UNIVERSITÄT
BAYREUTH

Der Schoof-Algorithmus

Bachelorarbeit

von

Dominik Köhler

FAKULTÄT FÜR MATHEMATIK, PHYSIK UND INFORMATIK
MATHEMATISCHES INSTITUT

Datum: 16. Mai 2018

Betreuung:
Prof. Dr. M. Dettweiler

Inhaltsverzeichnis

Einleitung

In den letzten Jahrzehnten wurden immer mehr mathematische Methoden entdeckt Daten zu verschlüsseln. Dabei nutzt man oft endliche Gruppen. In den 1980er Jahren entdeckten Victor S. Miller und Neal Koblitz unabhängig voneinander die Möglichkeit, auf elliptischen Kurven eine Addition zu definieren, mit deren Hilfe sich eine (endliche) Gruppe auf den elliptischen Kurven über endlichen Körpern definieren lässt. Dabei hat sich herausgestellt, dass Kryptographie auf elliptischen Kurven deutlich sicherer ist als in den herkömmlichen Verfahren, welche die Restklassenringe $\mathbb{Z}/n$ benutzen. Um geeignete Gruppen für die kryptographische Anwendung zu finden, benötigt man deren Ordnung. Für die Gruppe auf den elliptischen Kurven gab es dafür lange keine Möglichkeit, dies in polynomialer Laufzeit zu bestimmen. Erst 1985 hat *René Schoof* den ersten Algorithmus dazu vorgestellt. Dabei nutzte er die Tatsache aus, dass die Anzahl der Gruppenelemente nach oben beschränkt ist. Mithilfe des chinesischen Restsatzes genügt es somit, die Gruppenordnung modulo einiger teilerfremden Zahlen zu bestimmen, bis eine eindeutige Lösung im Bereich der möglichen Lösungen existiert.
Im Folgenden wird zuerst die Menge der elliptischen Kurven charakterisiert. Durch die Definition einer Addition erhält diese Menge eine Gruppenstruktur. In Kapitel 2 betrachten wir kurz die Sicherheit der Verschlüsselung mit elliptischen Kurven. Nachdem in Kapitel 3 theoretische Grundlagen für den Schoof-Algorithmus besprochen werden, erfolgt in Kapitel 4 die Formulierung des Algorithmus, sowie eine Laufzeitanalyse in Kapitel 5. Im Anhang findet sich eine Implementierung des Algorithmus in dem Programm „Magma“.

Kapitel 1

Einführung in die elliptischen Kurven

Zuerst wollen wir elliptische Kurven in der affinen Ebene betrachten. Danach gehen wir zur *projektiven* Ebene über, um das für die Gruppendefinition notwendige neutrale Element zu finden. Daraus lässt sich dann eine Gruppe bilden, sowie Formeln für die Berechnung der Gruppenoperation, der Addition, finden. Orientierung bietet uns dabei vor allem das Buch „Elliptische Kurven in der Kryptographie“ von Annette Werner, [Wer13].

1.1 Elliptische Kurven in der affinen Ebene

Die affine Ebene ist der zweidimensionale Vektorraum über einem Körper. Darin lassen sich Kurven als Nullstellenmengen von Polynomen in zwei Variablen definieren, siehe [Wer13]:

Definition 1.1.1 (Affine Ebene)
Die *affine Ebene* über einem Körper $\mathbb{F}$ ist der 2-dimensionale Vektorraum $\mathbb{F} \times \mathbb{F}$. Wir schreiben dafür $\mathbb{A}^2$.

Definition 1.1.2 (Affine Kurve)
Eine affine Kurve in der Ebene $\mathbb{A}^2$ ist die Nullstellenmenge eines Polynoms $P[x, y] \in \mathbb{F}[x, y]$ in der affinen Ebene, also die Menge

$$C := \{(x, y) \in \mathbb{A}^2 : C(x, y) = 0\}.$$

Wir schreiben C für die Kurve, die aus dem Polynom $C[x, y]$ hervorgeht.

Falls das Polynom $P[x, y]$ reduzibel ist, also in zwei nicht triviale Faktoren zerlegt werden kann, so zerfällt auch die Kurve P in zwei einzelne Kurven.

Im Folgendem sollen nur irreduzible Kurven, also Kurven über irreduziblen Polynomen, betrachtet werden.
Eine weitere Einschränkung folgt aus der späteren Definition der Addition. Dabei wird die Verdoppelung eines Punktes über die Tangente an diesem Punkt beschrieben. Dazu muss die Tangente allerdings eindeutig definiert sein. Das heißt, die Kurve darf sich nicht schneiden und keine Spitzen besitzen. Betrachtet man die elliptischen Kurven als reelle Funktionen, so steht die Tangente immer senkrecht auf dem Gradienten in dem gegebenen Punkt. Die Tangente ist also für alle Werte des Gradienten eindeutig definiert, außer dieser ist der Nullvektor. Somit ist für elliptische Kurven erforderlich, dass der Gradient in allen Punkten ungleich Null ist.
Daraus ergeben sich die beiden nächsten Definitionen, ebenfalls nach [Wer13]:

Definition 1.1.3 (Singulärer Punkt)
Ein Punkt $P = (x_0, y_0)$ auf einer algebraischen Kurve C ist *singulär*, falls $\frac{\partial C}{\partial x}(x_0, y_0) = \frac{\partial C}{\partial y}(x_0, y_0) = (0, 0)$. Eine *singuläre Kurve* ist eine Kurve mit mindestens einem *singulären* Punkt.

Hiermit kann man den affinen Anteil elliptischer Kurven über die sogenannte (lange) Weierstraß-Gleichung auf $\mathbb{A}^2$ definieren, wie in [SZP03], Def. 1.2:

Definition 1.1.4 (Affine Elliptische Kurven)
Elliptische Kurven in der affinen Ebene sind nicht-singuläre Kurven über den Lösungen der Gleichung

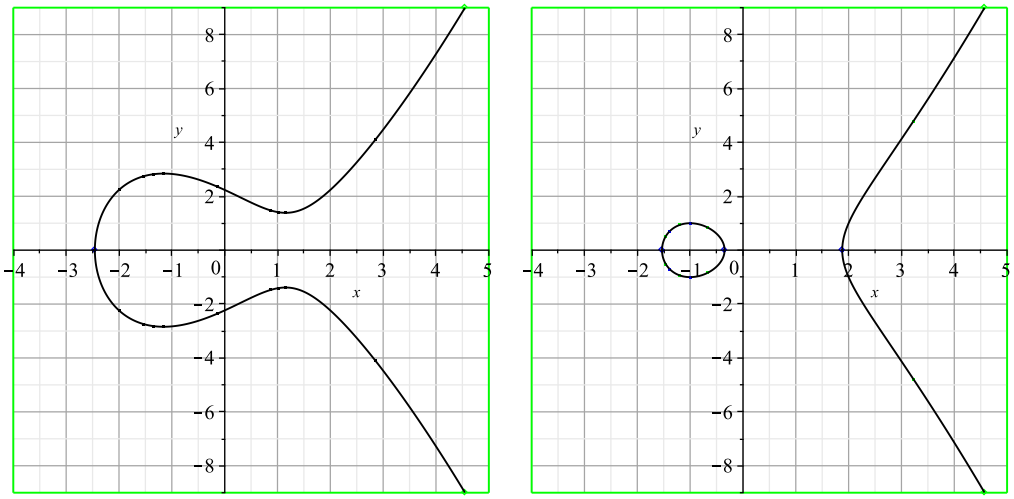
$$y^2 + c_1xy + c_3y = x^3 + c_2x^2 + c_4x + c_6. \tag{1.1}$$

Die obere Gleichung (1.1) heißt auch (lange) *Weierstraß-Gleichung*. Dies beschreibt nur den Anteil der elliptischen Kurven in der affinen Ebene. Die vollständige Definition folgt später.

Wir wollen für unsere Zwecke elliptische Kurven über Körpern mit Charakteristik 2 oder 3 ausschließen. Das führt u.a. zu einer Vereinfachung der (langen) Weierstraß-Gleichung zu der sogenannten kurzen *Weierstraß-Gleichung*, wie auch in [Was08] ab S. 10 beschrieben wird.

Lemma 1.1.5
Falls $\text{char}(\mathbb{F}) \neq 2, 3$, dann findet man Koeffizienten $a, b, c \in \mathbb{F}$, sodass sich die lange Weierstraß-Gleichung zu der kurzen Weierstraßgleichung $y^2 = x^3 + ax + b$ umschreiben lässt.

Beweis. Der Beweis erfolgt durch Nachrechnen und Koordinatentransformationen.



(a) $y^2 = x^3 - 4x + 5$ (b) $y^2 = x^3 - 3x - 1$

Abbildung 1.1.1: Zwei elliptische Kurven in der affinen Ebene

Zuerst betrachten wir den Fall $\text{char}(\mathbb{F}) \neq 2$. Somit lässt sich durch 2 teilen. Dann kann man die Koordinaten so umschreiben, dass man die vereinfachte Gleichung $y^2 = x^3 + b_2x^2 + b_4x + b_6$ erhält. Dafür setzt man:

- $b_2 = c_2 + \frac{c_1^2}{4}$
- $b_4 = c_4 + \frac{c_1c_3}{2}$
- $b_6 = c_6 + \frac{c_3^2}{4}$

Man bemerke, dass diese Koordinatentransformation eine bijektive Umformung ist. Nachrechnen auf der rechten Seite ergibt die Form:

$$\begin{aligned} x^3 + b_2x^2 + b_4x + b_6 &= x^3 + (c_2 + \frac{c_1^2}{4})x^2 + (c_4 + \frac{c_1c_3}{2})x + c_6 + \frac{c_3^2}{4} \\ &= x^3 + c_2x^2 + \frac{c_1^2}{4}x^2 + c_4x + \frac{c_1c_3}{2}x + c_6 + \frac{c_3^2}{4} \\ &= (\frac{c_1}{2}x + \frac{c_3}{2})^2 + x^3 + c_2x^2 + c_4x + c_6 \end{aligned}$$

Mithilfe der ebenfalls bijektiven Variablentransformation $y \mapsto y + \frac{c_1}{2}x + \frac{c_3}{2}$

führt dies auf der rechten Seite zu der Form:

$$\begin{aligned}(y + \frac{c_1}{2}x + \frac{c_3}{2})^2 &= y^2 + 2y(\frac{c_1}{2}x + \frac{c_3}{2}) + (\frac{c_1}{2}x + \frac{c_3}{2})^2 \\ &= y^2 + c_1xy + c_3y + (\frac{c_1}{2}x + \frac{c_3}{2})^2.\end{aligned}$$

Durch die Subtraktion des Terms $(\frac{c_1}{2}x + \frac{c_3}{2})^2$ auf beiden Seiten ergibt die gewünschte Gleichung.
Nun wollen wir noch zusätzlich den Fall $\text{char}(\mathbb{F}) \neq 3$ betrachten. Dabei können wir durch 3 teilen und somit lässt sich die Gleichung auf die Form $y^2 = x^3 + ax + b$ bringen, mit

- $a = b_4 - \frac{b_2^2}{3}$
- $b = b_6 - a\frac{b_2}{3} - \frac{b_2^3}{27} = b_6 - \frac{b_2b_4}{3} + \frac{2b_2^3}{27}$

Dabei führt man folgende Variablentransformation durch:

$$x \mapsto x + \frac{b_2}{3} \Rightarrow$$

$$\begin{aligned}x^3 + ax + b &= (x + \frac{b_2}{3})^3 + a(x + \frac{b_2}{3}) + b \\ &= x^3 + b_2x^2 + \frac{b_2^2}{3}x + \frac{b_2^3}{27} + ax + a\frac{b_2}{3} + b \\ &= x^3 + b_2x^2 + (\frac{b_2^2}{3} + b_4 - \frac{b_2^2}{3})x + \frac{b_2^3}{27} + a\frac{b_2}{3} + b_6 - a\frac{b_2}{3} - \frac{b_2^3}{27} \\ &= x^3 + b_2x^2 + b_4x + b_6x\end{aligned}$$

□

Bemerkung 1.1.6
Man erkennt an der kurzen Weierstraß-Gleichung, dass die Kurve achsensymmetrisch zur x-Achse ist, da für jede Lösung (x, y) auch $(x, -y)$ die Gleichung erfüllt.

Im Folgenden werden wir nur elliptische Kurven betrachten, die mit der kurzen Weierstraß-Gleichung beschrieben werden können. Des Weiteren definieren wir nur elliptische Kurven über endlichen Körpern $\mathbb{F}_q$. Allgemein können elliptische Kurven aber über beliebigen Körpern betrachtet werden.
Um algorithmisch zu bestimmen, ob eine Kurve singulär ist, hilft das nachfolgende Lemma. Es findet sich mit einem Beweis über die *Diskriminante* auch auf S. 30 f. in [Wer13].

Lemma 1.1.7
Eine Kurve, beschrieben mit der (kurzen) Weierstraß-Gleichung über einem Körper $\mathbb{F}_q$ mit $\text{char}(\mathbb{F}_q) \neq 2, 3$, ist genau dann nicht-singulär, wenn die rechte Seite drei paarweise verschiedene Nullstellen im algebraischen Abschluss von $\mathbb{F}_q$, $\overline{\mathbb{F}}_q$ besitzt.

Beweis. Ein Punkt (x_0, y_0) auf der elliptischen Kurve ist genau dann singulär, wenn $\frac{\partial f}{\partial x}(x_0, y_0) = \frac{\partial f}{\partial y}(x_0, y_0) = (0, 0)$ erfüllt ist. Die Kurve ist gegeben durch die kurze Weierstraß-Gleichung $f(x, y) = x^3 + ax + b - y^2$ mit der partiellen Ableitung in y: $\frac{\partial f}{\partial y}(x_0, y_0) = -2y_0$. Diese ist also nur gleich 0, falls $y_0 = 0$. Damit sieht die Kurve in dem Punkt $(x_0, 0)$ so aus: $0 = x_0^3 + ax_0 + b$. Somit ist der Punkt singulär, falls die partielle Ableitung nach x eine Nullstelle auf der Kurve besitzt, also falls ein x_0 existiert, sodass $f(x_0, 0) = 0$ und $\frac{\partial f}{\partial x}(x_0, 0) = 0$. Dies ist genau dann der Fall, falls x_0 eine doppelte Nullstelle von $f(x) = x^3 + ax + b$ ist, denn dann lässt sich die Gleichung umschreiben auf $f(x) = a'(x - x_0)^2(x - x_1)$. Die Ableitung ist dann

$$\begin{aligned} f'(x) &= 2a'(x - x_0)(x - x_1) + a'(x - x_0)^2 \\ &= a'(x - x_0)(2(x - x_1) + (x - x_0)). \end{aligned}$$

□

Somit kann man mithilfe der *Diskriminante* überprüfen, ob eine Kurve singulär ist, siehe [Wer13], S. 31. Eine andere Idee um zu überprüfen, ob eine Nullstelle doppelt ist, folgt aus der Beobachtung, dass für eine doppelte Nullstelle x_0 von $f(x)$ gilt, dass $f(x)$ und die Ableitung $f'(x)$ diese Nullstelle besitzen und damit auch den gemeinsamen Teiler $(x - x_0)$ haben. Mithilfe der *Resultante* lässt sich berechnen, ob zwei Polynome teilerfremd sind. Wendet man diese also auf $f(x)$ und $f'(x)$ an, erhält man die Aussage, ob die Kurve singulär ist. Dieser Ansatz wird zum Beispiel in [Ann14] verfolgt.

1.2 Elliptische Kurven in der projektiven Ebene

Um auf elliptischen Kurven eine Gruppe zu definieren, benötigt man einen eindeutigen Punkt auf dieser Kurve, der als neutrales Element dienen kann. Diesen wollen wir in der *projektiven Ebene* suchen. Dort wird die geometrische Idee verwirklicht, dass sich alle Geraden der Ebene paarweise in genau einem Punkt schneiden. Ebenfalls soll die affine Ebene in die projektive Ebene eingebettet sein, sodass die projektive Ebene eine Erweiterung der affinen

Ebene um die Schnittpunkte paralleler Geraden ist. Diese Punkte nennt man die Punkte im Unendlichen. Vergleiche hierzu den Abschnitt 2.3 aus [Wer13].

Definition 1.2.1 (Projektive Ebene)
Die *projektive Ebene* über einem Körper $\mathbb{F}_q$, geschrieben $\mathbb{P}^2$, ist die Menge

$$\{(x,y,z) \in (\overline{\mathbb{F}}_q^3)\}\backslash\{(0,0,0)\}_{/\sim},$$

mit der Äquivalenzrelation $\sim$, definiert durch:

$$a \sim b \Leftrightarrow \exists \lambda \in \mathbb{F}_q : a = b\lambda, \forall a, b \in \mathbb{F}_q^3.$$

Die projektive Ebene ist somit die Menge aller Ursprungsgeraden im $\mathbb{F}_q^3$. Der Nullpunkt bildet keine Ursprungsgerade, wird also ausgenommen. Zwei Vektoren gelten als äquivalent, wenn diese linear abhängig sind, also in dieselbe Richtung zeigen. Somit sind alle Ursprungsgeraden der Form $0 = ax + by + cz$ mit $z \neq 0$ eindeutig definiert über ihren Schnittpunkt mit der Ebene $(x, y, 1)$. Die affine Ebene kann also in diese Ebene eingebettet werden. Betrachtet man nun zwei parallele Geraden in der eingebetteten affinen Ebene, so erkennt man den Schnittpunkt der beiden Geraden in der projektiven Ebene:

$$\begin{aligned} g_1 &: 0 = ax + by + cz \\ g_2 &: 0 = ax + by + dz \\ g_1 \cap g_2 &= \{(x,y,z) \in g_1 : ax + bz - y = ax + cz - y\} \\ &= \{(x,y,z) : z = 0, x = b, y = -a\} = \{(a,b,0)\}. \end{aligned}$$

Dabei beschreibt die Gerade $p \in \mathbb{P}^2 : p = (x, y, 0), (x, y) \in \mathbb{F}_q^2$ die unendliche Gerade, auf der die unendlich fernen Punkte liegen.
Folglich besitzt die (kurze) Weierstraß-Gleichung in der projektiven Ebene die Form, siehe [Wer13], S. 25:

$$Y^2Z = X^3 + aX^2Z + bXZ^2 + cZ^3$$

Bemerkung 1.2.2
Die projektive Ebene ist somit eine Erweiterung der affinen Ebene, auch genannt der *projektive Abschluss* der affinen Ebene. Dabei gelten also auch dieselben Axiome, insbesondere, dass jede Gerade eindeutig durch zwei Punkte beschrieben werden kann.

Somit ergibt sich die Definition für elliptische Kurven für einen Körper mit Charakteristik größer als 3:

Definition 1.2.3 (Elliptische Kurven)
Elliptische Kurven über einem Körper mit $\text{char}(\mathbb{F}_q) \neq 2, 3$ sind definiert als die Nullstellenmenge der kurzen Weierstraß-Gleichung

$$y^2z = x^3 + ax^2z + bxz^2 + cz^3$$

in $\mathbb{P}^2$, wobei zusätzlich gilt, dass die Gleichung $0 = x^3 + ax + b$ drei verschiedene Lösungen in $\overline{\mathbb{F}}_q$ besitzt.

Jetzt folgt noch eine wichtige Beobachtung, mit der wir eine Gruppe auf elliptischen Kurven definieren können:

Lemma 1.2.4
Jede elliptische Kurve schneidet die unendliche Gerade $(x, y, 0)$ eindeutig im Punkt $(0, 1, 0)$. Dieser Punkt ist gleichzeitig der Schnittpunkt aller Parallelen zur y-Achse. Wir bezeichnen den Punkt im Folgenden als $\mathcal{O}$.

Beweis. Setze $z = 0$, dann folgt: $0 = x^3 \Leftrightarrow x = 0$. Zusätzlich folgt mithilfe von $(x, y, z) \in \mathbb{P}^2 \Leftrightarrow (x, y, z) \neq (0, 0, 0)$, dass $y \neq 0$. Da alle Punkte der Form $(0, y, 0) \in \mathbb{P}^2$ äquivalent sind, identifizieren wir ihn hier mit $(0, 1, 0)$.
Dieser Punkt ist der Schnittpunkt aller Parallelen zur y-Achse:
Allgemein ist eine Parallele zur y-Achse in der affinen Ebene gegeben durch die Gleichung

$$x = a, a \in \mathbb{F}_q.$$

In der projektiven Ebene wäre dies dann

$$x = az.$$

Betrachtet man zwei verschiedene Parallele der y-Achse, also einmal $x = az$ und dann $x = bz$, mit $b \neq a, b \in \mathbb{F}_q$, dann schneiden diese sich auf der unendlichen Gerade $(x, y, 0)$ und der Schnittpunkt berechnet sich durch

$$z = 0 \Rightarrow x = 0.$$

Die Lösungsmenge für den Schnittpunkt bildet also die y-Achse. Diese lässt sich, wie oben, wieder mit dem Punkt $(0, 1, 0)$ identifizieren. Folglich schneiden sich alle Parallelen der y-Achse im Punkt $(0, 1, 0)$.

□

Bemerkung 1.2.5
Der Punkt $\mathcal{O}$ ist nur für theoretische Zwecke interessant, um eine Gruppe zu definieren. In praktischen Anwendungen elliptischer Kurven in der Kryptografie wird dieser nicht benötigt, die Anwendungen finden in affinen Ebenen statt.

Damit kann man jetzt eine Addition auf elliptischen Kurven definieren: Eine Gerade durch zwei verschiedene Punkte, sowie die Tangente an einem Punkt auf der elliptischen Kurve erhält noch genau einen weiteren Schnittpunkt mit der elliptischen Kurve. Spiegelt man diesen an der x-Achse, so erhält man die Summe der beiden Punkte. Der Punkt im Unendlichen bleibt in diesem Fall gleich. Addiert man folglich drei Punkte, die auf einer Geraden liegen, so erhält man den Punkt $\mathcal{O}$.

1.3 Gruppenstruktur auf elliptischen Kurven

Es soll gezeigt werden, dass die obigen Behauptungen stimmen. Diese sind in der folgenden Definition noch einmal zusammengefasst, der Satz danach zeigt dann die Existenz und gibt eine Formel für die Addition an. Ebenfalls wird gezeigt, dass die Definition eindeutig ist. Man vergleiche hierzu [Ann14], Kapitel 2.4.

Definition 1.3.1 (Addition auf elliptischen Kurven)
Seien $P = (x_P, y_P), Q = (x_Q, y_Q)$ affine Punkte einer elliptischen Kurve, dann definiert sich die Addition von Punkten auf der elliptischen Kurve $E(\mathbb{F}_q)$ wie folgt. Man unterscheiden vier Fälle:

1. $x_P \neq x_Q, y_P \neq y_Q$: Dann trifft die Gerade durch P und Q einen weiteren Punkt $R' = (x_R, y_R)$ in der affinen Ebene. Somit definiert man $P + Q := -R' = (x_R, -y_R)$.

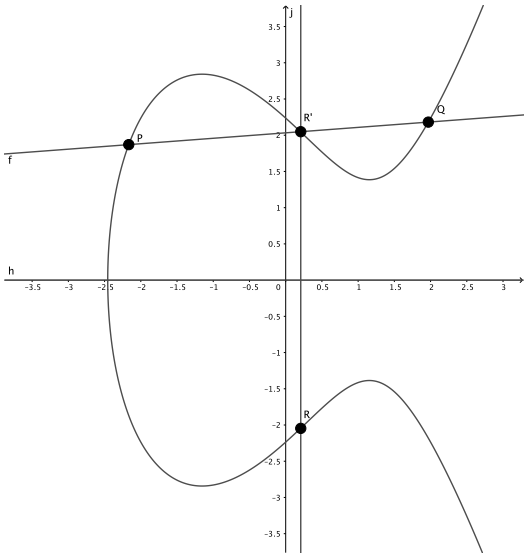
2. $x_P = x_Q, y_P = y_Q \neq 0$: Hier addiert man den Punkt P zu sich selber. Dafür nimmt man die Tangente an P. Diese trifft einen weiteren Punkt R' in der affinen Ebene, mit dem sich die Addition wie oben definiert: $P + Q := -R' = (x_R, -y_R)$.

3. $x_P = x_Q, y_P \neq y_Q$: Dann ist die Gerade durch P und Q parallel zur y-Achse, folglich schneidet diese die elliptische Kurve im Punkt $\mathcal{O}$. Die Addition ist dann ebenfalls definiert als $P + Q := -\mathcal{O} = \mathcal{O}$. Somit gilt $P = -Q$.

4. $x_P = x_Q, y_P = y_Q = 0$: Hier wird der Punkt P, wie in Fall 1, wieder zu sich selbst addiert, diesmal ist die Tangente allerdings eine Parallele der y-Achse. Somit gilt für die Addition: $P + Q := \mathcal{O}$.

Falls $P = \mathcal{O}$, $Q \in E(\mathbb{F}_q)$, dann gilt $P + Q = Q + P := Q$.

Abbildung 1.3.1: Das Bild veranschaulicht die Definition der Addition 1.3.6:



Satz 1.3.2
Sei $E(\mathbb{F}_q)$ eine elliptische Kurve, beschrieben durch $y^2 = F(x) = x^3 + ax + b, a, b \in \mathbb{F}_q$. Seien $P = (x_P, y_P), Q = (x_Q, y_Q) \in E(\mathbb{F}_q)$ und $R = (x_R, y_R) := P + Q$. Dann ist $R \in E(\mathbb{F}_q)$ eindeutig definiert und über folgende Formel gegeben. Dabei unterscheidet man wieder die Fälle aus der obigen Definition. Wir bezeichnen dabei die Gerade durch P und Q als L.

1. $P + Q = R = (x_R, y_R)$ ist gegeben durch:

$$\begin{aligned} x_R &= \mu^2 - x_P - x_Q, \\ y_R &= y_P + \mu(x_R - x_P), \\ \text{mit } \mu &= \frac{y_B - y_A}{x_B - x_A}. \end{aligned}$$

2. $P + Q = P + P = R = (x_R, y_R)$ ist gegeben durch:

$$\begin{aligned} x_R &= \mu^2 - x_P - x_Q \\ y_R &= -y_P - \mu(x_R - x_P), \\ \text{mit } \mu &= \frac{3x_P^2 + a}{2y_P}. \end{aligned}$$

3. $P + Q = Q + P = \mathcal{O}$: Dies ist der eindeutige weitere Schnittpunkt von $E(\mathbb{F}_q)$ und L.

4. $P + Q = P + P = \mathcal{O}$: Dies ist der eindeutige weitere Schnittpunkt von $E(\mathbb{F}_q)$ und L.

Falls $P = \mathcal{O}$, dann gilt für $Q \in E(\mathbb{F}_q)$: $P + Q = Q + P = Q$.

Beweis. Die ersten beiden Fälle können wir zu einem Fall zusammenfassen. Es ist lediglich erforderlich, die unterschiedlichen Werte für μ herzuleiten, die auch zu verschiedenen Geraden $L = \mu x + \tau$ führen.

1. Hier stellt man eine Gerade durch die Punkte P, Q auf. Diese ist durch die 2-Punkte-Form gegeben durch $y = \frac{y_B - y_A}{x_B - x_A}(x - x_P) + y_P$. Mit dem vorher definierten μ und mit $\tau = y_P - \mu x_P$ folgt: $y = mx + \tau$.

2. Falls die beiden Punkte gleich sind, stellt man die Tangente an diesem Punkt zur elliptischen Kurve auf. Die Tangente steht senkrecht auf dem Gradienten $\left(3x_P^2 + a, \ \ -2y_P\right)$. Die Senkrechte darauf ist folglich $\begin{pmatrix} 2y_P \\ 3x_P^2 + a \end{pmatrix}$. In Koordinatenform ergibt dies die Steigung $\mu = \frac{3x_P^2 + a}{2y_P}$. Der y-Abschnitt berechnet sich durch $\tau = y_P - x_P$. Somit ergibt sich wieder die Gleichung $y = \mu x + \tau$, wie in (1.), nur mit anderen Werten für μ und τ.

In beiden Fällen lässt sich aber ein weiterer Schnittpunkt von L und $E(\mathbb{F}_q)$ berechnen:
Setzt man die Gerade in die Gleichung für $E(\mathbb{F}_q)$ ein, erhält man

$$(\mu x + \tau)^2 = x^3 + ax + b.$$

Umgeformt in eine Gleichung dritten Grades ergibt dies

$$x^3 - \mu^2 x^2 + (a - 2\mu\tau)x + (b - c^2) = 0,$$

was in die Form

$$\begin{aligned} 0 &= (x - x_1)(x - x_2)(x - x_3) \\ &= x^3 + (-x_1 - x_2 - x_3)x^2 + (x_1x_2 + x_2x_3 + x_1x_3)x - x_1x_2x_3 \end{aligned}$$

mit den Nullstellen x_1, x_2, x_3 umgeschrieben werden kann. Allerdings sind die beiden Nullstellen x_P, x_Q, bzw. im Fall der Verdoppelung des Punktes P die doppelte Nullstelle bei x_P, bereits bekannt. Gemäß des Fundamentalsatzes der Algebra existiert also noch höchstens eine Lösung dieser Gleichung. Ebenfalls sieht man durch einen Vergleich der Terme zweiten Grades, dass μ durch $\mu^2 = x_1 + x_2 + x_3$ gegeben ist. Bei zwei bekannten Lösungen ist die dritte Lösung folglich gegeben durch $x_R = x_3 = \mu^2 - x_P - x_Q$. Nachdem

dies alles Berechnungen im Körper $\mathbb{F}_q$ sind, liegt diese Koordinate auch in $\mathbb{F}_q$. Folglich ist der Punkt in $E(\mathbb{F}_q)$ und insgesamt eindeutig definiert.
In den letzten beiden Fällen lässt sich die Existenz in einer gemeinsamen Rechnung zeigen:

3. In diesem Fall verläuft die Gerade L parallel zur y-Achse. Diese schneidet also den Punkt $\mathcal{O}$ und die elliptische Kurve. Zu zeigen ist nur, dass die Gerade keinen weiteren Punkt der elliptischen Kurve schneidet. Dies folgt aus der unten stehenden Gleichung (1.2). Diese besitzt höchstens zwei verschiedene Nullstellen für ein festes x.

4. Falls $P = Q, y_P = 0$, dann ist die Tangente an diesem Punkt ebenfalls parallel zur y-Achse. Dies sieht man durch eine Rechnung, denn der Gradient an diesem Punkt ist

$$\begin{pmatrix} 3x_P^2 + a & -2y_P \end{pmatrix} = \begin{pmatrix} 3x_P^2 + a \\ 0 \end{pmatrix}.$$

   Folglich hat die Tangente die Form $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ und ist somit eine Parallele zur y-Achse und schneidet als dritten Punkt noch $\mathcal{O}$. Die Gerade trifft auf keinen weiteren Punkt der elliptischen Kurve, denn betrachtet man den Schnittpunkt der Parallelen zur $y-$Achse, $x = x_p$, mit der elliptischen Kurve, so erhält man die Gleichung

$$y^2 = x_P^3 + ax_P + b. \tag{1.2}$$

   Diese kann mal als Funktion in y schreiben. Somit lässt sich die Gleichung umformen zu

$$\begin{aligned} 0 &= y^2 - x_P^3 - ax_P - b \\ &= (y - \sqrt{x_P^3 - ax_P - b})(y + \sqrt{x_P^3 - ax_P - b}). \end{aligned}$$

   Der Term unter der Wurzel ist allerdings 0, da der Punkt $(x_P, 0)$ auf der elliptischen Kurve liegt. Somit ist der einzige weitere Schnittpunkt der Punkt $\mathcal{O}$.

Die Eindeutigkeit der letzten beiden Fälle ergibt sich somit durch Einsetzen in die Gleichung der elliptischen Kurve. Parallelen der y-Achse haben die Form $x = x_P$, folglich erhalten wir die Form

$$y^2 = x_P^3 + ax_P + b.$$

Diese Gleichung hat folglich genau zwei Lösungen für $x_P^3 + ax_P + b \neq 0$, ansonsten eine. Diese sind die bekannten Punkte auf der Kurve $\pm y_P$. Somit ist die Eindeutigkeit auch erfüllt.
Falls $P = \mathcal{O}$, $Q \neq \mathcal{O}$, dann ist die Addition dieser Punkte $-Q$. □

Bemerkung 1.3.3
Dies ist auch ein Spezialfall des Satzes von Bezout, welcher besagt, dass sich zwei glatte Kurven vom Grad n und m in der projektiven Ebene $\mathbb{P}^2$ über einen algebraisch abgeschlossenen Körper $\mathbb{K}$ genau nm mal, gerechnet der Vielfachheiten, schneiden. Vergleiche hierzu auch den Abschnitt 5.3 in [Ful08].
Jede Gerade schneidet also jede elliptische Kurve genau drei mal, gerechnet der Vielfachheiten. Die Tangente an einen Punkt schneidet diesen mit der Vielfachheit zwei.

Mit dem folgenden Satz lässt sich mithilfe dieser Addition eine Gruppe auf elliptischen Kurven beschreiben.

Satz 1.3.4
Eine elliptische Kurve E über einen endlichen Körper $\mathbb{F}_q$ bildet zusammen mit der oben beschriebenen Addition und dem neutralen Element $\mathcal{O}$ eine abelsche Gruppe: $(E, +)$.
Es gelten also folgende Regeln:

1. Die Gruppe ist kommutativ: $\forall a, b \in (E, +) : a + b = b + a$

2. $\mathcal{O}$ ist das neutrale Element: $\forall a \in (E, +) : a + \mathcal{O} = \mathcal{O} + a = a$

3. Alle Elemente besitzen ein inverses Element: $\forall a \in (E, +) \exists \tilde{a} \in (E, +) : a + \tilde{a} = \tilde{a} + a = e$

4. Assoziativität: $\forall a, b, c \in (E, +) : (a + b) + c = a + (b + c)$

Beweis. Die ersten drei Punkte folgen direkt aus der Definition. Der Beweis für die Assoziativität ist allerdings komplizierter.

Wir beweisen den Satz mithilfe dem *Satz von Cayley-Bacharach*, wie auch in [Sto09], Lemma 9.3, beschrieben. Für den Beweis definieren wir zuerst

sechs projektive Geraden, und bilden den Schnitt mit der elliptischen Kurve:

$$\begin{aligned} L_1 \cap E &= \{P, Q, S'\} \\ M_1 \cap E &= \{\mathcal{O}, S, S'\} \\ L_2 \cap E &= \{\mathcal{O}, U, U'\} \\ M_2 \cap E &= \{Q, R, U'\} \\ L_3 \cap E &= \{S, R, T'\} \\ M_3 \cap E &= \{P, U, T''\} \end{aligned}$$

Damit erkennt man, dass $P + Q = S$ gilt, sowie $Q + R = U$. Wir wollen zeigen, dass $(P + Q) + R = P + (Q + R)$, also $S + R = P + U$ gilt. Da die Gleichungen $S' = -S$ und $U' = -U$ erfüllt sind, genügt es zu zeigen, dass sich die Gerade L_3 durch S und R sich mit der Geraden M_3 durch P und U in einem gemeinsamen Punkt T' auf E schneiden. Dazu betrachten wir den unten stehenden Satz 1.3.6. Für den Beweis wird noch folgendes Lemma benötigt, ein weiterer Spezialfall des *Satzes von Bezout*:

Lemma 1.3.5
Sei C eine glatte Kurve, gegeben durch das Polynom $F(x, y, z)$, mit Dimension $d \in \{2, 3\}$ in der projektiven Ebene $\mathbb{P}^2$, und G eine Gerade, die mindestens einen Punkt besitzt, der nicht auf C liegt. Dann haben C und G höchstens d gemeinsame Schnittpunkte.

Beweis. Wir führen den Beweis analog zu [Fis94], Satz 2.5. Durch Koordinatentransformation lässt sich erreichen, dass die Gerade G die Form $z = 0$ besitzt. Somit sind die Schnittpunkte der Geraden mit der Kurve C gegeben durch ein Polynom $g(x, y) = F(x, y, 0)$. Damit lassen sich $A_1(x, y), \ldots, A_{d-1}(x, y)$ finden, sodass sich F schreiben lässt als:

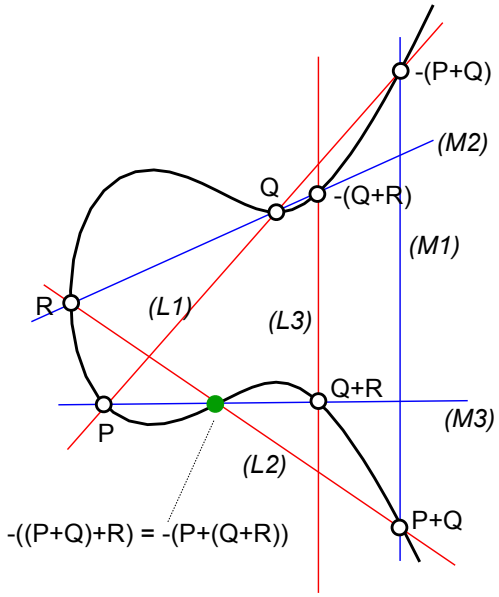
$$F(x, y, z) = A_0 z^d + A_1 z^{d-1} + \ldots + A_{d-1} z + g.$$

Gilt nun $g = 0$, so ist die gesamte Gerade G in C, denn dann lässt sich $F(x, y, z)$ durch z teilen. Ist $g \neq 0$, so existiert nach der homogenen Form des *Fundamentalsatzes der Algebra* eine Zerlegung

$$g = (b_1 x - a_1 y)^{k_1} \cdot \ldots \cdot (b_m x + a_m y)^{k_m},$$

mit $\sum_{k=1}^{m} = d$ und eindeutig bestimmten Punkten (a_i, b_i), mit $1 \leq i \leq m$. Somit ist die Anzahl der Schnittpunkte einer Kurve von Grad d mit einer Geraden höchstens d. Falls es mehr Schnittpunkte gibt, so liegt die Gerade auf der Kurve. □

Abbildung 1.3.2: Das Bild veranschaulicht den Beweis von Satz 1.3.6:



Satz 1.3.6
Seien G_i, G'_j, $1 \leq i, j \leq 3$ paarweise verschiedene Geraden in der projektiven Ebene $\mathbb{P}^2$. Seien P_{ij} die paarweise verschiedenen Schnittpunkte von G_i und G'_j mit einer Kurve C vom Grad 3. Sei weiterhin C' eine ebene projektive Kurve von Grad 3, die nur die acht Punkte $P_{ij}, (i,j) \neq (3,3)$ enthält. Dann liegt auch der neunte Punkt P_{33} auf C'.

Beweis. Wir zeigen zuerst, dass sich auf den Punkten $P_{ij}, (i,j) \neq (3,3)$ ein Vektorraum aus Polynomen der Dimension 2 definieren lässt.

Lemma 1.3.7
Sei V gegeben durch die Punkte P_{ij}:

$$V = \{F \in \mathbb{R}[x,y,z]_3 | F(P_{ij}) = 0, \forall\, 1 \leq i,j \leq 3, (i,j) \neq (3,3)\}.$$

Dann ist V ein Vektorraum der Dimension 2.

Beweis. Eine allgemeine algebraische Kurve der Dimension 3 in der projektiven Ebene $\mathbb{P}$ ist gegeben durch 10 Koeffizienten:

$$\begin{aligned} D(x,y,z) =& a_1x^3 + a_2x^2y + a_3x^2z + a_4xy^2 + a_5xyz \\ &+ a_6xz^2 + a_7y^3 + a_8y^2z + a_9yz^2 + a_{10}z^3. \end{aligned}$$

Die Bedingungen, dass D für die Punkte $P_{ij}, (i,j) \neq (3,3)$ verschwinden muss, führt zu einem linearen Gleichungssystem für die Koeffizienten. Diese sind abhängig von mindestens zwei frei wählbaren Koeffizienten. Somit hat der Vektorraum V mindestens die Dimension 2. Wir wollen nun zeigen, dass die Dimension genau 2 ist:
Nehmen wir an, die Dimension wäre mindestens 3. Das heißt, dass es drei Polynome in V gibt, die linear unabhängig sind. Alle Polynome in V bestehen aus höchstens zehn Monomen und alle haben 8 gemeinsame Nullstellen. Finden wir also ein drittes Polynom, das zu zwei bereits vorhandenen Polynomen linear unabhängig ist, so hat dieses mindestens eine weitere Nullstelle, die die ersten beiden jeweils nicht haben. Ebenfalls existiert eine Nullstelle, die auch eine Nullstelle eines der anderen beiden Polynomen sein könnte.

Somit können wir zwei weitere Punkte wählen, die auf D liegen sollen. Wählen wir P auf der Geraden G_1, sodass dieser auf keinem Schnittpunkt mit anderen Geraden liegt, und einen Punkt Q, der auf keiner Geraden liegt. Wir nehmen dabei an, dass der Körper groß genug ist, um diese Punkte enthalten zu können. Betrachten wir nun die Gerade G_1, gegeben durch $L_1(x,y,z) = 0$. Da G_1 und D in vier Punkten übereinstimmen, muss nach Lemma 1.3.5, G_1 komplett in D liegen, das heißt L_1 muss F teilen. Schreiben wir also $F = LF'$, mit F' einem Polynom vom Grad 2. Die von F' definierte Kurve schneidet wiederum die Gerade G_2 in den drei Punkten P_{21}, P_{22}, P_{23}. Somit muss die Gerade $G_2 = \{(x,y,z) \in \mathbb{F}_q \mid L_1(x,y,z) = 0\}$ wieder komplett in D liegen, und $L_2 \mid F'$. Somit gilt $F' = L_2F''$, wobei F'' eine Gerade beschreibt. Diese hat mit der Geraden G_3 die beiden Punkte P_{31} und P_{32} gemeinsam, die Geraden stimmen also überein. Damit ist die Kurve D aber durch die Geraden G_1, G_2, G_3 gegeben, also muss der Punkt Q auch auf einer dieser Geraden liegen. Allerdings konnten wir diesen so wählen, dass dies nicht gilt, wodurch sich ein Widerspruch ergibt.

□

Damit ist die Dimension des Vektorraums 2 und wir können das Lemma 1.3.5 beweisen:
Da C in V liegt, existieren Polynome $G, H \in \mathbb{R}[x,y,z]_3$, die sich in genau den neun Punkten P_{ij} schneiden und linear unabhängig sind. Diese bilden eine Basis von V. Sei die Kurve C' durch das Polynom F' beschrieben. Somit lässt sich C' als Linearkombination von G und H schreiben und damit verifizieren, dass $P_{33} \in C'$ gilt.

$$\begin{gathered} F' = aG + bH, \text{mit Konstanten } a, b, \\ \Rightarrow F(P_{33}) = a \cdot G(P_{33}) + b \cdot H(P_{33}) = a \cdot 0 + b \cdot 0 = 0. \end{gathered}$$

Damit erhält man das Ergebnis, dass der Punkt P_{33} ebenfalls auf C' liegt.

□

Daraus folgt nun der Beweis des eigentlichen Satzes 1.3.4. Definiert man nun $C = (L_1 \cup L_2 \cup L_3) \cap E$ und $C' = (M_1 \cup M_2 \cup M_3) \cap E$, so erhält man, dass C und C' sich genau durch die Punkte T', T'' unterscheiden, die anderen acht Punkte aber gleich sind. Nach dem obigen Satz gilt also, dass diese beiden Kurven gleich sind, folglich auch $T' = T''$ und unsere Behauptung für die Assoziativität der Addition auf elliptischen Kurven ist bewiesen.

□

Bemerkung 1.3.8
Es ist möglich, die Assoziativität auch anderweitig zu beweisen. In [Fri17] wird dies direkt nachgerechnet. In [Sil13], Thm. III.3.4e, wird beschrieben, wie man dies mithilfe des sogenannten *Satzes von Riemann-Roch* beweisen kann.

Somit haben wir eine kommutative Gruppe auf elliptischen Kurven. Im folgenden Kapitel sollen Anwendungen gezeigt werden, um die Notwendigkeit deutlich zu machen, die Anzahl der Punkte auf den elliptischen Kurven, also die Ordnung der Gruppe, zu kennen.

Kapitel 2

Verschlüsselung auf elliptischen Kurven

In diesem Kapitel soll kurz beschrieben werden, wie man elliptische Kurven zur Verschlüsselung von Daten nutzen kann. Grundlegend dazu ist das Konzept der *Public-Key-Kryptographie*.
Hier betrachten wir Möglichkeiten, Zahlen zu verschlüsseln, sodass nur berechtigte Personen diese ohne viel Aufwand wieder entschlüsseln können. Eine spezielle Zahl, der sogenannte *private Schlüssel*, ermöglicht die schnelle Entschlüsselung. Ist man nicht im Besitz des Schlüssels, ist es sehr rechenaufwendig die ursprüngliche Zahl zu finden. Funktionen, die diese Eigenschaft erfüllen, heißen *Einweg*-Funktionen. Verschlüsselt wird mit dem sogenannten *öffentlichen Schlüssel*. Um zu sehen, wie dies funktioniert, betrachten wir endliche Gruppen. Wie wir gesehen haben, kann man auf elliptischen Kurven eine Gruppe definieren. Um diese Gruppe für Verschlüsselungen von ganzen Zahlen zu nutzen, betrachten wir also Gruppen auf elliptischen Kurven, die isomorph zu endlichen Untergruppen von $\mathbb{Z}$ sind.
Viele Verfahren lassen sich dadurch auf elliptische Kurven übertragen. Wir werden im Folgenden ein solches Verfahren, das sogenannte *ElGamal-Verfahren*, genauer betrachten und bemerken, wieso dies mit elliptischen Kurven besonders sicher ist. Zuerst beschreiben wir dazu das sogenannte *diskrete Logarithmus Problem*, auf dem die maximale Sicherheit dieser Verfahren beruht. Dabei sei angemerkt, dass sich der private Schlüssel mit viel Aufwand berechnen lässt. Die Sicherheit beruht also auf der Annahme, dass es den Aufwand nicht wert wäre, diese Berechnungen durchzuführen. Als Quelle für dieses Kapitel dient das Buch „Kryptographie: Grundlagen, Algorithmen, Protokolle“ von Dietman Wätjen [Wät08]. Der Abschnitt über Public-Key-Kryptographie, 2.1, folgt aus dem Kapitel 5.2 in [Wät08], die anderen beiden Abschnitte, 2.2 und 2.3, aus dem siebten Kapitel in [Wät08].

2.1 Die Public-Key-Kryptographie

Bei der Public-Key-Kryptographie ist die Art der Verschlüsselung jedem bekannt. Mithilfe des jedem zugänglichen *öffentlichen Schlüssels* ist es möglich, Daten für den passenden *privater Schlüssel* zu verschlüsseln. Mit dem *privaten Schlüssel* lässt sich die Nachricht wieder entschlüsseln. Wie der Name vermuten lässt, ist dieser nur legitimierten Personen bekannt.

2.2 Das diskrete Logarithmus Problem

Die Sicherheit vieler Verfahren beruht darauf, dass es sehr schwer ist die Berechnungen umzukehren. Multipliziert man in einer multiplikativen Gruppe a-mal das Element P mit sich selbst, erhält man das Element P^a. Versucht man aus diesem Element und dem Wert P wieder die Anzahl der Multiplikationen a zu berechnen, entspricht dies dem diskreten Logarithmus $\log_P(P^a)$, genannt *diskreter Logarithmus*, da nur ganze Zahlen als Lösung in Frage kommen. Übertragen auf elliptische Kurven berechnet man mit dem diskreten Logarithmus die Anzahl der Additionen eines Punktes mit sich selbst, also den Wert a aus der Gleichung $Q = a \cdot P$, mit bekannten Punkten Q und P. Auf elliptischen Kurven ist die Addition aufwendiger als in den ganzen Zahlen, dementsprechend ist der diskrete Logarithmus dort noch schwieriger zu lösen. Es existieren für den diskreten Logarithmus auf endlichen Gruppen keine bekannten Algorithmen, die diesen in polynomialer Laufzeit lösen können. Auf der anderen Seite existieren Algorithmen die das Potenzieren und Multiplizieren auf elliptischen Kurven in polynomialer Laufzeit schaffen. Somit sind Funktionen mit dieser Eigenschaft *Einweg*-Funktionen.
Es wird im Allgemeinen angenommen, dass kein polynomialer Algorithmus für das Lösen des diskreten Logarithmus existiert, auch wenn dies bis jetzt nicht bewiesen werden konnte. Diese Fragestellung nennt man das *diskrete Logarithmus Problem* (DLP).

2.3 Beispiel: ElGamal-Verfahren

Das von ElGamal entwickelte Verfahren benutzt endliche zyklische Gruppen zur Verschlüsselung. Die Sicherheit dieser Methode beruht auf dem diskreten Logarithmus Problem. Wir wenden dies nun auf elliptische Kurven an:
Die Berechnungen werden auf der elliptischen Kurve $E(\mathbb{F}_q)$ durchgeführt. Die Gruppenordnung soll eine Primzahl p sein. Somit ist die Gruppe zyklisch.

1. Schlüsselerzeugung:

(a) Wähle $a \in \{1, \ldots, p-1\}$, $\text{ggt}(a, p) = 1$. Dieser Wert bleibt geheim, es ist der geheime Schlüssel.

(b) Berechne $A = a \cdot G$. Somit bildet $(G, A, \mathbb{F}_q)$ den öffentlichen Schlüssel.

2. Verschlüsselung einer Nachricht $m \in E(\mathbb{F}_q)$:

   (a) Wähle $r \in \{1, \ldots, p-1\}$, $\text{ggt}(r, p) = 1$.

   (b) Berechne die verschlüsselte Nachricht $(R, c) = (r \cdot G, r \cdot A + m)$.

3. Entschlüsseln einer Nachricht:

   (a) Berechne $m = c - a \cdot R$.

Dabei sieht man, dass die Sicherheit des Verfahrens, also der benötigte Aufwand, den privaten Schlüssel zu berechnen, auf dem DLP beruht. Ebenfalls lässt sich durch Nachrechnen verifizieren, dass das Verfahren mathematisch korrekt ist.

Bemerkung 2.3.1
Es hat sich herausgestellt, dass das DLP für elliptische Kurven besonders schwierig zu lösen ist. Bei einer Schlüssellänge von 256 Bits braucht es bereits 2^{128} Schritte, um das DLP zu lösen. Dazu gibt es eine Vielzahl an Analysen, zum Beispiel [BC13] oder [LM15], S. 524-547.

Wie wir an dem Beispiel gesehen haben, ist es für die Verschlüsselung auf elliptischen Kurven interessant, die Gruppenordnung bestimmen zu können. Ist diese prim, so ist die Gruppe über elliptischen Kurven zyklisch, und wir können das ElGamal-Verfahren anwenden.

Kapitel 3

Theorie für den Schoof-Algorithmus

Wir wollen im Folgenden die Anzahl der Punkte auf einer elliptischen Kurve über einem endlichen Körper bestimmen. Dafür hat René Schoof den *Frobenius-Endomorphismus* benutzt. Dieser beschreibt eine Abbildung auf einer elliptischen Kurve, die die Punkte von $E(\mathbb{F}_q)$ auf sich selbst abbildet und über $E(\overline{\mathbb{F}}_q)$ einen Homomorphismus beschreibt. Das Besondere an diesem Endomorphismus ist, dass man ihn als $2 \times 2-$Matrix schreiben kann, aus der sich die Anzahl der Lösungen berechnen lässt.
Da dies für große Zahlen sehr rechenaufwendig wird, berechnen wir die Anzahl der Punkte modulo einiger Primzahlen. Mithilfe des *Chinesischen Restsatzes* lässt sich eine Lösung modulo dem Produkt der Primzahlen berechnen. Dies ist ausreichend, falls das Produkt größer als die theoretisch mögliche Anzahl an Punkten ist. Dafür gibt uns der *Satz von Hasse* eine scharfe Obergrenze. Die Existenz einer Obergrenze ist schnell ersichtlich, da wir Punkte in einer Ebene über einem endlichen Körper betrachten.
Dafür müssen wir überprüfen, ob ein Punkt auf der elliptischen Kurve modulo einer Primzahl auch auf der Kurve $E(\mathbb{F}_q)$ liegt. Dabei helfen uns die sogenannten *Divisionspolynome*. Dies sind Polynome, die genau für die Lösungen modulo $l < q$ eine Nullstelle besitzen, wenn der Punkt auch auf der elliptischen Kurve $E(\mathbb{F}_q)$ liegt. Ebenfalls kann man mit diesen, zusammen mit der *Weil-Paarung*, den *Satz von Hasse* beweisen, wie auch eine wichtige Eigenschaft des *Frobenius-Endomorphismus*, womit sich die Gruppenordnung von $E(\mathbb{F}_q)$ bestimmen lässt. Das Kapitel orientiert sich an dem Buch „Elliptic Curves: Number Theory and Cryptography“ von L.C. Washington, [Was08]. Das Buch [Sil13] beschäftigt sich ebenfalls ausführlich mit dem Thema.

3.1 Divisionspolynome

Wir wollen die *Divisionspolynome* über die expliziten Formeln, wie in [Was08], Abschnitt 3.2, einführen und beschreiben.

Definition 3.1.1 (Divisionspolynome)
Divisionspolynome sind Polynome $\Psi_n \in \mathbb{Z}[x, y, a, b]$ der Form:

$$\begin{aligned}
\Psi_0 &= 0 \\
\Psi_1 &= 1 \\
\Psi_2 &= 2y \\
\Psi_3 &= 3x^4 + 6ax^2 + 12bx - a^2 \\
\Psi_4 &= 4y(x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - 8b^2 - a^3) \\
\Psi_{2n+1} &= \Psi_{n+2}\Psi_n^3 - \Psi_{n-1}\Psi_{n+1}^3, \text{ für } n \geq 2 \\
\Psi_{2n} &= \frac{1}{2y}\Psi_n(\Psi_{n+2}\Psi_{n-1}^2 - \Psi_{n-2}\Psi_{n+1}^2), \text{ für } n \geq 3.
\end{aligned}$$

Lemma 3.1.2
Bei Punkten (x, y) auf einer elliptischen Kurve, gegeben durch die kurze Weierstraß-Gleichung, gilt für die Divisionspolynome:

$$\begin{aligned}
\Psi_l(x, y) &\in \mathbb{Z}[x, a, b] \text{ für } l \text{ ungerade} \\
\Psi_l(x, y) &\in 2y\mathbb{Z}[x, a, b] \text{ für } l \text{ gerade.}
\end{aligned}$$

Beweis. Beweis über Induktion, in der man die Behauptung in den einzelnen Fällen nachrechnet. Wir nutzen dabei die kurze Weierstraß-Gleichung, um y^2 durch $x^3 + ax + b$ zu ersetzen. Folglich zeigen wir, dass für gerade l die Divisionspolynome Ψ_l eine ungerade Ordnung in y und den Vorfaktor 2 haben, und genauso für ungerade l die Ψ_l eine gerade Ordnung in y besitzen. Für $l \leq 4$ ist dies offensichtlich wahr. Sei also $l > 4$. Ist l gerade, so schreiben wir $l = 2n$. Dann gilt für $l > 4$ auch $2n > n + 2$, folglich können wir die Induktionsvoraussetzung für alle Polynome $\Psi_1, \ldots, \Psi_{n+2}$ annehmen. Bei ungeraden l schreiben wir $l = 2n + 1$, worauf genauso folgt, dass die Induktionsvoraussetzung für $\Psi_1, \ldots, \Psi_{n+2}$ als wahr angenommen werden kann.

- Für l gerade gilt:
  $l = 2n \Rightarrow \Psi_l = \Psi_{2n} = \frac{1}{2y}\Psi_n(\Psi_{n+2}\Psi_{n-1}^2 - \Psi_{n-2}\Psi_{n+1}^2)$.
  Nun unterscheiden wir zwischen n gerade und n ungerade:

  - n gerade:

    $$\frac{1}{2y}\Psi_n \in \mathbb{Z}[x, a, b].$$

In der Klammer ergibt sich bei beiden Termen ein Vorfaktor $2y$, da $\Psi_{n-1}, \Psi_{n+1} \in \mathbb{Z}[x, a, b]$, und die jeweils anderen Faktoren nach Induktionsvoraussetzung den Vorfaktor $2y$ besitzen. Folglich gilt in diesem Fall $\Psi_l \in 2y\mathbb{Z}[x, a, b]$.

- n ungerade:
  Dabei gilt für die Terme in der Klammer:

$$\Psi_{n+2}\Psi_{n-1}^2, \Psi_{n-2}\Psi_{n+1}^2 \in 4y^2\mathbb{Z}[x, a, b].$$

  Multiplizieren mit $\Psi_n \in \mathbb{Z}[x, a, b]$ ändert dies nicht. Multipliziert man am Ende mit $\frac{1}{2y}$ erhält man das gewünschte Ergebnis:

$$\Psi_l \in 2y\mathbb{Z}[x, a, b].$$

- Für l ungerade:
  - n gerade:

$$\Psi_{n+2}\Psi_n^3 \in (2y)^4\mathbb{Z}[x, a, b].$$

    Setzt man die besprochene Kongruenz $y^2 \equiv x^3 + ax + b$ ein, erhält man ein Polynom in $\mathbb{Z}[x, a, b]$. Das Polynom $\Psi_{n-1}\Psi_{n+1}^3$ ist nach Induktionsvoraussetzung in $\mathbb{Z}[x, a, b]$.
  - n ungerade:

$$\Psi_{n+2}\Psi_n^3 \in \mathbb{Z}[x, a, b], \Psi_{n-1}\Psi_{n+1}^3 \in (2y)^4\mathbb{Z}[x, a, b].$$

    Setzt man die besprochenen Kongruenzen ein, erhält man zwei Polynome in $\mathbb{Z}[x, a, b]$, die voneinander subtrahiert auch ein Element in $\mathbb{Z}[x, a, b]$ ergeben.

□

Wir wollen nun die skalare Multiplikation auf elliptischen Kurven mit Divisionspolynomen ausdrücken. Dafür definieren wir zuerst die dadurch entstehende Gruppe, analog zu [Was08], Abschnitt 3.1.

Definition 3.1.3
Sei $E(\mathbb{F}_q)$ eine elliptische Kurve, $l \in \mathbb{Z}, m \neq 0$. Dann ist die l-Torsionsgruppe, geschrieben $E[l]$ eine Gruppe mit der Eigenschaft:

$$E[l] = \{P \in E : l \cdot P = \mathcal{O}\}$$

Die einzelnen Punkte in der Gruppe heißen *l-Torsionspunkte*.

Die jeweiligen Gruppengesetze folgen aus den Rechengesetzen für Punkte auf elliptischen Kurven.

Satz 3.1.4
Sei P ein Punkt auf der elliptischen Kurve $E(\mathbb{F}_q)$, gegeben durch die kurze Weierstraß-Gleichung $y^2 = x^3 + ax + b$. Dann gilt für $l \in \mathbb{N}$ die Gleichung:

$$l \cdot P = (\frac{\phi_l(P)}{\Psi_l^2(P)}, \frac{\omega_l(P)}{\Psi_l^3(P)}),$$

mit den Polynomen

$$\begin{aligned} \phi_l(x,y) &:= x\Psi_l^2 - \Psi_{l+1}\Psi_{l-1} \\ \omega_l(x,y) &:= \tfrac{1}{4y}(\Psi_{l+2}\Psi_{l-1}^2 - \Psi_{l-2}\Psi_{l+1}^2). \end{aligned}$$

Dies ist wohldefiniert, das heißt für die Nullstellen $(x_i, *)$ von Ψ_l gilt:

$$l \cdot (x_i, *) = \mathcal{O}.$$

Beweis. Für den Beweis betrachtet man die elliptischen Kurven über dem Körper der komplexen Zahlen. Dies geht über den Rahmen dieser Bachelorarbeit hinaus, wird aber in [Was08], Thm. 9.33, bewiesen. □

Daraus folgt nun direkt die gewünschte Eigenschaft:

Korollar 3.1.5
Für die Divisionspolynome Ψ_l gilt:

$$\forall P = (x,y) \in E(\overline{\mathbb{F}_q}) : P \in E[l] \iff \Psi_l(x) = 0.$$

Für spätere Anwendungen wollen wir ebenfalls den Grad der Divisionspolynome beschreiben:

Lemma 3.1.6
Für den Grad des $l-$ten Divisionspolynoms Ψ_l in x gilt:

$$\deg(\Psi_l) = \begin{cases} \frac{l^2-4}{2} & \text{für } l \text{ gerade} \\ \frac{l^2-1}{2} & \text{für } l \text{ ungerade.} \end{cases}$$

Beweis. Der Beweis erfolgt durch Induktion. Dabei schreibt man $l = 2n+1$, beziehungsweise $l = 2n$ und unterscheidet jeweils zwischen n gerade und n ungerade. Dies ist sehr rechenaufwendig, aber geradlinig. Es wird nur die Gleichheit $y^2 = x^3 + ax + b$ der elliptischen Kurve benötigt, um dies richtig aufzulösen. Vergleiche für das Einsetzen die Divisionspolynome aus Definition 3.1.1.

1. Sei l gerade, also von der Form $l = 2n$:

   - Sei n gerade. Dann zeigen wir, dass $\Psi_l(x, y)$ von der Form $y(x^{\frac{l^2-4}{2}} + \ldots)$ ist:

   $$\begin{aligned}\Psi_{2n} &= \frac{1}{2y}\Psi_n(\Psi_{n+2}\Psi_{n-1}^2 - \Psi_{n-2}\Psi_{n+1}^2) \\ &= yx^{\frac{n^2-4}{2}}(dx^{\frac{(n+2)^2+4}{2}+\frac{2(n-1)^2-2}{2}} + ex^{\frac{(n-2)^2-4}{2}+\frac{2(n+1)^2-2}{2}}) + \ldots \\ &= yx^{\frac{n^2-4}{2}}(dx^{\frac{3n^2}{2}} + ex^{\frac{3n^2}{2}}) + \ldots \\ &= (d+e)yx^{\frac{4n^2-4}{2}} + \ldots \\ &= (d+e)yx^{\frac{l^2-4}{2}} + \ldots\end{aligned}$$

   für beliebige Koeffizienten $d, e \in \mathbb{Z}$.
   Teilen durch $(d + e)$ bringt das gewünschte Ergebnis.

   - Sei n ungerade. Dann folgt die Gleichheit ebenso mit Einsetzen der Formeln.

2. Sei l nun ungerade, also $l = 2n + 1$.

   - Sei n gerade.

   $$\begin{aligned}\Psi_{2n+1} &= \Psi_{n+2}\Psi_n^3 - \Psi_{n-1}\Psi_{n+1}^3 \\ &= dy^4x^{\frac{((n+2)^2-4)+(3n^2-12)}{2}} + \ldots + ex^{\frac{(2n+1)^2-1}{2}} + \ldots\end{aligned}$$

   Einsetzten der Gleichung $y^2 = ax^3 + bx + c$ bringt:

   $$\begin{aligned}&= d'y^4x^{3\frac{((n+2)^2-4)+(3n^2-12)}{2}} + ex^{\frac{(2n+1)^2-1}{2}} + \ldots \\ &= d'y^4x^{6+\frac{4n^2+4n-12}{2}} + ex^{\frac{(2n+1)^2-1}{2}} + \ldots \\ &= (d'+e)x^{\frac{(2n+1)^2-1}{2}}.\end{aligned}$$

   für beliebige Koeffizienten $d, d', e \in \mathbb{Z}$.
   Teilen durch $(d' + e)$ bringt wieder das gewünschte Ergebnis.

   - Sei l nun ungerade, also $l = 2n + 1$, mit $2 \nmid n$. Dann folgt das Resultat genauso wie oben, denn Ψ_n ist ein Term nur in x und $\Psi_{n\pm1}$ hat wieder jeweils ein y im Term.

□

3.2 Endomorphismen auf einer Elliptischen Kurve

Wir werden die Anzahl der Punkte vor allem mit dem nachfolgenden Endomorphismus, benannt nach Ferdinand Frobenius, bestimmen. Dabei orientieren wir uns an [Was08], Abschnitt 4.2. Zuerst wiederholen wir einen Satz aus der Zahlentheorie, welchen wir in [Was08], Appendix C.1, finden.

Satz 3.2.1
Sei $q = p^k, p$ prim und $\overline{\mathbb{F}}_q$ die algebraisch abgeschlossene Körpererweiterung von $\mathbb{F}_q$. Dann gilt:

$$\mathbb{F}_q = \{\alpha \in \overline{\mathbb{F}}_q : \alpha^q = \alpha\}.$$

Beweis. Es ist klar, dass die obige Menge aus den Nullstellen des Polynoms $g(x) = x^q - x$ über $\overline{\mathbb{F}_q}$ besteht. Es bleibt zu zeigen, dass das Polynom $g(x)$ keine weiteren Nullstellen besitzt. Dazu müsste der größte gemeinsame Teiler des Polynoms und der Ableitung des Selbigen allerdings deg > 1 besitzen. Allerdings ist der ggt 1, also vom Grad 0, da man mit $q = 0$ in $\mathbb{F}_q$ erhält: $(x^q - x)' = qx^{q-1} - 1 = -1$.
Somit besitzt das Polynom keine doppelten Nullstellen, und aufgrund des *Fundamentalsatzes der Algebra* auch nicht mehr als q Elemente. Somit ist $\mathbb{F}_q$ durch die Nullstellen des Polynoms $g(x), x \in \overline{\mathbb{F}}_q$ eindeutig gegeben. □

Satz 3.2.2 (Frobenius-Endomorphismus)
Sei $\overline{\mathbb{F}}_q$ der algebraische Abschluss von $\mathbb{F}_q$. Dann gilt für die Abbildung

$$\phi_q : \overline{\mathbb{F}}_q \to \overline{\mathbb{F}}_q, (x, y) \mapsto (x^q, y^q) :$$

1. ϕ_q ist ein Endomorphismus auf $E(\overline{\mathbb{F}}_q)$
2. $(x, y) \in E(\mathbb{F}_q) \Leftrightarrow \phi_q(x, y) = (x, y)$.

Außerdem ist ϕ_q die Identität auf $E(\mathbb{F}_q)$ und ein Homomorphismus auf $E(\mathbb{F}_q) \cup \mathcal{O}$.

Beweis. Siehe [Was08], Lemma 4.5.

1. ϕ_q ist ein Homomorphismus, was man durch Nachrechnen und Einsetzen der Additionsformel auf elliptischen Kurven zeigen kann.
   Für einen Körper $\mathbb{F}_q, q = p^k, p$ prim, gilt: $(a + b)^q = a^q + b^q$, sowie $a^q = a$. Dies setzt man in die (kurze) Weierstraß-Gleichung ein und erhält:
   $$y^{2^q} = (x^3 + ax + b)^q = (y^q)^2 = (x^q)^3 + ax^q + b$$
   Dies zeigt, dass der Punkt (x^q, y^q) tatsächlich in E liegt.

2. Nach Satz 3.2.1 gilt für $x \in \overline{\mathbb{F}}_q$:

$$x \in \mathbb{F}_q \Leftrightarrow \phi_q(x) = x^q = x.\tilde{}$$

Damit folgt:

$$\begin{aligned}(x,y) \in E(\mathbb{F}_q) &\Leftrightarrow x, y \in \mathbb{F}_q \\ &\Leftrightarrow \phi_q(x) = x, \phi_q(y) = y \\ &\Leftrightarrow \phi_q(x,y) = (x,y).\end{aligned}$$

Dies beschreibt einen Endomorphismus, wie man durch Nachrechnen und Einsetzen der Additionsformel auf elliptischen Kurven beweist. Dass $(x,y) \in E(\mathbb{F}_q) \Leftrightarrow \phi_q(x,y) = (x,y)$ gilt, folgt aus dem Einsetzen in die (kurze) Weierstraß-Gleichung. Insbesondere folgt aus dem zweiten Punkt:

$$\#E(\mathbb{F}_q) = \#\ker(\phi_q - 1). \tag{3.1}$$

□

Damit lässt sich der *Satz von Hasse* beweisen, der eine wichtige Abschätzung für die Anzahl der Punkte auf elliptischen Kurven gibt.
Berechnungen werden dabei in den sogenannten *Torsionsgruppen* durchgeführt. Diese sind wie folgt definiert:
Hat man eine elliptische Kurve E über einen Körper $\mathbb{F}_q$ sowie eine Primzahl p, $\text{ggt}(p,q) = 1$, dann kann man über Divisionspolynome ([Was08], Thm. 3.2) beweisen, dass gilt: $E[p] \cong \mathbb{Z}/p \times \mathbb{Z}/p$. Da p prim ist, ist $\mathbb{Z}/p$ zyklisch, somit lässt sich jeder Endomorphismus darauf als Koordinatentransformation ausdrücken, darstellbar als $2 \times 2 -$ Matrix, die von links an $\begin{pmatrix} x \\ y \end{pmatrix} \in E[p]$ multipliziert wird.
Bevor wir das beweisen, definieren wir noch den Grad eines Endomorphismus und einen separablen Endomorphismus. Vergleiche dazu [Was08], S. 51.

Definition 3.2.3 (Grad eines Endomorphismus)
Sei $\alpha(x,y)$ ein Endomorphismus, schreibe $\alpha(x,y) = (r_1(x), r_2(x)y)$, und $r_1(x) = \frac{p(x)}{q(x)}$. Dann ist der Grad von α definiert duch
$\deg(\alpha) = \text{Max}\{\deg(p(x)), \deg(q(x))\}$.

Definition 3.2.4 (Separabler Endomorphismus)
Der oben beschriebene Endomorphismus $\alpha(x,y) = (r_1(x)r_2(x)y) \neq 0$ heißt separabel, falls $r_1'(x) \neq 0$.

Damit können wir zeigen, dass eine l-Torsionsgruppe die Ordnung l^2 besitzt. Gruppen der Ordnung l^2, mit l prim, sind abelsch. Also ist die l-Torsionsgruppe nach dem *Hauptsatz für endlich erzeugte abelsche Gruppen* isomorph zu $\mathbb{Z}/l \times \mathbb{Z}/l$.

Satz 3.2.5
Der Endomorphismus $n \cdot P, P \in E(\mathbb{F}_q)$, der die $n-$Torsionsgruppe beschreibt, hat den Grad n^2.

Beweis. Wir folgen für den Beweis [Was08], Korollar 3.7. Wie wir gesehen haben, ist die skalare Multiplikation auf der elliptischen Kurve mit dem Punkt $P = (x, y)$ gegeben durch $l \cdot P = (\frac{\phi_l(x)}{\Psi_l^2(x)}, \frac{\omega_l(x,y)}{\Psi_l(x,y)^3})$, mit den in Satz 3.1.4 definierten Polynomen ϕ_l und ω_l. Dies beschreibt einen Endomorphismus. Um den Grad davon zu bestimmen, betrachten wir also den Bruch in der ersten Komponente. Dafür geben wir folgendes Lemma an. In [Was08] findet man dieses als Lemma 3.5.

Lemma 3.2.6
Für die Polynome ϕ_l und Ψ_l^2 gelten:

$$\begin{aligned} \deg(\phi_l(x)) &= l^2, \\ \deg(\Psi_l^2(x)) &= l^2 - 1. \end{aligned}$$

Beweis. Beweis durch Induktion.
Die zweite Aussage folgt direkt aus Lemma 3.1.6. Daraus folgt die Aussage für ϕ_l. Aus der Definition von ϕ_l,

$$\phi_l(x, y) := x\Psi_l^2 - \Psi_{l+1}\Psi_{l-1},$$

lässt sich folgern, dass $x\Psi_l^2$ den Grad l^2 und $\Psi_{l+1}\Psi_{l-1}$ nach Lemma 3.1.6 einen Grad kleiner gleich $\frac{(l+1)^2+(l-1)^2-2}{2} = \frac{2l^2}{2} = l^2$ besitzt.

□

Damit ist der Grad von ϕ_l größer als der Grad des Divisionspolynoms Ψ_l. Somit ist der Grad des durch die skalare Multiplikation beschriebenen Endomorphismus l^2, falls die beiden Polynome teilerfremd sind. Das wollen wir im Folgenden durch Widerspruch zeigen.
Nehmen wir an n wäre der kleinste Index, für den die beiden Polynome eine gemeinsame Nullstelle besitzen. Nun unterscheiden wir, ob n gerade oder ungerade ist und führen die Behauptung in beiden Fällen zum Widerspruch:

- $n = 2m$: Somit können wir den Punkt $2m \cdot P, P = (x, y)$ berechnen, indem wir zuerst P mit 2 multiplizieren und dann mit m. Wir erhalten für die Multiplikation mit 2:

$$\phi_2(x) = x^4 - 2ax^2 - 8bx + a^2$$
$$\Psi_2(x)^2 = 4y^2 = 4(x^3 + ax + b).$$

Somit erhalten wir:

$$\frac{\phi_{2m}}{\Psi_{2m}^2} = \frac{\phi_2(\frac{\phi_m}{\Psi_m}^2)}{\phi_2^2(\frac{\Psi_m}{\phi_m^2})} = \frac{\phi_m^4 - 2a\phi_m^2\Psi_m^4 - 8b\phi_m\Psi_m^6 + a^2\Psi_m^8}{4\Psi_m^2(\phi_m^3 + a\phi_m\Psi_m^4 + b\Psi_m^6)} = \frac{U}{V}. \quad (3.2)$$

Jetzt zeigen wir, dass die Polynome U und V keine gemeinsame Nullstelle haben. Dafür hilft uns die folgende Rechnung. In [Was08] ist dies Lemma 3.8.

Lemma 3.2.7
Sei $\Delta = 4a^3 + 27b^2$ und

- $F(x, z) := x^4 + 2ax^2z^2 - 8bxz^3 + a^2z^4$
- $G(x, z) := 4z(x^3 + axz^2 + bz^3)$
- $f_1(x, z) := 12x^2z + 16az^3$
- $g_1(x, z) := 3x^3 - 5axz^2 - 27bz^3$
- $f_2(x, z) := 4\Delta x^3 - 4a^2bx^2z + 4a(3a^3 + 22b^2)xz^2 + 12b(a^3 + 8b^2)z^3$
- $g_2(x, z) := a^2bx^3 + a(5a^3 + 32b^2)x^2z + 2b(13a^3 + 96b^2)xz^2 - 3a^2(a^3 + 8b^2)z^3$

Dann gilt: $Ff_1 - Gg_1 = 4\Delta z^7$ und $Ff_2 + Gg_2 = 4\Delta x^7$.

Beweis. Dies wird durch Nachrechnen gezeigt. Die Polynome erhält man aus dem *erweiterten euklidischen Algorithmus* für Polynome. Die beiden Polynome $F(x, 1), G(x, 1)$ haben keine gemeinsame Nullstelle, damit findet man die Polynome $f_1(x), g_1(x)$, sodass

$$F(x, 1)f_1(x) + G(x, 1)g_1(x) = 1.$$

Ersetzt man x mit $\frac{x}{z}$, multipliziert mit z^7 und mit 4Δ, so erhält man die erste Gleichung. Die zweite folgt genauso, nur durch das Ersetzen von z durch $\frac{z}{x}$ und anschließendem Multiplizieren von $4x^7\Delta$.

□

$$U \cdot f_1(\phi_m, \Psi_m^2) - V \cdot g_1\phi_m, \Psi_m^2) = 4\Psi_m^{14}\delta$$
$$U \cdot f_2(\phi_m, \Psi_m^2) + V \cdot g_2\phi_m, \Psi_m^2) = 4\Psi_m^{7}\delta$$

Somit gilt: Falls die Terme U und V eine gemeinsame Nullstelle besitzen, so auch die beiden Terme Ψ_m und ϕ_m. Allerdings ist $n = 2m$ der erste Index, bei dem dies der Fall wäre, womit wir in diesem Fall einen Widerspruch erhalten.
Somit müssen wir nur noch zeigen, dass aus Gleichung (3.2) folgt, dass $U = \phi_{2m}$ und $V = \Psi_{2m}^2$ gilt:
Wir wissen, dass $\frac{U}{V} = \frac{\phi_{2m}}{\Psi_{2m}^2}$ und dass U, V keine gemeinsame Nullstelle besitzen. Somit ist Ψ_{2m} ein Vielfaches von U und Ψ_{2m}^2 ein Vielfaches von V.
Mithilfe von dem obigen Lemma 3.2.6 verifiziert man, dass $\deg(U) = 4m^2$. Somit gilt, dass $\deg(U) = \deg(\phi_{2m})$. Daraus folgt, dass $U = \phi_{2m}$ gilt, und somit auch $V = \Psi_{2m}^2$.

- $n = 2m + 1$: Sei r eine gemeinsame Nullstelle von ϕ_n und Ψ_n^2.
  Da $\phi_n = x\Psi_n - \Psi_{n-1}\Psi_{n+1}$ nach Definition gilt, sowie auf der elliptischen Kurve $y^2 = x^3 + ax + b$ der Term $\Psi_{n-1}\Psi_{n+1}$ ein Term nur in x ist, folgt damit, dass $\Psi_{n-1}\Psi_{n+1}(r) = 0$. Somit folgt, dass $\Psi_{n+1}^2\Psi_{n-1}^2(r) = 0 \Rightarrow \Psi_{n\pm1}^2(r) = 0$.
  Sei $\delta = \pm 1$. Somit, nach Lemma 3.1.2, ist sowohl Ψ_n, als auch $\Psi_{n+2\delta}$ ein Polynom nur in x. Dann folgt: $(\Psi_n\Psi_{n+2\delta})^2(r) = \Psi_n^2(r)\Psi_{n+2\delta}^2(r) = 0 \Rightarrow \Psi_n(r)\Psi_{n+2\delta}(r) = 0$. Damit gilt auch $\phi_{n+\delta}(r) = 0$, da $\phi_{n+\delta} = x\Psi_{n+\delta}^2 - \Psi_n\Psi_{n+2\delta}$. Somit haben die beiden Polynome $\phi_n, \phi_{n+\delta}^2$ eine gemeinsame Nullstelle. Allerdings ist $n+\delta$ gerade, womit wir den ersten Fall, $n+\delta = 2m'$, vorliegen haben. Da n der kleinste Index sein soll, für den die beiden Polynome eine gemeinsame Nullstelle haben, erhalten wir die Ungleichung:
  $$\frac{n+\delta}{2} \geq n.$$
  Daraus folgt aber $(\delta, n) \in \{(-1, 1), (1, 1)\}$. Somit gilt $n = 1$. Damit können wir direkt aus der Definition einen Widerspruch ableiten, denn es gilt: $\Psi_1(x) = 1 \Rightarrow \Psi_1^2(x) = 1$. Also hat Ψ_1^2 keine Nullstelle, insbesondere keine gemeinsame mit $\phi_{1\pm1}^2(x)$.

Somit ist die Behauptung bewiesen, und der Endomorphismus, der das Multiplizieren eines Punktes auf der elliptischen Kurve mit einem Skalar l beschreibt, hat den Grad l^2.

□

Wir wollen nun die Ordnung der Gruppe $E[l]$ bestimmen. Dazu wollen wir im nächsten Satz die Gleichheit $\deg(\alpha) = \#\ker(\alpha)$ zeigen, falls α ein separabler Endomorphismus ist. Dabei stellen wir fest, dass die Multiplikation mit $n \in \mathbb{Z}$ separabel ist.

Dafür wollen wir einen allgemeinen Homomorphismus auf elliptischen Kurven anders schreiben. In [Was08] findet man dies auf S. 51.

Lemma 3.2.8
Man kann einen Homomorphismus $\alpha : E(\overline{\mathbb{F}}_q) \to E(\overline{\mathbb{F}}_q)$,
$\alpha(x, y) = (R_1(x, y), R_2(x, y))$ auch schreiben als $(r_1(x), r_2(x)y)$, $r_1(x) = \frac{p(x)}{q(x)}$,
mit teilerfremden Polynomen $p(x), q(x)$.

Beweis. Wir können eine beliebige rationale Funktion $R(x, y) \in \overline{\mathbb{F}}_q[x, y]$ auch auf der elliptischen Kurve $E(\overline{\mathbb{F}}_q)$ schreiben, indem wir die Gleichung der elliptischen Kurve, $y^2 = x^3 + ax + b$, einsetzen. Wir rechnen also in dem Polynomring ${}^{\overline{\mathbb{F}}_q[x,y]}\!/_{(x^3+ax+b-y^2)}$. Somit erhalten wir

$$R(x, y) \equiv \frac{p_1(x) + p_2(x)y}{p_3(x) + p_4(x)y} \bmod x^3 + ax + b - y^2,$$

für bestimmte Polynome $p_1(x), p_2(x), p_3(x), p_4(x)$.
Durch Erweitern mit den Term $p_3(x) - p_4(x)y$ erhält man somit Polynome $q_1(x), q_2(x), q_3(x)$, sodass gilt:

$$R(x, y) = \frac{q_1(x) + q_2(x)y}{q_3(x)}.$$

Da α ein Homomorphismus auf einer elliptischen Kurve ist, gilt ebenfalls:

$$\alpha(x, -y) = \alpha(-(x, y)) = -\alpha(x, y).$$

Daraus folgt, dass für $\alpha(x, y) = (R_1(x, y), R_2(x, y))$ gilt:

$$R_1(x, -y) = R_1(x, y), \text{ und } R_2(x, -y) = -R_2(x, y).$$

Zusätzlich kann man, wie oben gezeigt, $R_1(x, y) = \frac{q_1'(x)+q_2'(x)y}{q_3'(x)}$ schreiben. Daraus folgt $q_2'(x) = 0$, und $R_1(x, y)$ ist in der Tat eine Funktion nur in x. Somit erhält man $r_1(x), r_2(x)$, sodass für den Homomorphismus gilt:

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

□

Damit kann man jetzt den nächsten Satz beweisen. In [Was08] ist dies die Präposition 2.21.

Satz 3.2.9
Sei α ein separabler Endomorphismus auf einer elliptischen Kurve $E(\mathbb{F}_q)$. Dann gilt:

$$\deg(\alpha) = \#\mathrm{Ker}(\alpha),$$

wobei $\mathrm{Ker}(\alpha)$ der Kern des Homomorphismus $\alpha : E(\overline{\mathbb{F}}_q) \to E(\overline{\mathbb{F}}_q)$ ist.

Beweis. Schreibe $\alpha(x, y) = (r_1(x), r_2(x)y)$, mit $r_1(x) = \frac{p(x)}{q(x)}$, wobei die Polynome $p(x)$ und $q(x)$ teilerfremd seien. Damit gilt für die Ableitung:

$$r_1(x)' \neq 0 \Rightarrow p'q - qp' \neq 0.$$

Sei $S = \{x \in \overline{\mathbb{F}}_q \mid (pq' - qp')(x)q(x) = 0\}$ und $(a, b) \in E(\mathbb{F}_q)$ so, dass

1. $a, b \neq 0, (a, b) \neq \mathcal{O}$,
2. $\deg(p(x) - aq(x)) = \max\{\deg(p), \deg(q)\} = \deg(\alpha)$,
3. $a \notin r_1(S)$ und
4. $(a, b) \in \alpha(E(\overline{\mathbb{F}}_q))$.

Solch ein (a, b) existiert, da:

- $p'q - q'p \neq 0$, folglich ist die Menge S endlich und somit auch $\alpha(S)$.
- Die Funktion $r_1(x)$ nimmt unendlich viele verschiedene Punkte über $\overline{\mathbb{F}}_q$ an. Dafür wiederholen wir das folgende Lemma. Man vergleiche dazu zum Beispiel die Übungsaufgabe 4 aus Abschnitt 3.4 in [Bos13].

  Lemma 3.2.10
  Algebraisch abgeschlossene Körper haben unendlich viele Elemente.

  Beweis. Wir zeigen, dass endliche Körper, bezeichnet als $\mathbb{F}_q$, nicht algebraisch abgeschlossen sein können. Dabei orientieren wir uns an Euklids Beweis für die Existenz von unendlich vielen Primzahlen. Wir definieren die Funktion $f(x) = (\prod_{x \in \mathbb{F}_q}(x - a)) + 1$. Diese müsste nach dem *Fundamentalsatz der Algebra* eine Nullstelle besitzen, allerdings gilt $\forall a \in \mathbb{F}_q : f(a) = 1$. Dies ist ein Widerspruch. □

- $\forall x \in \overline{\mathbb{F}}_q \; \exists (x, y) \in E(\overline{\mathbb{F}}_q)$, somit ist die Menge $\alpha(E(\overline{\mathbb{F}}_q))$ unendlich.

Behauptung: Es existieren genau $\deg(\alpha)$ verschiedene Punkte (x_1, y_1) auf der Kurve $E(\overline{\mathbb{F}}_q)$, sodass $\alpha(x_1, y_1) = (a, b)$.
Wir zeigen, dass wir nur die verschiedene Werte für x_1 zeigen müssen:

Für solch einen Punkt haben wir dann $(a, b) = (\frac{p(x_1)}{q(x_1)}, y_1 r_2(x_1))$.
Nachdem $(a, b) \neq \mathcal{O}$, gilt $q(x_1) \neq 0$. Wir zeigen kurz, dass $r_2(x_1)$ definiert ist. Man findet dies auch als Aufgabe 2.19 in [Was08]. Daraus folgt, mit $b \neq 0$, dass $y_1 = \frac{b}{r_2(x_1)}$ gilt und somit y_1 nur von x_1 abhängt.

Lemma 3.2.11
Der oben angegebene Endomorphismus, $\alpha(x, y) = (\frac{p(x)}{q(x)}, y r_2(x))$ ist wohldefiniert in der y-Koordinate. Dafür schreiben wir $r_2(x) = \frac{s(x)}{t(x)}$.

Beweis. Es gilt, dass $p(x), q(x)$ keine gemeinsame Nullstelle besitzen, ebenso wie $s(x)$ und $t(x)$. Ebenfalls nehmen wir $q(x) \neq 0$ an. Es lässt sich somit zeigen, dass es ein zu $q(x)$ teilerfremden Polynom $u(x)$ gibt, sodass gilt:

$$\frac{(x^3 + ax + b)s^2(x)}{t(x)^2} = \frac{u(x)}{q(x)^3}.$$

Dabei nutzt man aus, dass der Endomorphismus einen Punkt auf der Kurve angibt. Man erhält also die Gleichung:

$$\begin{aligned}
\frac{s^2}{t^2} y^2 &= (\frac{p}{q})^3 + a\frac{p}{q} + b \Leftrightarrow \\
\frac{s^2}{t^2}(x^3 + ax + b) &= (\frac{p}{q})^3 + a\frac{p}{q} + b \Leftrightarrow \\
\frac{u}{q^3} &= \frac{p^3 + aq^2p + bq^3}{q^3} \Leftrightarrow \\
\frac{u}{q^3} &= \frac{p^3 + aq^2p}{q^3} + b.
\end{aligned}$$

Damit lässt sich erkennen, dass die beiden Polynome q und u genau dann eine gemeinsame Nullstelle besitzen, falls p und q ebenfalls eine gemeinsame Nullstelle besitzen.

Daraus folgert man:

$$\exists x_0 : t(x_0) = 0 \Rightarrow q(x_0) = 0.$$

□

Somit können wir den Beweis von Satz 3.2.9 abschließen. Nach der zweiten Annahme für die Menge S reicht es zu zeigen, dass das Polynom $p(x) - aq(x)$

keine doppelte Nullstelle besitzt.
Wir nehmen also an, $p(x) - aq(x)$ hätte eine doppelte Nullstelle. Dann gäbe es $x_0 \in E(\overline{\mathbb{F}}_q)$, sodass die folgenden beiden Gleichungen gelten:

$$p(x_0) - aq(x_0) = 0 \text{ und } p'(x_0) - aq'(x_0) = 0.$$

Für $a \neq 0$ heißt das, dass x_0 eine Nullstelle von $pq' - p'q$ sein muss, also gilt $x_0 \in S$. Damit wäre aber $a = r_1(x_0) \in r_1(S)$ wahr. Dies ist ein Widerspruch zur Behauptung, also besitzt das Polynom $p - aq$ insgesamt $\deg(\alpha)$ verschiedene Nullstellen. Somit existieren auch $\deg(\alpha)$ verschiedene Punkte (x_1, y_1), sodass $\alpha(x_1, y_1) = (a, b)$, womit auch der Kern des Homomorphismus α $\deg(\alpha)$ verschiedene Elemente besitzt. Da α ein Homomorphismus ist, gilt dies $\forall (a, b) \in \alpha(E(\overline{\mathbb{F}}_q))$.

□

Die skalare Multiplikation eines Punktes auf einer elliptischen Kurve ist separabel. Nach den beiden Sätzen 3.2.5 und 3.2.9 folgt, dass im Kern der Abbildung für die Multiplikation eines Punktes mit einer Primzahl $l \nmid q$ insgesamt l^2 Elemente sind. Mit dem *Hauptsatz für endlich erzeugte abelsche Gruppen* folgt dann, dass $E[l] \equiv \mathbb{Z}/l \times \mathbb{Z}/l$, $l \nmid q$, gilt.
Im Schoof-Algorithmus benutzt man für l nur Primzahlen. Somit sind die l-Torsionsgruppen immer isomorph zu der Gruppe $\mathbb{Z}/l \times \mathbb{Z}/l$, wobei $\mathbb{Z}/l$ zyklisch ist. Zyklische Gruppen haben insbesondere einen Erzeuger, womit man einen Endomorphismus über den Torsionsgruppen auch als lineare Transformation schreiben kann, siehe auch [Was08], S. 89:

Korollar 3.2.12
Seien $\beta_1, \beta_2 \in E[l]$ die Punkte, die die l-Torsionsgruppe über einer elliptischen Kurve $E(\mathbb{F}_q)$ erzeugen. Dann existieren für jeden Punkt $P \in E[l]$ zwei Elemente m_1 und m_2 aus $\mathbb{Z}/l$, sodass man P als Linearkombination der beiden Erzeuger schreiben kann: $P = m_1\beta_1 + m_2\beta_2$. Zusätzlich lässt sich jeder Endomorphismus auf $E[l]$ als Lineartransformation der Basis (β_1, β_2) von $E[l]$ schreiben.

Somit existiert zu jedem Homomorphismus ϕ eine Matrix A, sodass $\phi(P) = A \cdot P$ mit $P \in E(\mathbb{F}_q)$.
Die Eigenschaft, einen Endomorphismus auf l-Torsionsgruppen auch als 2×2-Matrix zu schreiben, wird noch oft benötigt. Damit lässt sich die *Weil-Paarung* anwenden, um den *Satz von Hasse* beweisen. Ebenfalls kann man Sätze für 2×2 - Matrizen somit auf Endomorphismen anwenden, um weitere Aussagen über den *Frobenius-Endomorphismus* zu treffen.

3.3 Der Satz von Hasse und die Spur des Frobenius

Sowohl für den *Satz von Hasse* als auch den *Frobenius-Endomorphismus*, wird der folgende Satz benötigt. Den Beweis findet man in [Was08], Thm. 11.7.

Satz 3.3.1 (Weil-Paarung)
Die Abbildung

$$e_n : E[n] \times E[n] \to \mu_n \tag{3.3}$$

wird Weil-Paarung genannt, wenn sie die folgenden Eigenschaften erfüllt:

1. e_n ist bilinear. Das bedeutet, dass
   $e_n(S_1 + S_2, T) = e_n(S_1, T)e_n(S_2, T)$ und
   $e_n(S, T_1, T_2) = e_n(S, T_1)e_n(S, T_2)$
   $\forall S, S_1, S_2, T, T_1, T_2 \in E[n]$.

2. e_n ist nicht ausgeartet, dies bedeutet:
   $\forall T \in E[n] : e_n(S, T) = 1 \Rightarrow S = \infty$,
   sowie
   $\forall S \in E[n] : e_n(S, T) = 1 \Rightarrow T = \infty$.

3. $e_n(T, T) = 1 \; \forall \, T \in E[n]$.

4. $e_n(T, S) = e_n(S, T)^{-1} \; \forall \, S, T \in E[n]$.

5. $e_n(\sigma S, \sigma T) = \sigma(e_n(S, T))$ für alle Automorphismen $\sigma : \overline{\mathbb{F}_q} \to \overline{\mathbb{F}_q}$, wobei σ die Koeffizienten der kurzen Weierstraß-Gleichung für E wieder auf sich selbst abbildet.

6. $e_n(\alpha(S), \alpha(T)) = e_n(S, T)^{\deg(\alpha)}$ für alle separablen Endomorphismen α von E. Falls die Koeffizienten von E Elemente aus dem Körper $\mathbb{F}_q$ sind, dann gilt dies auch für den Frobenius-Endomorphismus.

Jetzt können wir eine Abschätzung über die Anzahl der Punkte, den *Satz von Hasse*, beweisen.
Zuerst zeigen wir eine Möglichkeit, die Anzahl der Punkte auf einer elliptischen Kurve zu berechnen. Daraus folgern wir dann den *Satz von Hasse*, sowie eine konkrete Berechnung für die Anzahl der Punkte, die sogenannte *Spur des Frobenius*.
Den *Satz von Hasse* beweisen wir in den folgenden Schritten, nach dem Buch [Was08]:

1. Es gilt: $\det(\alpha) = \deg(\alpha)$ für einen separablen Endomorphismus α, oder den Frobenius-Endomorphismus.

2. Es gilt: $\deg(\phi_q - 1) = \#E(\mathbb{F}_q)$, mit ϕ_q Frobenius-Endomorphismus auf $E(\mathbb{F}_q)$.

3. Es gilt: $\deg(r\phi_q - s) = r^2q + s^2 - rs(q + 1 - \#E)$.

4. Der *Satz von Hasse* folgt dann aus der Abschätzung $\deg(r\phi_q - s) \geq 0$.

Die erste Aussage liefert folgendes Lemma. Es findet sich auch in [Was08], Präposition 3.15.

Lemma 3.3.2
Sei $\text{ggt}(n, q) = 1$, α_l ein beliebiger Endomorphismus auf der l-Torsionsgruppe $E[l]$ mit Basis (T_1, T_2), geschrieben als Matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, mit $a, b, c, d \in \mathbb{Z}/l$. Dann gilt: $\det(\alpha_l) \equiv \deg(\alpha) \bmod l$.

Beweis. Aus der Weil-Paarung folgt:

$$\begin{aligned} e_n(T_1, T_2)^{\deg(\alpha)} &= e_n(\alpha(T_1), \alpha(T_2)) = e_n(aT_1 + cT_2, bT_1 + dT_2) \\ &= e_n(T_1, T_1)^{ab} e_n(T_1, T_2)^{ad} e_n(T_2, T_1)^{cb} e_n(T_2, T_2)^{c,d} \\ &= e_n(T_1, T_2)^{ad-bc} = e_n(T_1, T_2)^{\det(\alpha)} \text{ (mod n)}. \end{aligned}$$

Da n beliebig groß sein kann, folgt ebenfalls Gleichheit. □

Die zweite Aussage, $\deg(\phi_q - 1) = \#E(\mathbb{F}_q)$, folgt aus dem folgenden Satz. Dieser zeigt, dass $\phi_q - 1$ einen separablen Endomorphismus beschreibt. Somit können wir dann den Satz 3.2.9 anwenden, und erhalten

$$\#\ker(\phi_q - 1) = \deg(\phi_1 - 1).$$

Die Punkte auf der elliptischen Kurve $E(\mathbb{F}_q)$ sind aber nach Gleichung 3.1 im Beweis von Satz 3.2.2 genau die Punkte aus dem Kern der Abbildung $\phi_q - 1$. Für den nächsten Satz werden noch einige allgemeinere Aussagen über Endomorphismen auf elliptischen Kurven benötigt, die im Laufe des Beweises gezeigt werden. Den Satz findet man auch in [Was08], Präposition 2.29.

Satz 3.3.3
Sei $E(\mathbb{F}_q)$ eine elliptische Kurve, mit $q = p^k, k \in \mathbb{Z}$. Seien $r, s \in \mathbb{Z}$, nicht beide gleich Null. Dann ist der Endomorphismus $r\phi_q + s$ genau dann separabel, wenn $p \nmid s$.

Beweis. Dafür wollen wir erst zwei Lemmata für Endomorphismen auf elliptischen Kurven zeigen, mit denen man einen Endomorphismus umschreiben kann. Damit folgt dann die Behauptung. Das erste Lemma findet man in [Was08], Lemma 2.26, kombiniert mit der Bemerkung 2.27. Das zweite Lemma findet man in [Was08], Lemma 2.24.

Lemma 3.3.4
Seien $\alpha_1, \alpha_2, \alpha_3 \neq 0$ Endomorphismen auf einer elliptischen Kurve $E(\mathbb{F}_q)$, mit $\alpha_1 + \alpha_2 = \alpha_3$. Schreibe diese in der Form

$$\alpha_i(x, y) = (R_{\alpha_i}(x), yS_{\alpha_i}(x)).$$

Nimmt man zusätzlich an, dass es Konstanten $c_{\alpha_1}, c_{\alpha_2}$ gibt, sodass

$$\frac{R'_{\alpha_1}(x)}{S_{\alpha_1}(x)} = c_{\alpha_1}, \frac{R'_{\alpha_2}(x)}{S_{\alpha_2}(x)} = c_{\alpha_2},$$

so gilt:

$$\frac{R'_{\alpha_1}(x)}{S_{\alpha_1}(x)} = c_{\alpha_1} + c_{\alpha_2}.$$

Beweis. Wir müssen, um zu überprüfen, ob der Ausdruck separabel ist, die Ableitung einer (noch genauer zu bestimmenden) Funktion untersuchen. Für diese müssen wir die Gleichung der elliptischen Kurve nach x ableiten. Schreiben wir also:

$$\begin{aligned} y^2 = x^3 + ax + b \Leftrightarrow y &= \pm\sqrt{x^3 + ax + b} \\ \Rightarrow y' &= \frac{x^3 + a}{2y} \\ \Rightarrow 2yy' &= x^3 + a \\ \Rightarrow \frac{df}{dx}(x, y) &= \frac{\partial f}{\partial x}(x, y) + \frac{\partial f}{\partial y}(x, y) = 3x^2 + a + 2y \end{aligned}$$

Damit stellen wir analog zu Lemma 2.24 in [Was08] für die Addition von Punkten auf einer elliptischen Kurve folgendes fest:

Lemma 3.3.5
Sei $E(\mathbb{F}_q)$ gegeben durch $y^2 = x^3 + ax + b$ (u, v) ein Punkt auf der Kurve $E(\mathbb{F}_q)$. Schreibe

$$(x, y) + (u, v) = (f(x, y), g(x, y)),$$

wobei $f(x, y), g(x, y)$ rationale Funktionen in x, y sind, sowie y eine Funktion in x, mit $\frac{dy}{dx} = \frac{3x^2+a}{2y}$ sein soll.
Dann gilt:

$$\frac{\frac{\partial f}{\partial x}(x, y)}{g(x, y)} = \frac{1}{y}.$$

Beweis. Die Formeln für die Addition auf elliptischen Kurven liefern:

$$f(x,y) = (\frac{y-v}{x-u})^2 - x - u$$
$$g(x,y) = \frac{-(y-v)^3 + x(y-v)(x-u)^2 + 2u(y-v)(x-u)^2 - v(x-u)^3}{(x-u)^3}$$
$$\frac{\partial f}{\partial x}(x,y) = \frac{2y'(y-v)(x-u) - 2(y-v)^2 - (x-u)^3}{(x-u)^3}.$$

Man kann daraus und mit der Gleichung $2yy' = 3x^2 + a$ direkt berechnen, dass gilt:

$$(x-u)^3(y\frac{\partial f}{\partial x}(x,y) - g(x,y)) = (v-y)(au + u^3 - v^2 - ax - x^3 + y^2).$$

Ebenfalls gelten, da $(u,v),(x,y) \in E(\mathbb{F}_q)$, die beiden Gleichungen

$$v^2 = u^3 + au + b,$$
$$y^2 = x^3 + ax + b.$$

Somit wird der obige Ausdruck vereinfacht zu:

$$(v-y)(-b+b) = 0.$$

Somit gilt: $y\frac{\partial f}{\partial x}(x,y) = g(x,y)$, und damit die Behauptung, $\frac{\frac{\partial f}{\partial x}(x,y)}{g(x,y)} = \frac{1}{y}$. □

Nun können wir den Beweis von Lemma 3.3.5 fortsetzen. Seien dazu $(x_1,y_1),(x_2,y_2) \in E(\mathbb{F}_q)$, mit $(x_1,y_1) + (x_2,y_2) = (x_3,y_3)$. Schreibe $(x_1,y_1) = \alpha(x,y)$, sowie $(x_2,y_2) = \alpha_2(x,y)$. Somit sind die Variablen x_3, y_3 determiniert durch x_1, y_1, x_2, y_2, welche wiederum determiniert sind durch x, y. Somit können wir $(x_3,y_3) = (f(x,y), g(x,y))$ schreiben. Mit dem obigen Lemma 3.3.5, sowie mit $(u,v) = (x_2,y_2)$, gilt:

$$\frac{\partial f}{\partial x_1} + \frac{\partial f}{\partial y_1}\frac{dy_1}{dx_1} = \frac{y_3}{y_1}.$$

Analog folgt auch:

$$\frac{\partial f}{\partial x_2} + \frac{\partial f}{\partial y_2}\frac{dy_2}{dx_2} = \frac{y_3}{y_2}.$$

Somit gilt nach der Annahme, dass

$$\frac{dx_1}{dx} = c_{\alpha_1}\frac{y_1}{y}, \frac{dx_2}{dx} = c_{\alpha_2}\frac{y_2}{y}.$$

Es gilt außerdem:

$$\frac{df}{dx} = R'_{\alpha_3(x)}, \frac{y}{g(x,y)} = \frac{y}{yS_{\alpha_3}} = \frac{1}{S_{\alpha_3}}.$$

Mit der Kettenregel erhalten wir dann das Ergebnis:

$$\begin{aligned}\frac{df}{dx} &= \frac{\partial f}{\partial x_1}(x,y)\frac{dx_1}{dx} + \frac{\partial f}{\partial y_1}(x,y)\frac{dy_1}{dx_1}\frac{dx_1}{dx} + \frac{\partial f}{\partial x_2}(x,y)\frac{dx_2}{d_x} + \frac{\partial f}{\partial x_2}(x,y)\frac{dy_2}{dx_2}\frac{dx_2}{dx} \\ &= \frac{g(x,y)}{y_1}\frac{y_1}{y}c_{\alpha_1} + \frac{g(x,y)}{y_2}\frac{y_2}{y}c_{\alpha_2} \\ &= (c_{\alpha_1} + c_{\alpha_2})\frac{g(x,y)}{y}.\end{aligned}$$

Teilt man nun durch $S_{\alpha_3} = \frac{y}{g(x,y)}$, erhält man das gewünschte Ergebnis. □

Damit zeigt man nun das folgende Lemma. In [Was08] ist dies die Präposition 2.28.

Lemma 3.3.6
Sei $E(\mathbb{F}_q)$ eine elliptische Kurve. Sei $n \in \mathbb{Z}\backslash\{0\}$, und sei die Multiplikation eines Punktes auf $E(\overline{\mathbb{F}}_q)$ gegeben durch:

$$n(x,y) = (R_n(x), yS_n(x)),$$

mit rationalen Funktionen R_n, S_n. Dann gilt:

$$\frac{R'_n(x)}{S_n(x)} = n.$$

Daraus folgt auch, dass die Multiplikation mit n genau dann separabel ist, wenn $n \nmid p, q = p^k, p$ prim, gilt, also n nicht die Charakteristik von $\mathbb{F}_q$ teilt.

Beweis. Wir müssen dies nur für positive n zeigen. Daraus folgt direkt die Eigenschaft für $n < 0$, denn:

$$R_{-n} = R_n, S_{-n} = -S_n \Rightarrow \frac{R'_{-n}}{S_{-n}} = -\frac{R'_n}{S_n}.$$

Beweis durch Induktion:
Für $n = 1$ gilt: $R_1(x) = 1, S_1(x) = 1 \Rightarrow \frac{R'_1(x)}{S_1(x)} = 1$. Aus Lemma 3.3.4 folgt der Induktionsschritt: $\frac{R'_{n+1}}{S_{n+1}} = \frac{R'_n}{S_n} + \frac{R'_1}{S_1} = \frac{R'_n}{S_n} + 1 = n + 1$,
$\Rightarrow \forall n \in \mathbb{N}$ gilt: $\frac{R'_n(x)}{S_n(x)} = n$. Somit ist die Abbildung separabel, denn es gilt

$$R'_n(x) \neq 0 \Leftrightarrow p \nmid n.$$

□

Damit lässt sich nun der eigentliche Satz 3.3.3 beweisen:
Wir schreiben die Multiplikation eines Punktes (x, y) mit einem Skalar r als:

$$r(x, y) = (R_r(x), yS_r(x))$$

Dann gilt für die Multiplikation eines Punktes mit r Endomorphismen ϕ_q:

$$\begin{aligned}(R_{r\phi_q}(x), yS_{r\phi_q}(x)) = (\phi_q r)(x, y) = (R_r^q(x), y^q S_r^q(x)) \\ = (R_r^q(x), y(x^3 + ax + b)^{\frac{q-1}{2}} S_r^q(x)) \\ \Rightarrow c_{r\phi_q} = \frac{R'_{r\phi_q}}{Sr\phi_q} = qR_r^{q-1}\frac{R'_r}{S_{r\phi_q}} = 0.\end{aligned}$$

Damit gilt $c_s = \frac{R'_s}{S_s} = s$ nach Lemma 3.3.6. Mit Lemma 3.3.4 gilt zusätzlich:

$$\frac{R'_{r\phi_q+s}}{Sr\phi_q + s} = c_{r\phi_q+s} = c_{r\phi_q} + c_s = 0 + s = s.$$

Damit folgt die Behauptung, dass $R'_{r\phi_q+s} \neq 0 \Leftrightarrow p \nmid s$. □

Im dritten Punkt wollen wir nun zeigen, dass die Gleichung

$$\deg(r\phi_q - s) = r^2 q + s^2 - rs(q + 1 - \#E)$$

gilt. Dafür können wir ausnutzen, dass man einen Endomorphismen auf $E[l]$ als 2×2 - Matrix schreiben kann. Für diese haben wir noch folgende Eigenschaften, welche sich aus dem Beweis zu Thm. 4.10 in [Was08] ergeben.

Lemma 3.3.7
Sei A eine 2×2 - Matrix. Dann gilt:

1. $A^2 - \mathrm{Spur}(A)A + \det(A)I = 0$
2. $\mathrm{Spur}(A) = 1 + \det(A) - \det(I - A)$

Dabei ist I die Identitätsmatrix.

Beweis. Der Beweis erfolgt durch Nachrechnen für die allgemeine Matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$:

1. $\begin{pmatrix} a & b \\ c & d \end{pmatrix}^2 - (a + d)\begin{pmatrix} a & b \\ c & d \end{pmatrix} + (ad - bc)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} =$
   $\begin{pmatrix} a^2 + bc & ab + bd \\ ac + cd & bc + d^2 \end{pmatrix} - \begin{pmatrix} a^2 + ad & ab + db \\ ac + dc & ad + d^2 \end{pmatrix} + \begin{pmatrix} ad - bc & ad - bc \\ ad - bc & ad - bc \end{pmatrix} =$
   $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$

2. $1+(ad-bc)-((1-a)(1-d)-bc) = 1+ad-bc-1+a-ad+d+bc = a+d = \mathrm{Spur}(A)$.

□

Nun lässt sich die gewünschte Gleichung beweisen. Bemerke, dass der Grad des Frobenius-Endomorphismus nach Definition den Wert q hat. Das nächste Lemma folgt aus [Was08], Lemma 4.8.

Lemma 3.3.8
Für $r, s \in \mathbb{Z}$ und $\mathrm{ggt}(s, q) = 1$ gilt:

$$\deg(r\phi_q - s) = r^2q - rst + s^2, \text{ mit } t = 1 + q - \#E.$$

Beweis. Sei $l \in \mathbb{Z}$ eine Primzahl mit $\mathrm{ggt}(l, q) = 1$. Nach Lemma 3.3.2 gilt die Äquivalenz „$\det(\alpha) \equiv \deg(\alpha)$“ für Endomorphismen α auf $E[l]$. Wir schreiben den Frobenius-Endomorphismus als Matrix:

$$\phi_q = A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Daraus lässt sich die Behauptung analog zu der Präposition 3.16 in [Was08] folgern:

$$\begin{aligned} \deg(r\phi_q - s) &= \det\left(r \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix} - s \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) \\ &= \det\begin{pmatrix} ar - s & br \\ cr & dr - s \end{pmatrix} \\ &= (ar - s)(dr - s) - bcr^2 = r^2(ad - bc) - rs(a + d) + s^2, \end{aligned}$$

wobei für die Spur einer 2×2 - Matrix gilt:

$$\mathrm{Spur}(A) = 1 + \det(A) - \det(I - A).$$

Somit kann man die Spur $a + d$ der Matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ auch schreiben als:

$$a + d = 1 + (ad - bc) - \det(I - A).$$

Da $\deg(I - A) = \#E(\mathbb{F}_q)$, sowie $\deg(\phi_q) = q$, erhalten wir:

$$\deg(r\phi_q - s) = r^2q - rst + s^2, \text{ mit } t = 1 + q - \#E \tag{3.4}$$

□

Damit lässt sich nun, wie im vierten Punkt beschrieben, der *Satz von Hasse* beweisen, wie auch in [Was08] auf Seite 100:

Satz 3.3.9 (Satz von Hasse)
Für die Anzahl der Punkte einer Elliptischen Kurve $E(\mathbb{F}_q)$ gilt:

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

Beweis. Wir betrachten den Grad des Endomorphismus $r\phi_q - s$. Dieser muss nach Definition immer nicht-negativ sein, somit erhalten wir die Abschätzung:

$$\begin{aligned} \deg(r\phi_q - s) &= r^2 q + s^2 + rs(q + 1 - t - q - 1) \\ &= r^2 q + s^2 - rst \geq 0. \end{aligned}$$

Teilen wir nun durch s^2, und ersetzten den Bruch $\frac{r}{s}$ mit x, erhalten wir die Abschätzung für den quadratischen Term:

$$qx^2 - tx + 1 \geq 0.$$

Die Diskriminante der quadratischen Gleichung ist gegeben durch $t^2 - 4q$. Damit der Term nicht-negativ bleibt, darf dieser höchstens eine Nullstelle besitzen. Folglich muss die Diskriminante ≤ 0 sein, womit wir die Abschätzung

$$t^2 \leq 4q \Leftrightarrow |t| \leq \sqrt{4q} = 2\sqrt{q}$$

erhalten.

□

Daraus folgt ebenfalls eine wichtige Berechnung für die Anzahl der Punkte, die *Spur des Frobenius*. Dies folgt aus Thm. 4.10 und Präposition 4.11 in [Was08].

Korollar 3.3.10
Für den Frobenius-Endomorphismus über $E(\mathbb{F}_q) \cup \mathcal{O}$ gilt:

$$\phi^2 - t\phi + q = 0 \Rightarrow t = q + 1 - \#E(\mathbb{F}_q).$$

Beweis. Wir schreiben den Endomorphismus wieder als 2×2 - Matrix A. Nach Lemma 3.3.7 gilt:

$$A^2 - \mathrm{Spur}(A)A + \det(A)I = 0. \tag{3.5}$$

Dabei gilt für die Spur der Matrix:

$$\mathrm{Spur}(A) = 1 + \det(A) - \det(I - A).$$

Wie im Beweis von 3.3.8, Gleichung (3.4) gezeigt, ist die Spur somit $q + 1 - E(\mathbb{F}_q)$. Setzt man die Spur wieder in Gleichung (3.5) ein, und ersetzt die Matrix mit dem Frobenius-Endomorphismus, so erhält man das gewünschte Ergebnis:

$$\phi^2 - (q + 1 - \#E(\mathbb{F}_q))\phi + q = 0.$$

□

Dies ist nun ausreichend, um den Schoof-Algorithmus für das Punktezählen auf elliptischen Kurven über endlichen Körpern zu beschreiben.

Kapitel 4

Der Schoof-Algorithmus

Wie bereits im Kapitel 2 erörtert, ist es für einige kryptographische Verfahren sehr wichtig, die Gruppenordnung auf elliptischen Kurven berechnen zu können, um geeignete Gruppen für Verschlüsselungsverfahren zu finden. Dazu stellte René Schoof 1985 den ersten Algorithmus mit polynomialer Laufzeit vor [Sch85]. Der nach ihm benannte Schoof-Algorithmus soll in diesem Kapitel vorgestellt werden. Orientierung bietet uns dabei der Abschnitt 4.5 in [Was08].

4.1 Abschätzung für die Anzahl der Punkte

Mit der Spur des Frobenius ist ein erster Anhaltspunkt gegeben, um $\#E(\mathbb{F}_q)$ berechnen zu können. Da es im Allgemeinen kompliziert ist, t aus der Spur des Frobenius berechnen zu können, wird t_l modulo einiger Primzahlen $l \in \{ p_1, \ldots, p_r\}$ mit $l \nmid q$ berechnet. Mithilfe des *chinesischen Restsatzes* lässt sich dann aus $t_{p_1}, \ldots, t_{p_r}$ die gewollte Lösung t mod $\prod_{i=1}^{r} p_r$ berechnen. Ist das Produkt größer als $4\sqrt{q}$, ist dies nach dem Satz von Hasse, 3.3.9, ausreichend, um $\#E(\mathbb{F}_q)$ berechnen zu können, da laut diesem $\mid t \mid \leq 2\sqrt{q}$ gilt. Die Punkte mod p_l sind die *l-Torsionspunkte*, wie in Korollar 3.1.5 beschrieben.

4.2 Formulierung des Schoof-Algorithmus

Wir wollen nun die Spur des Frobenius in den l-Torsionsgruppen bestimmen. Weil die x-Koordinate eines Punktes, der sowohl in der l-Torsionsgruppe $E[l]$, als auch auf der elliptischen Kurve $E(\mathbb{F}_q)$ liegt, eine Nullstelle im l-ten Divisionspolynom Ψ_l ist, können wir die Spur des Frobenius auf Kongruenz zu 0 mod Ψ_l testen. Somit wollen wir im Folgenden möglichst effizient eine

Lösung der Gleichung

$$\phi_q^2(x,y) - t\phi(x,y) + q(x,y) \equiv 0 \pmod{\Psi_l} \tag{4.1}$$

in der x-Koordinate finden. Man bemerke dabei, dass l dazu ungerade sein muss, damit das Polynom Ψ_l nur in x geschrieben werden kann. Deswegen behandeln wir den Fall $l = 2$ gesondert:

Nachdem wir angenommen haben, dass $\mathrm{char}(\mathbb{F}_q) \neq 2, 3$ ist, ist q ungerade, somit gilt $t = -(\#E(\mathbb{F}_q) + q + 1)$. Ebenfalls gilt mit der Spur des Frobenius:

$$t_2 \equiv -(\#E(\mathbb{F}_q) + q + 1) \equiv -\#E(F_q) \pmod{2}.$$

Somit gilt $t_2 = 0$ genau dann, wenn es einen Punkt in $E(\mathbb{F}_q)$ der Ordnung zwei gibt, denn dann muss, gemäß dem *Satz von Lagrange*, 2 die Gruppenordnung von $E(\mathbb{F}_q)$ teilen. Aufgrund der Achsensymmetrie der elliptischen Kurven zur x-Achse muss ein Punkt der Ordnung 2 die Form $(x_0, 0)$ besitzen. Somit existiert so ein Punkt genau dann, wenn die rechte Seite der Weierstraß-Gleichung für $E(\mathbb{F}_q)$ eine Nullstelle in $\mathbb{F}_q$ besitzt, also

$$t_2 = \begin{cases} 0, & \mathrm{ggt}(x^3 + ax + b, x^q - x) \neq 1 \\ 1, & \mathrm{ggt}(x^3 + ax + b, x^q - x) = 1. \end{cases} \tag{4.2}$$

Betrachten wir nun Primzahlen ≥ 3. Wir wollen dabei den ersten und den letzten Teil der Gleichung (4.1) berechnen und daraus die Lösung für t_l ableiten. Dazu betrachten wir im Folgenden nur Punkte, die sowohl in der l-Torsionsgruppe $E[l]$ sind, als auch in der elliptischen Kurve $E(\mathbb{F}_q)$. Wir rechnen also in dem Quotientenring

$$Q = {}^{\mathbb{F}_q[x,y]}\!\diagup_{(\Psi_l, y^2 - x^3 - ax - b)}.$$

Ebenfalls sei $q_l \equiv q \pmod{l}$. Da wir Punkte in $E[l]$ betrachten, macht dies für das Ergebnis keinen Unterschied.

Existiert ein Punkt, für den die beiden Terme nicht gleich sind, also falls

$$\phi_q^2(x_0, y_0) \not\equiv \pm q_l(x_0, y_0)$$

gilt, so ist die Berechnung aufwendiger. Da dies aber der häufigere Fall ist, betrachten wir den Fall als Erstes. Falls jedoch für alle Punkte (x, y) Gleichheit gilt, also

$$\phi_q^2(x, y) \equiv \pm q_l(x, y),$$

so lässt sich die Berechnung vereinfachen. Dies behandeln wir in dem darauf folgenden Abschnitt, 4.2.2. Dazu führen wir die Notation

$$(x_j, y_j) := j(x, y)$$

ein, um die aus der skalaren Multiplikation entstehenden Koordinaten gesondert behandeln können.

4.2.1 Fall 1: $l > 2, \exists(x, y) : (x^{q^2}, y^{q^2}) \neq \pm q_l(x, y)$

Für alle $1 \leq j \leq \frac{l-1}{2}$ überprüft man die Korrektheit der Gleichung

$$(x^{q^2}, y^{q^2}) + q_l(x, y) \equiv j(x^q, y^q) \pmod{\Psi_l}$$

in der x-Koordinate. Man bemerke, dass für Punkte auf der elliptischen Kurve die Gleichheit

$$j(x^q, y^q) = (j(x, y))^q$$

gilt. Zuerst berechnen wir die linke Seite:

$$(X(x, y), Y(x, y)) := (x^{q^2}, y^{q^2}) + q_l(x, y).$$

Dabei ist $X(x, y)$ ein Polynom nur in der Variable x:
Aus der Formel für die Addition auf elliptischen Kurven erhalten wir:

$$X(x, y) = (\frac{y^{q^2} - y_{q_l}}{x^{q^2} - x_{q_l}})^2 - x^{q^2} - x_{q_l}.$$

Somit müssen wir noch zeigen, dass die beiden Funktionen $(y^{q^2} - y_{q_l})$, sowie x_{q_l} Funktionen nur in x sind. Bei x_{q_l} ergibt sich dies aus der Symmetrie der elliptischen Kurve zur y-Achse, sowie der geometrischen Anschauung der Addition auf elliptischen Kurven. Für den Term $(y^{q^2} - y_{q_l})$ betrachten wir die kurze Weierstraß-Gleichung, $y^2 = x^3 + ax + b$, sowie die Formeln für das skalare Multiplizieren auf elliptischen Kurven aus dem Satz 3.1.4. Aus letzteren erhalten wir:

$$y_{q_l} = \frac{\Psi_{2q_l}}{2\Psi_{q_l}^4}.$$

Dabei ist q_l gerade, den es gilt $q_l \equiv q \bmod l$, mit ungeraden q, l. Mit Lemma 3.1.2 folgt:

$$\begin{aligned} &\Psi_{2q_l} \in y\mathbb{F}_q[x, a, b], \\ &\Psi_{q_l}^4 \in y^4\mathbb{F}[x, a, b] \Rightarrow \Psi_{q_l}^4 \in \mathbb{F}[x, a, b]. \end{aligned}$$

Ebenfalls können wir den Term umschreiben:

$$y^{q^2} - y_{q_l} = y^2(y^{q^2-1} - \frac{y_{q_l}}{y})^2.$$

Dies sind alles Terme, in denen y nur gerade Exponenten besitzt, wobei bei dem hinteren Bruch das y gekürzt wird. Damit lassen sich alle y durch Einsetzen der kurzen Weierstraß-Gleichung eliminieren und im Schoof-Algorithmus auf die Gleichheit von

$$X(x) \equiv x_j^q$$

in dem besprochenen Quotientenring Q testen. Falls für ein j Gleichheit erreicht wird, können wir dann mithilfe der y-Koordinate das Vorzeichen bestimmen.

Die y-Koordinate $Y(x, y)$ lässt sich ebenfalls mit der bekannten Formel für die Addition auf elliptischen Kurven berechnen.
Damit lässt sich dann für jedes j explizit überprüfen, ob $X(x, y) \equiv jx^q \bmod \Psi_l(x)$ gilt. Erfüllt ein $\pm t_l := j$ diese Gleichheit, muss man nur noch das Vorzeichen von t_l bestimmen. Dabei vergleicht man die y-Koordinaten: Gilt $Y(x, y) = jy^q$, so ist das Vorzeichen von t_l positiv, ansonsten negativ.

4.2.2 Fall 2: $l > 2, \forall(x, y) : (x^{q^2}, y^{q^2}) = \pm q_l(x, y)$

Nehmen wir zuerst an, dass $(x^{q^2}, y^{q^2}) = +q_l(x, y)$ gilt. Dann vereinfacht sich die Spur des Frobenius, Gleichung (4.1), auf die Form:

$$2q_l(x, y) + t\phi(x, y) = 0.$$

Somit sind wir an der Lösung j für die folgende Gleichung interessiert:

$$2q_l(x, y) = j(x^q, y^q) = j\phi(x, y).$$

Betrachten wir q_l, bemerken wir, dass für $q_l \neq 0$ die Ungleichheit $q_l \neq -q_l \pmod{l}$ gelten muss, da q_l gerade und l eine ungerade Zahl ist, die q_l nicht teilt. Daraus folgt $t_l \neq 0$, sonst hätte die Gleichung keine Lösung. Quadriert man die Gleichung, setzt daraufhin die obige Gleichheit wieder ein und dividiert durch q_l, erhält man:

$$(2q_l)^2(x, y) = j^2\phi^2(x, y) = j^2(x^{q^2}, y^{q^2}) = j^2 q_l(x, y) \pmod{l}$$

$$\Leftrightarrow j^2 \equiv 4q_l(\bmod l).$$

Da $4 = 2^2 \bmod l$ für alle $l \geq 2$ gilt, stimmt die erste Annahme genau dann, wenn q_l eine Quadratzahl mod l ist. Mit dem folgenden Satz, wie in [IR13], Präposition 4.2.1 beschrieben, kann man dies direkt bestimmen:

Satz 4.2.1
Für $a \in \mathbb{Z}/q$, $q = p^k, p$ prim, mit ggt(a, q)= 1, gilt:

$$\exists x \in \mathbb{Z}/q : x^2 \equiv a \bmod q \Leftrightarrow a^{\frac{\phi(q)}{2}} \equiv 1 \bmod q,$$

wobei $\phi(q) := \#\mathbb{F}_q^* = p(p^{k-1} - \frac{1}{p})$ die *Euler'sche ϕ-Funktion* darstellt.

Beweis.
" $\Rightarrow$ " : Aus dem *Satz von Euler* folgt: $a^{\phi(q)} \equiv 1 \bmod q$ für alle $a \in \mathbb{Z}/q^*$. Sei nun $a \equiv x^2 \bmod q$, dann gilt:

$$a^{\frac{\phi(q)}{2}} \equiv x^{2^{\frac{\phi(q)}{2}}} \equiv x^{\phi(q)} \equiv 1 \bmod q.$$

" $\Leftarrow$ " : Sei nun $a^{\frac{\phi(q)}{2}} \equiv 1 \bmod q$. $\mathbb{F}_q^*$ ist zyklisch, mit dem Erzeuger g. Schreibe also $a := g^b$. Dann gilt (mod q):

$$a^{\frac{\phi(q)}{2}} \equiv g^{b^{\frac{\phi(q)}{2}}} \equiv g^{\frac{b\phi(q)}{2}} \Rightarrow 2 \mid b \text{ oder } \phi(q) \mid \frac{b\phi(q)}{2}.$$

Falls $2 \mid b$, so kann man $b = 2k$ schreiben und es gilt $a \equiv g^b \equiv g^{2k} \equiv (g^k)^2$. Im anderen Fall gilt $\phi(q) \mid \frac{b\phi(q)}{2} \Leftrightarrow 2\phi(q) \mid b\phi(q) \Leftrightarrow 2 \mid b$. Somit geht dies in den ersten Fall über. □

In Magma ist jedoch bereits eine Funktion implementiert, die überprüft, ob eine Zahl ein Quadrat modulo einer Primzahl ist. Diese nutzen wir später für die Implementierung.

Falls q_l eine Quadratzahl mod l ist, so existiert ein $w \in \mathbb{Z}/l$, sodass

$$q_l \equiv w^2 \pmod{l}$$

gilt. Dabei lässt sich w als Wurzel modulo einer Primzahl explizit berechnen, wie zum Beispiel in [Coh13], Abschnitt 1.5 gezeigt wird. Auch in [Wät08], Kapitel 9.2, wird ein Algorithmus dazu beschrieben. In Magma ist ein solcher Algorithmus ebenfalls bereits implementiert, sodass wir diesen nutzen können. Mit der Gleichung $j^2 = 4q_l$ erhält man dann

$$\pm t_l := j = \pm 2w.$$

Das Vorzeichen bestimmt man wieder über die y-Koordinate:

- $w(x^q, y^q) = (x^{q^2}, y^{q^2}) \Rightarrow t_l = 2w$
- $w(x^q, y^q) = (x^{q^2}, -y^{q^2}) \Rightarrow t_l = -2w$

Um die Berechnungen durchführen zu können, müssen wir erst feststellen, ob wir in diesem Fall sind, also ob die folgende Gleichheit für die berechnete Wurzel $w = \sqrt{q_l}$ gilt:

$$(x^q, y^q) = \pm w(x, y) = (x_w, \pm y_w). \tag{4.3}$$

Dabei reicht es hier nicht aus, dies auf Kongruenz zu 0 modulo Ψ_l zu testen, da die Gleichheit nicht für alle Punkte in $E[l]$ gelten muss. Ein Beispiel

hierfür ist die Kurve $E(\mathbb{F}_{1997})$, gegeben durch $y^2 = x^3 + 46x + 74$. Dort verschwindet die Spur des Frobenius nicht modulo $l = 7$, obwohl die besprochene Gleichheit aus Fall 1 nicht für alle Punkt erfüllt ist. Dies wird hier nicht weiter ausgeführt, kann aber mit dem implementierten Algorithmus überprüft werden. Wir betrachten das Problem aber theoretisch.
Schreibt man die Gleichung in Matrixform, beschreibt $\pm w$ genau den Eigenwert zu dem Vektor (x, y). Ist der Frobenius-Endomorphismus durch eine Matrix A gegeben, die einen eindimensionalen Eigenraum besitzt, so existieren Punkte (x, y), sodass $A(x, y) \neq \pm w(x, y)$ gilt. Deswegen überprüfen wir, ob der Term

$$(x^q, y^q) - (x_w, \pm y_w)$$

Nullstellen in $E[l]$ besitzt.

Betrachten wir zuerst die x-Koordinate. Es bietet sich an, mithilfe des *Euklidischen Algorithmus* zu überprüfen, ob die beiden Terme $x^q - x_w$ und Ψ_l gemeinsame Nullstellen besitzen. Existiert eine solche, dann existiert ein Punkt, für den die gewünschte Gleichheit aus der Gleichung (4.3) erfüllt ist. Wir können den *Euklidischen Algorithmus* nur für Polynome anwenden, jedoch ist der Term x_w nach den Rechenregeln für die skalare Multiplikation, siehe Satz 3.1.4, durch rationale Funktionen gegeben. Für die Nullstellen von rationalen Funktionen reicht es, nur den Zähler zu betrachten. Deswegen bringen wir $x^q - x_w$ auf einen Nenner, und betrachten daraufhin den Zähler. Bezeichnen wir diesen als $D_x(x, y)$. Ebenfalls wollen wir mithilfe der kurzen Weierstraß-Gleichung den Zähler als ein Polynom in x schreiben, da Ψ_l auch ein Polynom in l ist. Dazu berechnen wir:

$$x^q - x_w = x^q - \frac{x\Psi_w^2 - \Psi_{w+1}\Psi_{w-1}}{\Psi_w^2}.$$

Erweitert man x^q um Ψ_w^2, ergibt sich für $D_x(x)$:

$$D_x(x, y) = x^q\Psi_w^2 - x\Psi_w^2 - \Psi_{w+1}\Psi_{w-1} = (x^q - x)\Psi_w^2 - \Psi_{w+1}\Psi_{w-1}.$$

Wir sehen, dass sowohl für w gerade als auch ungerade, alle y in diesem Term einen geraden Exponenten besitzen müssen. Nehmen wir an, dass die Divisionspolynome in der Form aus Lemma 3.1.2 gegeben sind. In unserer Implementierung in Magma werden wir die Polynome auch direkt so berechnen, siehe den Algorithmus im Abschnitt A.1.2. Somit ist 2 der größte vorkommende Exponent für y. Um nun y^2 mit $x^3 + ax + b$ zu ersetzen, unterscheiden wir zwischen w gerade und ungerade.

Ist w gerade, so multiplizieren wir $\frac{x^3+ax+b}{y^2}$ an den Term Ψ_w^2, um das y mit $x^3 + ax + b$ zu ersetzen.

Ist w ungerade, so multiplizieren wir den Bruch $\frac{x^3+ax+b}{y^2}$ an den Term $\Psi_{w+1}\Psi_{w-1}$. Somit erhalten wir:

$$D_x(x) = \begin{cases} (x^q - x)\Psi_w^2 \frac{x^3+ax+b}{y^2} - \Psi_{w+1}\Psi_{w-1}, & w \equiv 0 \pmod 2 \\ (x^q - x)\Psi_w^2 - \Psi_{w+1}\Psi_{w-1}\frac{x^3+ax+b}{y^2}, & w \equiv 1 \pmod 2. \end{cases}$$

Nun können wir die Existenz einer gemeinsamen Nullstelle von $D(x)$ und $\Psi_l(x)$ mit dem *Euklidischen Algorithmus* überprüfen. Liefert dieser ein Ergebnis ungleich 1, so gibt es Punkte, die Gleichung (4.3) erfüllen. Dann bestimmen wir das Vorzeichen von w über die y-Koordinate. Dafür prüfen wir wieder, ob Punkte existieren, die für $+w$ die Gleichung (4.3) erfüllen, also eine gemeinsame Nullstelle von $y^q - y_w$ und Ψ_l sind. Dazu bringen wir $y^q - y_w$ auf einen Nenner und betrachten wie oben den Zähler des entstehenden Bruches. Bezeichnen wir diesen als $D_y(x)$ und schreiben ihn als eine Funktion in x. Dazu berechnen wir, mit Satz 3.1.4:

$$y^q - y_w = y^q - \frac{\Psi_{w+2}\Psi_{w-1}^2 - \Psi_{w-2}\Psi_{w+1}^2}{4y\Psi_w^3}.$$

Erweitert man y^q um $4y\Psi_w^3$ und kürzt den Bruch um y, so erhält man für $D_y(x)$:

$$D_y(x,y) = 4y^q\Psi_w^3 - \frac{\Psi_{w+2}\Psi_{w-1}^2 - \Psi_{w-2}\Psi_{w+1}^2}{y}.$$

Wie oben nutzen wir die kurze Weierstraß-Gleichung, um $D_y(x,y)$ nur in x zu schreiben. Somit erhalten wir:

$$D_y(x) = \begin{cases} 4(x^3+ax+b)^{\frac{q-3}{2}}\frac{\Psi_w^3}{y^3} - \frac{\Psi_{w+2}}{y}\Psi_{w-1}^2 - \frac{\Psi_{w-2}}{2}\Psi_{w+1}^2, & w \equiv 0 \pmod 2 \\ 4y^{q-1}\Psi_w^3 - (\Psi_{w+2}\frac{\Psi_{w-1}^2}{y^2} - \Psi_{w-2}\frac{\Psi_{w+1}^2}{y^2})(x^3+ax+b), & w \equiv 1 \pmod 2. \end{cases}$$

Um nicht unnötig große Polynome zu vergleichen, berechnen wir, wie auch im Abschnitt 5.2 in [Mue91] beschrieben wird, den euklidischen Algorithmus nur für die beiden Argumente $D_x(x)$ mod Ψ_l und Ψ_l. Diese haben die selben gemeinsamen Nullstellen wie $D_x(x)$ und Ψ_l. Dasselbe gilt für $D_y(x)$.

Tritt keiner der oben genannten Fälle ein, ist die erste Annahme falsch. Folglich gilt $(x^{q^2}, y^{q^2}) = -q_l(x,y)$. Setzt man dies in die Spur des Frobenius ein und lässt diese verschwinden, erhält man:

$$\begin{aligned}(x^{q^2}, y^{q^2}) + j(x^q, y^q) - q_l(x,y) &= (x^{q^2}, y^{q^2}) + j(x^q, y^q) - (x^{q^2}, y^{q^2}) \\ &= j(x^q, y^q) = 0 \\ &\Rightarrow j = 0.\end{aligned}$$

Es folgt: $t_l := j = 0$.

4.2.3 Der restliche Algorithmus

Hat man die obigen Schritte für alle r Primzahlen ausgeführt, berechnet man mithilfe des nachfolgenden Satzes, dem *chinesischen Restsatz* für ganze Zahlen, die Spur des Frobenius t aus dem Kongruenzsystem $t \equiv t_l \bmod p_l$, $1 \leq l \leq r$. Dafür benötigen wir zuerst noch den erweiterten *Euklidischen Algorithmus*. Mit diesen berechnet man für zwei ganze Zahlen a, b den größten gemeinsamen Teiler der beiden, sowie die *Bezout-Koeffizienten*, also zwei ganze Zahlen s, t, sodass $\text{ggt}(a, b) = sa + tb$. Den Algorithmus findet man zum Beispiel in [Coh13], Algorithmus 1.3.6. Den chinesischen Restsatz findet man in [Coh13], Algorithmus 1.3.12.

Algorithmus 1 Euklidischer Algoritmus

1: $(r_1, s_1, t_1) \leftarrow (a, 1, 0)$
2: $(r_2, s_2, t_2) \leftarrow (b, 0, 1)$
3: $k \leftarrow 2$
4: **while** $r_k \neq 0$ **do**
5: $\quad q_k \leftarrow r_{k-1}$ div r_k
6: $\quad (r_{k+1}, s_{k+1}, t_{k+1}) \leftarrow (r_{k-1} - q_k r_k, s_{k-1} - q_k s_k, t_{k-1} - q_k t_k)$
7: $\quad k \leftarrow k + 1$
return $(r_{k-1}, s_{k-1}, t_{k-1})$

Dabei beschreibt „div“ die ganzzahlige Division.

Beweis. Der Beweis erfolgt durch Induktion:
Wir wollen verifizieren, dass $r_k = s_k a + t_k b$ in jedem Schritt gilt. Setzen wir die Werte aus dem Algorithmus ein, erhalten wir in jedem Schritt die Gleichung:

$$r_{k-1} - q_k r_k = (s_{k-1} - q_k s_k)a + (t_{k-1} - q_k t_k)b.$$

Als Induktionsvoraussetzung nehmen wir an, dass

$$r_{k-1} = s_{k-1} a + t_{k-1} b$$

gilt. Subtrahiert man dies aus der Gleichung, erhält man

$$-q_k r_k = -q_k s_k a + -q_k t_k b.$$

Dividieren durch $(-q_k)$ bringt das gewünschte Ergebnis.
Der Algorithmus terminiert, da für alle k gilt:

$$r_{k+1} = r_{k-1} \bmod r_k \Rightarrow r_{k+1} < r_k \text{ für } r_k \neq 0.$$

Gilt $r_k = 0$, ist der Algorithmus bereits terminiert.

□

Satz 4.2.2 (Chinesischer Restsatz auf den ganzen Zahlen)
Seien $p_1, \ldots, p_r$ paarweise teilerfremde ganze Zahlen. Dann besitzt das folgende Kongruenzsystem für $x, t_1, \ldots, t_r \in \mathbb{Z}$ genau eine Lösung modulo $\prod_{i=1}^{r} p_i$:

$$x \equiv t_i \bmod p_i.$$

Beweis. Wir zeigen dies für den Fall $r = 2$. Für größere Werte von r kann man das Kongruenzsystem iterativ lösen, indem man die ersten zwei Kongruenzen löst, und deren Lösung mit der dritten Kongruenz löst, usw.
Mit dem *euklidischen Algorithmus* erhält man ganze Zahlen a, b, sodass $ap_1 + bp_2 = \text{ggt}(p_1, p_2) = 1$ gilt. Die Lösung x lässt sich nun mit der folgenden Formel berechnen:

$$x \equiv bt_1p_1 + at_2p_2 \bmod p_1p_2$$

Dies wird verifiziert durch:

$$t_1p_1 \equiv 1 \bmod p_2 \Rightarrow bt_1p_1 + at_2p_2 \equiv bt_1p_1 \equiv b(t_1p_1) \equiv b \bmod p_2$$

Genauso erhält man:

$$t_2p_2 \equiv 1 \bmod p_1 \Rightarrow bt_1p_1 + at_2p_2 \equiv at_2p_2 \equiv a(t_2p_2) \equiv a \bmod p_1$$

Die Lösung ist eindeutig, da für eine weitere Lösung x' gelten muss:

$$\begin{aligned} x - x' &\equiv 0 \bmod p_1, \\ x - x' &\equiv 0 \bmod p_2 \\ \Rightarrow x - x' &\equiv 0 \bmod p_1p_2. \end{aligned}$$

Somit erhalten wir auch gleich eine praktische Formel zur Implementierung im Schoof-Algorithmus. Nach jedem errechneten t_l erhalten wir ein neues Kongruenzsystem, bestehend aus der bisherigen Lösung und der neu hinzugewonnen Kongruenz $t_l \equiv t \bmod l$. Damit können wir das gesamte Kongruenzsystem iterativ lösen. □

Die Spur des Frobenius t ist dann eindeutig in dem Intervall $\mid t \mid \leq \frac{\prod_{k=1}^{r} p_k}{2}$ gegeben.
Zusammengefasst ergibt sich der Schoof-Algorithmus:

Algorithmus 2 Der Schoof-Algorithmus

Input: **Elliptische Kurve** $E[x,y] := x^3 + ax + b - y^2$**, Endlicher Körper** $\mathbb{F}_q$

Output: Gruppenordung $\#E(\mathbb{F}_q)$

Bestimme ungerade Primzahlen $p_1 \dots p_r$, $p_l \nmid q$, sodass $\prod_{l=1}^{r} p_r > 4\sqrt{q}$
if $\text{ggt}(x^q - x, x^3 - ax - b) \neq 1$ **then**
    $t_2 \leftarrow 0$
else
    $t_2 \leftarrow 1$
for $l := 1$ to r **do**
    $q_l \leftarrow q \bmod p_l$
    Berechne das l-te Divisionspolynom Ψ_l
Die weiteren Berechnungen der Schleife werden im Ring $\mathbb{F}_q[x,y]\backslash(E[x,y], \Psi_l)$ ausgeführt:
    Berechne $(x^q, y^q), (x^{q^2}, y^{q^2}), (x_{q_l}, y_{q_l}) := q_l(x,y)$
    if $x^{q^2} \neq x_{q_l}$ **then**
        $(x', y') \leftarrow (x^{q^2}, y^{q^2}) + (x_{q_l}, y_{q_l})$
        for $1 \leq j \leq \frac{p_l - 1}{2}$ **do**
            if $x' = jx^q$ **then**
                if $y' = jy^q$ **then**
                $t_l \leftarrow j$
                else
                $t_l \leftarrow -j$
    else
        if q_l ist Quadrat mod p_l **then**
            $w \leftarrow \sqrt{(}q_l) \bmod p_l$
            if ggt(Zähler$(x^q - x_w)$,$\Psi_l) = 1$ **then**
                $t_l \leftarrow 0$
            else
                if ggt(Zähler$(y^q - y_w)$,$\Psi_l) = 1$ **then**
                $t_l \leftarrow -2w$
                else
                $t_l \leftarrow 2w$
Berechne t mit dem chinesischen Restsatz aus $t_{p_1} \dots t_{p_r}, \mid t \mid \leq \frac{\prod_{k=1}^{r} p_k}{2}$.
return $\#E(\mathbb{F}_q) := q + 1 - t$

Kapitel 5

Aufwand für den Schoof-Algorithmus

Zuerst wollen wir eine Notation für die Obere Schranke der Laufzeit einführen, die sogenannte $O-$Notation. Diese wurde 1894 von Paul Bachmann eingeführt, siehe [Bac94].

Definition 5.0.1
Seien f, g Folgen reeller Zahlen, dann ist $f \in O(g)$, wenn $\limsup_{x \to 0} \left| \frac{f(x)}{g(x)} \right| < \infty$.

Die Folge $f(x)$ hat also die Größe $O(g)$, wenn sie kleiner ist als g oder nur um einen konstanten Faktor größer wird.

5.1 Laufzeit

Die Wahl der Primzahlen ist entscheidend für die Laufzeit. Sind diese sehr groß gewählt, muss man größere und mehr Divisionspolynome berechnen, was zu wesentlich längerer Laufzeit und mehr Speicherbedarf führt. Deswegen versucht man die Primzahlen möglichst klein zu wählen. Um die notwendige Anzahl und Größe der Primzahlen zu bestimmen, helfen die folgenden zwei Beobachtungen. Zuerst bestimmen wir die Größe der größten zu verwendenden Primzahl. Damit gibt uns der *Primzahlsatz* eine Abschätzung für die Anzahl der zu verwendenden Primzahlen an. Vergleiche dazu auch [Mue91], Satz 6.3. Wir werden für die Laufzeit den Algorithmus verwenden, um die Anzahl der Ziffern einer Zahl zu bestimmen. Da der Algorithmus auf Computern mit binärer Arithmetik implementiert werden soll, verwenden wir hierzu den dualen Logarithmus.

Lemma 5.1.1
Die größte im Schoof-Algorithmus zu verwendende Primzahl ist in der Größenordnung $O(\log(q))$.

Beweis. Dazu definieren wir, wie in der Einleitung zu [RS62], die folgende Funktion:

$$\theta(x) := \log(\Pi_{p \leq x} p)$$

Für die Laufzeit betrachten wir vor allem sehr große Körper $\mathbb{F}_q$, auf der wir die elliptische Kurve definieren. Insbesondere gilt also $q \geq 41$. Somit folgt mit der Abschätzung (3.16) in [RS62]:

$$\theta(x) > x(1 - \frac{1}{\log x}), \text{ für} x \geq 41.$$

Damit erhalten wir:

$$\log(\Pi_{i=1}^{k} p_i) > p_k \cdot (1 - \frac{1}{\log p_k})$$

Wählt man p_k größer als $2\log(4\sqrt{q})$ und nützt man die Ungleichung

$$1 - \frac{1}{\log(p_k)} > \frac{1}{2} \text{ für } k \geq 3,$$

erhält man:

$$\begin{aligned}
\log(\Pi_{i=1}^{k} p_i) &> p_k \cdot (1 - \frac{1}{\log(p_k)}) \\
&> 2\log(4\sqrt{q}) - \frac{2\log(4\sqrt{q})}{\log(2\log(4\sqrt{q}))} \\
&> 2\log(4\sqrt{q}) - \log(4\sqrt{q}) \\
&= \log(4\sqrt{q}).
\end{aligned}$$

Insbesondere, da der Logarithmus streng monoton steigend ist, erhält man also die Abschätzung über die Argumente:

$$\Pi_{i=1}^{k} p_i > 4\sqrt{q}.$$

Dies ist die gewollte Abschätzung, die wir erreicht haben, indem wir p_k in der Größe $O(\log(q))$ gewählt haben. Somit ist das Lemma bewiesen. □

Satz 5.1.2
Gebe die Funktion $\pi : \mathbb{N} \to \mathbb{N}, n \mapsto \pi(n)$ die Anzahl der Primzahlen $\leq n$ an. Dann gilt:

$$log(n) \cdot \pi(n) \sim n$$

Beweis. Der Beweis dazu ist für diese Arbeit zu umfangreich, dafür sei auf Literatur verwiesen. Ein elementarer Beweis wird in der Bachelorarbeit von Jonas Oechsner, [Öc11], erörtert. Die erste Anleitung für einen Beweis hatte jedoch Bernhard Riemann in der Funktionentheorie gefunden. 1896 wurde der Satz von Jacques Hadamard und Charles-Jean de La Valle Poussin der Primzahlsatz auf zwei unterschiedliche Arten vollständig bewiesen. Eine Vereinfachung der Beweise dazu findet man zum Beispiel in dem Buch [Apo98], Kapitel 13. □

Damit erhalten wir nun das folgende Korollar:

Korollar 5.1.3
Es müssen im Schoof-Algorithmus insgesamt bis zu $O(\log(q))$ Primzahlen der Größe von bis zu $O(\log(q))$ verwendet werden.

Beweis. Nach Lemma 5.1.1 ist die größte zu verwendende Primzahl in der Größenordnung $\log(q)$. Sei dies die k-te Primzahl. Dann folgt mit dem Primzahlsatz, Satz 5.1.2, für die Anzahl der Primzahlen kleiner als p_k, $\pi(p_k)$:

$$p_k \sim k \cdot \pi(p_k) \sim k \cdot log(q).$$

Damit ist die Anzahl der zu verwendenden Primzahlen im Schoof-Algorithmus ebenfalls $O(\log(q))$. □

Nun betrachten wir die Komplexität der im Algorithmus vorkommenden Berechnungen. Vergleiche dazu [Mue91], Kapitel 6. Die Laufzeit der Polynomdivision betrachten wir im folgenden Lemma. Die Laufzeit der Multiplikation und Addition von Polynomen folgen aus [Mue91], Satz 6.1. Die Größe der Divisionspolynome ist durch Lemma 3.1.6 gegeben.

Die Berechnungen werden in einem Quotientenring durchgeführt, in dem die Polynome modulo einem Polynom betrachtet werden. Dazu müssen wir den Rest modulo einem Polynom berechnen.

Lemma 5.1.4
Den Rest von einem Polynom mit Grad n durch ein Polynom mit grad m in einem Körper $\mathbb{F}_q$ zu berechnen benötigt mit $n > m + 1$ insgesamt $O((n - m + 1)\log^2(q))$ Bit-Operationen.

Beweis. Dies folgt aus dem *Divisionsalgorithmus* für Polynome. Man muss $n - m + 1$ Divisionen durchführen. Dabei sind die Koeffizienten jeweils aus $\mathbb{F}_q$, somit erfordert jede Division $\log^2(q)$ Operationen. □

Damit können wir jetzt die Anzahl der Berechnungen im Schoof-Algorithmus nachvollziehen:

- Berechnung der Divisionspolynome:
  Da die größte zu verwendende Primzahl in der Größenordnung $O(\log(q))$ ist, müssen wir $O(\log(q))$ verschiedene Divisionspolynome berechnen. Nach den Formeln aus Definition 3.1.1 für die Berechnung der Divisionspolynome benötigt dies nur Multiplikationen und Additionen. Bezeichnet d den größten Grad der Divisionspolynome, benötigen wir dazu $O(d^2\log^2(q))$ Operationen. Die größten Divisionspolynome sind vom Grad $O(l^2) = O(\log^2(q))$, somit benötigt die Berechnung eines Divisionspolynoms $\log^6(q)$ Bit-Operationen. Für $\log(q)$ Divisionspolynome ergibt dies insgesamt $\log^7(q)$ Bit-Operationen, um alle Divisionspolynome zu berechnen.

- Berechnung von x^q, x^{q^2} für jede Primzahl l in dem Ring

  $$R = \mathbb{F}_q[x,y]/(\Psi_l, y^2 - x^3 - ax - b) :$$

  Die Berechnung von x^{q^2} ist komplizierter. Man muss den Rest der Division von x^q durch jeweils Ψ_l und $y^2 + x^3 + ax + b$ berechnen. Dabei dominiert eindeutig die Division durch Ψ_l, da $\deg(\Psi_l) = \frac{l^2-1}{2}$. Aus dem obigen Lemma 5.1.4 folgt, dass $O(q^2 - \frac{l^2-1}{2} + 1)\log^2(q)$ Operationen benötigt werden. Da l auch die Länge $\log(q)$ besitzt, sind dies insgesamt $O(\log^6(q))$ Operationen. Dies führt man für insgesamt $\log(q)$ Divisionspolynome durch, erhält also insgesamt $O(\log^7(q))$ Operationen. Diese werden einmal berechnet und anschließend im gesamten Algorithmus verwendet.

- Test, ob $(x', y') = (x^{q^2}, y^{q^2}) + q_l(x,y) = jx^q$ für $1 \leq j \leq \frac{l-1}{2}$:
  Nach den Regeln der Addition von Punkten auf elliptischen Kurven erfordert dies Multiplikationen im Ring

  $$R := \mathbb{F}_q[x,y]/(y^2 - x^3 - ax - b, \Psi_l(x,y)).$$

  Multiplikation einer Zahl mit n Bits benötigt ohne Nutzung schneller Arithmetik n^2 Bit-Operationen. Dabei gilt $\deg(\Psi_l) = \frac{l^2-1}{2}$, somit haben die Elemente in R den Grad $O(l^2)$, mit Koeffizienten in $\mathbb{F}_q$. Diese besitzen die Länge $\log(q)$, damit hat ein Element in R die Größe $O(l^2\log(q))$. Somit benötigt es $O(l^2\log(q))^2$ Bit-Operationen für die Addition.
  Dies wird nun für bis zu $\log(q)$ Primzahlen berechnet. Für jede dieser Primzahlen werden dann bis zu $\frac{l-1}{2}$ Tests durchgeführt.
  Dies ist die aufwendigste Berechnung im Algorithmus. Man führt dies für insgesamt $\log(q)$ Primzahlen der Länge $\log(q)$ durch und erreicht somit eine Komplexität von $O(\log^8(q))$.

- Berechnung des größten gemeinsamen Teilers mit dem Euklidischen Algorithmus:
  Dieser benötigt für Polynome vom Grad $O(n)$ insgesamt $O(n^2 \log(q)^2)$ Operationen. Wir berechnen den größten gemeinsamen Teiler für Polynome von Grad $O(l^2)$ für die Primzahlen l in der inneren Schleife des Algorithmus. Somit benötigt das Berechnen insgesamt $O(\log^3(q))$ Operationen.

- Der restliche Algorithmus wird von den obigen Berechnungen stark dominiert:

  - Primzahlen finden: Man benötigt insgesamt $O(\log(q))$ Primzahlen. In der Praxis wird $\log(q)$ in etwa 250 sein, sodass man nur die ersten 250 Primzahlen berechnen muss. Diese kann man zum Beispiel mit den *Sieb des Atkin* bestimmen. Dies ist ein effizientes Verfahren, um alle Primzahlen bis zu einer gegebenen Größe n zu berechnen, das das *Sieb des Eratosthenes* optimiert. Es benötigt eine Laufzeit von $O(\log(n))$. Das Verfahren wird hier nicht vorgestellt, dafür wird auf [AB04] verwiesen.
  - Das Berechnen einer Wurzel modulo einer Primzahl:
    Berechnet man dies primitiv, muss man für jede Primzahl l insgesamt bis zu $\frac{l-1}{2}$ verschiedene Zahlen quadrieren, und auf Gleichheit mit $j \bmod l$ testen. Division mit Rest erfordert $O(\log^2(q))$ Operationen, das Quadrieren von Zahlen ebenfalls, somit erfordert dies insgesamt für $\log(q)$ verschiedene Zahlen $O(\log^3(q))$ Bit-Operationen.
  - Berechnungen für den chinesischen Restsatz: Dabei löst man $\log(q)$ Kongruenzsysteme mit je zwei Kongruenzen. In diesem berechnet man jeweils mit dem euklidischen Algorithmus die Bezout-Koeffizienten und führt daraufhin Multiplikationen und Additionen mit Zahlen der Größe $O(\log(q))$ durch. Insgesamt benötigt dies also $O(\log^2(q))$ Operationen.
  - Test, ob eine Zahl eine Quadratzahl modulo einer Primzahl ist: Dazu berechnet man nach Satz 4.2.1 $a^{\frac{\phi(q)}{2}} \pmod q$. Dies benötigt insgesamt $O(\log^3(q))$ Operationen, siehe [Yan07], Thm. 1.3.13.

Das Berechnen der Divisionspolynome und das Multiplizieren benötigt also die meisten Berechnungen. Somit erreicht man für den Algorithmus ohne schnelle Arithmetik eine Laufzeit von $O(\log^8(q))$.
Mit schneller Arithmetik lässt sich eine Komplexität von $O(\log^5(q))$ erreichen.

Bemerkenswerte Beschleunigung wird aber erst durch den Schoof-Elkies-Atkin Algorithmus erreicht, siehe [Dew98]. Dabei werden nur bestimmte Primzahlen verwendet, die die Berechnungen der Divisionspolynome vereinfachen. Damit erreicht man eine Laufzeit von $O(\log^3(q))$.

5.2 Speicherbedarf

Der meiste Speicher wird von den Divisionspolynomen benötigt. Wie weiter oben bemerkt, müssen wir $O(\log(q))$ Divisionspolynome berechnen. Für jedes Divisionspolynom Ψ_l müssen wir l^2 Koeffizienten abspeichern. Jeder Koeffizient ist aus dem Körper $\mathbb{F}_q$, folglich mit einem Speicherbedarf von je $\log(q)$ Bits. Da die Primzahlen l bis zu $O(\log(q))$ groß sind, benötigen wir für jedes Divisionspolynom Ψ_l einen Speicherbedarf von $O(\log^2(q) \cdot \log(q)) = O(\log^3(q))$. Für alle $\log(q)$ Divisionspolynomen ergibt dies einen Speicherbedarf von $O(\log^4(q))$.
Ansonsten müssen nur konstante Größen abgespeichert werden, also benötigt der Schoof-Algorithmus insgesamt einen Speicherbedarf von $O(\log^4(q))$ Bits.

5.3 Laufzeittest

In der Tabelle 5.1 ist die Laufzeit des Schoof-Algorithmus für zwei verschiedene elliptische Kurven über 35 verschiedenen Primzahlkörpern beschrieben. Man erkennt dabei, dass die Laufzeit vor allem von den Primzahlen abhängt. Die Berechnungen wurden an den von Magma zu Verfügung gestellten online-Calculator ausgeführt. Diesen findet man unter `http://magma.maths.usyd.edu.au/calc/`. Dieser benötigt bereits für Primzahlen in der Größenordnung von 2^{20} eine Laufzeit von circa sieben Sekunden. Der effizientere in Magma implementierte SEA-Algorithmus, siehe Abschnitt A.2.2, berechnet in dieser Zeit die Gruppenordnung für Kurven über Körpern mit circa 2^{250} Elementen, siehe Tabelle 5.2. Die Primzahlen stammen von der Internetseite `http://primes.utm.edu`.

Tabelle 5.1: Laufzeit des Schoof-Algorithmus:
CPU Sekunden

Primzahl q	$y^2 = x^3 + 46x + 74$	$y^2 = x^3 + 97x + 199$	$\lfloor \log_2(q) \rfloor$
16427	1.02	0.72	14
17971	1.37	1	14
18181	1.31	1.13	14
24509	1.03	1.32	14
27611	1.25	0.95	14
30029	1.25	1.27	14
32057	0.76	0.74	14
32323	1.25	1.31	14
32423	1.32	1.34	14
32467	1.33	1.20	14
65537	0.80	1.28	16
67061	1.15	1.51	16
77377	1.18	1.23	16
86969	1.50	1.43	16
87811	1.25	1.31	16
90053	1.40	1.23	16
100003	0.82	1.42	16
100151	1.44	1.64	16
100153	1.58	1.51	16
111119	1.33	1.63	16
262553	1.12	1.08	18
279421	1.87	1.58	18
304807	1.54	1.54	18
333667	3.54	2.76	18
343141	4.39	4.94	18
351551	4.09	3.84	18
373757	4.23	3.24	18
401101	3.54	4.24	18
463003	3.21	3.89	18
511111	3.46	5.75	18
1062881	6.71	3.53	20
1120211	3.22	5.39	20
1242421	5.42	6.74	20
1556551	9.12	9.45	20
1831381	13.14	4.18	20

Tabelle 5.2: Laufzeit des Schoof-Algorithmus:

Primzahl q	CPU Sekunden für $y^2 = x^3 + 1993x + 1997$	$\lfloor \log_2(q) \rfloor$
58507257027668292914913707121362860099486421251314361133428157864445670000000000000000130000000000000011000000000009000000000700000005000003000118181818181818181818181818181818181818181818181818181818181818181818181818181818181818185323955009471744507093833849366798683834244431140567946328078240579623316397777777777777777777777777777777777772227777777777777777777777777777777777777	3.58	231
	4,14	243
	4.79	253
	7.87	263
	6,5	275

Anhang A

Implementierung in Magma

Magma ist ein mächtiges Algebra-Programm, das viele Algorithmen effizient umsetzt. Es wurde von der „Computational Algebra Group“ an der Universität Sydney entwickelt. Weitere Informationen findet man unter `http://magma.maths.usyd.edu.au/magma/`. Das Programm lässt sich mit begrenzten Ressourcen online nutzen, siehe `http://magma.maths.usyd.edu.au/calc/`. Um es auch an einen eigenen Rechner verwenden zu können, benötigt man allerdings eine kostenpflichtige Lizenz.

Im Folgenden wollen wir den Schoof-Algorithmus implementieren. Dazu betrachten wir zuerst einige notwendige Berechnungen, um später den Schoof-Algorithmus, wie im Algorithmus 2 beschrieben, zu implementieren. Dabei definieren wir zuerst den Körper und den Polynomring, in dem die Operationen ausgeführt werden. Die Variable „qglob“ gibt dabei die Anzahl der Elemente des Körpers $\mathbb{F}_{qglob}$ an, in dem die elliptische Kurve definiert sein soll. Der angegebene Code berechnet die Anzahl der Punkte der elliptischen Kurve, gegeben durch das Polynom $E[x, y] : y^2 - x^3 - ax - b$, für frei wählbare Koeffizienten a, b im Körper $\mathbb{F}_{997}$.

```
1 qglob  := 997;
2 Fq := FiniteField(qglob);
3 P<x,y> := PolynomialRing(Fq,2);
```

A.1 Methoden für den Schoof-Algorithmus

Als erste Methode berechnen wir eine Menge an Primzahlen, die nach Kapitel 4.1 ausreichend für den Schoof-Algorithmus ist. Dazu fügen wir solange Primzahlen zu einer Liste hinzu, bis deren Produkt größer als $4\sqrt{q}$ ist. Daraufhin entfernen wir so lange kleine Primzahlen wieder aus der Liste, wie das Produkt weiterhin größer als $4\sqrt{q}$ bleibt. Dabei betrachten wir in der

Liste nur ungerade Primzahlen. Die Berechnung für die Primzahl 2 führen wir direkt am Anfang aus. In der Liste sind die Primzahlen der Größe nach aufsteigend geordnet.

A.1.1 Algorithmus für die Bestimmung der Primzahlen

```
1  BerechnePrimzahlMenge := function(q)
2  prod := 2*3;
3  a := [3];
4  i := 1;
5  testvar := 4*Sqrt(q);
6  while prod lt testvar do
7        p := NextPrime(a[i]);
8        Append(~a, p);
9        i := i+1;
10       prod := prod*p;
11 end while;
12 //kleine Primzahlen entfernen:
13 while prod/a[1] ge testvar do
14       prod := prod/a[1];
15       a := Reverse(Prune(Reverse(a)));
16 end while;
17 return a;
18 end function;
```

Außerdem ist es für den Schoof-Algorithmus notwendig, die Divisionspolynome zu berechnen. Damit wir diese später mit dem euklidischen Algorithmus auf gemeinsame Teiler testen können, berechnen wir die Divisionspolynome gleich in der Form angeben, die in Lemma 3.1.2 beschrieben wird, sodass die ungeraden Divisionspolynome als Funktionen in x und die geraden Divisionspolynome in der Form $f(x)y$ gegeben sind. Die Divisionspolynome werden in den restlichen Algorithmen immer mit der Liste „D“ aufgerufen. Nachdem Magma nicht den nullten Eintrag einer Liste beschreiben kann, sind in der Liste nur die restlichen benötigten Divisionspolynome gespeichert.

A.1.2 Algorithmus zur Bestimmung der Divisionspolynome

```
1  BerechneDivisionspolynome := function(a,b, n)
2  D := [1+0*x, 2*y, 3*x^4+6*a*x^2+12*b*x-a^2,
3  4*y*(x^6+5*a*x^4+20*b*x^3-5*(a*x)^2
4  -4*a*b*x-8*b^2-a^3)];
```

```
5  i := 5;
6  while i le n do
7        j := (i-1) div 2;
8        D[i] := D[j+2]*D[j]^3-D[j-1]*D[j+1]^3 ;
9        j := j+1;
10       D[i+1] := D[j]*(D[j+2]*D[j-1]^2
11       -D[j-2]*D[j+1]^2)*(1/(2*y));
12       i := i+2;
13 end while;
14 //y^2 mit x^3+ax+b ersetzen:
15 i := 1;
16 E := D;
17 while i le n+1 do
18       MD := Monomials(D[i]);
19       C := Coefficients(D[i]);
20       E[i] := 0;
21       j := 1; //Innere Schleife
22       leave := false;
23       while j le #MD do
24             while leave eq false do
25                   if IsDivisibleBy(MD[j],y^2) eq true then
26                         MD[j] := (MD[j]/(y^2))*(x^3+a*x+b);
27                   else
28                         leave := true;
29                         E[i] := E[i]+C[j]*MD[j];
30                   end if;
31             end while;
32             j := j+1;
33             leave := false;
34       end while;
35       i := i+1;
36 end while;
37 return E;
38 end function;
```

Eine Berechnung die mehrmals im Schoof-Algorithmus benötigt wird ist das skalare Multiplizieren von Punkten. Dazu implementieren wir die Funktion wie im Satz 3.1.4. Man bemerke, dass wir das nullte Divisionspolynom nicht in die Liste D der Divisionspolynome mit aufgenommen haben. Somit müssen wir die skalare Multiplikation mit 2 getrennt betrachten.

A.1.3 Algorithmus für die skalare Multiplikation

```
1  SkalareMultiplikation := function(l,D)
2  if l eq 1 then
3       return [x,y];
4  elif l eq 2 then
5       return [x-((D[l-1]*D[l+1])/D[l]^2),
6       (D[l+2]*D[l-1]^2)/(4*y*D[l]^3)];
7  else
8       return [x-((D[l-1]*D[l+1])/D[l]^2),
9       (D[l+2]*D[l-1]^2-D[l-2]*D[l+1]^2)/(4*y*D[l]^3)];
10 end if;
11 end function;
```

Ebenfalls werden wir Punkte auf der elliptischen Kurve addieren müssen. Dazu verwenden wir die im Satz 1.3.2 gezeigten Formeln.

A.1.4 Algorithmus für das Addieren von Punkten

```
1  PunktAddition := function(P,R)
2  mu := (R[2]-P[2])/(R[1]-P[1]);
3  x3 := mu^2-P[1]-R[1];
4  y3 := P[2]+mu*(x3-P[1]);
5  return [x3,y3];
6  end function;
```

A.2 Implementierung des Schoof-Algorithmus

Damit können wir nun den Schoof-Algorithmus implementieren. Wir verwenden dabei die bool-Variable „leave“ um festzustellen, ob der Algorithmus bereits eine Lösung gefunden hatte. Die meisten Berechnungen werden in dem Quotientenring P modulo (E, Ψ_l) ausgeführt. Dieser wird im Algorithmus als „Q“ bezeichnet. Dort können wir effizient den Test auf Gleichheit im ersten Fall, wie in Kapitel 4.2.1 beschrieben, durchführen. Für den Test auf den größten gemeinsame Teiler aus 4.2.2 in den Zeilen 73 und 86 werden die benötigten Terme in einem Polynomring mit einer Variable gespeichert, um die Berechnung „modulo“ durchzuführen. Dazu existiert in Magma die Funktion „UnivariatePolynom“. Ebenfalls verwenden wir im zweiten Fall, Zeile 51, die Magma-Funktion „IsSquare (F ! ql)“, um zu bestimmen, ob ql ein Quadrat im endlichen Körper über der Primzahl l ist. Dies ersetzt den Satz 4.2.1.

Funktionen, um den größten gemeinsamen Teiler von zwei Polynomen zu finden („GreatestCommonDivisor“) sowie eine Wurzel modulo einer Primzahl („Sqrt“) zu bestimmen, und der chinesische Restsatz („CRT“) sind ebenfalls bereits in Magma implementiert. Für jede Primzahl wird am Ende eine neue Zwischenlösung für die Anzahl der Punkte mithilfe des chinesischen Restsatzes berechnet, wie im Beweis von Satz 4.2.2 beschrieben. Die Lösung wird in der Variable „talt“ gespeichert. In „lprod“ wird das Produkt der bisher verwendeten Primzahlen gespeichert.

A.2.1 Der Schoof-Algorithmus

```
1  Schoof := function(a,b,q)
2  S := BerechnePrimzahlMenge(q);
3  //Für l eq 2:
4  talt := 1;
5  Testx := UnivariatePolynomial(x^q-x);
6  TestE := UnivariatePolynomial(x^3+a*x+b);
7  if GreatestCommonDivisor(Testx mod TestE, TestE) ne 1
       then
8        talt := 0;
9  end if;
10 lprod := 2;
11 // Für l ge 3:
12 Q<x0,y0> := quo< P |y^2-x^3-a*x-b >;
13 D :=BerechneDivisionspolynome(a,b,S[#S]+2);
14 for l in S do
15       tl := 0;
16       Q<x0,y0> := quo< P |y^2-x^3-a*x-b, D[l] >;
17       ql := q mod l;
18       //Berechnen von (x,y)^q, (x,y)^(q^2),ql(x,y) in Q
19       xq2 := x0^(q^2);
20       yq2 := y0^(q^2);
21       xyql:= SkalareMultiplikation(ql, D);
22       xql := Q ! xyql[1];
23       yql := Q ! xyql[2];
24       leave := false;
25       //Fall 2: Test auf  x0^(q^2) ne xql:
26       if xq2-xql ne 0 then
27            //Berechnung von (X,Y) = (xq2,yq2)+(xql,yql):
28            Add := PunktAddition([xq2,yq2],[xql,yql]);
```

```
29          X := Q ! Add[1];
30          Y := Q ! Add[2];
31          //while-Schleife: Test auf X eq xj^q
32          j := 1;
33          while j le (l-1) div 2 and leave eq false do
34               xyj := SkalareMultiplikation(j, D);
35               xj := Q ! xyj[1];
36               yj := Q ! xyj[2];
37               if X - xj^q eq 0  then
38                    leave := true;
39                    if Y - yj^q eq 0 then
40                         tl := -j;
41                    else
42                         tl := j;
43                    end if;
44               end if;
45               j := j+1;
46          end while;
47     end if;
48 //Fall 3: Test, ob q_l ein Quadrat mod l ist
49 F := FiniteField(l);
50 I := Integers();
51 if IsSquare(F ! ql) eq true and leave eq false then
52     w := I ! Sqrt(F ! ql);
53     if w lt 0 then
54          w := w+w+w;
55     end if;
56     if w gt (l div 2) and w ne l then
57          w := l-w;
58     end if;
59     if w eq 1 then
60          Testx := x^q-x;
61     else
62          if w mod 2 eq 0 then
63               Testx := ((x^q-x))*(D[w] div y)^2*(x^3+a
                      *x+b)+(D[w-1]*D[w+1]);
64          else
65               Testx := (x^q-x)*D[w]^2+(D[w-1] div y)*(
                      D[w+1] div y)*(x^3+a*x+b);
66          end if;
67     end if;
```

```
68      //Berechnungen nur mod der Gleichung für die
           elliptische Kurve:
69      //Funktion nur in x schreiben:
70      Rx<x1> := PolynomialRing(Fq);
71      Testx := UnivariatePolynomial(Testx);
72      Dl := UnivariatePolynomial(D[l]);
73      if GreatestCommonDivisor(Testx mod Dl, Dl) ne 1
           then
74          if w eq 1 then
75              w := l-w;
76          end if;
77          if w eq 2 then
78              Testy := 4*(x^3+a*x+b)^((q+3) div 2)*(D
                   [2] div y)^3-(D[4] div y)*D[1]^2;
79          elif w mod 2 eq 0 then
80              Testy := 4*(x^3+a*x+b)^((q+3) div (2))*(
                   D[w] div y)^3-(D[w+2] div y)*D[w
                   -1]^2+(D[w-2] div y)*D[w+1]^2;
81          else
82              Testy := 4*(x^3+a*x+b)^((q-1) div (2))*D
                   [w]^3-D[w+2]*(D[w-1] div y)^2+D[w
                   -2]*(D[w+1] div y)^2;
83          end if;
84          //Funktion nur in x schreiben:
85          Testy := UnivariatePolynomial(Testy);
86          if  GreatestCommonDivisor(Testy mod Dl, Dl)
               ne 1
87              then tl :=  2*w;
88          else
89              tl := -2*w;
90          end if;
91      end if;
92  end if;
93  //Mit dem chinesischen Restsatz die Zwischenlösung
       berechnen:
94  talt := CRT([talt,tl],[lprod,l]);
95  lprod := l*lprod;
96  end for;
97  if(talt gt (lprod div 2)) then
98      talt := talt-lprod;
99  end if;
```

```
100 return q+1-talt;
101 end function;
```

Die Funktion wird aufgerufen mit

```
1 Schoof(a,b,qglob);
```

für die elliptische Kurve $y^2 = x^3 + ax + b$ über den Körper $\mathbb{F}_{qglob}$.
In Magma ist ebenfalls ein Algorithmus für das Bestimmen der Gruppenordnung auf elliptischen Kurven über einen Körper $\mathbb{F}_q$. Dieser beruht auf Verbesserungen des Schoof-Algorithmus durch Noam Elkies und A.O.L. Atkin. Mit der Funktion „EllipticCurve([FldFin | RngInt, RngInt])“ erstellt man in Magma eine elliptische Kurve über einen endlichen Körper, beschrieben durch die kurze Weierstraß-Gleichung $y^2 = x^3 + ax + b$. Die Gruppenordnung einer solchen elliptischen Kurve bestimmt man mit „#“.

A.2.2 Der SEA-Algorithmus von Magma

```
1 SEA := function(q,a,b)
2 F := FiniteField(q);
3 E := EllipticCurve([F | a, b]);
4 return #E;
5 end function;
```

Literaturverzeichnis

[AB04] A.O.L. Atkin and D.J. Bernstein. Prime sieves using binary quadratic forms. *Math.Comp.*, 73:1023–1030, 2004. Online erhältlich unter `http://www.ams.org/journals/mcom/2004-73-246/S0025-5718-03-01501-1/S0025-5718-03-01501-1.pdf`; Zuletzt aufgerufen am 28.04.2018.

[Ann14] Samuele Anni. Elliptic Curves. Skript zur Vorlesung. Universität Heidelberg, Fakultät für Mathematik, 2014. Online erhältlich unter `http://www.iwr.uni-heidelberg.de/groups/arith-geom/anni/MA426.pdf`; Zuletzt aufgerufen am 28.04.2018.

[Apo98] T.M. Apostol. *Introduction to Analytic Number Theory.* Undergraduate Texts in Mathematics. Springer New York, 1998.

[Bac94] P. Bachmann. *Zahlentheorie: th. Die Elemente der Zahlentheorie.* Zahlentheorie: versuch einer Gesammtdarstellung dieser Wissenschaft in ihren Haupttheilen. B. G. Teubner, 1894.

[BC13] Joppe Bos and Craig Costello. Elliptic and hyperelliptic curves: a practical security analysis. Technical report, October 2013. Online erhältlich unter `https://www.microsoft.com/en-us/research/publication/elliptic-and-hyperelliptic-curves-a-practical-security-analysis/`; Zuletzt aufgerufen am 28.04.2018.

[Bos13] S. Bosch. *Algebra.* Springer-Lehrbuch. Springer Berlin Heidelberg, 2013.

[Coh13] H. Cohen. *A Course in Computational Algebraic Number Theory.* Graduate Texts in Mathematics. Springer Berlin Heidelberg, 2013.

[Dew98] L. Dewaghe. Remarks on the Schoof-Elkies-Atkin algorithm. *Math.Comp.*, 67:1247–1252, 1998. Online erhältlich unter `https:`

//doi.org/10.1090/S0025-5718-98-00962-4; Zuletzt aufgerufen am 28.04.2018.

[Fis94] G. Fisher. *Ebene Algebraische Kurven*. Vieweg Studium; Aufbaukurs Mathematik Series. Vieweg+Teubner Verlag, 1994.

[Fri17] Stefan Friedl. An Elementary Proof of the Group Law for Elliptic Curves. 9, 2017. Online erhältlich unter http://www.uni-regensburg.de/Fakultaeten/nat_Fak_I/friedl/papers/elliptic_2017.pdf; Zuletzt aufgerufen am 28.04.2018.

[Ful08] William Fulton. Algebraic Curves, An Introduction to Algebraic Geometry. 2008.

[IR13] K. Ireland and M. Rosen. *A Classical Introduction to Modern Number Theory*. Graduate Texts in Mathematics. Springer New York, 2013.

[LM15] J. Lopez and C.J. Mitchell. *Information Security: 18th International Conference, ISC 2015, Trondheim, Norway, September 9-11, 2015, Proceedings*. Lecture Notes in Computer Science. Springer International Publishing, 2015.

[Mue91] Die Berechnung der Punktanzahl von elliptischen Kurven über endlichen Primkörpern. Diplomarbeit. Universität des Saarlandes, Saarbrücken, 1991.

[RS62] J. Rosser and L. Schoenfeld. Approximate Formulas for some Functions of Prime Numbers. *Illinois Journal of Mathematics*, 6:64–94, 1962. Online erhältlich unter https://projecteuclid.org/euclid.ijm/1255631807; Zuletzt aufgerufen am 28.04.2018.

[Sch85] René Schoof. Elliptic Curves over Finite Fields and the Computation of Square Roots mod *p*. *Math. Comp.*, 44(170):483–494, 1985. Online erhältlich unter http://www.mat.uniroma2.it/~schoof/ctpts.pdf; Zuletzt aufgerufen am 28.04.2018.

[Sil13] J.H. Silverman. *The Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics. Springer New York, 2013.

[Sto09] Michael Stoll. Arithmetik elliptischer kurven mit anwendungen. Vorlesungsskript. Mathematisches Institut der Universität Bayreuth, 2009. Online erhältlich unter http://www.mathe2.uni-

bayreuth.de/stoll/teaching/EllKA/Skript-EllK.pdf; Zuletzt aufgerufen am 28.04.2018.

[SZP03] S. Schmitt, H.G. Zimmer, and A. Pethö. *Elliptic Curves: A Computational Approach*. De Gruyter Studies in Mathematics. De Gruyter, 2003.

[Was08] L.C. Washington. *Elliptic Curves: Number Theory and Cryptography, Second Edition*. Discrete Mathematics and Its Applications. CRC Press, 2008.

[Wät08] D. Wätjen. *Kryptographie: Grundlagen, Algorithmen, Protokolle*. Spektrum Akademischer Verlag, 2008.

[Wer13] A. Werner. *Elliptische Kurven in der Kryptographie*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2013.

[Yan07] S.Y. Yan. *Cryptanalytic Attacks on RSA*. Springer US, 2007.

[Öc11] J. Öchsner. Ein elementarer Beweis des Primzahlsatzes. Ausführung und historische Perspektive. Bachelorarbeit. Universität Würzburg, Fakultät für Mathematik, 2011. Online erhältlich unter https://www.mathematik.uni-wuerzburg.de/~steuding/oechsner.pdf; Zuletzt aufgerufen am 28.04.2018.

Abbildungsverzeichnis

Tabellenverzeichnis

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie wörtlich oder inhaltlich übernommene Stellen als solche kenntlich gemacht habe.

Ort, Datum