbb{F}_p^\times$ and some integer k. This means that g be can taken as a generator of the group $\mathbb{F}_p^\times$, and the discrete logarithm problem can be defined in terms of this generator g. Before describing the algorithm of the Index Calculus, two additional definitions are needed [10].

Definition 6.1. *Let B be an integer. Then an element $x \in \mathbb{F}_p$ is called* B-smooth *if all its prime factors are less than or equal to B.*

Definition 6.2. *For an integer B, the* factor base *is the set that contains all primes and prime powers less than or equal to B. The factor base will be denoted by $\mathbb{B}$.*

Now the algorithm for finding the discrete logarithm $\log(h)$ with respect to a generator g and the prime p will be described.

The Index Calculus algorithm takes as input the two elements g and h and outputs the discrete logarithm $\log(h)$, which satisfies $g^{\log(h)} \equiv h \bmod p$.

The first step is to select an integer B and to solve the discrete logarithm problem

$$g^k \equiv m \bmod p \quad \text{for all elements} \quad m \in \mathbb{B}.$$

The next step is to compute $g^{-j} \cdot h$ for some arbitrary values of j until a B-smooth element is obtained. For this value of j it holds that

$$g^{-j} \cdot h \equiv \prod_{m \in \mathbb{B}} m^{a_m} \bmod p,$$

for certain exponents a_m, depending on the element m. Then the problem can be rewritten as

$$\log(h) \equiv j + \sum_{m \in \mathbb{B}} a_m \cdot \log(m) \bmod p - 1.$$

This solves the discrete logarithm problem in the group $\mathbb{F}_p^\times$.

One difficulty that remains is solving the discrete logarithm problem for the elements in the factor base. These discrete logarithms can be solved by first finding x such that g^x is B-smooth,

$$g^x \equiv \pm \prod_{m \in \mathbb{B}} m^{a_m} \bmod p. \tag{11}$$

Finding such values of x can be done by computing g^x for some random exponents and only storing the B-smooth elements. These equations can then be transformed into discrete logarithm equations, that is

$$x \equiv \pm \sum_{m \in \mathbb{B}} a_m \log(m) \bmod p-1. \tag{12}$$

If the number of such equations is equal to, or higher than the number of elements in the factor base, then the linear that consists of all these equations can be solved. In other words, linear algebra can be used to find the solutions for $\log(m)$ from the linear system. This will yield $\log(m)$ for all $m \leq B$.

There is one subtlety. The system of equations (12) is a system of equations modulo $p-1$. When solving this linear system with standard linear algebra methods such as Gaussian elimination, a lot of inverses need to be computed. However, since $p-1$ is composite, there are a lot of numbers that do not have an inverse modulo $p-1$. This problem can be solved by applying the Chinese remainder theorem to deal with simultaneous congruences [26] [10].

Theorem 6.3 (Chinese Remainder Theorem)**.** *Let $p_1, p_2, \ldots, p_k$ be pairwise distinct primes. Let $a_1, a_2, \ldots, a_k$ be arbitrary integers and $m_i = p_i^{e_i}$ for some positive integer exponents e_i. Then the system*

$$x \equiv a_1 \ mod\ m_1, \quad x \equiv a_2 \ mod\ m_2, \quad \ldots, x \equiv a_k \ mod\ m_k$$

has an integer solution $x = c$.

Hence to find a solution to the system of equations (12), first the congruences can be solved modulo r, for each prime r dividing $p-1$. If r^i divides $p-1$ for some integer i, then the solution can be found in $\mathbb{Z}/r^i\mathbb{Z}$ rather than $\mathbb{Z}/r\mathbb{Z}$. As a final step, the Chinese remainder can be applied to combine the different solutions into a solution modulo $p-1$.

The following examples shows how Index Calculus works in practice.

Example 6.4. Consider the prime $p = 607$ and $g = 5$. The goal is to solve the discrete logarithm problem $5^x \equiv 31 \bmod 607$. Pick $B = 11$, this means the factor base is the set $\{2, 3, 5, 7, 11\}$. The first step is to find elements 5^x that are B-smooth for some integers x. This yields the following relations modulo 607:

$$\begin{aligned}
5^5 &\equiv 2 \cdot 3^2 \cdot 5 \bmod 607, \\
5^{22} &\equiv 2^5, \\
5^{32} &\equiv 11, \\
5^{75} &\equiv 3^3 \cdot 7, \\
5^{80} &\equiv 2 \cdot 7, \\
5^{93} &\equiv 2^4 \cdot 3 \cdot 7.
\end{aligned}$$

Changing these relations into a system of discrete logarithm equations gives the following system of congruences modulo 606:

$$\begin{aligned} 5 &\equiv \log(2) + \log(3) + \log(5) \bmod 606, \\ 22 &\equiv 5\log(2), \\ 32 &\equiv \log(11), \\ 75 &\equiv 3\log(3) + \log(7), \\ 80 &\equiv \log(2) + \log(7), \\ 93 &\equiv 4\log(2) + \log(3) + \log(7). \end{aligned}$$

The number 606 is a composite number, it can be factored as $606 = 2 \cdot 3 \cdot 101$. Hence the linear system needs to be solved modulo 2, modulo 3 and modulo 101. This can be done easily by performing Gaussian elimination. The solutions to the three linear systems are given by

$$\begin{aligned} (\log(2), \log(3), \log(5), \log(7), \log(11)) &= (0, 1, 1, 0, 0) \bmod 2, \\ (\log(2), \log(3), \log(5), \log(7), \log(11)) &= (2, 1, 1, 0, 2) \bmod 3, \\ (\log(2), \log(3), \log(5), \log(7), \log(11)) &= (65, 20, 1, 15, 32) \bmod 11. \end{aligned}$$

Then the Chinese remainder theorem can be used to combine the solutions into a solution modulo 606:

$$(\log(2), \log(3), \log(5), \log(7), \log(11)) = (368, 121, 1, 318, 32).$$

The goal in this example was to compute $5^x \equiv 31 \bmod 607$. The next step in Index Calculus is to compute $31 \cdot 5^{-x}$ for natural numbers x until a B-smooth number is obtained. After a few attempts, this gives

$$31 \cdot 5^{-37} = 2^4 \cdot 11 \bmod 607,$$

or equivalently, in terms of discrete logarithms

$$\log(31) = 37 + 4\log(2) + \log(11) \bmod 606.$$

Now the previously solved discrete logarithms can be substituted to obtain the final answer:

$$\log(31) = 37 + 4 \cdot 368 + 32 \equiv 329 \bmod 606.$$

To check the solution, compute $5^{329} \equiv 31 \bmod 607$. ■

The choice of B is essential. If B is too small then it will be very complicated to find powers of g that factor into primes in the factor base. On the other hand, if B is too large, then it will not be hard to find suitable powers of g, but a lot of linear algebra will be needed to compute the discrete logarithms.

Remark. Finding B-smooth numbers is not a trivial task. For small examples such as Example 6.4, B-smooth number can be found with trial division. This is not very efficient in a large field. Section 6.2.1 will mention and briefly explain sieving methods, such as the quadratic sieve, as an efficient method for finding B-smooth numbers. Moreover selecting the optimal size of B is not a trivial task. Section 6.2.2 will explain more about the optimal choice of B.

6.2.1 Sieving methods

A major part of the Index Calculus relies on finding B-smooth numbers. Not only in the first step of the Index Calculus, where smaller discrete logarithm problems need to be solved, but also when computing $g^{-j} \cdot h$. There are different methods for finding all B-smooth numbers in a field. Most of these methods are based on a sieving method or sieve. Sieving methods were already described by the ancient Greeks. The sieve of Erastosthenes is a well-known ancient Greek sieving method for making a list of prime numbers less than a certain integer [10]. The idea of this method is to mark all composite numbers iteratively, starting with the numbers divisible by the first prime, 2. Next mark all the numbers divisible by the second prime, 3, and so on. By adapting the Sieve of Erastosthenes slightly, a sieving method for prime factorization can be found.

Commonly used sieving method in the Index Calculus are the Quadratic Sieve and the Number Field Sieve [10]. The Quadratic Sieve works quite well for relatively small numbers. The Number field Sieve on the other hand is a lot more complicated, but works faster for really big numbers.

6.2.2 Running time

There are a few steps in the Index Calculus that can influence the running time of the method. In the first place finding enough equations such that g^x is B-smooth, i.e. relations of the form of equation (11). Finding these equations depends mainly on the choice of B. From [10], the optimal choice of B can be found.

It turns out that there is an algorithm such that finding the suitable number of equations , i.e. as many as the number of elements in the factor base or more, of the form

$$g^x = (\text{product of primes in the factor base})$$

takes sub-exponential time.

Another time consuming step in the Index Calculus method is checking whether elements are B-smooth. A sieving method like the number field sieve can decrease the running time of this step enormously.

Overall, the running time of the Index Calculus can be approximated by $c \cdot \exp\{\sqrt{2 \ln p \ln \ln p}\}$, where c is a constant, hence the Index Calculus is a sub-exponential solving algorithm for the discrete logarithm problem in a finite field.

Note that all the best known algorithms to solve general discrete logarithm problems are exponential. This means that if there is the possibility of reducing the discrete logarithm problem to a problem over a finite field, the complexity of the discrete logarithm problem might decrease a lot. This is a crucial observation regarding the elliptic curve discrete logarithm problem, as will become clear in Section 6.4.

6.3 Elliptic curve discrete logarithms

Based on two examples, it will now be shown that especially the elliptic curve discrete logarithm problem for an elliptic curve defined over a finite field is interesting for cryptographic applications. The examples will compare the complexity of the discrete logarithm problem for an elliptic curve defined over a field of infinite and finite order.

Example 6.5. Consider the elliptic curve $E\colon y^2 = x^3 - x + 1$ over $\mathbb{Q}$. An obvious point on this curve is the point $(1, 1)$. Start adding the point $P = (1, 1)$ to itself. To find the point $2 \cdot P$, the third point of intersection of the tangent line at P and the elliptic curve E has to be found. The tangent line at P is given by $y = x$. This line intersects the elliptic curve in the points $P = (1, 1)$ (intersection point of multiplicity 2) and $(-1, -1)$. Hence the x-coordinate of the point $2 \cdot P$ is given by -1. To find the x-coordinate of $3 \cdot P$, the third point of intersection of the line through P and $2 \cdot P$ needs to be found. The line through P and $2P$ is given by $y = 1$, which intersects the curve in the points $P = (1, 1)$, $2 \cdot P = (-1, 1)$ and $(0, 1)$. Hence the x-coordinate of $3 \cdot P$ is equal to 0, etc. The x-coordinates of the points $k \cdot P$ for $k = 1, \ldots, 12$ are given in Table 3.

Multiples of P	x-coordinate
$1{\cdot}P$	1
$2{\cdot}P$	-1
$3{\cdot}P$	0
$4{\cdot}P$	3
$5{\cdot}P$	5
$6{\cdot}P$	1/4
$7{\cdot}P$	-11/9
$8{\cdot}P$	19/25
$9{\cdot}P$	56
$10{\cdot}P$	159/121
$11{\cdot}P$	-255/361
$12{\cdot}P$	-223/784

Table 3: The complexity of multiples of $P \in E(\mathbb{Q})$

Keep adding P to itself and observe that the size of the x-coordinate of $k \cdot P$ grows. For points of infinite order, this happens in general. If R is an arbitrary point of infinite order on an elliptic curve, the size of the coordinates of $k \cdot R$ get larger as k gets larger.

Now consider the discrete logarithm problem for two points P and Q on an elliptic curve, i.e. the goal is to find an integer n such that $n \cdot P = Q$. The previous observation implies that to find the range of n, one can look at the size of the x-coordinate of P and Q. This means that solving the discrete logarithm problem for an elliptic curve over $\mathbb{Q}$ is not computationally expensive. ■

On the other hand, when working over a finite field, this property is absent and therefore cannot be used to solve the discrete logarithm problem. This makes the problem a lot harder to solve over a finite field. The next example will demonstrate this.

Example 6.6. Consider the elliptic curve $E\colon y^2 = x^3 - x + 1$ defined over the finite field $\mathbb{F}_{97}$. Adding the point $P = (1, 1)$ on the elliptic curve to itself now yields the x-coordinates given in Table 4. The strategy for finding these points is similar to the previous example, but the computations are now over the field $\mathbb{F}_{97}$.

Multiples of P	x-coordinate
$1 \cdot P$	1
$2 \cdot P$	96
$3 \cdot P$	0
$4 \cdot P$	3
$5 \cdot P$	5
$6 \cdot P$	73
$7 \cdot P$	85
$8 \cdot P$	90
$9 \cdot P$	56
$10 \cdot P$	43
$11 \cdot P$	31
$12 \cdot P$	57

Table 4: The complexity of multiples of $P \in E(\mathbb{F}_{97})$

There seems to be almost no structure to the x-coordinates of subsequent multiples, and there is no obvious way, other than brute-force searching, to gain information about the integer k when looking at the x-coordinate of $k \cdot P$. ■

Examples like this suggest that the discrete logarithm problem for an elliptic curve defined over a finite field is more complicated and hence it is a more suitable problem for cryptographic applications.

6.4 The MOV attack

Finally the MOV can be properly introduced. The idea of the MOV attack is to reduce discrete logarithms in the group of points of an elliptic curve over a finite field to logarithms in the multiplicative group of a (perhaps larger) finite field. The MOV attack is based on establishing an isomorphism between a subgroup of $E(\mathbb{F}_q)$ and a subgroup of the field K. The Weil pairing introduced in Section 3.4 will be essential in the MOV attack. Recall that for a point $P \in E(\mathbb{F}_q)$ of order N, the Weil pairing is described by a special bilinear map that associates an N'th root of unity with a given pair of N-torsion points. So if $\gcd(N, q) = 1$ and $R, S \in E[N]$, then $e_N(R, S)$ is an N'th root of unity. Furthermore, for a field K, recall that the group of N'th roots of unity is given by $\mu_N = \{x \in \overline{K} \mid x^N = 1\}$. This is a cyclic group of order N. The N'th root of unity ζ_N is a primitive N'th root of unity if and only if $\zeta^N = 1$ and $\zeta^k = 1$ for all $1 \leq k < N$.

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$, let P and Q be two points in $E(\mathbb{F}_q)$. Let N be the order of the point P, and to apply the Weil pairing, assume that $\gcd(N, q) = 1$.

6.4.1 Embedding degree

The embedding degree of an integer in a finite field is the basis of the MOV attack [21].

Definition 6.7. *The embedding degree of an integer N in the finite field $\mathbb{F}_q$ is defined as the smallest positive integer d such that*

$$\mu_N \subseteq F_{q^d}^{\times}.$$

Remark. The embedding degree can be defined equivalently as the smallest positive integer d such that

$$q^d \equiv 1 \bmod N.$$

This holds since the group $\mathbb{F}_{q^d}^{\times}$ is cyclic of order $q^d - 1$.

Lemma 6.8. *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ and assume that $\gcd(N, q) = 1$. Let d be the embedding degree of the integer N in the field $\mathbb{F}_q$. Then*

$$E[N] \subset E(\mathbb{F}_{q^d}).$$

Proof. By showing that a basis of $E[N]$ is contained in $E(\mathbb{F}_{q^d})$, it will be shown that $E[N] \subset E(\mathbb{F}_{q^d})$.

Let $P \in E(\mathbb{F}_q)$ be a point of order N and pick $T \in E[N]$ such that $\{P, T\}$ forms a basis for $E[N]$. As usual, let ϕ_q denote the Frobenius endomorphism. The goal is to show that $\phi_{q^d}(T) = T$, because this would imply that $T \in E(\mathbb{F}_{q^d})$. For the Frobenius endomorphism it holds that

$$\phi_q(P) = P,$$

since $P \in E(\mathbb{F}_q)$, and

$$\phi_q(T) = a \cdot P + b \cdot T,$$

for some elements $a, b \in \mathbb{Z}/N\mathbb{Z}$.

Since $\{P, T\}$ forms a basis for $E[N]$, the Weil pairing $e_N(P, T)$ is a primitive N'th root of unity, as in Corollary 9. By the properties of the Weil pairing from Theorem 3.4, the Weil pairing of P and T satisfies

$$\begin{aligned} e_N(P,T)^q &= \phi_q(e_N(P,T)) \\ &= e_N(\phi_q(P), \phi_q(T)) \\ &= e_N(P, a \cdot P + b \cdot T) \\ &= e_N(P,P)^a e_N(P,T)^b \\ &= e_N(P,T)^b. \end{aligned}$$

The fact that $e_N(P, T)$ is a primitive N'th root of unity now implies that $b \equiv q \bmod N$. Therefore

$$\begin{aligned} \phi_q(T) &= a \cdot P + q \cdot T \\ \phi_q(\phi_q(T)) &= a \cdot P + q(a \cdot P + q \cdot T) \\ &= a \cdot P + qa \cdot P + a^2 \cdot T \qquad (13) \\ & \qquad (14) \end{aligned}$$

In general, it holds that

$$(\phi_q \circ \cdots \circ \phi_q)(T) = (a(q + \cdots + q^{d-1})) \cdot P + q^d \cdot T.$$

The integer d is the embedding degree of N in $\mathbb{F}_q$, so

$$q^d \equiv 1 \bmod N,$$

and furthermore

$$q + q + q^2 + \cdots + q^{d-1} \equiv 0 \bmod N.$$

This last congruence holds since $x^d - 1$ can be factored as $(x-1)(1+x+x^2+\cdots+x^{d-1})$ in $\mathbb{F}_q$. From equation 13 the Frobenius endomorphism satisfies $\phi_{q^d}(T) = (a(q+\cdots+q^{d-1}))\cdot P + q^d \cdot T$. The reasoning above implies that $\phi_{q^d}(T) = T$, which proves that $T \in E(\mathbb{F}_{q^d})$.

Therefore the basis of $E[N]$ given by $\{P, T\}$ is contained in $E(\mathbb{F}_{q^d})$ and hence $E[N] \subset E(\mathbb{F}_{q^d})$.

□

6.4.2 The algorithm

Recall that E is an elliptic curve over $\mathbb{F}_q$, P and Q are two points in $E(\mathbb{F}_q)$ and N is the order of the point P. The goal of the MOV attack is to find the integer n such that $Q = n \cdot P$. Before describing the algorithm for the MOV attack, it will be verified that such an integer exists.

Lemma 6.9. *There exists an integer n such that $Q = n \cdot P$ if and only if $N \cdot Q = \infty$ and $e_N(P, Q) = 1$, where e_N is the Weil pairing.*

Proof. Assume that $Q = n \cdot P$. Then $N \cdot Q = nN \cdot P = \infty$. By using the bilinearity of the Weil pairing and property 3 of the Weil pairing from Theorem 3.4, the Weil pairing of P and Q satisfies

$$\begin{aligned} e_N(P, Q) &= e_N(P, n \cdot P) \\ &= e_N(P, P)^n \\ &= 1^n = 1 \end{aligned}$$

For the other direction of the proof assume that $N \cdot Q = \infty$ and $e_N(P, Q) = 1$. This immediately implies that $Q \in E[N]$. For the N-torsion points of the elliptic curve E it holds that $E[N] \cong \mathbb{Z}/N\mathbb{Z} \oplus \mathbb{Z}/N\mathbb{Z}$, since the characteristic of F_q does not divide N (see Theorem 3.2). Hence the element Q can be written in the form $Q = a \cdot P + b \cdot R$, for some integers a and b, where R is chosen such that $\{P, R\}$ forms a basis for $E[N]$. This means that if it can be shown that $b \equiv 0 \bmod N$, then the lemma follows. By Corollary 3.5, $\zeta \coloneqq e_N(P, R)$ is a primitive N'th root of unity. From the assumption that $e_N(P, Q) = 1$ it now follows that

$$\begin{aligned} 1 &= e_N(P, Q) \\ &= e_N(P, a \cdot P + b \cdot R) \\ &= e_N(P, P)^a e_N(P, R)^b \\ &= e_N(P, R)^b \\ &= \zeta^b, \end{aligned}$$

where moreover the bilinearity and property 3 of the Weil pairing as in Theorem 3.4 were used. Since ζ is a primitive root N'th root of unity it follows that $b \equiv 0 \bmod N$, and therefore $Q = a \cdot P$. □

Thus, to use the MOV attack to solve the discrete logarithm problem $Q = n \cdot P$, let P and Q be two N-torsion points on the elliptic curve E. First the basic idea of the MOV attack will be described.

Let d be the embedding degree of the integer N in $\mathbb{F}_q$. Suppose that it is possible to find an element $T \in E[N]$ such that P and T generate $E[N]$, or equivalently $e_N(P,T)$ has order N. From the definition of the Weil pairing the elements $e_N(P,T)$ and $e_N(Q,T)$ are N'th roots of unity and since d is the embedding degree, $e_N(Q,T)$ and $e_N(P,T)$ are elements in the multiplicative group $\mathbb{F}_{q^d}^{\times}$. Now the discrete logarithm problem $e_N(Q,T) = e_N(P,T)^n$ is a discrete logarithm problem in the multiplicative group of a finite field extension. Then, to solve the discrete logarithm problem, all the computations are in $\mathbb{F}_{q^d}$ since by Lemma 6.8 it follows that the embedding degree d of the integer N in $\mathbb{F}_q$ is the smallest integer d such that $E[N] \in E(\mathbb{F}_{q^d})$.

This method solves the correct discrete logarithm problem since by the bilinearity of the Weil pairing, it follows that

$$\zeta_1 = e_N(Q,T) = e_N(n \cdot P, T) = e_N(P,T)^n = \zeta_2^n.$$

The complete algorithm for the MOV attack also contains a method on how to find the point T such that $e_N(P,T)$ has order N. It takes as input the two points N-torsion points P and Q of the elliptic curve and outputs the integer n such that $Q = n \cdot P$. It can be found in [17] and [26]. The algorithm that implements the MOV attack consists of the following steps.

MOV attack algorithm

1. Choose a random point $T \in E(\mathbb{F}_{q^d})$, where d is the embedding degree of the integer N in $\mathbb{F}_q$.
2. Compute the order M of the point T.
3. Compute $r = \gcd(M, N)$ and $T_1 = (M/r)T$. Then the point T_1 is a point of order r. Since r divides N this means that $T_1 \in E[N]$.
4. Compute $\zeta_1 = e_N(Q, T_1)$ and $\zeta_2 = e_N(P, T_1)$. Then both ζ_1 and ζ_2 are N'th roots of unity, hence $\zeta_1, \zeta_2 \in \mathbb{F}_{q^d}^{\times}$.
5. Solve the discrete logarithm problem $\zeta_1 = \zeta_2^n$ in $F_{q^d}^x$. This gives as a solution $n \bmod r$.
6. Continue the same process with random points T until the least common multiple of the subsequent values of r is equal to N. This determines the value of $n \bmod N$.

Remark. When the least common multiple of the subsequent values of r is equal to N, the element $e_N(P,T)$ has order N. Using the Chinese remainder theorem, the value of $n \bmod N$ can be determined.

Remark. The algorithm does not always find a point T such that P and T generate $E[N]$ on the first attempt, potentially $r = 1$ could occur often. However, this is not the case as the algorithm finds the desired point T with a very high probability [26]. This means that only after a few iterations of the algorithm, n will be found and the discrete logarithm problem will be solved.

The embedding degree d of the integer N in the field $\mathbb{F}_q$ determines the complexity and running time of the the MOV attack. In [21] it was shown that for an arbitrary chosen elliptic curve E over $\mathbb{F}_q$, the embedding degree of an integer N, where N is a large prime divisor of the order of $E(\mathbb{F}_q)$, is proportional to N. This implies that for an arbitrary elliptic curve E over a finite field, the MOV attack reduces the discrete logarithm problem in $E(\mathbb{F}_q)$ to a discrete logarithm problem in a much larger finite field, so the MOV attack has exponential running time. Therefore, in general, the MOV attack does not give computational advantage for solving the discrete logarithm problem, compared to solving the discrete logarithm problem with a collision algorithm. However, if the embedding degree of the integer d in $\mathbb{F}_q$ is reasonably small then the MOV attack could possibly yield a large computational advantage for solving the elliptic curve discrete logarithm problem.

6.4.3 The MOV attack for supersingular curves

The MOV attack is only a feasible method for solving the elliptic curve discrete logarithm problem if the field $\mathbb{F}_{q^d}$ is not much larger than the original field $\mathbb{F}_q$. Elliptic curves for which this holds are called pairing-friendly elliptic curves [21]. For pairing-friendly elliptic curves, Index Calculus methods can be applied to efficiently solve the discrete logarithm problem in $\mathbb{F}_{q^d}$. This will result in a much faster solving algorithm for the elliptic curve discrete logarithm problem. An interesting family of pairing friendly elliptic curves is the family of supersingular elliptic curves. In the course of this section it will be shown that for supersingular elliptic curve, the embedding degree of an integer N in the field $\mathbb{F}_q$ can in general be taken as a very small integer. This means that for supersingular elliptic curves, the extension field $\mathbb{F}_{q^d}$ of $\mathbb{F}_q$ is not much larger than the original field, as desired.

For supersingular elliptic curves with trace identically zero, the embedding degree takes a conveniently small value.

Proposition 6.10. *Let E be an elliptic curve defined over the finite field $\mathbb{F}_q$, and suppose that $a = q + 1 - \#E(\mathbb{F}_q) = 0$. If there exists a point P of order N in $E(\mathbb{F}_q)$, where N is a positive integer, then $E[N] \subseteq E(\mathbb{F}_{q^2})$.*

Proof. First note that for the Frobenius endomorphism ϕ_q it holds that $(x, y) \in E(\mathbb{F}_q)$ if and only if $\phi_q(x, y) = (x, y)$. This follows from a property of the Frobenius homomorphism on a finite field: $x \in \mathbb{F}_q$ if and only if $\phi_q(x) = x$. The curve E is a supersingular and $E(\mathbb{F}_q)$ has $q+1$ points. Let S be an N-torsion point on the elliptic curve. If the point S satisfies $\phi_{q^2}(S) = S$, then by the previous argument $S \in E(\mathbb{F}_{q^2})$. Recall from Theorem 4.4 that the Frobenius trace a of an elliptic curve is the unique integer that satisfies

$$\phi_q^2 - a\phi_q + q = 0.$$

As a is equal to zero, this equality is equivalent to

$$\phi_q^2 = -q.$$

Substituting the N-torsion point S into this equality gives

$$\phi_q^2(S) = -qS.$$

The fact that there exists a point of order N on the elliptic curve implies that N divides $q+1$, that is $q \equiv -1 \bmod N$. Therefore

$$\phi_q^2(S) = S,$$

which concludes the proof. □

For a prime $p < 5$ there are supersingular elliptic curves defined over the field $\mathbb{F}_p$ with trace not identically equal to zero. This was shown in Corollary 5.2. In cryptographic applications based on the discrete logarithm problem, such curves will never occur, but for completeness it is good to mention what values the embedding degree can take for these curves. For such supersingular elliptic curves, the argument is similar to before, but the embedding degree d can take the values in $\{2, 3, 6\}$. Details of this argument can be found in [17].

The following two results from [17] fully determine the running time of the MOV attack for supersingular elliptic curves.

Theorem 6.11. *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. Then the reduction of the discrete logarithm problem for $E(\mathbb{F}_q)$ to a discrete logarithm problem an extension $\mathbb{F}_{q^m}$ of the finite field $\mathbb{F}_q$ takes probabilistic polynomial time.*

Theorem 6.12. *By using the MOV attack the discrete logarithm problem for a supersingular elliptic curve E defined over a finite field $\mathbb{F}_q$ can be solved in probabilistic sub-exponential time.*

Full proofs can be found in [17]. The next section contains examples of the MOV attack on supersingular curves over finite fields with large order, which will show the validity of the theorems in particular examples.

It can be concluded that for supersingular elliptic curves, the elliptic curve discrete logarithm problem can in most cases be translated to a discrete logarithm problem over the finite field $\mathbb{F}_{q^2}$. In this field the discrete logarithm can be solved in sub-exponential time using Index Calculus methods. Hence, from the MOV attack it follows that supersingular elliptic curves are not safe to use in cryptographic applications base on the discrete logarithm problem.

7 Examples of the MOV attack

The algorithm of the MOV attack for solving the discrete logarithm problem in case of supersingular elliptic curves has been described theoretically in the previous section. Several different statements were made regarding the complexity of the elliptic curve discrete logarithm problem when the MOV attack is applied. The efficiency of this strategy will now be shown in practice. In other words, for different examples of supersingular elliptic curves, the running time of the MOV attack will be compared to the the running time of a general solving method for the discrete logarithm problem.

Note that to actually compare the running times of different algorithms, the supersingular elliptic curve needs to be constructed over a finite field that is considerably large. Otherwise, the running times will be too short.

7.1 Generating supersingular elliptic curves

In order to perform the MOV attack on a discrete logarithm problem for supersingular elliptic curves, suitable supersingular elliptic curves need to be generated. In Section 5, several different characterizations of supersingular curves were given. In particular, Proposition 5.4 allowed for constructing supersingular elliptic curves over a finite field $\mathbb{F}_q$ with a certain restriction on q. These special supersingular elliptic curves could be used to verify the efficiency of the MOV attack. Also complex multiplication methods according to [24] can be used to generate supersingular elliptic curves. The method described by Sutherland in [24] is probably the best known method nowadays for generating supersingular elliptic curves.

However, in this thesis, a slightly different approach is taken to construct supersingular elliptic curves that are suitable for demonstrational purposes. In other words, supersingular elliptic curves with small coefficients are generated over a large field $\mathbb{F}_p$. This means that 'new' supersingular elliptic curves are generated, the curves are 'new' in the sense that they are not constructed by means of the previously mentioned methods.

Together with a fellow student, Sven Hofman [11], who was studying anomalous elliptic curves, an algorithm was implemented that generates supersingular and anomalous elliptic curves. Our approach for finding these supersingular and anomalous elliptic curves can be described in the following way. For a prime p in a range of large primes, Weierstrass equations with coefficients A, B both ranging from -100 to 100 are constructed over the field $\mathbb{F}_p$. For each pair of coefficients it has to be verified whether the corresponding Weierstrass equation defines an elliptic curve. Recall from the introduction and Section 2, that only a Weierstrass equation with coefficients A, B that satisfy the condition $4A^3 + 27B^2 \neq 0$ defines an elliptic curve. Then, by using Schoof's algorithm, the order of the elliptic curve can be computed. If the order of the elliptic curve is equal to $p + 1$, then a supersingular elliptic curve is found (Corollary 5.2). If the order of the elliptic curve is equal to p, then an anomalous elliptic curve is found [11]. Then exactly the same steps are repeated for the next prime in the range of primes. As for the starting prime, a prime in the range 2^{48} was chosen to make it feasible to find multiple examples of supersingular elliptic curves within a few weeks.

Remark. To verify the number of points of an elliptic curves, Schoof's algorithm is not the only method that can be used. Another method that can be used is based on selecting random point on the elliptic curve.

If the elliptic curve has prime order p (which happens if the curve is anomalous) then points on the elliptic curve can only have order 1 or p. If multiplying a random point of the elliptic curve by p gives the identity element then indeed the curve has order p. However, this method is not very efficient in case of supersingular elliptic curves, since the order of supersingular elliptic curve is a composite number. For a supersingular curve, the points on the curve can have as order every divisor of $p+1$. Therefore multiplying a random point by $p+1$ does not give much information in this case. The only thing that is certain is that if multiplying a random point on the elliptic curve by $p+1$ does not give the identity element, then the curve can not have order $p+1$.

Because of this slight inconvenience for supersingular elliptic curves when using random points, the implementation of the method for generating supersingular (and anomalous) elliptic curves combines multiplying by random points with Schoof's algorithm to determine the order of the elliptic curves.

7.1.1 Finding the prime range

It will be briefly explained here why a prime of size 2^{48} was used in the code.

Assume that the method for finding supersingular elliptic curves is implemented in a code that will run for 15 days. What is desired range of primes to look in to almost certainly find supersingular elliptic curves and anomalous curves by using the method as described above? In other words, the goal is to find the right balance between the available time and the size of the prime such that supersingular elliptic curves (and anomalous curves) are found with a high probability. For the details of finding anomalous elliptic curves, see [11].

To find this balance one has to look at the complexity of the implemented code. The complexity is mainly determined by the complexity of Schoof's algorithm. All the other steps in the code have a running time that is negligible. The running time of Schoof's algorithm depends on the order of magnitude of the prime. Let us say that Schoof's algorithm in the field $\mathbb{F}_p$ takes approximately m steps to find the order of the elliptic curve.

For a prime of size $p \approx 2^n$, the probability of finding a supersingular curve is approximately equal to [24]:

$$Prob = c \cdot \frac{1}{4\sqrt{2^n}},$$

where c is a constant. Therefore, for a prime of size $p \approx 2^n$ the number of curves that needs to be checked approximately to find one supersingular curve can be given in terms of n and $Prob$. This exact value will not be given here. Let us denote the number of curves that need to be checked for $p \approx 2^n$ by R.

For each of these curves, finding the order of the elliptic curve requires approximately m steps. This gives a total number of steps that is needed to find at least one supersingular elliptic curve for a prime of size $p \approx 2^n$, namely for $p \approx 2^n$ approximately $R \cdot m$ steps are needed. The desired

value of n is the value such that the required number of curves can be checked in the given amount of time. In other words, this is the value n such that the total amount of steps that is needed to find the desired curves, $R \cdot m$, can be executed in the given amount of time.

It turns out that for primes of size 2^{50}, the code would be feasible with the software on the available computers. To be completely safe, primes in the range 2^{48} are used.

7.1.2 Implementing the MOV attack

The code is implemented in SageMath, see Appendix A.4.1. There are a few subtleties regarding the implementation of this code.

First of all, using Schoof's algorithm to find the order of an elliptic curve can be seen as an experiment with outcome success (if the curve is supersingular) or failure (if the curve is not supersingular). If the experiments are not independent, the code does not work as desired. Here two experiments are independent if there is not relation between the number of points for different elliptic curves. Therefore, elliptic curves that are isomorphic need to be avoided; for such curves there clearly is a relation between the order of the elliptic curves. From Theorem 2.2 it is clear how isomorphism between elliptic curves are defined. An elliptic curve defined over a field $\mathbb{F}_p$ defined by the Weierstrass equation with coefficients (A, B) is isomorphic to an elliptic curve defined by the Weierstrass equation with coefficients (d^2A, d^3B), over the field $\overline{\mathbb{F}}_p$, where $d \in \overline{\mathbb{F}}_p$. Moreover, two elliptic curves with coefficients (A, B) and (d^2A, d^3B) are isomorphic over $\mathbb{F}_p$ if d is a square in $\mathbb{F}_p$ [2].

For supersingular elliptic curves there two important statements regarding isomorphisms. Any curve isomorphic to a supersingular elliptic curve is still supersingular and furthermore any twist of a supersingular elliptic curve is still supersingular and in particular also has $p + 1$ points. This can be seen from equation (10) in Section 2.2, which describes a relation for the number of points of two elliptic curves with the same j-invariant. It also follows from the fact that supersingular elliptic curves have a quaternion endomorphism ring. So, if E is a supersingular curve, E has $p + 1$ points, then any elliptic curve with the same j-invariant also has $p + 1$ points. For supersingularity is does not matter if the curve is defined over $\mathbb{F}_p$, $\mathbb{F}_q$ or an algebraic closure, the endomorphism ring either is a quaternion algebra or it is not.

This means that in the code in SageMath, for a certain prime p, Schoof's algorithm is only applied to an elliptic curves with a j-invariant that did not occur before.

Furthermore, elliptic curves with complex multiplication need to be taken into account, these curves do not need to be checked. In Table 1, all j-invariants of elliptic curves over $\mathbb{Q}$ that have complex multiplication are given. Therefore, in the code in A.4.1, the j-invariant of the elliptic curve over $\mathbb{Q}$ is checked before actually constructing the elliptic curve over $\mathbb{F}_p$.

After running the code for a few weeks, the code resulted in nine supersingular elliptic curves, which are listed in Table 5.

	Prime p	Supersingular curve
E_1	$p = 2^{48} + 180307$	$y^2 = x^3 - 44x + 9$
E_2	$p = 2^{48} + 188323$	$y^2 = x^3 + 76x + 25$
E_3	$p = 2^{48} + 228055$	$y^2 = x^3 - 11x + 67$
E_4	$p = 2^{48} + 241647$	$y^2 = x^3 - 8x + 20$
E_5	$p = 2^{48} + 244627$	$y^2 = x^3 - 69x + 16$
E_6	$p = 2^{48} + 293053$	$y^2 = x^3 + 32x + 90$
E_7	$p = 2^{48} + 356437$	$y^2 = x^3 - 83x + 27$
E_8	$p = 2^{48} + 389671$	$y^2 = x^3 + 65x + 51$
E_9	$p = 2^{48} + 417547$	$y^2 = x^3 - 11x + 89$

Table 5: Supersingular elliptic curves

Only the first two curves are suitable for demonstrational purposes since these curves are generated by an integral point with small coefficients over $\mathbb{F}_p$. The other curves do not have useful integral points or convenient generators.

7.1.3 Comparing the running times

Example 7.1. Consider the supersingular elliptic curve $E_1 \colon y^2 = x^3 - 44x + 9$ from Table 5. The group of points $E_2(\mathbb{Q})$ is generated by $P = (0, 3)$ and $Q = (8, 13)$. Over $\mathbb{Q}$, these two points are independent, meaning that there does not exist an integer n such that $Q = n \cdot P$ or $P = n \cdot Q$. However, $E_2(\mathbb{F}_p)$ is generated by the point P. This means that over $\mathbb{F}_p$, the integer n such that $Q = n \cdot P$ does exist. The MOV attack is applied to find this integer. Appendix A.4.2 contains the implemented MOV attack for this particular example. From Appendix A.4.2, the solution is given by

$$n = 2141618601636,$$

The solution can be verified by computing $n \cdot P$ for this value of n, and observing that this equal to Q, as expected. ■

Example 7.2. Consider the supersingular elliptic curve $E_3 \colon y^2 = x^3 + 76x + 25$. The points $P = (0, 5)$ and $Q = (7, 30)$ are two points on this curve. As in the previous example, the group of points $E(\mathbb{Q})$ is generated by P and Q. Over the field $\mathbb{Q}$, these two points are independent, so there does not exist an integer n such that $Q = n \cdot P$ or $P = n \cdot Q$. However, over the finite field $\mathbb{F}_p$ for $p = 2^{48} + 188323$, the group of points of the elliptic curve is generated by P. Therefore, the MOV attack can be applied to find the integer n such that $Q = n \cdot P$. In Appendix A.4.3, the MOV attack is implemented for this particular example. From Appendix A.4.3, the solution is given by

$$n = 142967077962838,$$

The solution is verified by checking the equality $Q = n \cdot P$ for $n = 142967077962838$. ■

Of course, the goal in this section is to compare the running time of the MOV attack for these two particular examples to the running time of a general solving algorithm. To this end, the baby step, giant step method will now be applied to the elliptic curve discrete logarithm problem for the elliptic curves E_1 and E_2 with the points P and Q as described in the examples.

Appendix A.4.2 implements the baby step, giant step method for Example 7.1 and Appendix A.4.3 implements the baby step, giant step method for Example 7.2.

It turns out that for the Examples 7.1 and 7.2 the running time of the MOV attack is approximately equal to

$$t_{run} = 0.1s,$$

while the running time of the baby step, giant step method for Example 7.1 is approximately equal to

$$t_{run} = 12761.5s \approx 3.5h.$$

The MOV attack thus yields a significantly faster solving algorithm for the elliptic curve discrete logarithm problem compared to the baby step, giant step method.

Remark. The implementations of the MOV attack and the baby step, giant step method in Appendix are naive implementations; the running times of these codes are not optimal.

Remark. One thing that was not mentioned before is that for a cyclic group G, whose order is a composite integer N, the discrete logarithm problem can be reduced to a small number of simpler problems by means of the Pohlig-Hellman method, which was not discussed in this thesis. The Pohlig-Hellman method reduces the problem to a discrete logarithm problem in different groups with prime-power order. For details on the Pohlig-Hellman method [26] can be consulted. The Pohlig-Hellman method implies that the complexity of the discrete logarithm problem for supersingular elliptic curves (but also in general groups) also depends on the factorization of $p+1$. If the largest prime factor of $p+1$ is considerably smaller than $p+1$, then Pohlig-Hellman can efficiently solve the discrete logarithm problem. The largest prime factor of $p+1$ determines the running time of the Pohlig-Hellman method.

In Example 7.1, it holds that $N = p + 1 = 2^2 \cdot 3 \cdot 41 \cdot 6661 \cdot 85888547$. The Pohlig-Hellman method can efficiently solve the discrete logarithm problem by reducing the problem to 5 smaller discrete logarithm problems in groups of order respectively $2^2, 3, 41, 6661, 85888547$. In Example 7.2 the same thing happens. Here $N = p + 1 = 2^2 \cdot 3^2 \cdot 5 \cdot 5701 \cdot 274293961$. This means that Pohlig-Hellman reduces the problem to discrete logarithms problems over groups of size $2^2, 3^2, 5, 5701, 274293961$. This implies that it is also necessary to compare the running time of the MOV attack to the running time of the Pohlig-Hellman method. This will be done for both examples. It turns out that for Example 7.1 the running time of the Pohlig-Hellman method is approximately equal to

$$t_{run} = 10.5s.$$

For Example 7.2, the running time of the Pohlig-Hellman method is approximately equal to

$$t_{run} = 6.5s.$$

The Pohlig-Hellman method is a lot faster than the baby step, giant step method. However, the MOV attack still yields a significantly faster method for solving the elliptic curve discrete logarithm problem.

Therefore, as is also clear from the examples, by means of the MOV attack, the elliptic curve discrete logarithm problem for supersingular elliptic curves can be solved too fast for cryptographic applications.

8 Conclusion and discussion

Elliptic curves have been used in cryptography for a several decades. Following the suggestion of Koblitz and Miller the elliptic curve discrete logarithm problem has been used for this purpose. Elliptic curves defined over a finite field have proven to give an interesting and complex discrete logarithm problem. Supersingular elliptic curves, the special family of elliptic curves characterized by having Frobenius trace equal to zero in the finite field, were once thought to be very promising to use for this purpose (because of the ease of computations on them).

However, the algorithm introduced in 1996 by Menezes, Okamoto and Vanstone, known as the MOV attack yields a sub-exponential solving algorithm for solving the elliptic curve discrete logarithm problem for supersingular elliptic curves. By means of the Weil pairing, the MOV attack reduces the elliptic curve discrete logarithm problem to a discrete logarithm problem in a finite field. In case of supersingular elliptic curves, the discrete logarithm problem in the finite field can be solved efficiently by applying Index Calculus methods since the embedding degree is small.

By constructing examples of supersingular elliptic curves, the power of the MOV attack was shown in practice. This allowed for comparing the running times of the MOV attack to the running time of a general method for solving the elliptic curve discrete logarithm problem. From these comparisons, the theoretical results were confirmed; the MOV solved the elliptic curve discrete logarithm problem significantly faster. Therefore, to guarantee the safety of a cryptographic system, supersingular elliptic curves should not be used in cryptographic applications based on the discrete logarithm problem.

It is important to note that the implementation of the methods in SageMath does not give a completely accurate representation of the running time of the described methods. This has to do with the fact that SageMath has a lot of pre-implemented methods available, some of which are used in the codes in the Appendix A.4.1. The running times of these pre-implemented methods of course slightly influences the running time of the complete program, which could lead to a perturbation of the results. However, the results are assumed to be good enough to at least say something about the running times of both methods.

It is also worth noting that supersingular elliptic curves have proven to be very useful in other areas of cryptography. In particular the supersingular isogeny key exchange [13] is a very promising cryptographic system suitable for post-quantum cryptography.

It will be very interesting to follow the developments and the “fall and rise and fall and rise” [7] of supersingular elliptic curves in cryptography in the future.

References

[1] Stephen Abbott et al. *Understanding Analysis*. Springer, 2015.

[2] Ian Blake, Gerald Seroussi, Gadiel Seroussi, and Nigel Smart. *Elliptic Curves in Cryptography*, volume 265. Cambridge university press, 1999.

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

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

[5] Noam D. Elkies. Distribution of supersingular primes. *Astérisque*, 198(200):127–132, 1991.

[6] Andreas Enge. *Elliptic Curves and Their Application to Cryptography, An Introduction*. Springer Science+Business Media New York, first edition, 1999.

[7] Steven Galbraith. The fall and rise and fall and rise of supersingular elliptic curves (in cryptography). *University of Auckland, New Zealand*, 2018.

[8] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer Science & Business Media, 2006.

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

[10] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. *An Introduction to Mathematical Cryptography*, volume 1. Springer, 2008.

[11] Sven Hofman. The discrete logarithm problem on anomalous elliptic curves. Bachelor's thesis, University of Groningen, Faculty of Science and Engineering, July 2020.

[12] James Ivory. Demonstration of a theorem respecting prime numbers. *New series of The mathematical repository*, 1, 1806.

[13] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International Workshop on Post-Quantum Cryptography*, pages 19–34. Springer, 2011.

[14] Neal Koblitz, Alfred Menezes, and Scott Vanstone. The state of elliptic curve cryptography. *Designs, codes and cryptography*, 19(2-3):173–193, 2000.

[15] Martijn Maas. Pairing-based cryptography. Master's thesis, Technische Universiteit Eindhoven, Department of Mathematics and Computing Science, January 2004.

[16] Alfred J. Menezes, Jonathan Katz, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC press, 1996.

[17] 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.

[18] Victor S. Miller. The weil pairing, and its efficient calculation. *Journal of cryptology*, 17(4):235–261, 2004.

[19] Steffen Müller and Jaap Top. *Group Theory*. University of Groningen, 2018.

[20] René Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Mathematics of computation*, 44(170):483–494, 1985.

[21] Joseph H. Silverman. *Arithmetic of Elliptic Curves*, volume 2. Springer, 1986.

[22] Joseph H. Silverman. *Advanced Topics in the Arithmetic of Elliptic Curves*, volume 151. Springer Science & Business Media, 1994.

[23] Harold M. Stark et al. A complete determination of the complex quadratic fields of class-number one. *The Michigan Mathematical Journal*, 14(1):1–27, 1967.

[24] Andrew V. Sutherland. Identifying supersingular elliptic curves. *LMS Journal of Computation and Mathematics*, 15:317–325, 2012.

[25] Jaap Top. *Security & Codes*. University of Groningen, 2020.

[26] Lawrence C. Washington. *Elliptic Curves, Number Theory and Cryptography*. Chapman & Hall/CRC, Taylor and Francis Group, second edition, 2008.

A Appendix

A.1 Finite abelian groups

This section gives a brief overview of the most important statements regarding finitely generated abelian groups.

Theorem A.1. *Let G be a finite abelian group. Then G has the following structure*

$$G \cong \mathbb{Z}/n_1\mathbb{Z} \oplus \mathbb{Z}/n_2\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/n_s\mathbb{Z},$$

where the integers $n_1, n_2, \ldots n_s$ satisfy $n_i | n_{i+1}$ for $i = 1, 2, \ldots s-1$.

Definition A.2. *The group G is called finitely generated if there exist integers $m_1, m_2, \ldots m_k$ and elements $g_1, g_2, \ldots, g_k$ from G such that every element $g \in G$ can be written in the form*

$$g = m_1 g_1 + m_2 g_2 + \cdots + m_k g_k.$$

Theorem A.3. *Let G be a finitely generated abelian group, then G has the following structure*

$$G \cong Z^r \oplus \mathbb{Z}/n_1\mathbb{Z} \oplus \mathbb{Z}/n_2\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/n_s\mathbb{Z}$$

The integer n is called the rank of the group G and the subgroup of G that is isomorphic to $\mathbb{Z}/n_1\mathbb{Z} \oplus \mathbb{Z}/n_2\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/n_s\mathbb{Z}$ is called the torsion-subgroup of G.

For more details, and the proof of Theorem A.1 and Theorem A.3, [19] can be used.

A.2 Fermat's little theorem

Fermat's little theorem is a theorem that was first stated in 1640, it can be very useful for computations in a field of nonzero characteristic [25].

Theorem A.4. *If p is a prime and $\gcd(a, p) = 1$, then $a^{n-1} \equiv a \ mod \ p$.*

There exist a lot of different proofs of Fermat's little theorem. For example, Ivory [12] proved Fermat's little theorem using the Euler totient function.

A.3 Division polynomials

Division polynomials were essential in the proof of Proposition 3.2, where the general structure of the group of n-torsion points of an elliptic curve was proven. This section provides a short introduction to division polynomials and it states the most important results. For more details, and the proofs of the statements, see [26].

Division polynomials are polynomials in $\mathbb{Z}[x, y, A, B]$ that are defined recursively in the following way

$$\begin{aligned}
\psi_0 &= 0 \\
\psi_1 &= 1 \\
\psi_2 &= 2y \\
\psi_3 &= 3x^4 + 6Ax^2 + 12Bx - A^2 \\
\psi_{2m+1} &= \psi_{m+2}\psi_m^3 - \psi_{m-1}\psi_{m+1}^3 \quad \text{for} \quad m \geq 2 \\
\psi_{2m} &= (2y)^{-1}(\psi_m)(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2) \quad \text{for} \quad m \geq 3.
\end{aligned}$$

Lemma A.5. *The polynomial ψ_n is a polynomial in $\mathbb{Z}[s, y^2, A, B]$ when n is odd, and ψ_n is a polynomial in $2y\mathbb{Z}[x, y^2, A, B]$ when n is even.*

Furthermore, define the polynomials

$$\begin{aligned}
\phi_m &= x\psi_m^2 - \psi_{m+1}\psi_{m-1} \\
\omega_m &= (4y)^{-1}(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2).
\end{aligned}$$

For an elliptic curve $E\colon y^2 = x^3 + Ax + B$, every term y^2 in these polynomials can be replaced by a polynomial in x. This means that the polynomials ϕ and ψ^2 are polynomials only in x.

Lemma A.6.

$$\begin{aligned}
\phi_n(x) &= x^{n^2} + \textit{terms of lower degree} \\
\psi_n^2(x) &= n^2 x^{n^2-1} + \textit{terms of lower degree}
\end{aligned}$$

The multiplication by n endomorphism can be described in terms of the division polynomials ψ_n and the polynomials ϕ_n and ω_n.

Theorem A.7. *Let $E\colon y^2 = x^3 + Ax + B$ be an elliptic curve defined over some field K with characteristic not equal to 2. Let $P = (x, y)$ be some point on the elliptic curve E and let n be a positive integer. Then*

$$n \cdot P = \Big(\frac{\phi_n(x)}{\psi_n^2(x)}, \frac{\omega_n(x, y)}{\psi_n(x, y)^3}\Big)$$

A.4 Codes

The codes in Appendices A.4.1 and A.4.4 were constructed in collaboration with fellow student Sven Hofman who was working on anomolous elliptic curves at the time [11].

A.4.1 SageMath - "curvefinder"

This code in SageMath implements the method described in Section 7.1 to find supersingular (and anomolous) elliptic curves. Note that `avoidInvariant` is constructed to avoid checking isomorphic curves and `CM_invariants` is constructed to avoid checking curves that have complex multiplication over $\mathbb{Q}$.

```
import time
P = Primes();
p = 1009 #16225927682921336339157801029503 1
CM_invariants = cm_j_invariants(QQ)

tStart=time.time()
while p < 2000:
    print('prime:',p);
    K = GF(p)
    avoidInvariant = set()

    for i in range(1,100):
        for j in range(1,100):

            for power1 in range(0,2):
                for power2 in range(0,2):

                    coeff1 = (-1)^(power1)*i;
                    coeff2 = (-1)^(power2)*j;

                    discriminant = K((4*coeff1^3+27*coeff2^2));

                    if discriminant != 0:
                        jInvariant = K(1728*(4*coeff1^3/discriminant));

                        if jInvariant not in avoidInvariant
                                    and jInvariant not in CM_invariants:
                            E = EllipticCurve(GF(p),[coeff1,coeff2])
                            avoidInvariant.add(jInvariant)
                            cardinality = E.cardinality();

                            if cardinality in [p,p+2]:
                                print('anomalous:',[coeff1,coeff2,p]);
                            elif cardinality == p+1:
```

```
                                print('supersingular:',[coeff1,coeff2,p]);

    p = P.next(p);

tEnd=time.time()
print("totalTime = " + str(tEnd-tStart))
```

A.4.2 SageMath - Example 2

The MOV attack

This SageMath code implements the MOV attack as described in Section 6.4.2 for the first example in Section 7.1.3.

```
from sage.arith.functions import LCM_list
import time

p = 2^48+180307
F.<t> = GF(p^2)
E = EllipticCurve(F,[-44, 9])
P = E(0,3)
N = P.order() # N = p+1
Q = E(8,13)
print("P = " + str(P.xy()) + " is the fixed generator, of order " + str(N))
print("Q = " + str(Q.xy()))
lcm=N+1
L1=[]
L2=[]

tStart=time.time()
while True:
    T = E.random_point()
    M = T.order()
    d = gcd(M,N)
    T = ZZ(M/d)*T
    print("T = " + str(T.xy()) + ", a (random) point of order " + str(d))
    a = P.weil_pairing(T,N)
    n = a.multiplicative_order()

    if 0 not in [x%n for x in L2]:
        b = Q.weil_pairing(T,N)
        l=log(b,a)
        L1.append(l)
        L2.append(n)
        if LCM_list(L2)==N:
            m=crt(L1,L2)
```

```
            print("a = e_N(T,P) = " + str(a))
            print("b = e_N(T,Q) = " + str(b))
            print("b = a^"+str(l))
            print('solution = ' + str(m))
            break
tEnd=time.time()
tTotal=tEnd-tStart
print("Total time = " + str(tTotal))
```

Baby step, giant step method

This SageMath code implements the baby step, giant step method as described in Section 6.1.1 for the first example in Section 7.1.3.

```
reset()
import time
from sage.groups.generic import bsgs

p=2^48+180307
F.<t> = GF(p^2)
E = EllipticCurve(F,[-44,9])
P = E(0,3)
N = P.order() # N = p+1
Q = E(8,13)

tStart=time.time()
a=bsgs(P,Q,[0,p],operation='+')
tEnd=time.time()
totalTime=tEnd-tStart
print('time',totalTime)
print('solution',a%(p+1))
```

A.4.3 SageMath - Example 3

The MOV attack

This SageMath code implements the MOV attack as described in Section 6.4.2 for the second example in Section 7.1.3.

```
from sage.arith.functions import LCM_list
import time

p = 2^48+188323
F.<t> = GF(p^2)
E = EllipticCurve(F,[76,25])
P = E(0,5)
N = P.order() # N = p+1
Q = E(7,30)
```

```
print("P = " + str(P.xy()) + " is the fixed generator, of order " + str(N))
print("Q = " + str(Q.xy()))
lcm=N+1
L1=[]
L2=[]

tStart=time.time()
while True:
    T = E.random_point()
    M = T.order()
    d = gcd(M,N)
    T = ZZ(M/d)*T
    print("T = " + str(T.xy()) + ", a (random) point of order " + str(d))

    a = P.weil_pairing(T,N)
    n = a.multiplicative_order()

    if 0 not in [x%n for x in L2]:
        b = Q.weil_pairing(T,N)
        l = log(b,a)
        L1.append(l)
        L2.append(n)
        if LCM_list(L2) == N:
            m=crt(L1,L2)
            print("a = e_N(T,P) = " + str(a))
            print("b = e_N(T,Q) = " + str(b))
            print("b = a^"+str(l))
            print('solution = ' + str(m))
            break
tEnd=time.time()
tTotal=tEnd-tStart
print("Total time = " + str(tTotal))
```

Baby step, giant step method

This SageMath code implements the baby step, giant step method as described in Section 6.1.1 for the second example in Section 7.1.3.

```
reset()
import time
from sage.groups.generic import bsgs

p=2^48+188323
F.<t> = GF(p^2)
E = EllipticCurve(F,[-44,9])
```

```
P = E(0,5)
N = P.order() # N = p+1
Q = E(7,30)

tStart=time.time()
a=bsgs(P,Q,[0,p],operation='+')
tEnd=time.time()
totalTime=tEnd-tStart
print('time',totalTime)
print('solution',a%(p+1))
```

A.4.4 Mathematica

The Mathematica code below computes the desired order of magnitude of the starting prime for the method of finding supersingular elliptic curves. It takes as input `NumberOfOperationsAvailable`, which is the number of seconds in 15 days, and outputs `i`. The desired order of the prime is equal to 2^i.

```
ClearAll["Global'*"]
NumberOfOperationsAvailable=15*86500;
GuessPrimeRange=2^40;
NumberOfOperationsPerSecond=10^13;
prob=1;
i=Log[2,GuessPrimeRange];
While[prob>0.95,
i=i+1;
SizeOfPrime=2^i;
NumberOfCurvesAvailable=N[(NumberOfOperationsAvailable)/
                                        NumberOfOperationsPerSecond^-1,50];
prob1=N[(1 - 1/(4Sqrt[SizeOfPrime]))^(NumberOfCurvesAvailable), 50];
prob=N[1-prob1,10];
];
i
```