Berechnung von Divisorenklassengruppen von globalen Funktionenkörpern mittels der Tate-Lichtenbaum Paarung

Diplomarbeit

von

Jens-Dietrich Bauch

Angefertigt am Institut für Mathematik der
Technischen Universität Berlin

Sommer 2009

Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbständig und ohne fremde Hilfe angefertigt und mich anderer, als der im beigefügten Verzeichnis angegebenen Hilfsmittel, nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht.

Berlin, den 7. September 2009

_____

(Jens-Dietrich Bauch)

Inhaltsverzeichnis

Einleitung

Eine der grundlegendsten Strukturen der Mathematik sind Gruppen. Eine davon ist die Divisorenklassengruppe eines globalen Funktionenkörpers F/K. Sie ist eine abelsche Gruppe und ihre Untergruppe, die Klassengruppe, ist sogar endlich.
Das diskrete Logarithmus Problem für zwei Elemente g_1 und g_2 einer endlichen abelschen Gruppe beschreibt, deren linearen Zusammenhang $kg_1 = g_2$ mit einer kleinsten nicht negativen ganzen Zahl k zu bestimmen. Für Verschlüsselungsverfahren in der Kryptographie sind gerade solche abelschen Gruppen geeignet, in denen das diskrete Logarithmus Problem schwer zu lösen ist. Die Klassengruppe von globalen Funktionenkörpern und speziell die von elliptischen globalen Funktionenkörpern bilden bis auf Isomorphie die Basis der im Moment effizientesten Public-Key-Verfahren.

Die Gruppenstruktur der Jacobischen Varietät einer glatten projektiven algebraischen Kurve über dem endlichen Körper K ist isomorph zur Klassengruppe eines globalen Funktionenkörpers F/K. Unter Ausnutzung dieser Isomorphie ist es möglich, die Berechnung von Endomorphismenringen solcher Jacobischen Varietäten unter bestimmten Voraussetzungen auf die Bestimmung von Klassengruppen von globalen Funktionenkörpern zu übersetzen.

Die vorliegende Arbeit beschäftigt sich mit der Berechnung von Divisorenklassengruppen beziehungsweise Klassengruppen von globalen Funktionenkörpern. Im Gegensatz zu den schon bekannten Berechnungsverfahren beruht der neue Ansatz darauf, mittels der Tate-Lichtenbaum Paarung, Methoden aus der linearen Algebra für die Berechnung der Divisorenklassengruppe zu verwenden.

Sie gliedert sich wie folgt:

Im ersten Kapitel werden alle benötigten Grundlagen behandelt. Nachdem wir den Zusammenhang von endlichen abelschen Gruppen und Moduln beziehungsweise Vektorräumen kennengelernt haben, führen wir in die Theorie der Funktionenkörper ein. Anschließend werden die wichtigsten Invarianten zu diesem Thema, das Geschlecht, die Divisorenklassengruppe und die Klassengruppe eines Funktionenkörpers beschrieben. Zum Abschluss dieses Kapitels legen wir noch die algorithmischen Rahmenbedingungen für diese Arbeit fest.

Das zweite Kapitel beginnt mit der Definition der Tate-Lichtenbaum Paarung und erläutert die Zusammenhänge zu nicht-ausgearteten bilinearen Abbildungen auf Vek-

torräumen. Anschließend beschreiben wir einige Eigenschaften von Vektorräumen und nicht-ausgearteter bilinearer Abbildungen auf diesen. Danach wird ein Verfahren zur Bestimmung von linear unabhängigen Divisorenklassen vom Grad 0 mittels der Tate-Lichtenbaum Paarung vorgestellt. Zuletzt präsentieren wir Alternativen dieses Verfahrens für all die Fälle, in denen die Tate-Lichtenbaum Paarung nicht definiert ist.

Im dritten Kapitel werden wir mehrere Algorithmen zur Berechnung der Klassengruppe eines globalen Funktionenkörpers kennenlernen. Nachdem die zu diesem Thema relevanten Algorithmen vorgestellt wurden, beschreiben wir die neuentwickelten. Dabei wird zwischen einer probabilistischen Variante sowie einer nicht-probabilistischen unterschieden. Grundlegend basieren beide Varianten auf der Idee über Untervektorräume der l-Torsionsgruppe der Klassengruppe für einen Primteiler l der Klassenzahl die l-Sylow-Gruppe der Klassengruppe und damit die Klassengruppe selbst zu berechnen.

Für die probabilistische Variante beschreiben wir zuerst die wahrscheinlichkeitstheoretischen Grundlagen um Erzeugendensysteme von Vektorräumen in der Klassengruppe zu produzieren. Danach wird mit Hilfe der im zweiten Kapitel entwickelten Methoden erklärt, wie Basen daraus berechnet werden können. Abschließend fügen wir die Ergebnisse zusammen, stellen einen Algorithmus zur Berechnung von l-Sylow-Gruppen der Klassengruppe eines globalen Funktionenkörper vor und zeigen, wie damit die Klassengruppe bestimmt werden kann.

Bezüglich der nicht-probabilistischen Variante setzen wir voraus, dass ein Erzeugendensystem der Klassengruppe zur Verfügung steht. Als erstes erklären wir, wie für einen Primteiler l der Klassenzahl aus dem Erzeugendensytem der Gruppe ein Erzeugendensystem von Untervektoräumen der l-Torsionsgruppe der Klassengruppe konstruiert werden kann. Anschließend entwickeln wir analog zur probabilistischen Variante darüber einen Algorithmus zur Berechnung der Klassengruppe.

Nachdem die Komplexität der Algorithmen miteinander verglichen wurde, stellen wir zum Abschluss von Kapitel drei noch ein Verfahren zur Berechnung des diskreten Logarithmus in der Klassengruppe eines globalen Funktionenkörper mit der Tate-Lichtenbaum Paarung vor.

Im vierten Kapitel vergleichen wir an praktischen Beispielen einige der vorgestellten Algorithmen, indem wir für konkrete globale Funktionenkörper die Klassengruppe von diesen berechnen lassen. Zum Abschluss der Arbeit geben wir im fünften Kapitel ein Anwendungsbeispiel der entwickelten Verfahren mit Ausblick.

Kapitel 1

Grundlagen

In diesem Kapitel beschreiben wir das mathematische Grundgerüst für die vorliegende Arbeit. Für die Theorie der algebraischen Funktionenkörper verweisen wir auf [Sti93].

1.1 Darstellung von abelschen Gruppen

Wir werden in dieser Arbeit alle abelschen Gruppen G additiv darstellen, dass heißt die Verknüpfung von Gruppenelementen g_1 und g_2 schreiben wir als $g_1 + g_2$. Das Inverse Elemente von $g \in G$ ist $-g$ und 0 bezeichne das neutrale Element in G. Sei g ein Element von G sowie $k \in \mathbb{Z}$, dann definiert

$$kg := \begin{cases} \underbrace{g + \cdots + g}_{k-mal} & wenn\ k > 0, \\ -(\underbrace{g + \cdots + g}_{k-mal}) & wenn\ k < 0, \\ 0 & wenn\ k = 0 \end{cases}$$

die Multiplikation von Elementen aus G mit ganzen Zahlen.

Definition 1.1. *Für ein Element g einer abelschen Gruppe G bezeichne* $\mathrm{Ord}(g)$ *die kleinste positive ganze Zahl k, so dass $kg = 0$ gilt. Wir nennen diese Zahl die Ordnung von g in G.*

Definition 1.2. *Für eine abelsche Gruppe G und eine positive ganze Zahl m definieren wir die Untergruppe $G[m] := \{g \in G \mid mg = 0\}$ und nennen sie die m-Torsionsgruppe von G.*

Nach dem Homomorphiesatz der Gruppentheorie sind die Gruppen $G[m]$ und G/mG isomorph. Ist g ein Element von $G[m]$ oder G/mG, dann gilt $mg = 0$. Also ist

$$kg = (k + \gamma m)g$$

für alle ganzen Zahlen k und γ. Wir betrachten statt k die ganze Zahl $\tilde{k}$ zwischen 0 und $m-1$, die durch $k = jm + \tilde{k}$ eindeutig definiert ist. Fassen wir $\{0, \ldots, m-1\}$ als Vetretersystem des Ringes $\mathbb{Z}/m\mathbb{Z}$ auf, dann wird aus $G[m]$ beziehungsweise G/mG ein freier $\mathbb{Z}/m\mathbb{Z}$-Modul. Ist m eine Primzahl, dann bezeichnet $\mathbb{F}_m := \mathbb{Z}/m\mathbb{Z}$ einen Körper und $G[m]$ sowie G/mG werden zu $\mathbb{F}_m$-Vektorräumen. Damit können wir die Begriffe und Methoden der Linearen Algebra auf die Untergruppen $G[m]$ und G/mG übertragen.
Nach dem Hauptsatz über endlich erzeugte abelsche Gruppen gilt die folgende Isomorphie:

$$G \cong \mathbb{Z}/c_1\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/c_n\mathbb{Z},$$

wobei $c_1 \mid c_2 \mid \cdots \mid c_n$. Wir nennen die ganzen Zahlen $c_1, \ldots, c_n$ die Gruppeninvarianten von G. Den zu $\mathbb{Z}/c_i\mathbb{Z}$ isomorphen Teil von G mit $i \in \{1, \ldots, n\}$ bezeichnen wir als die i-te zyklische Komponente von G. Eine Teilmenge S der Gruppe G heißt ein Erzeugendensystem von G, wenn sich jedes Element $g \in G$ als $\mathbb{Z}$-Linearkombination mit Elementen aus S darstellen läßt. Wir schreiben in diesem Kontext

$$G = \langle S \rangle .$$

Bestimmt $\{g_1, \ldots, g_n\}$ die Menge S, so dass g_i die i-te zyklischen Komponente von G erzeugt, dann gilt offensichtlich $G = \langle S \rangle$. Wir nennen in diesem Fall S ein minimales Erzeugendensystem von G.

1.2 Funktionenkörper

Definition 1.3. *Sei K ein Körper. Ein algebraischer Funktionenkörper F/K in einer Variable über K ist eine algebraische Körpererweiterung F von $K(x)$, wobei $x \in F$ über K ein transzendentes Element ist. In diesem Zusammenhang nennen wir K den Konstantenkörper. Bezeichne $\overline{K}$ den algebraischen Abschluss von K in F, dann heißt dieser Körper exakter Konstantenkörper. Ist der Konstantenkörper endlich, so heißt der Funktionenkörper dazu globaler Funktionenkörper.*

Der Einfachheit halber schreiben wir, statt algebraischer Funktionenkörper F über dem Konstantenkörper K in einer Variablen, einfach Funktionenkörper F/K. Wenn klar ist welcher Konstantenkörper gemeint ist, schreiben wir kurz F.

Definition 1.4. *Ein algebraischer Funktionenkörper F'/K' heißt algebraische Erweiterung eines algebraischen Funktionenkörpers F/K, wenn F'/F eine algebraische Körpererweiterung bestimmt und K in K' enthalten ist.*
Die algebraische Erweiterung F'/K' von F/K nennen wir Konstantenkörpererweiterung, wenn $F' = FK' := F'(K')$ gilt.

Definition 1.5. *Ein Bewertungsring des Funktionenkörpers F/K ist ein Ring $\mathcal{O}$ mit folgenden Eigenschaften:*

(i) $K \subsetneq \mathcal{O} \subsetneq F$ *und*

(ii) für jedes $z \in F$ *gilt* $z \in \mathcal{O}$ *oder* $z^{-1} \in \mathcal{O}$.

Sei $\mathcal{O}$ ein Bewertungsring von F/K und bezeichne $\mathcal{O}^\times$ die Menge der invertierbaren Elemente davon. Dann ist der Ring $\mathcal{O}$ lokal und die Menge $\mathrm{P} = \mathcal{O} \backslash \mathcal{O}^\times$ dessen eindeutiges maximales Ideal. Die Menge $\mathcal{O}_\mathrm{P} := \{z \in F \mid \ z^{-1} \notin \mathrm{P}\}$ ist ein Bewertungsring. Dieser wird durch P eindeutig bestimmt und heißt Bewertungsring von P.

Definition 1.6. *Eine Stelle* P *eines Funktionenkörpers* F/K *ist das maximale Ideal eines Bewertungsringes* $\mathcal{O}$ *von* F/K. *Die Menge aller Stellen des Funktionenkörpers* F/K *wird durch*

$$\mathbb{P}_F := \{\mathrm{P} \mid \mathrm{P} \textit{ ist Stelle von } F/K\}$$

bezeichnet.

Sei P eine Stelle von F/K und $\mathcal{O}_\mathrm{P}$ der korrespondierende Bewertungsring, dann ist $F_\mathrm{P} := \mathcal{O}_\mathrm{P}/\mathrm{P}$ ein Körper und $deg(\mathrm{P}) := [F_\mathrm{P} : K]$ heißt der Grad von P. Diese Zahl ist für jede Stelle von F/K endlich. Für $x \in \mathcal{O}_\mathrm{P}$ bezeichne $x(\mathrm{P})$ dessen Klasse $x + \mathrm{P}$ in F_P. Indem wir $x(\mathrm{P}) := \infty$ für $x \in F \setminus \mathcal{O}_\mathrm{P}$ setzen, erhalten wir eine Abbildung von F nach $F_\mathrm{P} \cup \{\infty\}$.

Definition und Satz 1.7. *Sei* $\mathrm{P} \in \mathbb{P}_F$ *und* $\mathcal{O}_\mathrm{P}$ *der dazugehörige Bewertungsring von* F/K, *dann existiert ein* $t \in \mathcal{O}_\mathrm{P}$ *mit* $\mathrm{P} = t\mathcal{O}_\mathrm{P}$. *Wir nennen* t *eine Uniformisierende von* P. *Fixieren wir ein solches* t, *so hat jedes Element* $0 \neq z \in F$ *eine eindeutige Darstellung der Form* $z = t^n u$ *mit* $u \in \mathcal{O}_\mathrm{P}^\times$ *und* $n \in \mathbb{Z}$. *Wir definieren damit für* $\mathrm{P} \in \mathbb{P}_F$ *die Abbildung* $v_\mathrm{P} : F \longrightarrow \mathbb{Z} \cup \{\infty\}$ *durch*

$$v_\mathrm{P}(z) := n \textit{ und } v_\mathrm{P}(0) := \infty.$$

Beweis. Beweise für diese Aussagen finden sich in [Sti93, S.4-5]. □

Die letzte Definition hängt ausschließlich von P ab und nicht von der Wahl von t.

Definition 1.8. *Sei* F'/K' *eine algebraische Erweiterung von* F/K. *Wir sagen, dass eine Stelle* $\mathrm{P}' \in \mathbb{P}_{F'}$ *über der Stelle* $\mathrm{P} \in \mathbb{P}_F$ *liegt, falls* P *in* P' *enthalten ist.*

Liege $\mathrm{P}' \in \mathbb{P}_{F'}$ über der Stelle $\mathrm{P} \in \mathbb{P}_F$, dann gibt es eine positive ganze Zahl e mit der Eigenschaft

$$v_{\mathrm{P}'}(x) = e \cdot v_\mathrm{P}(x) \textit{ für alle } x \in F.$$

Ist die Zahl e gleich 1, dann heißt die Körpererweiterung F'/F an der Stelle P unverzweigt, andernfalls nennen wir die Erweiterung verzweigt bei P. Für P' über P gilt $\mathcal{O}_\mathrm{P} \subseteq \mathcal{O}_{\mathrm{P}'}$. Folglich können wir den Restklassenkörper F_P in $F'_{\mathrm{P}'}$ einbetten und damit $F'_{\mathrm{P}'}/F_\mathrm{P}$ als eine algebraische Körpererweiterung auffassen. Für den Fall, dass der Grad dieser Körpererweiterung gleich 1 ist, bezeichnen wir F'/F an der Stelle P als rein verzweigt.

1.3 Divisorenklassengruppe und Geschlecht

In diesem Abschnitt beschreiben wir die wichtigsten Invarianten von Funktionenkörpern:

- Die Divisorenklassengruppe,
- die Klassengruppe und
- das Geschlecht

eines Funktionenkörpers F über dem endlichen Konstantenkörper $K := \mathbb{F}_q$ mit $q = p^k$ Elementen. Hierbei bezeichnet p immer eine Primzahl und k eine positive ganze Zahl. Wenn es nicht explizit erwähnt wird, ist der Konstantenkörper K beliebig.

Definition 1.9. *Die freie abelsche Gruppe, die von den Stellen von F/K erzeugt wird, heißt die Divisorengruppe von F/K. Wir bezeichnen sie mit $\mathcal{D}_F$. Die Elemente $D \in \mathcal{D}_F$ heißen Divisoren von F/K und sind formale Summen der folgenden Form*

$$D = \sum_{\mathrm{P} \in \mathbb{P}_F} n_{\mathrm{P}} \mathrm{P}$$

mit $n_{\mathrm{P}} \in \mathbb{Z}$ und fast allen $n_{\mathrm{P}} = 0$. Der Exponent $v_{\mathrm{P}}(D)$ einer Stelle P in D ist der Wert n_{P}. Sind alle n_{P} gleich 0, dann heißt D der Nulldivisor und wir schreiben für D einfach 0.
Die Menge $\{\mathrm{P} \in \mathbb{P}_F \mid v_{\mathrm{P}}(D) \neq 0\}$ heißt Träger von D, wir kürzen sie mit $\mathrm{supp}(D)$ ab. Zwei Divisoren D_1 und D_2 nennen wir koprim, wenn der Schnitt derer Träger leer ist. Seien $D_1 := \sum_{\mathrm{P} \in \mathbb{P}_F} n_{\mathrm{P}} \mathrm{P}$ und $D_2 := \sum_{\mathrm{P} \in \mathbb{P}_F} m_{\mathrm{P}} \mathrm{P}$ Divisoren von F/K. Mittels

$$D_1 \leq D_2 :\Longleftrightarrow n_{\mathrm{P}} \leq m_{\mathrm{P}} \text{ für alle } \mathrm{P} \in \mathbb{P}_F$$

wird eine Halbordnung auf $\mathcal{D}_F$ definiert. Den Grad eines Divisors bezeichnen wir durch

$$deg(D) := \sum_{\mathrm{P} \in \mathbb{P}_F} v_{\mathrm{P}}(D) \deg(\mathrm{P}).$$

Für ein Element $0 \neq z \in F$ sind die Mengen $Z(z) := \{\mathrm{P} \in \mathbb{P}_F \mid v_{\mathrm{P}}(z) > 0\}$ und $N(z) := \{\mathrm{P} \in \mathbb{P}_F \mid v_{\mathrm{P}}(z) < 0\}$ endlich.

Definition 1.10. *Für jedes $z \in F$ ungleich 0 bezeichnen wir*

$$(z)_0 := \sum_{\mathrm{P} \in Z(z)} v_{\mathrm{P}}(z)\mathrm{P} \text{ und } (z)_\infty := \sum_{\mathrm{P} \in N(z)} (-v_{\mathrm{P}}(z))\mathrm{P}$$

als den Nullstellendivisor beziehungsweise Polstellendivisor von z. Der Hauptdivisor von z wird durch

$$(z) := (z)_0 - (z)_\infty$$

definiert. Die Menge

$$\mathcal{P}_F := \{(z) \mid 0 \neq z \in F\}$$

nennen wir die Gruppe der Hauptdivisoren des Funktionenkörpers F/K.

Die Gruppe der Hauptdivisoren $\mathcal{P}_F$ ist eine Untergruppe der Divisorengruppe $\mathcal{D}_F$. Durch die Relation $D \sim D' :\Longleftrightarrow D = D' + (z)$ mit D und D' aus $\mathcal{D}_F$ sowie (z) aus $\mathcal{P}_F$ erhalten wir eine Äquivalenzrelation.

Definition 1.11. *Die Faktorgruppe $C_F := \mathcal{D}_F/\mathcal{P}_F$ heißt Divisorenklassengruppe von F. Die Divisorenklasse von $D \in \mathcal{D}_F$ in C_F bezeichnen wir mit $[D]$. Der Grad eines Elements aus C_F entspricht dem Grad eines Repräsentanten aus dessen Klasse.*

Definition 1.12. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper. Die Menge*

$$C_F^0 := \{[D] \in C_F \mid \deg([D]) = 0\}$$

ist eine Untergruppe von C_F und heißt Divisorenklassengruppe vom Grad 0 von $F/\mathbb{F}_q$ oder kurz Klassengruppe von F. Wenn klar ist welcher Funktionenkörper gemeint ist, sagen wir einfach Klassengruppe.

Definition und Satz 1.13. *Die Klassengruppe C_F^0 eines globalen Funktionenkörpers F ist endlich. Die Ordnung dieser abelschen Gruppe bezeichnen wir durch h_F und nennen sie die Klassenzahl von F.*

Beweis. Ein Beweis findet sich in [Sti93, S.159]. □

Definition 1.14. *Für einen Divisor D eines Funktionenkörpers F/K definieren wir den Riemann-Roch-Raum durch*

$$\mathcal{L}(D) := \{\ z \in F \mid (z) \geq -D\} \cup \{0\}.$$

Lemma 1.15. *Sei D ein Divisor eines Funktionenkörpers F/K, dann ist $\mathcal{L}(D)$ ein K-Vektorraum.*

Beweis. In [Sti93, S.17] wurde die Aussage bewiesen. □

Definition 1.16. *Das Geschlecht g eines Funktionenkörper F/K ist definiert durch*

$$g := \max\{\deg(D) - \dim_K(\mathcal{L}(D)) + 1 \mid D \in \mathcal{D}_F\}.$$

Das Geschlecht eines Funktionenkörpers ist eine nicht-negative ganze Zahl. Wir werden in dieser Arbeit ausschließlich Funktionenkörper vom Geschlecht $g \geq 1$ betrachten.

Beispiel 1.17. *Funktionenkörper F/K vom Geschlecht $g = 1$ heißen elliptische Funktionenkörper, wenn ein Divisor vom Grad 1 von F/K existiert. Ist das Geschlecht $g = 2$, so nennen wir den Funktionenkörper hyperelliptischer Funktionenkörper.*

Satz 1.18. *Die Klassenzahl eines Funktionenkörpers $F/\mathbb{F}_q$ vom Geschlecht g ist kleiner gleich $(\sqrt{q}+1)^{2g}$.*

Beweis. In [SD06, S.237] ist ein Beweis nachzulesen. □

1.3.1 Struktur der Divisorenklassengruppe und der m-Torsionsgruppe

Wir wollen nun die Divisorenklassengruppe eines Funktionenkörpers F/K etwas genauer betrachten. Unser Interesse ist es, diese im Fall eines endlichen Konstantenkörpers K zu berechnen. Der folgende Satz zeigt, dass es zur Berechnung der Divisorenklassengruppe eines globalen Funktionenkörpers ausreicht dessen Klassengruppe zu bestimmen.

Satz 1.19. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper. Bezeichne C_F die Divisorenklassengruppe und C_F^0 und die Klassengruppe von F, dann gilt die folgende Isomorphie:*

$$C_F \cong C_F^0 \oplus \mathbb{Z}.$$

Beweis. Ein Beweis findet sich in [Hes99, S.3]. □

Sei F ein Funktionenkörper vom Geschlecht g über dem endlichen Körper $\mathbb{F}_q$ mit $q = p^k$ Elementen und einer Primzahl p. Bezeichne $\overline{F}$ einen fest gewählten algebraischen Abschluss von F und $\overline{\mathbb{F}}_q$ den algebraischen Abschluss von $\mathbb{F}_q$ in $\overline{F}$. Wir betrachten die Konstantenkörpererweiterung $F' := F\overline{\mathbb{F}}_q$ über $\overline{\mathbb{F}}_q$.

Satz 1.20. *Für $m \in \mathbb{Z}^{>0}$ mit $p \nmid m$ und $r \in \mathbb{Z}^{>0}$ ist*

(i) $C_{F'}^0[m] \cong (\mathbb{Z}/m\mathbb{Z})^{2g}$ *und*

(ii) $C_{F'}^0[p^r] \cong (\mathbb{Z}/p^r\mathbb{Z})^s$ *mit* $0 \leq s \leq g$.

Beweis. In [Hes99, S.44] finden sich Verweise auf Beweise. □

Da F' eine Konstantenkörpererweiterung von F ist, können wir die Elemente von C_F^0 in $C_{F'}^0$ einbetten und somit C_F^0 als Untergruppe von $C_{F'}^0$ auffassen. Also folgt unmittelbar mit Satz 1.20, dass

$$C_F^0 \cong (\mathbb{Z}/c_1\mathbb{Z}) \oplus \cdots \oplus (\mathbb{Z}/c_f\mathbb{Z}) \text{ mit } f \leq 2g \tag{1.1}$$

gilt.

1.4 Differentiale

In dieser Arbeit wollen wir die Divisorenklassengruppe beziehungsweise die Klassengruppe eines globalen Funktionenkörpers F/K berechnen, dessen Konstantenkörper $q = p^k$ Elemente hat, wobei p eine Primzahl und k eine positive ganze Zahl bezeichnet. Für einen Primteiler l der Klassenzahl werden wir dazu die Struktur der l-Torsionsgruppe von C_F^0 ausnutzen. Ist dieser Primteiler gleich p, dann verwenden wir die Theorie der Differentiale.

Bezeichne in diesem Abschnitt F/K einen Funktionenkörper über dem vollkommenen Konstantenkörper K. Für die nachfolgende Einführung halten wir uns an [Sti93, Kap. IV.1.]. Darin wird zusätzlich verlangt, dass K algebraisch abgeschlossen ist. Diese Voraussetzung spielt bei den folgenden Definitionen und Überlegungen aber keine Rolle.

Definition 1.21. *Sei M ein F-Modul. Eine K-lineare Abbildung $\delta : F \longrightarrow M$ heißt Ableitung von F/K, wenn $\delta(uv) = u\delta(v) + v\delta(u)$ für alle $u, v \in F$ gilt.*

Die Elemente $x \in F$, für die $F/K(x)$ eine seperable algebraische Körpererweiterung ist, heißen seperierende Elemente von F/K. Nach [Sti93, S.128] ist die Existenz eines solchen Elements garantiert.

Definition und Satz 1.22. *Sei $x \in F$ ein seperierendes Element des Funktionenkörpers F/K. Dann existiert eine eindeutige Ableitung $\delta_x : F \longrightarrow F$ von F/K mit der Eigenschaft $\delta_x(x) = 1$. Sie heißt eindeutige Ableitung nach x.*

Beweis. Ein Beweis ist in [Sti93, S.135] nachzulesen. □

Lemma 1.23. *Seien x und y seperierende Elemente von F/K, dann gilt $\delta_y = \delta_y(x)\delta_x$.*

Beweis. Die Aussage wurde in [Sti93, S.137] bewiesen. □

Mit Hilfe des letzten Lemmas, können wir auf der Menge

$$Z := \{(x, y) \in F \times F \mid y \text{ ist seperierendes Element von } F/K\}$$

durch $(x_1, y_1) \sim (x_2, y_2) :\Longleftrightarrow x_2 = x_1\delta_{y_2}(y_1)$ eine Äquivalenzrelation definieren. Die Äquivalenzklassen $(x, y) \in Z$ werden durch xdy bezeichnet.

Definition 1.24. *Die Menge $\Omega(F/K) := \{xdy \mid x \in F \text{ und } y \in F \text{ ist seperierend}\}$ heißt der Raum der Differentiale von F/K. Die Elemente $xdy \in \Omega(F/K)$ werden Differentiale von F/K genannt.*

Der Raum der Differentiale von $\Omega(F/K)$ ist ein F-Modul. Wir definieren die Multiplikation für $xdy \in \Omega(F/K)$ und $z \in F$ durch

$$z(xdy) := (zx)dy.$$

Wählt man ein beliebiges seperierendes Element z von F/K, dann gibt es für x_1dy_1 und x_2dy_2 aus $\Omega(F/K)$ die Darstellung $x_1dy_1 = (x_1\delta_z(y_1))dz$ und $x_2dy_2 = (x_2\delta_z(y_2))dz$. Nun können wir die Addition mittels

$$x_1dy_1 + x_2dy_2 = (x_1\delta_z(y_1))dz + (x_2\delta_z(y_2))dz := (x_1\delta_z(y_1) + x_2\delta_z(y_2))dz$$

definieren. Wegen Lemma 1.23 ist diese Definition unabhängig von der Wahl des seperierenden Elements z. Indem wir $dt := 0$ für jedes nicht-seperierende Element t von F/K setzen, erhalten wir die Abbildung

$$F \longrightarrow \Omega(F/K) \text{ mit } t \longmapsto dt.$$

Satz 1.25.

(i) Der Raum der Differentiale $\Omega(F/K)$ ist ein freier F-Modul vom Rang 1.

(ii) Die Abbildung $d: F \longrightarrow \Omega(F/K)$ mit $t \longmapsto dt$ ist eine Ableitung von F/K.

Beweis. In [Sti93, S.138] wurden beide Aussagen bewiesen. □

Satz 1.26. *Sei F ein Funktionenkörper über dem vollkommenen Konstantenkörper K und t eine Uniformisierende einer Stelle P von F/K. Dann ist t ein seperierendes Element von F/K.*

Beweis. Ein Beweis ist in [Coh91, S.138] nachzulesen. □

Nach den letzten beiden Sätzen, können wir jedes $\omega \in \Omega(F/K)$ schreiben als $\omega = xdt$, wobei $x \in F$ und t eine Uniformisierende einer Stelle $\mathrm{P} \in \mathbb{P}_F$ ist. Der Wert $v_{\mathrm{P}}(\omega) := v_{\mathrm{P}}(\frac{\omega}{dt})$ heißt die Ordnung von ω in P und ist unabhängig von der Wahl von t. Für jedes $\omega \in \Omega(F/K)$ und fast alle $\mathrm{P} \in \mathbb{P}_F$ ist $v_{\mathrm{P}}(\omega) = 0$. Folglich können wir mittels $(\omega) := \sum_{\mathrm{P}\in\mathbb{P}_F} v_{\mathrm{P}}(\omega) \cdot \mathrm{P} \in \mathcal{D}_F$ jedem Differential ungleich Null von F/K einen Divisor aus $\mathcal{D}_F$ zuweisen.

Definition 1.27. *Ein Differential $\omega \in \Omega(F/K)$ heißt holomorph, wenn $v_{\mathrm{P}}(\omega) \geq 0$ für alle Stellen P von F/K ist. Den Raum der holomorphen Differentiale von F/K bezeichnen wir mit $\Omega^0(F/K)$.*

Der Raum der holomorphen Differentiale von F/K ist ein g-dimensionaler K-Vektorraum, wobei g das Geschlecht des Funktionenkörpers F/K bezeichnet.

Zuletzt definieren wir noch einen für die spätere Betrachtungen wichtigen Teilraum des Raumes der Differentiale.

Definition 1.28. *Der Raum der logarithmischen Differentiale von F/K ist definiert durch*

$$\Omega^{\log}(F/K) := \{(1/y)dy \mid y \in F^\times\}.$$

1.5 Algorithmische Vorbemerkungen

1.5.1 $\mathcal{O}$- und $\widetilde{\mathcal{O}}$-Notation

Bei der Entwicklung von Algorithmen ist es notwendig abzuschätzen, welchen Rechenaufwand sie haben und welchen Speicherplatz sie benötigen. Um solche Analysen zu vereinfachen, verwenden wir in dieser Arbeit die $\mathcal{O}$-Notation sowie die $\widetilde{\mathcal{O}}$-Notation. Beschreibe fortan log den Logarithmus zur Basis 2.
Für zwei Funktionen $f, g : \mathbb{N} \longrightarrow \mathbb{R}$ schreiben wir $f(n) = \mathcal{O}(g(n))$, falls es Konstanten $c \in \mathbb{R}^+$ und $N \in \mathbb{N}$ gibt, so dass

$$|f(n)| \leq c \cdot g(n)$$

für fast alle $n \geq N$ gilt. Diese Notation läßt sich auch auf Funktionen in mehreren Variablen verallgemeinern. Existiert eine positive ganze Zahl k mit der Eigenschaft $f(n) = \mathcal{O}(g(n)\log^k(g(n)))$, dann schreiben wir der Einfachheit halber $f(n) = \widetilde{\mathcal{O}}(g(n))$. Für $\varepsilon \in (0,1)$ ist $1/(1-\varepsilon)$ beziehungsweise $1/(1-\sqrt{\varepsilon})$ in $]1,\infty[$. Lassen wir nur Werte für ε und $\sqrt{\varepsilon}$ der Form $(n-1)/n$ beziehungsweise $((n-1)/n)^2$ mit $n \in \mathbb{Z}^{>1}$ zu, dann ist $1/(1-\varepsilon) = n$ beziehungsweise $1/(1-\sqrt{\varepsilon}) = n$ und es ergibt Sinn, die eben beschriebene Notation auch für Funktionen in $1/(1-\varepsilon)$ oder $1/(1-\sqrt{\varepsilon})$ zu verwenden. Diese Festlegung ist aber nur bezüglich asymptotischer Analysen von Interesse, praktisch betrachten wir für ε beliebige reelle Zahlen zwischen 0 und 1.
Sei $\varepsilon \in (0,1)$ und $F/\mathbb{F}_q$ ein Funktionenkörper vom Geschlecht g sowie n eine positive ganze Zahl. Wir werden des Öfteren in dieser Arbeit Laufzeiten von Algorithmen betrachten, die polynomiell in $\log(1/(1-\varepsilon))$, $\log(q)$ und g sind sowie linear in n. Damit meinen wir, dass es Konstanten $k_1, k_2, k_3 \in \mathbb{Z}^{>1}$ gibt, so dass die Laufzeiten $\mathcal{O}((\log(1/(1-\varepsilon)))^{k_1}(\log(q))^{k_2}g^{k_3}n)$ entsprechen, wobei bei dieser Notation $1/(1-\varepsilon)$ wie zuvor aufzufassen ist. Werden die Konstanten nicht näher spezifiziert, so schreiben wir stattdessen $(\log(1/(1-\varepsilon)))^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}g^{\mathcal{O}(1)}\mathcal{O}(n)$.

1.5.2 Algorithmische Repräsentation von Funktionenkörper

Einen algebraischen Funktionenkörper F/K vom Geschlecht g realisieren wir algorithmisch als den Quotientenkörper des Restklassenringes

$$K[x][y]/f(x,y)K[x][y],$$

wobei $f(x,y) = y^n + f_1(x)y^{n-1} + \cdots + f_n(x) \in K[x][y]$ ein irreduzibles Polynom bezeichnet, welches normiert und seperabel in y ist. Wir nennen f mit diesen Eigenschaften ein definierendes Polynom von F. Bezeichne $\deg_x(f)$ und $\deg_y(f)$ den Grad eines Polynoms $f \in K[x][y]$ aufgefaßt als Polynom in x beziehungsweise y.

Lemma 1.29. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper vom Geschlecht $g \geq 1$ und $q = p^k$ mit einer Primzahl p und $k \in \mathbb{Z}^{>0}$. Dann existiert ein definierendes Polynom f aus*

$K[x][y]$ von F, so dass $\deg_x(f) < 2g^2 + 3g(\lceil 2\log_q(2g)\rceil + 1) + (\lceil 2\log_q(2g)\rceil + 1)^2$ *und* $\deg_y(f) < g + \lceil 2\log_q(2g)\rceil + 1$ *ist.*

Beweis. Sei P eine Stelle minimalen Grads von F und $\lambda \in \mathbb{Z}$ minimal gewählt, so dass $\lambda \cdot \deg(\mathrm{P}) \geq 2g$ gilt. Nach [Sti93, S.29 I.5.17.] existiert dann $x \in \mathcal{L}(\lambda\mathrm{P}) \setminus \mathcal{L}((\lambda-1)\mathrm{P})$. Das heißt, es gilt $v_{\mathrm{P}}(x) = -\lambda$. Teilt p die Zahl λ, dann wählen wir anstelle von λ einfach $\lambda+1$ und erhalten $x \in \mathcal{L}((\lambda+1)\mathrm{P}) \setminus \mathcal{L}(\lambda\mathrm{P})$ mit $v_{\mathrm{P}}(x) = -(\lambda+1)$, wobei p nun $(\lambda+1)$ nicht teilt. Gelte also ohne Einschränkung, dass $p \nmid \lambda$. Nach [Sti93, S.128 III.9.2.(a)] ist damit x ein seperierendes Element von F/K.
Wir wählen eine weitere Stelle Q mit minimalem Grad und $\mathrm{Q} \notin \mathrm{supp}((x))$ sowie $\mu \in \mathbb{Z}$ minimal, so dass $\mu \cdot \deg(\mathrm{Q}) \geq g$ ist. Dann existiert nach [Sti93, S.21 I.4.17.] ein Element ρ in $\mathcal{L}(\mu\mathrm{Q}) \setminus K$. Wegen der Wahl von x und ρ ist die Körpererweiterung $F/K(x)$ bei P rein verzweigt und $F/K(\rho)$ bei P unverzweigt. Wegen [Sti93, S.62 III.1.6.] gilt damit $[F : K(x,\rho)] = 1$, also $F = K(x,\rho)$. Da x seperierend ist, existiert ein irreduzibles Polynom $f \in K[x][t]$ mit $f(x,\rho) = 0$, welches in t seperabel ist. Da x ein Element von $\mathcal{L}(\lambda\mathrm{P}) \setminus \mathcal{L}((\lambda-1))$ ist und wegen $\rho \in \mathcal{L}(\mu\mathrm{Q})$, haben wir $(x)_\infty = \lambda\mathrm{P}$ und $(\rho)_\infty = \tilde{\mu}\mathrm{Q}$ mit $\tilde{\mu} \leq \mu$. Nach [Sti93, S.18 I.4.11.] ist

$$\begin{aligned}\deg((x)_\infty) &= \lambda \deg(\mathrm{P}) = [K(x) : K] = \deg_x(f) \text{ und}\\ \deg((\rho)_\infty) &= \tilde{\mu} \deg(\mathrm{Q}) = [K(\rho) : K] = \deg_t(f).\end{aligned}$$

Wegen [Hesed, S.14] sind die Grade der Stelle P und Q kleiner gleich $\lceil 2\log_q(2g)\rceil + 1$. Dann sind wegen der Wahl von λ und μ aber $\lambda \cdot \deg(\mathrm{P}) < 2g + \lceil 2\log_q(2g)\rceil + 1$ und $\tilde{\mu} \cdot \deg(\mathrm{Q}) < g + \lceil 2\log_q(2g)\rceil + 1$. Also ist $\deg_x(f) < 2g + \lceil 2\log_q(2g)\rceil + 1$ und $\deg_t(f) < g + \lceil 2\log_q(2g)\rceil + 1$. Das Polynom f ist noch nicht in t normiert. Habe f die Darstellung

$$f(x,t) = f_n(x)t^n + f_{n-1}(x)t^{n-1} + \cdots + f_0(x).$$

Wir multiplizieren die Gleichung mit $f_n(x)^{n-1}$ und definieren $y := f_n(x)t$. Damit erhalten wir

$$\tilde{f}(x,y) := f_n(x)^{n-1} f(x,t) = y^n + f_{n-1}(x)y^{n-1} + \cdots + f_n(x)^{n-1} f_0(x).$$

Das Polynom $\tilde{f} \in K[x][y]$ ist nun normiert in y. Die Irreduzibilität von f überträgt sich auf $\tilde{f}$. Da f bezüglich t seperabel war, ist es $\tilde{f}$ bezüglich y. Wegen $\tilde{f}(x,\rho) = 0$ bezeichnet also $\tilde{f}$ ein definierendes Polynom von F. Aus

$$\deg_x(\tilde{f}(x,y)) = \deg(f_n(x)^{n-1} f(x,t)) = (n-1)\deg(f_n(x)) + \deg_x(f(x,t))$$

und $n < g + \lceil 2\log_q(2g)\rceil + 1$ sowie $\deg(f_n(x)), \deg_x(f(x,t)) < 2g + \lceil 2\log_q(2g)\rceil + 1$ folgt, dass $\deg_x(\tilde{f}(x,y)) < 2g^2 + 3g(\lceil 2\log_q(2g)\rceil + 1) + (\lceil 2\log_q(2g)\rceil + 1)^2$ ist. Der Grad von $\tilde{f}$ in y ist gleich dem Grad von f in t und damit kleiner als $g + \lceil 2\log_q(2g)\rceil + 1$.

□

In dieser Arbeit ist K der endliche Körper $\mathbb{F}_q$ mit $q = p^k$ Elementen, wobei p eine Primzahl bezeichnet und k eine positive ganze Zahl. Wir nehmen an, dass alle Rechnungen in F bezüglich einer wie eben beschriebenen Darstellung durchgeführt werden. Nach dem letzten Lemma können wir auch voraussetzen, dass $n := [F : K(x)] = \mathcal{O}(g)$ und $C_f := max\{\lceil \frac{\deg(f_i)}{i} \rceil \mid 1 \leq i \leq n\}$ gleich $\mathcal{O}(g^2)$ ist. Detaillierte Informationen zu dieser Darstellung finden sich in [Hes99].

Die Komplexität der in dieser Arbeit beschriebenen Algorithmen werden in Recheneinheiten in $\mathbb{F}_q$, dem endlichen Konstantenkörper des Funktionenkörpers F, beschrieben. Dabei sei vorausgesetzt, dass alle Operationen $+, -, \cdot, /$ und das Testen auf Gleichheit, unabhängig von der Wahl der Elemente in $\mathbb{F}_q$, denselben Aufwand haben. Die auftretenden ganzen Zahlen werden klein sein. Aus diesem Grund werden wir die Kosten des Rechnens damit durch die Kosten für das Rechnen in $\mathbb{F}_q$ beschreiben.

Anstatt mit Divisorenklassen $[D]$ der Klassengruppe C_F^0 rechnen wir mit dessen Repräsentanten. Dazu wird angenommen, dass in dieser Arbeit ausschließlich Repräsentanten D betrachtet werden, die folgende Eigenschaften erfüllen:

1. Für $\mathrm{D} = \sum_i \lambda_i \mathrm{P}_i$ ist $h(\mathrm{D}) := \sum_{\lambda_i > 0} \lambda_i \deg(\mathrm{P}_i) + \sum_{\lambda_i < 0} -\lambda_i \deg(\mathrm{P}_i) = \mathcal{O}(g)$.
2. Die Kardinalität von supp(D) ist gleich $\mathcal{O}(g)$.
3. Das Maximum der Grade der Stellen aus supp(D) ist gleich $\mathcal{O}(g)$.

Sei $k \in \mathbb{Z}$. Statt mit $k[D] = [kD]$ befassen wir uns mit kD. Die Zahlen, die wir später betrachten, werden alle durch die Klassenzahl h_F nach oben beschränkt sein. Wegen Satz 1.18 gilt $h_F = \mathcal{O}(q^g)$. Also ist $h(kD) = \mathcal{O}(gq^g)$. Des Weiteren betrachten wir auch Linearkombinationen D' von Divisoren, die 1. bis 3. erfüllen und maximal $\mathcal{O}(g)$ Summanden haben. Dann ist $\#supp(D') = \mathcal{O}(g^2)$. Mittels [Hesed, Algorithmus 33] berechnen wir einen Divisor A vom Grad 1 mit einer Darstellung $A = l_1\mathrm{P}_1 + l_2\mathrm{P}_2$ und $l_i \deg(\mathrm{P_i}) < (\lceil 2\log_q(2g) \rceil + 1)^2$ mit $i \in \{1, 2\}$. Reduzieren wir kD beziehungsweise D' mittels A, wie in [Hes02, S.13-15] beschrieben, dann erhalten wir Divisoren $\widetilde{D}_1, \widetilde{D}_2 \geq 0$ und $r_1, r_2 \in \mathbb{Z}$ mit $[kD] = [\widetilde{D}_1 + r_1 A]$ und $[D'] = [\widetilde{D}_2 + r_2 A]$. Dabei erfüllt $\widetilde{D}_i + r_i A$ für $i \in \{1, 2\}$ wieder die zuvor beschriebenen drei Eigenschaften. Nach [Hes02, Remark 8.6] und den letzten Überlegungen können wir die Repräsentanten $\widetilde{D}_i + r_i A$ für $i \in \{1, 2\}$ mit einer in g und $\log(q)$ polynomiellen Komplexität berechnen.
In dieser Arbeit werden immer ‚statt kD oder D', die Repräsentanten der selben Divisorenklassen $\widetilde{D}_1 + r_1 A$ beziehungsweise $\widetilde{D}_2 + r_2 A$ betrachtet. Wollen wir zwei Divisorenklassen $[D_1]$ und $[D_2]$ vom Grad 0 addieren oder eine mit einer ganzen Zahl k multiplizieren, müssen wir deren Repräsentanten schlimmstenfalls erst im oberen Sinne reduzieren und dann addieren beziehungsweise mit k multiplizieren. Der Aufwand dafür ist dann polynomiell in g und $\log(q)$. Die Elemente $[D_1]$ und $[D_2]$ sind genau dann gleich, wenn $\deg(D_1) = \deg(D_2)$ und $\dim_{\mathbb{F}_q}(\mathcal{L}(D_1 - D_2)) = 1$ ist. Mit [Hes02, Algorithmus 8.8.]

können wir den Test auf Gleichheit realisieren. Wegen obiger Annahmen und [Hes02, Remark 8.9.] ist die Komplexität dafür ebenfalls polynomiell in g und $\log(q)$. Haben wir für einen Repräsentanten D einer Divisorenklasse vom Grad 0 schon die reduzierte Darstellung $\widetilde{D} + rA$ berechnet, dann ist $[D]$ gleich der Null in C_F^0, wenn $r = 0$ und $\widetilde{D}$ der Nulldivisor ist.

Bemerkung 1.30. *Sei F ein Funktionenkörper vom Geschlecht g über $\mathbb{F}_q$ und C_F^0 dessen Klassengruppe. Der Aufwand für das Rechnen in C_F^0 ist polynomiell in g und* $\log(q)$. *Damit meinen wir das Addieren und Vergleichen von zwei Elementen aus C_F^0 sowie das Multiplizieren einer Divisorenklasse vom Grad 0 mit einer ganzen Zahl.*

Eine zentrale Rolle dieser Arbeit ist das zufällige Wählen von Divisorenklassen vom Grad Null. In [Böt08] wird ein Algorithmus vorgestellt, der genau das für Funktionenkörper $F/\mathbb{F}_q$ vom Geschlecht g realisiert. Dessen Komplexität ist polynomiell in $\log(q)$. Für die Eingabe eines Divisors $A = l_1\mathrm{P}_1 + l_2\mathrm{P}_2$ vom Grad 1, wie zuletzt beschrieben, berechnet der Algorithmus einen Repräsentanten D einer Divisorenklasse der Klassengruppe der Form

$$D = P - dA,$$

wobei P eine Stelle von $F/\mathbb{F}_q$ ist mit $\deg(P) = d = \mathcal{O}(g)$. Der Ausdruck $l_i \deg(\mathrm{P_i}) < (\lceil 2\log_q(2g)\rceil + 1)^2$ ist gleich $\mathcal{O}(g)$ mit $i \in \{1, 2\}$. Also sind die Repräsentanten der zufällig gewählten Divisorenklassen von der Form

$$D = P - dl_1P_1 - dl_2P_2$$

und damit $h(D) = \mathcal{O}(g)$, $\#\operatorname{supp}(D) = \mathcal{O}(1)$ sowie das Maximum der Grade der Stellen von D gleich $\mathcal{O}(g)$. Das rechtfertigt unter anderem die Annahmen dieses Abschnittes.

Kapitel 2

Linear unabhängige Divisorenklassen

In diesem Kapitel sei F ein Funktionenkörper vom Geschlecht $g \geq 1$ über dem endlichen Körper $\mathbb{F}_q$ mit $q = p^k$ Elementen und p eine Primzahl sowie $k \in \mathbb{Z}^{>0}$. Es bezeichne dazu C_F^0 die Klassengruppe von F, h_F die Klassenzahl von F sowie für einen Teiler m der Klassenzahl $C_F^0[m]$ die m-Torsionsgruppe von C_F^0 . Wie in Abschnitt 1.1 beschrieben, fassen wir $C_F^0[m]$ und C_F^0/mC_F^0 als freie $\mathbb{Z}/m\mathbb{Z}$-Moduln auf. Wegen ihrer Isomorphie haben sie den gleichen Rang n, der wegen (1.1) kleiner gleich $2g$ ist. Unser Interesse in diesem Kapitel ist es, einen Algorithmus zu beschreiben, der aus einer Teilmenge von $C_F^0[m]$ die linear unabhängigen Elemente bestimmt. Wir beschränken uns dabei auf den Fall, dass $m = l$ ein Primteiler der Klassenzahl von F ist. Damit werden $C_F^0[l]$ und C_F^0/lC_F^0 zu $\mathbb{F}_l$-Vektorräumen. Im dritten Kapitel wollen wir Basen von Erzeugendensystemen bestimmter Untervektorräume von $C_F^0[l]$ berechnen, also ist die letzte Restriktion hinsichtlich des folgenden Kapitels sinnvoll. Zunächst betrachten wir den Fall, dass l und p verschieden sind. Am Ende dieses Kapitels beschreiben wir den Fall $l = p$.

2.1 Der Fall l ist ungleich p

Definition 2.1. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper und m ein Teiler der Klassenzahl. Die kleinste positive ganze Zahl $d(q, m)$ mit*

$$m \mid (q^{d(q,m)} - 1)$$

nennen wir den Einbettungsgrad von $F/\mathbb{F}_q$ bezüglich m. Ist klar welcher Funktionenkörper $F/\mathbb{F}_q$ und welches m gemeint sind, sagen wir einfach Einbettungsgrad.

Definition und Satz 2.2. *Sei F ein Funktionenkörper über $\mathbb{F}_q$ und m ein Teiler dessen Klassenzahl, der koprim zu q ist mit $d(q, m) = 1$, dann existiert eine nicht-ausgeartete*

bilineare Abbildung

$$T_m : C_F^0[m] \times C_F^0/mC_F^0 \to \mathbb{F}_q^\times/(\mathbb{F}_q^\times)^m.$$

Diese Abbildung heißt Tate-Lichtenbaum Paarung.

Beweis. Die Gruppe $\mathbb{F}_q^\times$ hat die Ordnung $q-1$ und ist zyklisch. Also existiert $\beta \in \mathbb{F}_q$ mit $\mathbb{F}_q^\times = \langle\beta\rangle$. Wegen $d(q,m) = 1$ teilt m die Ordnung von $\mathbb{F}_q^\times$. Dann ist $\gamma := \beta^{\frac{q-1}{m}}$, eine m-te primitive Einheitswurzel, in $\mathbb{F}_q$ enthalten und damit auch die Menge der m-ten Einheitswurzeln. Nach [Hes04, Theorem 3] folgt, dass T_m nicht-ausgeartet ist. □

Da $\mathbb{F}_q^\times$ eine zyklische Gruppe der Ordnung $q-1$ ist, gilt

$$\mathbb{F}_q^\times/(\mathbb{F}_q^\times)^m \cong \mathbb{Z}/c\mathbb{Z} \text{ mit } c := ggT(q-1,m).$$

Für $c > 1$ bezeichnet $\mathbb{Z}/c\mathbb{Z}$ einen nicht-trivialen Teilring von $\mathbb{Z}/m\mathbb{Z}$. Ist l ein Primteiler von h_F mit $\mathrm{ggT}(q,l) = d(q,l) = 1$, dann gilt insbesondere $\mathrm{ggT}(q-1,l) = l$ und wir können T_l als $\mathbb{F}_l$-bilineare Abbildung auf dem $\mathbb{F}_l$-Vektorraum $C_F^0[l] \times C_F^0/lC_F^0$ auffassen.

Wir benötigen ein Kriterium um auszusagen, welche Elemente einer Teilmenge von $C_F^0[l]$ linear unabhängig sind. Dazu betrachten wir zunächst den Fall $d(q,l) = 1$ und anschließend $d(q,l) > 1$.

2.1.1 Einbettungsgrad von $F/\mathbb{F}_q$ bezüglich l ist 1

Für $l = ggT(q-1,l)$ fassen wir T_l als eine nicht-ausgeartete $\mathbb{F}_l$-bilineare Abbildung auf. Wir werden nun erst einmal allgemeine Überlegungen zu $\mathbb{K}$-Vektorräumen V, W und einer auf $V \times W$ nicht-ausgearteten $\mathbb{K}$-bilinearen Abbildung f machen.

Satz 2.3. *Seien ein Körper $\mathbb{K}$ und $\mathbb{K}$-Vektorräume V, W der Dimension $n > 0$ sowie eine nicht-ausgeartete $\mathbb{K}$-bilineare Abbildung $f : V \times W \to \mathbb{K}$ gegeben. Sind V_1 und W_1 Untervektorräume von V beziehungsweise W der Dimension $0 < k \leq n$, so dass $f_{|_{V_1 \times W_1}}$ nicht-ausgeartet ist, dann existiert zu jedem Untervektorraum V_2 mit $V_1 \subseteq V_2 \subseteq V$ ein Untervektorraum $W_2 \subseteq W$, so dass W_1 in W_2 enthalten und $f_{|_{V_2 \times W_2}}$ nicht-ausgeartet ist.*

Beweis. Für $k = n$ müssen wir nichts beweisen. Gelte also $k < n$. Sei $B_V := \{v_1, \ldots, v_n\}$ eine Basis von V und $B_W := \{w_1, \ldots, w_n\}$ eine Basis von W. Es gelte ohne Einschränkung $V_1 = \mathrm{Spann}(\{v_1, \ldots, v_k\})$ und $W_1 = \mathrm{Spann}(\{w_1, \ldots, w_k\})$. Nun genügt es zu zeigen, dass für beliebiges $v \in \{v_{k+1}, \ldots, v_n\}$ ein $w_i \in \{w_{k+1}, \ldots, w_n\}$ existiert, so dass f eingeschränkt auf $\mathrm{Spann}(\{v_1, \ldots, v_k, v\}) \times \mathrm{Spann}(\{w_1, \ldots, w_k, w_i\})$ nicht-ausgeartet ist.
Angenommen die Behauptung wäre falsch, dann existiert für alle $i \in \{k+1, \ldots, n\}$ zur

Menge $B_i := \{w_1, \ldots, w_k, w_i\}$ ein $0 \neq x_i \in \text{Spann}(\{v_1, \ldots, v_k, v\})$ mit $f(x_i, b) = 0$ für alle $b \in B_i$. Es gilt $\bigcap_{i=k+1}^{n} B_i = \{w_1, \ldots, w_k\}$, folglich ist

$$\{w_1, \ldots, w_k\} \subset \text{Kern}(f(x_i, \cdot)) \text{ für } i = k+1, \ldots, n. \tag{2.1}$$

Hätte ein x_i keine v-Komponente, dann wäre $x_i \in V_1$ und somit $f_{|_{V_1 \times W_1}}$ im Widerspruch zur Voraussetzung ausgeartet. Also haben alle x_i einen nicht-trivialen v-Anteil. Betrachte x_i und x_j für beliebige $i, j \in \{k+1, \ldots, n\}$ mit $i \neq j$. Es gibt ein α aus $\mathbb{K}$, so dass $x_i - \alpha x_j \in V_1$ ist. Wegen (2.1) und der Bilinearität der Abbildung f gilt $f(x_i - \alpha x_j, b) = 0$ für alle $b \in W_1$. Nach Voraussetzung ist aber $f_{|_{V_1 \times W_1}}$ nicht-ausgeartet, also muss $x_i - \alpha x_j = 0$ beziehungsweise $x_i = \alpha x_j$ sein. Da die Wahl von i und j beliebig war, sind $x_{k+1}, \ldots, x_n \in \text{Spann}(\{v_s\})$ für ein $0 \neq v_s \in V_1$. Nun gilt für $x_i \neq x_j$ mit $B_i \subseteq Kern(f(x_i, \cdot))$ und $B_j \subseteq Kern(f(x_j, \cdot))$ sowie $x_i, x_j \in \text{Spann}(\{v_s\})$, dass $B_i \cup B_j \subseteq Kern(f(v_s, \cdot))$. Also ist $\bigcup_{i=k+1}^{n} B_i = \{w_1, \ldots, w_n\} = B_W \subseteq Kern(f(v_s, \cdot))$. Das steht im Widerspruch dazu, dass $f_{|_{V \times W}}$ nicht-ausgeartet ist. Demnach war die Annahme falsch und die Aussage ist bewiesen. □

Satz 2.4. *Seien $\mathbb{K}$ ein Körper und V, W $\mathbb{K}$-Vektorräume der Dimension $n > 0$ mit $B := \{v_1, \ldots, v_n\} \subset V$, $C := \{w_1, \ldots, w_n\} \subset W$ gegeben. Bezeichne $f : V \times W \to \mathbb{K}$ eine $\mathbb{K}$-bilineare Abbildung. Wenn*

$$\det\left(f(v_i, w_j)_{i,j=1..n}\right) \neq 0$$

gilt, sind die Teilmengen B und C Basen von V beziehungsweise W. Ist zusätzlich f nicht-ausgeartet, dann gilt auch die Umkehrung.

Beweis. Wir definieren $M := (f(v_i, w_j)_{i,j=1..n})$. Da $\det(M) \neq 0$ gilt, sind die Elemente von C ungleich 0. Angenommen C sei keine Basis, dann gibt es ohne Einschränkung $\lambda_1, \ldots, \lambda_{n-1}$ aus $\mathbb{K}$ mit $w_n = \sum_{i=1}^{n-1} \lambda_i w_i$. Wegen der Bilinearität von f erhalten wir äquivalent

$$(f(v_j, w_n)_{j=1,\ldots,n}) = \sum_{i=1}^{n-1} \lambda_i \left(f(v_j, w_i)_{j=1,\ldots,n}\right). \tag{2.2}$$

Das heißt, die Spalten der Matrix M sind linear abhängig und damit deren Determinante im Widerspruch zur Voraussetzung gleich Null. Analog kann man zeigen, dass die Zeilen von M linear abhängig sind, wenn B keine Basis ist.
Sei f nicht-ausgeartet sowie C und D Basen von V beziehungsweise W. Angenommen es gilt $\det(M) = 0$. Dass heißt, die Spalten von M sind linear abhängig und es gilt ohne Einschränkung äquivalent zu (2.2)

$$\sum_{i=1}^{n-1} \lambda_i \left(f(v_j, w_i)_{j=1,\ldots,n}\right) - \left(f(v_j, w_n)_{j=1,\ldots,n}\right) = 0 \in \mathbb{K}^n.$$

Wir definieren $w := \sum_{i=1}^{n} \lambda_i w_i - w_n$ und erhalten $f(v_1, w) = \cdots = f(v_n, w) = 0$ mit $w \neq 0$, da C eine Basis ist. Nach Voraussetzung ist f nicht-ausgeartet und $B = \{v_1, \ldots, v_n\}$ eine Basis, also liegt ein Widerspruch vor. □

Kriterium

Sei F wieder ein globaler Funktionenkörper über $\mathbb{F}_q$ mit $q = p^k$. Wir betrachten einen zu p verschiedenen Primteiler l der Klassenzahl h_F, so dass der Einbettungsgrad von F bezüglich l eins ist. Wir wenden nun die Resultate des letzten Teilabschnittes auf die speziellen $\mathbb{F}_l$-Vektorräume $C_F^0[l]$ und C_F^0/lC_F^0 sowie die nicht-ausgeartete bilineare Abbildung T_l an. Für ein Element $[D]+lC_F^0 \in C_F^0/lC_F^0$ schreiben wir im Folgenden der Einfachheit halber immer nur einen Repräsentanten $[D] \in C_F^0$ hin. Wir meinen aber, wenn es nicht explizit erwähnt wird, stets die Klasse in C_F^0/lC_F^0.

Korollar und Definition 2.5. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper und l ein Primteiler von h_F mit $\mathrm{ggT}(q,l) = d(q,l) = 1$. Bezeichne $D := \{[D_1], \ldots, [D_\kappa]\}$ eine Menge von Divisorenklassen vom Grad 0 und $E := \{[E_1], \ldots, [E_s]\}$ eine Teilmenge von C_F^0/lC_F^0.*

1. *Ist $s = \kappa$ und gilt $d_l(D,E) := \det(T_l([D_i],[E_j])_{i,j=1,\ldots,\kappa}) \neq 0$, dann sind sowohl die Elemente von D als auch von E linear unabhängig.*
2. *Wenn D eine Menge linear unabhängiger Elemente aus $C_F^0[l]$ ist und E ein Erzeugendensystem von C_F^0/lC_F^0, dann existiert eine Teilmenge $\widetilde{E}$ von E mit $d_l(D,\widetilde{E}) \neq 0$.*

Beweis. Nach Satz 2.2 ist die Tate-Lichtenbaum Paarung T_l mit den Voraussetzungen des Satzes wohldefiniert und eine auf $C_F^0[l] \times C_F^0/lC_F^0$ nicht-ausgeartete $\mathbb{F}_l$-bilineare Abbildung. Definiere $V_1 := \mathrm{Spann}(D)$ und $W_1 := \mathrm{Spann}(E)$. Offensichtlich ist $f_{|_{V_1 \times W_1}}$ $\mathbb{F}_l$-bilinear. Wegen $d_l(D,E) \neq 0$ folgt mit Satz 2.4 die erste Behauptung.
Nach Satz 2.3 gibt es ein Untervektorraum W von C_F^0/lC_F^0, so dass T_l eingeschränkt auf $\mathrm{Spann}(D) \times W$ nicht-ausgeartet ist. Da $\mathrm{Spann}(E) = C_F^0/lC_F^0$ gilt, existieren linear unabhängige $[E_1], \ldots, [E_\kappa] \in E$ mit $W = \mathrm{Spann}(\{[E_1], \ldots, [E_\kappa]\})$. Definieren wir $\widetilde{E}$ als die Menge dieser Elemente, dann folgt die zweite Behauptung. □

Nun haben wir ein Kriterium zur Überprüfung der linearen Unabhängigkeit von Elementen aus $C_F^0[l]$. Zuletzt folgt noch ein Korollar, welches grundlegend für den anschließenden Algorithmus zur Berechnung dieser ist.

Korollar 2.6. *Sei F ein Funktionenkörper über $\mathbb{F}_q$ und l ein Primteiler der Klassenzahl von F mit $\mathrm{ggT}(q,l) = d(q,l) = 1$. Bezeichne E ein Erzeugendensystem von C_F^0/lC_F^0 und existiere ein Untervektorraum V von $C_F^0[l]$ der Dimension $\kappa \geq 2$. Weiter sei für $1 \leq s < \kappa$ die Menge B_s bestehend aus den linear unabhängigen Elementen $[D_1], \ldots, [D_s]$ aus $C_F^0[l]$ zusammen mit der Menge $D \subseteq C_F^0[l]$ ein Erzeugendensystem von V. Gilt für $\{[E_1], \ldots, [E_s]\} \subseteq E$*

$$d_l(\{[D_1], \ldots, [D_s]\}, \{[E_1], \ldots, [E_s]\}) \neq 0,$$

dann gibt es $[D_{s+1}] \in D$ und $[E_{s+1}] \in E$ mit der Eigenschaft

$$d_l(\{[D_1], \ldots, [D_{s+1}]\}, \{[E_1], \ldots, [E_{s+1}]\}) \neq 0.$$

Beweis. Wegen $\kappa > s$ gibt es ein $[D_{s+1}] \in D$, so dass $\{[D_1], \ldots, [D_{s+1}]\}$ linear unabhängig sind. Nach Satz 2.3 existiert ein Untervektorraum W von C_F^0/lC_F^0 der Dimension $s+1$, so dass T_l eingeschränkt auf $\mathrm{Spann}(\{[D_1], \ldots, [D_{s+1}]\}) \times W$ nicht-ausgeartet ist und $\{[E_1], \ldots, [E_s]\} \subset W \subseteq \mathrm{Spann}(E)$. Nach Satz 2.5 sind $[E_1], \ldots, [E_s]$ linear unabhängig. Das heißt, $W \setminus \mathrm{Spann}(\{[E_1], \ldots, [E_s]\})$ hat die Dimension 1. Da E ein Erzeugendensystem von C_F^0/lC_F^0 ist, existiert $[E_{s+1}] \in E$ mit $\mathrm{Spann}([E_{s+1}]) = W \setminus \mathrm{Spann}(\{[E_1], \ldots, [E_s]\})$. Nach Satz 2.4 folgt dann die Behauptung.

□

2.1.2 Der Algorithmus

Wir betrachten den Funktionenkörper $F/\mathbb{F}_q$ und den Primteiler l der Klassenzahl von F mit $\mathrm{ggT}(q,l) = d(q,l) = 1$. Sei $n := \dim_{\mathbb{F}_l}(C_F^0[l]) = \dim_{\mathbb{F}_l}(C_F^0/lC_F^0)$ und $D := \{[D_1], \ldots, [D_{n_1}]\}$ eine Teilmenge von $C_F^0[l]$ sowie $E := \{[E_1], \ldots, [E_{n_2}]\}$ ein Erzeugendensystem von C_F^0/lC_F^0. Weiter sei $B_s := [D_{i_1}], \ldots, [D_{i_s}] \subseteq C_F^0[l]$ eine Menge aus linear unabhängigen Elementen. Wir definieren $V := \mathrm{Spann}(D \cup B_s)$ und $\kappa := \dim_{\mathbb{F}_l}(V)$, dabei gilt $s \leq \kappa \leq n$. Wir wollen nun einen Algorithmus beschreiben, der aus D mit Hilfe von B_s eine Basis von V bestimmt.
Für $s = \kappa$ ist nichts zu berechnen, also betrachten wir $s < \kappa$. Ist $s = 0$ oder $\kappa = 1$, dann wählen wir ein Element $[D_{i_1}]$ aus D ungleich Null und definieren $s := 1$ sowie $B_s := \{[D_{i_1}]\}$ (für $s = \kappa = 1$ sind wir fertig). Nach Korollar 2.5.2. gibt es $\widetilde{B}_s := \{[E_{i_1}], \ldots, [E_{i_s}]\} \subseteq E$ mit

$$d_l(B_s, \widetilde{B}_s) \neq 0.$$

Weil $s < \kappa$ ist, existiert ein $[D_{i_{s+1}}] \in D$, so dass $\{[D_{i_1}], \ldots, [D_{i_{s+1}}]\}$ linear unabhängig sind. Wegen Korollar 2.6 gibt es dazu $[E_{i_{s+1}}] \in E \setminus \widetilde{B}_s$ mit

$$d_l(B_{s+1}, \widetilde{B}_{s+1}) \neq 0,$$

wobei $B_{s+1} := \{[D_{i_1}], \ldots, [D_{i_{s+1}}]\}$ und $\widetilde{B}_{s+1} := \{[E_{i_1}], \ldots, [E_{i_{s+1}}]\}$ ist. Wir setzen s gleich $s+1$. Ist $s < \kappa$ wiederholen wir den letzten Schritt. Nach insgesamt $\kappa - s$ Schritten haben wir $B_\kappa := \{[D_{i_1}], \ldots, [D_{i_\kappa}]\}$ und $\widetilde{B}_\kappa := \{[E_{i_1}], \ldots, [E_{i_\kappa}]\}$ mit

$$d_l(\widetilde{D}_\kappa, \widetilde{E}_\kappa) \neq 0.$$

Nach Korollar 2.5.1. ist B_κ eine Basis von V und die Elemente aus $\widetilde{B}_\kappa$ sind linear unabhängig.
Sei V ein Untervektorraum von $C_F^0[l]$. Der Algorithmus bekommt als Eingabe die Mengen $E := \{[E_1], \ldots, [E_{n_1}]\}$ und $B := \{[B_1], \ldots, [B_s]\}$ aus $C_F^0[l]$, wobei die Elemente aus B linear unabhängig sind und $V = \mathrm{Spann}(B \cup E)$ gilt. Des Weiteren benötigt der Algorithmus die beiden Teilmengen $\widetilde{B} := \{[\widetilde{B}_1], \ldots, [\widetilde{B}_s]\}$ und $\widetilde{E} := \{[\widetilde{E}_1], \ldots, [\widetilde{E}_{n_2}]\}$ von C_F^0/lC_F^0. Dabei sind die Elemente der ersten Menge linear unabhängig mit $d_l(B, \widetilde{B}) \neq 0$ und $\mathrm{Spann}(\widetilde{B} \cup \widetilde{E}) = C_F^0/lC_F^0$. Die Mengen B und $\widetilde{B}$ können auch leer sein.

Algorithmus 1 : Berechnung linear unabhängiger Elemente in $C_F^0[l]$ mit $l \neq p$

Eingabe: Teilmengen $B, E \subseteq C_F^0[l]$ und $\widetilde{B}, \widetilde{E} \subseteq C_F^0/lC_F^0$, wie eben beschrieben, und $\kappa \in \mathbb{N}$ eine obere Schranke von $\dim_{\mathbb{F}_l}(V)$.

Ausgabe: Basis von V und $\dim_{\mathbb{F}_l}(V)$ linear unabhängige Elemente aus C_F^0/lC_F^0, so dass T_l eingeschränkt auf das Kreuzprodukt derer Erzeugnisse nicht-ausgeartet ist.

1: **for** $[E_i] \in E$ **do**
2: **if** $\#B = \kappa$ **then**
3: **return** B, $\widetilde{B}$
4: **end if**
5: $B \leftarrow B \cup \{[E_i]\}$
6: **for** $[\widetilde{E}_j] \in \widetilde{E}$ **do**
7: $\widetilde{B} \leftarrow \widetilde{B} \cup \{[\widetilde{E}_j]\}$
8: **if** $d_l(B, \widetilde{B}) \neq 0$ **then**
9: $\widetilde{E} \leftarrow \widetilde{E} \setminus \{[\widetilde{E}_j]\}$ und gehe zu 1 mit dem nächsten i
10: **else**
11: $\widetilde{B} \leftarrow \widetilde{B} \setminus \{[\widetilde{E}_j]\}$
12: **end if**
13: **end for**
14: $B \leftarrow B \setminus \{[E_i]\}$
15: **end for**
16: **return** B, $\widetilde{B}$

Satz 2.7. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper vom Geschlecht g und l ein zu q teilerfremder Primteiler der Klassenzahl von F mit $d(q, l) = 1$. Für Erzeugendensysteme $B \cup E$ eines Untervektorraumes V von $C_F^0[l]$ und $\widetilde{B} \cup \widetilde{E}$ von C_F^0/lC_F^0, wie zuletzt definiert, berechnet Algorithmus 1 eine Basis von V. Die Komplexität des Algorithmus ist gleich*

$$g^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(\#E\#\widetilde{E}).$$

Beweis. Da sowohl $\#E$ als auch $\#\widetilde{E}$ endlich sind, terminiert der Algorithmus nach endlich vielen Schritten. Aus den Überlegungen zuvor folgt unmittelbar die Korrektheit von Algorithmus 1.
Nach (1.1) gilt $\dim_{\mathbb{F}_l}(C_F^0[l]) \leq 2g$. Um $d_l(B, \widetilde{B})$ zu bestimmen, müssen wir maximal $(2g)^2$-mal die Tate-Lichtenbaum Paarung auswerten und die Determinante einer $2g \times 2g$-Matrix über $\mathbb{F}_l$ berechnen. Seien $[D]$ und $[E]$ Elemente aus $C_F^0[l]$ beziehungsweise C_F^0/lC_F^0. Mittels [Fra07, Algorithmus 3.3.] können wir $x \in F^\times$ berechnen, so dass die Divisoren $D + (x)$ und E koprim sind. Mit [Hes02, Algorithmus 8.5.] erhalten wir eine reduzierte Darstellung von $l(D + (x))$ und damit ein Element $a \in F^\times$ mit der Eigenschaft $l(D + (x)) = (a)$. Nun können wir die Tate-Lichtenbaum Paarung durch $T_l(D + (x), E) := a(E) = \prod_{P \in \mathrm{supp}(E)} N_{F_P/\mathbb{F}_q}(a(P))^{v_P(E)} \in \mathbb{F}_q^\times/(\mathbb{F}_q^\times)^l$ berechnen. Hierbei ist $N_{F_P/\mathbb{F}_q}(a(P))$ gleich der Determinante des $\mathbb{F}_q$-Vektorraumendomorphismus,

welcher der Multiplikation mit $a(P)$ entspricht. Für $\alpha := T_l(D+(x), E)$ müssen wir noch $\phi(\alpha)$ berechnen, wobei ϕ einen Isomorphismus zwischen $\mathbb{F}_q^\times/(\mathbb{F}_q^\times)^l$ und $\mathbb{F}_l$ bezeichnet. Wir realisieren ϕ auf die folgende Weise: Sei z ein Erzeuger der zyklischen Gruppe $\mathbb{F}_q^\times$ und $\gamma := \frac{q-1}{l}$. Wir definieren $\tilde{z} := z^\gamma$ und betrachten $\phi(\alpha) := \log_{\tilde{z}}(\alpha^\gamma)$, wobei $\log_a(b)$ den diskreten Logarithmus von b zur Basis a bezeichnet (Informationen dazu finden sich in [Buc03]). Eigentlich bildet ϕ in die Menge $\{0, \ldots, l-1\}$ ab. Wir fassen diese Menge als ein Vertretersystem von $\mathbb{F}_l$ auf.
In [Fra07, Algorithmus 3.3.] müssen Riemann-Roch-Räume von Divisoren D berechnet werden, für die $h(D) = \mathcal{O}(g)$ angenommen werden kann. Nach [Hes02, Remark 8.9.] ist die Komplexität dafür, wenn wir einen Divisor A vom Grad 1 wählen, logarithmisch in $h(D)$, linear in $\#\operatorname{supp}(D)$ und polynomiell in g und β, wobei β das Maximum der in $\operatorname{supp}(D)$ enthaltenen Stellen bezeichnet. Da die Komplexität zur Berechnung der Riemann-Roch-Räume die von [Fra07, Algorithmus 3.3.] dominiert, können wir teilerfremde D und E mit der eben beschriebenen Komplexität berechnen. Nach [Hes02, Remark 8.6] ist der Aufwand zur Berechnung von a ebenfalls wie eben. Zur Auswertung von a in E müssen wir Ganzheitsbasen von Dedekindringen (eine Definition kann in [Neu92, S.19] nachgelesen werden) und Normen beziehungsweise Determinanten in den Restklassenringen berechnen. Der Aufwand dafür ist polynomiell in g und $\log(q)$. Zur Berechnung von $\phi(\alpha)$ müssen wir einen diskreten Logarithmus in der von $\tilde{z}$ erzeugten Gruppe mit l Elementen bestimmen. Benutzen wir den Pohling-Hellman-Algorithmus, dann ist nach [Buc03, Theorem 11.5.4.] die Komplexität dafür gleich $\mathcal{O}(\log(l))$, da l eine Primzahl bezeichnet. Nach Satz 1.18 ist $l \leq h_{\mathbb{F}} \leq (\sqrt{q}+1)^{2g}$, also $\mathcal{O}(\log(l)) = \mathcal{O}(g\log(q))$. Letztendlich ist die Komplexität um $T_l(D, E)$ als Element in $\mathbb{F}_l$ zu berechnen wegen der Generalvoraussetzung aus Teilabschnitt 1.5.2 bezüglich der Darstellung der Repräsentanten von Elementen von C_F^0, polynomiell in g und $\log(q)$. Die Determinante einer $2g \times 2g$-Matrix über $\mathbb{F}_l$ können wir mit einem in g polynomiellen Aufwand bestimmen. Damit ist der Aufwand um $d_l(B, \widetilde{B})$ zu berechnen gleich $g^{\mathcal{O}(1)}\log(q)^{\mathcal{O}(1)}$. Maximal $\#E\#\widetilde{E}$ mal muss d_l berechnet werden. Also ist die Gesamtkomplexität von Algorithmus 1 gleich $g^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(\#E\#\widetilde{E})$. □

Bemerkung 2.8. *In Algorithmus 1 berechnen wir in jeder Schleife*

$$d_l(\{[B_1], \ldots, [B_s], [E_{i_1}] \ldots, [E_{i_a}]\}, \{[\widetilde{B}_1], \ldots, [\widetilde{B}_s], [\widetilde{E}_{i_1}], \ldots, [\widetilde{E}_{i_a}]\}).$$

Also bestimmen wir die Determinante einer Matrix M_a der Form

$$\begin{pmatrix}
T_m([B_1],[\widetilde{B}_1]) & \cdots & T_m([B_1],[\widetilde{B}_s]) & T_m([B_1],[\widetilde{E}_{i_1}]) & \cdots & T_m([B_1],[\widetilde{E}_{i_a}]) \\
\vdots & & \vdots & \vdots & & \vdots \\
T_m([B_s],[\widetilde{B}_1]) & \cdots & T_m([B_s],[\widetilde{B}_s]) & T_m([B_s],[\widetilde{E}_{i_1}]) & \cdots & T_m([B_s],[\widetilde{E}_{i_a}]) \\
T_m([E_{i_1}],[\widetilde{B}_1]) & \cdots & T_m([E_{i_1}],[\widetilde{B}_s]) & T_m([E_{i_1}],[\widetilde{E}_{i_1}]) & \cdots & T_m([E_{i_1}],[\widetilde{E}_{i_a}]) \\
\vdots & & \vdots & \vdots & & \vdots \\
T_m([E_{i_a}],[\widetilde{B}_1]) & \cdots & T_m([E_{i_a}],[\widetilde{B}_s]) & T_m([E_{i_a}],[\widetilde{E}_{i_1}]) & \cdots & T_m([E_{i_a}],[\widetilde{E}_{i_a}])
\end{pmatrix}.$$

Um im darauffolgenden Durchlauf die Determinante einer Matrix M_{a+1} zu berechnen, genügt es, M_a um die Spalte

$$(T_m([B_1],[\widetilde{E}_{i_{a+1}}]),\ldots,T_m([B_s],[\widetilde{E}_{i_{a+1}}]),T_m([E_{i_1}],[\widetilde{E}_{i_{a+1}}]),\ldots,T_m([E_{i_{a+1}}],[\widetilde{E}_{i_{a+1}}]))^T$$

und die Zeile

$$(T_m([E_{i_{a+1}}],[\widetilde{B}_1]),\ldots,T_m([E_{i_{a+1}}],[\widetilde{B}_s]),T_m([E_{i_{a+1}}],[\widetilde{E}_{i_1}]),\ldots,T_m([E_{i_{a+1}}],[\widetilde{E}_{i_{a+1}}]))$$

zu erweitern. Das heißt, die Anzahl der Auswertungen der Tate-Lichtenbaum Paarung ist in jeder Schleife von Algorithmus 1 kleiner gleich $4g-1$. Damit können wir Algorithmus 1 beschleunigen.

2.1.3 Der Einbettungsgrad von F bezüglich m ist ungleich 1

Die Tate-Lichtenbaum Paarung ist nur dann auf $C_F^0[l]\times C_F^0/lC_F^0$ definiert, wenn $l|h_F$ und $\mathrm{ggT}(q,l)=d(q,l)=1$ gilt. Wir betrachten nun den Fall $d(q,l)\neq 1$ ist. Bezeichne dabei l wieder einen zu q teilerfremden Primteiler der Klassenzahl von F.

Sei F' eine Konstantenkörpererweiterung von F vom Grad $d(q,l)$ und bezeichne $C_{F'}^0$ die Klassengruppe von F', dann kann C_F^0 als eine Untergruppe von $C_{F'}^0$ aufgefaßt werden. Also können wir $C_F^0[l]$ und C_F^0/lC_F^0 als Untervektorräume von $C_{F'}^0[l]$ beziehungsweise $C_{F'}^0/lC_{F'}^0$ interpretieren. Die Repräsentanten einer Divisorenklasse vom Grad 0 von F sind formale Linearkombinationen von Stellen von $F/\mathbb{F}_q$. Betrachten wir eine Repräsentation des Funktionenkörpers wie in Teilabschnitt 1.5.2, dann können wir die Erzeuger dieser Stellen durch Quotienten von Polynomen aus $\mathbb{F}_q[x][y]$ darstellen. Durch das Einbetten der Polynome aus $\mathbb{F}_q[x][y]$ in $\mathbb{F}_{q^{d(q,l)}}[x][y]$ sind wir in der Lage die Divisoren vom Grad 0 aus C_F^0 in $C_{F'}^0$ einzubetten. Oder anders ausgedrückt, erhalten wir darüber $\mathbb{F}_l$-lineare Einbettungen von $C_F^0[l]$ und C_F^0/lC_F^0 nach $C_{F'}^0[l]$ beziehungsweise $C_{F'}^0/lC_{F'}^0$.

Zunächst wollen wir eine obere Schranke für den Einbettungsgrad $d(q,l)$ bestimmen.

Lemma 2.9. *Seien $q,l\in\mathbb{Z}^{>0}$ mit $\mathrm{ggT}(q,l)=1$, dann teilt l*

$$q^{\varphi(l)}-1.$$

Dabei bezeichnet φ die Eulersche φ-Funktion.

Beweis. Es gilt $\#(\mathbb{Z}/l\mathbb{Z})^\times=\varphi(l)$. Das heißt, für $a\in(\mathbb{Z}/l\mathbb{Z})^\times$ ist $a^{\varphi(l)}\equiv 1 \bmod l$. Wegen $\mathrm{ggT}(q,l)=1$ ist $q\not\equiv 0 \bmod l$ und damit $q^{\varphi(l)}\equiv 1 \bmod l$. □

Satz 2.10. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper und l ein zu q koprimer Teiler der Klassenzahl von F. Dann ist der Einbettungsgrad $d(q,l)$ von F bezüglich l kleiner gleich $\varphi(l)$.*

Beweis. Die Behauptung folgt unmittelbar aus Lemma 2.9. □

Bemerkung 2.11.

(i) Mittels $d(q,l) := \min\{l \mid (q^j - 1) \mid 1 \leq j \leq \varphi(l)\}$ kann man den Einbettungsgrad eines Funktionenkörpers $F/\mathbb{F}_q$ bezüglich eines Teilers l der Klassenzahl von F definieren und auch einfach berechnen.

(ii) Für einen Primteiler l der Klassenzahl gilt $d(q,l) \leq l - 1$. Folglich ist der Einbettungsgrad eines Funktionenkörpers $F/\mathbb{F}_q$ bezüglich $l = 2$ immer gleich 1.

Sei für einen Primteiler $l \neq p$ der Klassenzahl V ein Untervektorraum von $C_F^0[l]$ und F' eine Konstantenkörpererweiterung von $F/\mathbb{F}_q$ vom Grad $d(q,l)$. Statt des $\mathbb{F}_l$-Untervektorraumes V betrachten wir den Untervektorraum V' von $C_{F'}^0[l]$, der entsteht, indem wir V in $C_{F'}^0[l]$ einbetten, etwa über die Einbettung ψ. Haben wir ein Erzeugendensytem E von V, so ist $E' := \psi(E)$ eines von V'. Offensichtlich haben V und V' die gleiche Dimension. Um mittels Algorithmus 1 eine Basis B von V zu bestimmen, berechnen wir eine Basis B' aus E' von V'. Dann ist $B := \psi^{-1}(B')$ eine Basis von V. Zur Berechnung von B' benötigen wir für Algorithmus 1 noch ein Erzeugendensystem von $C_{F'}^0/lC_{F'}^0$. Angemerkt sei hier noch, dass im Allgemeinen die Dimension von C_F^0/lC_F^0 kleiner gleich der von $C_{F'}^0/lC_{F'}^0$ ist. Dennoch sind beide Dimensionen durch $2g$ nach oben beschränkt.
Wir können also mittels Algorithmus 1 Basen von Untervektorräumen von $C_F^0[l]$ berechnen auch wenn $d(q,l) > 1$ ist, indem wir die Berechnung nach $C_{F'}^0[l]$ verlagern. In diesem Sinne ist auch der folgende Satz zu verstehen.

Satz 2.12. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper vom Geschlecht g und l ein Primteiler der Klassenzahl von F mit $d(q,l) > \mathrm{ggT}(q,l) = 1$. Es bezeichne F' eine Konstantenkörpererweiterung von F vom Grad $d(q,l)$. Für Erzeugendensysteme $B' \cup E'$ eines Untervektorraumes V' von $C_{F'}^0[l]$ und $\widetilde{B}' \cup \widetilde{E}'$ von $C_{F'}^0/lC_{F'}^0$, analog zu Satz 2.7, berechnet Algorithmus 1 eine Basis B' von V'. Die Komplexität dafür ist gleich*

$$g^{\mathcal{O}(1)} l^{\mathcal{O}(1)} \log(q)^{\mathcal{O}(1)} \mathcal{O}(\#E' \# \widetilde{E}').$$

Beweis. Die Korrektheit folgt aus den Bemerkungen zuvor. Da wir alle Rechnungen über $\mathbb{F}_{q^{d(q,l)}}$ anstatt über $\mathbb{F}_q$ durchführen und nach Satz 2.7 die Anzahl der Elemente des Konstantenkörpers logarithmisch in die Komplexität von Algorithmus 1 eingeht, ergibt sich, wegen $d(q,l) \leq \varphi(l) = l - 1$, die beschriebene Komplexität. □

2.2 Der Fall $l = p$

Sind l und q nicht koprim, dann ist die Tate-Lichtenbaum Paarung nicht definiert und das eben beschriebene Verfahren zur Bestimmung von linear unabhängigen Elementen

der l-Torsionsgruppe der Klassengruppe nicht anwendbar. Wir erklären nun ein alternatives Verfahren zur Bestimmung von linear unabhängigen Elementen aus $C_F^0[p]$. Für detaillierte Informationen verweisen wir auf [Hes99, Abschnitt 3.4].

Nach [Hes99, S.49] ist $C_F^0[p] \cong \Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$, wobei

$$\Phi : C_F^0[p] \longrightarrow \Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q) \text{ mit } pD = (a) \longmapsto (1/a)da$$

den $\mathbb{F}_p$-Vektorraumisomorphismus bezeichnet. Sei x ein seperierendes Element von $F/\mathbb{F}_q$. Nach [Lan73, S.311] können wir jedes $ydx \in \Omega(F/\mathbb{F}_q)$ schreiben als

$$ydx = (y_0^p + y_1^p x + \cdots + y_{p-1}^p x^{p-1})dx$$

mit $y_0, \ldots, y_{p-1} \in F$.

Definition 2.13. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper mit $q = p^k$ und x ein seperierendes Element von $F/\mathbb{F}_q$, dann heißt die Abbildung*

$$C : \Omega(F/\mathbb{F}_q) \longrightarrow \Omega(F/\mathbb{F}_q) \text{ mit } C(ydx) = y_{p-1}dx$$

der Cartier-Operator von $F/\mathbb{F}_q$.

Satz 2.14. *Es sei F ein Funktionenkörper über dem endlichen Körper $\mathbb{F}_q$ mit $q = p^k$ Elementen, dann gilt:*

(i) Der Cartier-Operator von $F/\mathbb{F}_q$ ist $\mathbb{F}_p$-linear und

(ii) $\mathrm{Kern}((1 - C)_{|\Omega^0(F/\mathbb{F}_q)}) = \Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$.

Beweis. Erklärungen und Verweise auf Beweise sind in [Hes99, S.49-50] zu finden. □

Nach dem letzten Satz können wir mit den Mitteln der linearen Algebra eine Basis des $\mathbb{F}_p$-Vektorraumes $\Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$ berechnen. Haben wir eine solche Basis B von $\Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$, dann ist die Vorgehensweise die Folgende:
Zunächst müssen wir die Bilder der Divisorenklassen $[\widetilde{D}_1], \ldots, [\widetilde{D}_s]$ aus $C_F^0[p]$ unter Φ bestimmen. Anschließend können wir darüber die Koordinatenvektoren $v_1, \ldots, v_s$ der Divisorenklassen $[\widetilde{D}_1], \ldots, [\widetilde{D}_s]$ in $\mathbb{F}_p^n$ mit $n := \#B$ berechnen und diese einfach auf lineare Unabhängigkeit untersuchen.
Der folgende Algorithmus realisiert das zuletzt beschriebene Verfahren.

Algorithmus 2 : Berechnung linear unabhängiger Elemente in $C_F^0[p]$

Eingabe: E Erzeugendensystem eines Untervektorraumes V von $C_F^0[p]$, $\widetilde{B}$ Basis eines Untervektorraumes von V oder die leere Menge, B Basis von $\Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$ und $\kappa \in \mathbb{Z}^{>0}$ eine obere Schranke von $\dim_{\mathbb{F}_p}(V)$.
Ausgabe: Basis von V.

1: $j \leftarrow \#\widetilde{B}$
2: **if** $j > 0$ **then**
3: $M \leftarrow (v_1, \ldots, v_j)$, $v_i \in \mathbb{F}_p^{\#B}$ Koordinatenvektoren der Elemente aus $\widetilde{B}$ sind
4: **else**
5: $M \leftarrow ()$
6: **end if**
7: **for** $[D] \in E$ **do**
8: **if** $j = \kappa$ **then**
9: **return** B
10: **end if**
11: Berechne $a \in F^\times$ mit $pD = (a)$
12: Berechne Koordinatenvektor $v \in \mathbb{F}_p^{\#B}$ von $(1/a)da$ bzgl. B
13: **if** $Rang(M, v) > j$ **then**
14: $M \leftarrow (M, v)$, $j \leftarrow j + 1$, $B \leftarrow B \cup \{[D]\}$
15: **end if**
16: **end for**
17: **return** B

Satz 2.15. *Sei F ein Funktionenkörper über dem endlichen Körper $\mathbb{F}_q$. Weiter sei B eine Basis von $\Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$ und E ein Erzeugendensystem eines $\mathbb{F}_p$-Untervektorraumes V von $C_F^0[p]$ sowie $\widetilde{B}$ eine Basis eines $\mathbb{F}_p$-Untervektorraumes von V. Dann berechnet Algorithmus 2 eine Basis von V. Dessen Komplexität ist gleich*

$$(\log(q))^{\mathcal{O}(1)} g^{\mathcal{O}(1)} \mathcal{O}(\#E).$$

Beweis. Die Korrektheit des Algorithmus resultiert aus den Bemerkungen zuvor. Das er terminiert, ist offensichtlich. Mit den Annahmen für die Repräsentanten D der Divisorenklassen vom Grad 0 in dieser Arbeit, können wir, wie im Beweis von Satz 2.7 beschrieben, $pD = (a)$ mit einem in g polynomiellen Aufwand berechnen. Die Anzahl der Rechenoperationen, um den Koordinatenvektor von $(1/a)da$ bezüglich B zu bestimmen, ist polynomiell in g und $\log(q)$.
Die Matrix M hat wegen $\dim_{\mathbb{F}_p}(C_F^0[p]) \leq g$ maximal g viele Zeilen und Spalten. Der Aufwand, um den Rang der Matrix zu bestimmen, ist nach [vzGG03, S.705] gleich $\mathcal{O}(t^3)$, wobei t das Maximum der Anzahl der Zeilen und Spalten von M bezeichnet. Folglich ist der Aufwand zur Berechnung des Ranges von M polynomiell in g. Zusammenfassend ist also die Komplexität von Algorithmus 2 gleich $(\log(g))^{\mathcal{O}(1)} g^{\mathcal{O}(1)} \mathcal{O}(\#E)$. □

Kapitel 3

Berechnung der Divisorenklassengruppe

Sei F ein Funktionenkörper vom Geschlecht g über $\mathbb{F}_q$ mit $q = p^k$ und bezeichne p eine Primzahl sowie k eine positive ganze Zahl. In diesem Kapitel beschreiben wir zuerst einen probabilistischen Algorithmus und anschließend einen nicht-probabilistischen zur Berechnung der Divisorenklassengruppe C_F von F. Nach [Hes99, S.3] gilt $C_F \cong C_F^0 \oplus \langle [D] \rangle$, wobei $[D]$ ein beliebiger Divisor vom Grad 1 ist. Die Existenz eines solchen Divisors wird durch [Sti93, S.164] garantiert. Haben wir einen Divisor vom Grad 1, dann genügt es also die Struktur der Klassengruppe C_F^0 von F zu berechnen. Damit meinen wir die Gruppeninvarianten $c_1, \ldots, c_n \in \mathbb{Z}^{>0}$ mit

$$C_F^0 \cong (\mathbb{Z}/c_1\mathbb{Z}) \oplus \cdots \oplus (\mathbb{Z}/c_n\mathbb{Z})$$

und den Divisoren $D_1, \ldots, D_n$ vom Grad 0, so dass $\{[D_1], \ldots, [D_n]\}$ ein minimales Erzeugendensystem der Gruppe C_F^0 ist. Wir nehmen für den Rest der Arbeit an, dass die Klassenzahl h_F von F bekannt ist, das heißt, wir kennen die Ordnung von C_F^0. Bevor wir einen groben Überblick der Idee der Algorithmen geben, betrachten wir zunächst die zu diesem Thema relevanten Verfahren.

3.1 Bestehende Algorithmen

1. [BS05, Algorithm 2]
2. [Tes98, Algorithm 2.1]
3. [Hes99, Algorithmus 5.5]

Ausgehend von einem Erzeugendensytem S einer beliebigen abelschen Gruppe G, dessen Gruppenoperationen bekannt sind, berechnet sowohl [BS05, Algorithm 2] als auch

[Tes98, Algorithm 2.1] die Struktur von G im zuvor beschriebenen Sinn. Für [Tes98, Algorithm 2.1] benötigen wir zusätzlich eine Funktion $f : G \longrightarrow \{1, \ldots, 20\}$ mit

$$\sum_{i=1}^{20} \left| \#\{a \in G \mid f(a) = i\} - \frac{\#G}{20} \right| = \mathcal{O}(\sqrt{\#G}), \tag{3.1}$$

welche effizient berechenbar ist. Diese Forderung ist heuristischer Natur. Wir stellen kurz eine praktische Realisierung einer solchen Funktion f im Fall $G = C_F^0$ vor:
Wählen wir einen Divisor A vom Grad 1, dann können wir jede Divisorenklasse aus C_F^0 eindeutig durch $D - \deg(D)A$ repräsentieren, wobei D einen Divisor minimalen Grads von F mit $D \geq 0$ bezeichnet. Indem wir dem Bitstring der eindeutig dargestellten Divisorenklasse mittels einer Hashfunktion (zum Beispiel SHA-1, zu finden in [Buc03, S.197]) eine ganze Zahl modulo 20 zuweisen, erhalten wir eine Abbildungsvorschrift, die heuristisch der Bedingung (3.1) genügt. Für detailliertere Informationen zur Realisierung und zum Zweck einer solchen Funktion f verweisen wir auf [Tes98, S. 1643].

Mit [Hes99, Algorithmus 5.5] wird ausschließlich die Struktur der Divisorenklassengruppe beziehungsweise der Klassengruppe von globalen Funktionenkörpern berechnet. Als Eingabe benötigt er aber kein Erzeugendensystem der Klassengruppe.

Im Anschluss beschreiben wir Verfahren, welche mit Hilfe der Tate-Lichtenbaum Paarung die Struktur der Klassengruppe eines globalen Funktionenkörpers berechnen. Wir wollen im Abschnitt 3.8 und Kapitel 4 die Algorithmen [BS05, Algorithm 2], [Tes98, Algorithm 2.1] und [Hes99, Algorithmus 5.5] mit den neuentwickelten vergleichen. Dazu listen wir die Komplexität der schon bekannten Algorithmen auf. Nach Bemerkung 1.30 ist die Komplexität sowohl für den Vergleich zweier Elemente aus C_F^0 als auch deren Addition sowie das Multiplizieren von Divisorenklassen vom Grad 0 mit ganzen Zahlen polynomiell in g und $\log(q)$. Das heißt, wir können die Gruppenoperationen in C_F^0 mit einem in g und $\log(q)$ polynomiellen Aufwand ausführen. Da nach Satz 1.18 die Klassenzahl $h_F = \#C_F^0 = \mathcal{O}(q^g)$ ist, ergeben sich die folgenden Komplexitäten der Algorithmen zur Berechnung der Klassengruppe eines Funktionenkörpers $F/\mathbb{F}_q$ vom Geschlecht g:

1. Nach [BS05, Theorem 1.1] ist die Komplexität von [BS05, Algorithm 2] gleich

   $$\mathcal{O}(\#S(\sqrt{q})^g). \tag{3.2}$$

   Die Anzahl der zu speichernden Elemente aus C_F^0 entspricht $\mathcal{O}(\#S(\sqrt{q})^g)$.

2. Nach [Tes98, Theorem 1.1] ist die erwartete Komplexität von Algorithmus [Tes98, Algorithm 2.1] ebenfalls gleich

   $$\mathcal{O}(\#S(\sqrt{q})^g). \tag{3.3}$$

   Die Anzahl der zu speichernden Elemente aus C_F^0 entspricht $\mathcal{O}(\#S)$.

3. Nach [Hes99, Satz 5.23] ist die erwartete Komplexität von Algorithmus [Hes99, Algorithmus 5.5] gleich

$$\exp\left(\sqrt{g\log(g)}\right)^{\left(5\sqrt{\log(q)/6}+\mathcal{O}(1)\right)}. \tag{3.4}$$

   Die Anzahl der zu speichernden Elemente aus C_F^0 entspricht $\mathcal{O}\left(gq^{\log_q(g)}\right)$.

Die jeweiligen Annahmen, die zu den erwarteten Komplexitäten von [Tes98, Algorithm 2.1] und [Hes99, Algorithmus 5.5] führen, können in [Tes98, s.1650-1656] beziehungsweise [Hes99, S.72] nachgelesen werden.

3.2 Idee der Algorithmen

Jede endliche abelsche Gruppe ist die direkte Summe ihrer l-Sylow-Gruppen. Beschreibe $C_F^0(l)$ die l-Sylow-Gruppe der Klassengruppe von F für einen Primteiler l der Klassenzahl h_F. Ist $h_F = \prod_i^n l_i^{e_i}$ die Primfaktorzerlegung der Klassenzahl, dann gilt also

$$C_F^0 = C_F^0(l_1) \oplus \cdots \oplus C_F^0(l_n).$$

Im Folgenden beschreiben wir die Grundidee der Algorithmen zur Berechnung der Struktur einer l-Sylow-Gruppen von C_F^0. Kennen wir die Struktur jeder l-Sylow-Gruppe der Klassengruppe, dann können wir durch die letzte Überlegung daraus auf die Struktur der gesamten Klassengruppe schließen und damit, wie Eingangs erwähnt, auf die der Divisorenklassengruppe von F.
Teile l^s mit $s \geq 1$ die Klassenzahl exakt, das heißt $l^s \mid h_F$ und $l^{s+1} \nmid h_F$. Nach dem Hauptsatz über endlich erzeugte abelsche Gruppen gilt

$$C_F^0(l) \cong \bigoplus_{i=1}^{z} (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$$

mit $0 \leq \delta_i$ für $i = 1, \ldots, z$. Wegen $\#C_F^0(l) = l^s$ gilt $\prod_{i=1}^z l^{i\delta_i} = l^s$ beziehungsweise $\sum_{i=1}^z i\delta_i = s$. Unser Interesse ist es, die Werte δ_i für $i = 1, \ldots, z$ zu berechnen, denn daraus erhalten wir die Gruppeninvarianten von $C_F^0(l)$. Zunächst noch folgende Definitionen und Überlegungen.

Wie wir in Abschnitt 1.1 beschrieben haben, können wir $C_F^0[m]$ als $\mathbb{Z}/m\mathbb{Z}$-Modul auffassen. In diesem Fall ist $m = l$ eine Primzahl, damit $\mathbb{F}_l := \mathbb{Z}/l\mathbb{Z}$ ein Körper und $C_F^0[l]$ ein $\mathbb{F}_l$-Vektorraum. Es gilt

$$C_F^0[l] \cong \bigoplus_{i=1}^{z} (\mathbb{Z}/l\mathbb{Z})^{\delta_i} = \mathbb{F}_l^n,$$

wobei $n := \dim_{\mathbb{F}_l}(C_F^0[l]) = \sum_{i=1}^{z} \delta_i \leq 2g$ ist. Wir definieren die Menge

$$U_i := \{l^{(i-1)}[D] \mid [D] \in C_F^0(l) \text{ mit } \mathrm{Ord}([D]) = l^i\} \cup \{[0]\} \text{ für } i \in \{1, \ldots, z\}. \tag{3.5}$$

und $U_{z+1} := \{\}$. Offensichtlich können wir die Mengen U_i als $\mathbb{F}_l$-Untervektorräume von $C_F^0[l]$ auffassen. Sie erfüllen

$$U_i \cong \bigoplus_{j=i}^{z} (\mathbb{Z}/l\mathbb{Z})^{\delta_i} = \mathbb{F}_l^{n_i}$$

mit $n_i := \dim_{\mathbb{F}_l}(U_i) = \sum_{j=i}^{z} \delta_j$. Die $\mathbb{F}_l$-Untervektorräume U_i mit $i \in \{1, \ldots, z\}$ von $C_F^0[l]$ bilden eine Inklusionskette. Es gilt $U_z \subseteq U_{z-1} \subseteq \cdots \subseteq U_1 = C_F^0[l]$. Setzen wir n_{z+1} gleich Null, dann erhalten wir

$$\dim_{\mathbb{F}_l}(U_i \setminus U_{i+1}) = n_i - n_{i+1} = \sum_{j=i}^{z} \delta_j - \sum_{j=i+1}^{z} \delta_j = \delta_i \tag{3.6}$$

für alle i. Folglich brauchen wir nur die Dimensionen n_i der Untervektorräume U_i bestimmen und bekommen darüber für i den Wert δ_i, also die Gruppeninvarianten von $C_F^0(l)$.
Nun können wir die Idee der in diesem Kapitel beschriebenen Algorithmen vorstellen.

Zuerst berechnen wir den Wert z, das Maximum von $\{\log_l(\mathrm{Ord}([D])) \mid [D] \in C_F^0(l)\}$. Ist z bekannt, dann ermitteln wir für jedes i den Wert δ_i sowie ein minimales Erzeugendensystem von $C_F^0(l)$ auf die folgende Weise:
Wir bestimmen für $i \in \{1, \ldots, z\}$, beginnend mit $i = z$, eine Teilmenge $\{[D_{i_1}], \ldots, [D_{i_m}]\}$ von Elementen aus $C_F^0(l)$ der Ordnung l^i, so dass die Menge

$$E_i := \{l^{i-1}[D_{i_1}], \ldots, l^{i-1}[D_{i_m}]\} \subseteq C_F^0[l]$$

ein Erzeugendensystem des $\mathbb{F}_l$-Untervektorraumes $U_i \setminus U_{i+1}$ ist. Mittels Algorithmus 1 und Algorithmus 2 aus Kapitel 2 bestimmen wir eine Basis B_i dieses Untervektorraumes der Form $B_i := \{l^{i-1}[D_{i_1}], \ldots, l^{i-1}[D_{i_{\delta_i}}]\}$. Spätestens nach z Schritten entsteht wegen der Inklusionskette der Vektorräume U_i eine Basis B_l von $C_F^0[l]$ der Form

$$B_l := B_z \cup \cdots \cup B_1 = \{l^{i-1}[D_{j_i}] \mid i \in \{1, \ldots, z\} \text{ und } j_i \in \{1, \ldots, \delta_i\}\}.$$

Da die Elemente $[D_{j_i}]$ für $i \in \{1, \ldots, z\}$ und $j_i \in \{1, \ldots, \delta_i\}$ die Ordnung l^i haben, bildet

$$\widetilde{B}_l := \{[D_{j_i}] \mid i \in \{1, \ldots, z\} \text{ und } j_i \in \{1, \ldots, \delta_i\}\} \tag{3.7}$$

ein minimales Erzeugendensystem der Gruppe $C_F^0(l)$.

Seien l_1 und l_2 verschiedene Primteiler der Klassenzahl. Für $i \in \{1, 2\}$ bezeichne

$\widetilde{B}_{l_i} := \{[D_{i_1}], \ldots, [D_{i_{\kappa_i}}]\}$ ein minimales Erzeugendensystem von $C_F^0(l_i)$, dessen Elemente bezüglich ihrer Ordnung aufsteigend sortiert sind. Ist ohne Einschränkung $\kappa_1 > \kappa_2$, dann definieren wir

$$\widetilde{B}_{l_1} + \widetilde{B}_{l_2} := \{[D_{1_1}], \ldots, [D_{1_{\kappa_1-\kappa_2}}], [D_{1_{\kappa_1-\kappa_2+1}} + D_{2_1}], \ldots, [D_{1_{\kappa_1}} + D_{2_{\kappa_2}}]\}. \tag{3.8}$$

Für $\kappa_1 = \kappa_2$ setzen wir $\widetilde{B}_{l_1} + \widetilde{B}_{l_2}$ gleich $\{[D_{1_i} + D_{2_i}] \mid i \in \{1, \ldots, \kappa_1\}\}$. Damit ist $\widetilde{B}_{l_1} + \widetilde{B}_{l_2}$ ein minimales Erzeugendensystem von $C_F^0(l_1) \oplus C_F^0(l_2)$.

Bemerkung 3.1. *Haben wir für jeden Primteiler l_i der Klassenzahl h_F ein minimales Erzeugendensystem $\widetilde{B}_{l_i}$ von $C_F^0(l_i)$, wie in (3.7), dann ist gemäß der in (3.8) definierten Addition*

$$\sum_i \widetilde{B}_{l_i}$$

ein minimales Erzeugendensystem der Klassengruppe.

3.3 Dimensionsschranken der Untervektorräume U_i

Für einen Primteiler l der Klassenzahl h_F des globalen Funktionenkörpers F vom Geschlecht g betrachten wir die l-Sylow-Gruppe $C_F^0(l) \cong \bigoplus_{i=1}^z (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ der Klassengruppe C_F^0. Sei $\#C_F^0(l)$ gleich l^s und $s \geq 1$. Wir wollen eine obere Schranke der Dimension $n_i := \sum_{j=i}^z \delta_j$ der in (3.5) definierten $\mathbb{F}_l$-Untervektorräume U_i von $C_F^0[l]$ bestimmen. Diese sind für spätere Betrachtungen von großer Bedeutung. Wir setzen $\delta_{z+1} := 0$ und definieren $s_i := s - \sum_{j=i+1}^z j\delta_j$ für $i \in \{1, \ldots, z\}$ sowie α_i durch die eindeutige Darstellung von

$$s_i = s - \sum_{j=i+1}^{z} j\delta_j = i\alpha_i + \beta_i \text{ mit } \beta_i < i.$$

Satz 3.2. *Für $i \in \{1, \ldots, z\}$ gilt $\delta_i \leq \alpha_i$.*

Beweis. Wegen der Isomorphie von $C_F^0(l)$ und $\bigoplus_{i=1}^z (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ bedeutet die Gleichung $s_i = s - \sum_{j=i+1}^z j\delta_j = \sum_{j=1}^i j\delta_j = i\alpha_i + \beta_i$ mit $\beta_i < i$, dass in $\bigoplus_{j=1}^i (\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ maximal $(\mathbb{Z}/l^i\mathbb{Z})^{\alpha_i}$ enthalten ist. Folglich muss $\delta_i \leq \alpha_i$ gelten. □

Nach (3.6) ist $n_i - n_{i+1} = \delta_i$ für $i \in \{1, \ldots, z\}$ oder äquivalent $n_i = n_{i+1} + \delta_i$. Da wir n_i von $i = z$ beginnend bis $i = 1$ berechnen werden, kennen wir im i-ten Schritt n_{i+1}. Mit dem letzten Satz erhalten wir

$$\begin{aligned} & n_{i+1} \leq n_i \leq n_{i+1} + \alpha_i \\ \iff & n_{i+1} \leq n_i \leq n_{i+1} + \lfloor \frac{s_i}{i} \rfloor. \end{aligned} \tag{3.9}$$

Zusätzlich benötigen wir noch eine adäquate obere Schranke der Dimension der Vektorräume $C_F^0[l] \setminus U_{i+1}$ für $i \in \{1, \ldots, z\}$. Nach (1.1) gilt $n := \dim_{\mathbb{F}_l}(C_F^0[l]) = \sum_{j=1}^{z} \delta_j \leq 2g$. Für $i < z$ erhalten wir

$$\dim_{\mathbb{F}_l}(C_F^0[l] \setminus U_{i+1}) = n - n_{i+1} = \sum_{j=1}^{z} \delta_j - \sum_{j=1}^{i+1} \delta_j = \sum_{j=1}^{i} \delta_j.$$

Lemma 3.3. *Für $i \in \{1, \ldots, z\}$ und $\delta_i \geq 1$ gilt*

$$s_i - i \geq n - \sum_{j=i+1}^{z} \delta_j - 1.$$

Beweis. Sei $i \in \{1, \ldots, z\}$ und $\delta_{z+1} := 0$. Wir führen den Beweis durch Widerspruch und nehmen an, dass $s_i - i < n - \sum_{j=i+1}^{z} \delta_j - 1$ gilt. Äquivalent erhalten wir

$$\begin{aligned}
& s - \sum_{j=i+1}^{z} j\delta_j - i < n - \sum_{j=i+1}^{z} \delta_j - 1 \\
\iff & \sum_{j=1}^{z} j\delta_j - \sum_{j=i+1}^{z} j\delta_j - i < \sum_{j=1}^{z} \delta_j - \sum_{j=i+1}^{z} \delta_j - 1 \\
\iff & \sum_{j=1}^{i} j\delta_j - i < \sum_{j=1}^{i} \delta_j - 1 \\
\iff & \sum_{j=1}^{i-1} j\delta_j + i(\delta_i - 1) < \sum_{j=1}^{i-1} \delta_j + \delta_i - 1
\end{aligned}$$

Nach Voraussetzung war aber $\delta_i - 1 \geq 0$ und damit ist die letzte Ungleichung widersprüchlich. □

Korollar 3.4. *Für $i \in \{1, \ldots, z-1\}$ ist $\sum_{j=1}^{i} \delta_j \leq s_i - i + 1$. Im Fall $i = z$ gilt $n \leq \min\{2g, s - z + 1\}$.*

Beweis. Sei $i \in \{1, \ldots, z-1\}$. Nach Lemma 3.3 ist $s_i - i \geq n - \sum_{j=i+1}^{z} \delta_j - 1$ oder äquivalent

$$s_i - i + 1 \geq n - \sum_{j=i+1}^{z} \delta_j = \sum_{j=1}^{z} \delta_j - \sum_{j=i+1}^{z} \delta_j = \sum_{j=1}^{i} \delta_j.$$

Wegen $s_z = s$ und $n \leq 2g$ folgt die Behauptung. □

Bemerkung 3.5. *Für den Fall $l = p$ ist $n := \dim_{\mathbb{F}_p}(C_F^0[p]) = \sum_{j=1}^{z} \delta_j \leq g$. Wir definieren s_i wie zuvor. Damit bleiben die eben beschriebenen Schranken für n_i mit $i \in \{1, \ldots, z\}$ auch für $l = p$ korrekt. Wenn B eine Basis von $\Omega^0(F/\mathbb{F}_q) \cap \Omega(F/\mathbb{F}_q)$ ist, dann folgt mit den Überlegungen aus Abschnitt 2.2, dass n und $\#B$ übereinstimmen. Damit können wir für jedes $i \in \{1, \ldots, z\}$ die Dimension $\dim_{\mathbb{F}_p}(C_F^0[p] \backslash U_{i+1}) = \sum_{j=1}^{i} \delta_j$ durch $\#B - \sum_{j=i+1}^{z} \delta_j$ exakt bestimmen.*

3.4 Probabilistischer Algorithmus

In diesem Abschnitt beschreiben wir einen probabilistischen und randomisierten Algorithmus zur Berechnung der Klassengruppe eines Funktionenkörpers $F/\mathbb{F}_q$ vom Geschlecht g mit $q = p^k$. Wie immer bezeichne dabei p eine Primzahl und k eine positive ganze Zahl. Für einen Primteiler l der Klassenzahl h_F von F sei $C_F^0(l)$ isomorph zu $\bigoplus_{i=1}^{z} (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ sowie $U_i := \{l^{(i-1)}[D] \mid [D] \in C_F^0(l) \text{ und } \mathrm{Ord}([D]) = l^i\} \cup \{[0]\}$ beziehungsweise $U_{z+1} := \{\}$. Zur Beschreibung geben wir folgende Unterteilung vor, dabei hängen alle Betrachtungen von einer Wahrscheinlichkeit $\varepsilon \in (0, 1)$ ab:

1. Berechnung des Maximums der Menge $\{\log_l(\mathrm{Ord}([D])) \mid [D] \in C_F^0(l)\}$.
2. Wahrscheinlichkeitstheoretische Grundlagen zur Bestimmung von Erzeugendensystemen der Untervektorräume $U_i \setminus U_{i+1}$ für $i \in \{1, \ldots, z\}$.
3. Berechnung von Basen der Untervektorräume $U_i \setminus U_{i+1}$ für $i \in \{1, \ldots, z\}$.
4. Berechnung von $C_F^0(l)$.
5. Berechnung von C_F^0 als Zusammenfassung der vorhergehenden Resultate.

3.4.1 Maximale Ordnung in $C_F^0(l)$

Wir suchen ein Element $[D]$ in $C_F^0(l)$, dessen Ordnung maximal ist. Haben wir ein solches Element gefunden, dann ist $z = \log_l(\mathrm{Ord}([D]))$. Sei $\varepsilon \in (0, 1)$ und $[D_1], \ldots, [D_k]$ zufällig gewählte Divisorenklassen vom Grad 0 aus $C_F^0(l)$ sowie $S_k := \{[D_1], \ldots, [D_k]\}$. Nach [ER65, Theorem 2] erzeugt S_k für

$$k \geq \log(\#C_F^0(l)) + 2\log(1/(1-\varepsilon)) + \log(\log(\#C_F^0(l))) + 5 =: B(\#C_F^0(l), \varepsilon)$$

die abelsche Gruppe $C_F^0(l)$ mit Wahrscheinlichkeit größer gleich ε. Ist S ein Erzeugendensystem einer endlichen abelschen Gruppe G, dann beinhaltet es auch ein Element maximaler Ordnung in G. Das heißt, wählen wir k als $\lceil B(\#C_F^0(l), \varepsilon) \rceil$, dann beinhaltet S_k ein Element maximaler Ordnung in $C_F^0(l)$ mit Wahrscheinlichkeit $\geq \varepsilon$. Offensichtlich ist $\mathcal{O}(B(\#C_F^0(l), \varepsilon)) = \mathcal{O}(\log(\#C_F^0(l)) + \log(1/(1-\varepsilon))$. Da $\#C_F^0(l) \leq h_F$ ist und nach Satz 1.18 die Klassenzahl durch $(\sqrt{q}+1)^{2g}$ beschränkt wird, gilt insgesamt

$$\lceil B(\#C_F^0(l), \varepsilon) \rceil = \mathcal{O}\left(g\log(q) + (\log(1/(1-\varepsilon)))\right). \tag{3.10}$$

Nun können wir einen Algorithmus beschreiben, der für $C_F^0(l) \cong \bigoplus_{i=1}^{z}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ probabilistisch den Wert z berechnet.

Algorithmus 3 : Maximale Ordnung in $C_F^0(l)$

Eingabe: Klassengruppe C_F^0, Primzahl l und $s \in \mathbb{Z}^{\geq 1}$ mit $h_F = l^s c$ und $\mathrm{ggT}(l, c) = 1$ sowie $\varepsilon \in (0, 1)$.

Ausgabe: $z \in \mathbb{Z}^{>0}$, Maximum der Menge $\{\log_l(\mathrm{Ord}([D])) \mid [D] \in C_F^0(l)\}$ mit Wahrscheinlichkeit $\geq \varepsilon$ korrekt.

```
1: z ← 0
2: for i = 1, ..., ⌈B(l^s, ε)⌉ do
3:    Wähle zufällig ein Element [D_i] aus C_F^0,  j ← 0
4:    while l^j c · [D_i] ≠ [0] do {Berechnung der Ordnung von [D_i]}
5:       j ← j + 1
6:    end while
7:    if j>z then
8:       z ← j
9:    end if
10: end for
11: return z
```

Satz 3.6. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper vom Geschlecht g und l ein Primteiler von h_F sowie $\varepsilon \in (0, 1)$. Algorithmus 3 berechnet das Maximum der Menge $\{\log_l(\mathrm{Ord}([D])) \mid [D] \in C_F^0(l)\}$ mit Wahrscheinlichkeit größer gleich ε korrekt. Die Anzahl der benötigten Rechenoperationen ist gleich*

$$g^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(\log(1/(1-\varepsilon))).$$

Beweis. Für eine zufällig gewählte Divisorenklasse $[D_i]$ vom Grad 0 ist wegen $h_F = l^s c$ mit $\mathrm{ggT}(l, c) = 1$ die Divisorenklasse $c[D_i] = [cD_i]$ ein zufällig gewähltes Element aus $C_F^0(l)$. Mit den vorherigen Überlegungen folgt damit die Korrektheitsaussage des Satzes. Der Aufwand für den Test $l^j c[D_i] = [0]$ ist nach Bemerkung 1.30 polynomiell in g und $\log(q)$. Da wir j minimal mit $l^j c[D_i] = [0]$ bestimmen, folgt $j \leq s \leq \log_l(h_F)$. Wegen Satz 1.18 gilt dann $j = \mathcal{O}(g\log(q))$. Die Anzahl der Tests in der While-Schleife ist demnach linear in g und $\log(q)$. Nach [Böt08, S.99] ist der Aufwand zur Berechnung einer zufälligen Divisorenklasse vom Grad 0 polynomiell in $\log(q)$. Wegen (3.10) folgt dann die Komplexitätsaussage des Satzes. □

3.4.2 Wahrscheinlichkeit der Erzeugendensystemen

Eine zentrale Rolle des probabilistischen Algorithmus zur Berechnung von C_F^0 wird die Bestimmung von Basen der Untervektorräume $U_i \setminus U_{i+1}$ mit $i \in \{1, \ldots, z\}$ sein. Über die Basen erhalten wir sowohl die Gruppeninvarianten als auch ein minimales

Erzeugendensystem von $C_F^0(l)$. Sei ε eine reelle Zahl zwischen 0 und 1. Wir werden nun die wahrscheinlichkeitstheoretischen Rahmenbedingungen schaffen, die notwendig sind, um in Abhängigkeit von ε Erzeugendensysteme E_i der Untervektorräume $U_i \setminus U_{i+1}$ für $i \in \{1, \ldots, z\}$ zu bestimmen.

Kombinatorische Vorbemerkung

Um mit Algorithmus 1 aus Kapitel 2 für einen Primteiler $l \neq p$ der Klassenzahl eine Basis aus einem Erzeugendensystem E_i eines Untervektorraumes von $C_F^0[l]$ zu berechnen, benötigen wir zusätzlich ein Erzeugendensystem des $\mathbb{F}_l$-Vektorraumes C_F^0/lC_F^0. Die Dimension n von C_F^0/lC_F^0 ist gleich der Dimension von $C_F^0[l]$. Wegen $U_i \cong \mathbb{F}_l^{n_i}$ mit $n_i \leq n$ und $C_F^0/lC_F^0 \cong \mathbb{F}_l^n$ beschreiben wir als Erstes mit $V := \mathbb{F}_l^n$ die Wahrscheinlichkeit für folgende Ereignisse:

1. Unter j vielen zufällig gewählten Elementen aus V ist ein Erzeugendensystem von V.
2. So wie 1. zusammen mit der Annahme, dass schon m viele linear unabhängige Elemente aus V gegeben sind, wobei $m < n$ ist.

Nach der zufälligen Wahl eines Elements, legen wir es immer zur Menge zurück.

Lemma 3.7. *Sei $V := \mathbb{F}_l^n$ mit $n \geq 1$ und $j \in \mathbb{Z}^{>0}$ mit $j \leq n$, dann gibt es*

$$\prod_{i=0}^{j-1}(l^n - l^i)$$

viele Möglichkeiten für $v_1, \ldots, v_j \in V$, so dass $\dim_{\mathbb{F}_l}(\mathrm{Spann}(\{v_1, \ldots, v_j\})) = j$ gilt.

Beweis. Wir definieren das Erzeugnis der leeren Menge als den Nullvektorraum. Damit $v_1, \ldots, v_j$ in V linear unabhängig sind, muss

$$\begin{aligned} &v_1 \notin \mathrm{Spann}(\{\}),\\ &v_2 \notin \mathrm{Spann}(\{v_1\}),\\ &v_3 \notin \mathrm{Spann}(\{v_1, v_2\}),\\ &\vdots\\ &v_j \notin \mathrm{Spann}(\{v_1, \ldots, v_{j-1}\}) \end{aligned}$$

gelten. Für $1 \leq i \leq j$ und linear unabhängige $v_1, \ldots, v_{i-1}$ ist die Anzahl der Möglichkeiten, dass $v_i \notin \mathrm{Spann}(\{v_1, \ldots, v_{i-1}\})$ gilt, wegen $\#\,\mathrm{Spann}(\{v_1, \ldots, v_{i-1}\}) = l^{i-1}$ und $\#\mathbb{F}_l^n = l^n$ gerade $l^n - l^{i-1}$. Da i beliebig war, ist die Gesamtanzahl an Möglichkeiten gleich dem Produkt $\prod_{i=0}^{j-1}(l^n - l^i)$. □

Definition und Satz 3.8. *Sei $V := \mathbb{F}_l^n$ mit $n \geq 1$ und $j \in \mathbb{Z}^{>0}$ mit $j \geq n$, dann gibt es*

$$H_1(n,j,l) := \prod_{i=0}^{n-1}(l^j - l^i)$$

viele Möglichkeiten für $v_1, \ldots, v_j \in V$, so dass $\text{Spann}(\{v_1, \ldots, v_j\}) = V$ *gilt.*

Beweis. Wir definieren die $n \times j$-Matrix $M := (v_1, \ldots, v_j)$. Bezeichne M^T die transponierte Matrix von M. Die Vektoren $v_1, \ldots, v_j$ erzeugen genau dann den Vektorraum V, wenn der Rang von M gleich n, der Dimension von V, ist.
Wegen $M \in \mathbb{F}_l^{n \times j}$ folgt $M^T \in \mathbb{F}_l^{j \times n}$. Da $Rang(M) = Rang(M^T)$ gilt, bilden die Spalten von M genau dann einen Erzeugendensystem von V, wenn die Zeilen von M ein n-dimensionalen Untervektorraum von $\mathbb{F}_l^j$ aufspannen. Folglich stehen die j-elementigen Erzeugendensysteme aus $\mathbb{F}_l^n$ in Bijektion zu den n-elementigen Teilmengen linear unabhängiger Vektoren aus $\mathbb{F}_l^j$. Nach Lemma 3.7 ist wegen $n \leq j$ die Anzahl dieser Teilmengen gerade $\prod_{i=0}^{n-1}(l^j - l^i)$. □

Korollar und Definition 3.9. *Sei $V := \mathbb{F}_l^n$ mit $n \geq 1$ und $j \in \mathbb{Z}^{>0}$ mit $j \geq n$. Wählt man zufällig $v_1, \ldots, v_j \in V$, dann ist die Wahrscheinlichkeit, dass diese Vektoren den Vektorraum V erzeugen gleich*

$$W_1(n,j,l) := \frac{H_1(n,j,l)}{l^{nj}}.$$

Beweis. Nach dem letzten Satz gibt es $H(n,j,l)$ viele Möglichkeiten, dass $v_1, \ldots, v_j$ den Vektorraum V erzeugen. Da $\#V = l^n$ ist und wir j viele Elemente auswählen, gibt es dafür l^{nj} viele Möglichkeiten. Der Quotient entspricht nun der gewünschten Wahrscheinlichkeit. □

Nun betrachten wir die gleiche Situation mit dem Unterschied, dass wir schon m viele linear unabhängige Vektoren aus V gegeben haben.

Definition und Satz 3.10. *Sei $V := \mathbb{F}_l^n$ mit $n \geq 1$ und seien $v_1, \ldots, v_m \in V$ linear unabhängig mit $n > m \geq 0$. Die Anzahl der Möglichkeiten für $\tilde{v}_1, \ldots, \tilde{v}_j \in V$ mit $j + m \geq n$, dass* $\text{Spann}(\{v_1, \ldots, v_m, \tilde{v}_1, \ldots, \tilde{v}_j\}) = V$ *gilt, ist*

$$H_2(n,j,m,l) := \sum_{i=n-m}^{n} B(n,i,m,l) H_1(i,j,l).$$

Dabei wird $B(n,i,m,l)$ definiert durch

$$B(n,i,m,l) := \frac{\prod_{k=0}^{i-(n-m)-1}(l^m - l^k)}{\prod_{k=0}^{i-(n-m)-1}(l^{i-(n-m)} - l^k)} \cdot \frac{\prod_{k=m}^{n-1}(l^n - l^k)}{\prod_{k=i-(n-m)}^{i-1}(l^i - l^k)}.$$

Beweis. Definiere $L := \text{Spann}\{v_1, \ldots, v_m\}$ und $U := \text{Spann}\{\tilde{v}_1, \ldots, \tilde{v}_j\}$. Wir zählen alle Möglichkeiten, dass die Dimension von U zwischen n und $n-m$ ist und $\text{Spann}\{v_1, \ldots, v_m, \tilde{v}_1, \ldots, \tilde{v}_j\} = V$ gilt. Wir fixieren $i := \dim_{\mathbb{F}_l}(U)$. Nach Satz 3.8 bezeichnet $H_1(i,j,l)$ die Anzahl der Möglichkeiten, dass j viele Vektoren in einem i-dimensionalen $\mathbb{F}_l$-Vektorraum diesen erzeugen. Deshalb muss nur noch gezeigt werden, dass $B(n,i,m,l)$ gleich der Anzahl der i-dimensionalen Untervektorräume U von V ist, die Folgendes erfüllen:

$$(i)\ U \cup L = V \text{ und } (ii)\ \dim_{\mathbb{F}_l}(U \cap L) \text{ ist minimal.}$$

Wegen $U \cap L = U \setminus (V \setminus L)$ ist $\dim_{\mathbb{F}_l}(U \cap L) = i - (n-m)$. Um die Anzahl der paarweise verschiedenen U mit den Eigenschaften (i) und (ii) zu bestimmen, zählen wir zunächst die Anzahl der Basen solcher Untervektorräume. Jede Basis dieser i-dimensionalen Untervektorräume muss wegen $\dim_{\mathbb{F}_l}(U \cap L) = i - (n-m)$ gerade $i-(n-m)$ viele Basiselemente aus dem m-dimensionalen Vektorraum L haben. Wegen (ii) müssen die restlichen $(n-m)$ Basisvektoren aus $V \setminus L$ sein. Dann ist die Anzahl der Basen solcher i-dimensionalen Untervektorräume gleich

$$(l^m - 1) \cdot \ldots \cdot (l^m - l^{i-(n-m)-1}) \cdot (l^n - l^m) \cdot \ldots \cdot (l^n - l^{n-1}).$$

Jetzt muss nur noch durch die Anzahl der Basen dividiert werden, welche die gleichen Untervektorräume U aufspannen. Dazu betrachten wir ein U, das (i) und (ii) erfüllt und Dimension i hat. In $U \cap L$ gibt es $(l^{i-(n-m)} - 1) \cdot \ldots \cdot (l^{i-(n-m)} - l^{i-(n-m)-1})$ viele Basen. Sind die Elemente $b_1, \ldots, b_{i-(n-m)}$ schon gewählt, dann gibt es in dem Untervektorraum U genau $(l^i - l^{i-(n-m)}) \cdot \ldots \cdot (l^i - l^{i-1})$ viele Kombinationen von $(n-m)$ vielen linear unabhängigen Vektoren, so dass sie zusammen mit $b_1, \ldots, b_{i-(n-m)}$ den Untervektorraum U erzeugen. Folglich gibt es zu einem festen U

$$(l^{i-(n-m)} - 1) \cdot \ldots \cdot (l^{i-(n-m)} - l^{i-(n-m)-1}) \cdot (l^i - l^{i-(n-m)}) \cdot \ldots \cdot (l^i - l^{i-1})$$

viele Basen die U erzeugen. Letztendlich ist die Anzahl der verschiedenen Untervektorräume U der Dimension i, die (i) und (ii) erfüllen

$$\frac{(l^m - 1) \cdot \ldots \cdot (l^m - l^{i-(n-m)-1}) \cdot (l^n - l^m) \cdot \ldots \cdot (l^n - l^{n-1})}{(l^{i-(n-m)} - 1) \cdot \ldots \cdot (l^{i-(n-m)} - l^{i-(n-m)-1}) \cdot (l^i - l^{i-(n-m)}) \cdot \ldots \cdot (l^i - l^{i-1})}$$

gleich $B(n,i,m,l)$. Summieren wir über alle i zwischen $n-m$ und n, dann erhalten wir die gewünschte Formel. □

Korollar 3.11. *Seien H_1 und H_2 wie zuvor definiert und $n, j \in \mathbb{Z}^{>0}$ sowie $m \in \mathbb{Z}^{\geq 0}$ mit $j + m \geq n > m$, dann gilt*

(i) $H_2(n,j,0,l) = H_1(n,j,l)$ *und*

(ii) $H_2(n,j,m,l) \geq H_1(n,j,l)$.

Beweis. Da $H_2(n,j,0,l) = B(n,n,0,l) \cdot H_1(n,j,l)$ ist und

$$B(n,n,0,l) = \frac{\prod_{k=0}^{n-n-1}(1-l^k)}{\prod_{k=0}^{n-n-1}(l^{n-n}-l^k)} \cdot \frac{\prod_{k=0}^{n-1}(l^n-l^k)}{\prod_{k=n-n}^{n-1}(l^n-l^k)} = \frac{\prod_{k=0}^{-1}(1-l^k)}{\prod_{k=0}^{-1}(l^{n-n}-l^k)} \cdot \frac{\prod_{k=0}^{n-1}(l^n-l^k)}{\prod_{k=0}^{n-1}(l^n-l^k)} = 1$$

folgt *(i)*. Da der letzte Summand von $H_2(n,j,m,l)$ schon ein positives Vielfaches von $H_1(n,j,l)$ ist, gilt auch die zweite Behauptung. □

Korollar und Definition 3.12. *Seien $v_1,\dots,v_m \in \mathbb{F}_l^n$ linear unabhängig mit $n \geq 1$. Die Wahrscheinlichkeit, dass für zufällig gewählte Vektoren $\tilde{v}_1,\dots,\tilde{v}_j$ aus $\mathbb{F}_l^n$ mit $j+m \geq n > m \geq 0$ die Menge $\{v_1,\dots,v_m,\tilde{v}_1,\dots,\tilde{v}_j\}$ den Vektorraum $\mathbb{F}_l^n$ erzeugt, ist*

$$W_2(n,j,m,l) := \frac{H_2(n,j,m,l)}{l^{nj}}.$$

Beweis. Nach Satz 3.10 ist die Anzahl der Erzeugendensysteme mit der beschriebenen Eigenschaft gleich $H_2(n,j,m,l)$. Weil $\#\mathbb{F}_l^n = l^n$ und wir j Elemente wählen, gibt es dafür l^{nj} viele Möglichkeiten. Also ist der Quotient gleich der gewünschten Wahrscheinlichkeit. □

Lemma 3.13. *Unter den Voraussetzungen des letzten Satzes gilt*

(i) $\lim_{j\to\infty} W_2(n,j,m,l) = 1$ *und*

(ii) $W_2(n,j,m,l)$ *ist bezüglich $n > m$ monoton fallend.*

Beweis. Nach Definition 3.10 ist $W_2(n,j,m,l)$ gleich

$$\begin{aligned}\frac{\sum_{i=n-m}^{n} B(n,i,m,l)H_1(i,j,l)}{l^{nj}} &= \frac{\sum_{i=n-m}^{n} B(n,i,m,l)\prod_{k=0}^{i-1}(l^j-l^k)}{l^{nj}} \\ &= \sum_{i=n-m}^{n} B(n,i,m,l)\prod_{k=0}^{i-1}\left(l^{(j-\frac{nj}{i})}-l^{(k-\frac{nj}{i})}\right)\end{aligned}$$

Für $i < n$ ist $\lim_{j\to\infty}\left(l^{(j-\frac{nj}{i})}-l^{(k-\frac{nj}{i})}\right) = 0$ und für $i = n$ geht der Ausdruck gegen 1. Also ist

$$\lim_{j\to\infty} W_2(n,j,m,l) = B(n,n,m,l) = 1.$$

Die Wahrscheinlichkeit, dass j Vektoren aus $\mathbb{F}_l^n$ einen $(n-m)$-dimensionalen Untervektorraum von $\mathbb{F}_l^n$ aufspannen, ist offensichtlich größer als die Wahrscheinlichkeit, dass gleich viele Vektoren aus $\mathbb{F}_l^{n+1}$ einen $(n-m+1)$-dimensionalen Untervektorraum von $\mathbb{F}_l^{n+1}$ erzeugen. □

Zusammenhang zur Klassengruppe

Sei weiterhin $C_F^0(l)$ isomorph zu $\bigoplus_{i=1}^{z}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$. Wir können die Untervektorräume $U_i = \{l^{(i-1)}[D] \mid [D] \in C_F^0(l) \text{ und } \mathrm{Ord}([D]) = l^i\} \cup \{[0]\}$ für $i \in \{1, \ldots, z\}$ von $C_F^0[l]$ auch als Untergruppe von $C_F^0(l)$ beziehungsweise der Klassengruppe auffassen. Wir definieren die Untergruppe C_i von $C_F^0(l)$ durch

$$C_i := l^{(i-1)} \cdot C_F^0(l) \cong \bigoplus_{j=i}^{z}(\mathbb{Z}/l^{j-(i-1)}\mathbb{Z})^{\delta_j} = \bigoplus_{j=1}^{\tilde{z}}(\mathbb{Z}/l^j\mathbb{Z})^{\beta_j},$$

wobei $\tilde{z} := z - (i-1)$ und $\beta_j := \delta_{j+(i-1)}$ gilt. Offensichtlich ist U_i in C_i enthalten. Die Elemente der Ordnung l in C_i sind die Elemente von $U_i \setminus \{[0]\}$. Wir wollen durch zufälliges Wählen von Elementen aus C_i ein Erzeugendensystem E_i von U_i in Abhängigkeit einer Wahrscheinlichkeit $\varepsilon \in (0,1)$ bestimmen. Im Folgenden beschreiben wir zunächst die Wahrscheinlichkeit, dass in einer Menge aus j vielen zufällig gewählten Elementen aus C_i ein Erzeugendensystem des Untervektorraumes U_i enthalten ist. Anschließend werden wir diese Wahrscheinlichkeit abschätzen.

Satz 3.14. *Sei $G = \bigoplus_{i=1}^{\tilde{z}}(\mathbb{Z}/l^i\mathbb{Z})^{\beta_i}$ eine Gruppe mit $\sum_{i=1}^{\tilde{z}}\beta_i = n$, dann kann man die Untergruppe $G[l] := \bigoplus_{i=1}^{\tilde{z}}(\mathbb{Z}/l\mathbb{Z})^{\beta_i}$ von G als n-dimensionalen $\mathbb{F}_l$-Vektorraum auffassen. Seien weiter $v_1, \ldots, v_m \in G[l]$ linear unabhängig mit $0 \le m < n$, dann gibt es*

$$\sum_{i=n-m}^{j}\binom{j}{i} \cdot H_2(n,i,m,l) \cdot (\#G - l^n)^{j-i}$$

viele Möglichkeiten, dass für $g_1, \ldots, g_j \in G$ mit $j + m \ge n$ unter $v_1, \ldots, v_m, g_1, \ldots, g_j$ ein Erzeugendensystem von $G[l]$ ist.

Beweis. Wir zählen für $n - m \le i \le j$ die Möglichkeiten, dass i viele Elemente der $g_1, \ldots, g_j \in G$ in $G[l]$ liegen und mit $v_1, \ldots, v_m$ ein Erzeugendensystem von $G[l]$ bilden sowie dass $(j-i)$ viele der $g_1, \ldots, g_j \in G$ nicht in $G[l]$ sind. Nach Satz 3.10 bezeichnet $H_2(n,i,m,l)$ die erste Anzahl. Die zweite entspricht $(\#G - \#G[l])^{j-i}$. Des Weiteren können die g_i noch vertauscht werden, aber lediglich so, dass sowohl die Reihenfolge der g_i in $G[l]$ als auch die der g_i in $G \setminus G[l]$ beibehalten wird. Folglich gibt es $\frac{j!}{i!(j-i)!} = \binom{j}{i}$ Vertauschungen. Insgesamt erhalten wir so für festes i wegen $\#G[l] = l^n$ genau

$$\binom{j}{i} \cdot H_2(n,i,m,l) \cdot (\#G - l^n)^{j-i}$$

viele Kombinationen. Da i von $n - m$ bis j läuft, müssen wir für jedes solche i die Kombinationen aufsummieren und erhalten so die gewünschte Formel. □

Korollar und Definition 3.15. *Seien G und $G[l]$ wie in Satz 3.14 definiert sowie $v_1, \ldots, v_m$ linear unabhängige Elemente aus $G[l]$ mit $0 \le m < n$. Weiter seien für*

$j+m \geq n$ die Elemente $g_1, \ldots, g_j$ zufällig aus G gewählt. Die Wahrscheinlichkeit, dass in $\{v_1, \ldots, v_m, g_1, \ldots, g_j\}$ ein Erzeugendensystem von $G[l]$ enthalten ist, entspricht

$$W_3(n, j, m, l, \#G) := \frac{\sum_{r=n-m}^{j} \binom{j}{r} \cdot H_2(n, r, m, l) \cdot (\#G - l^n)^{j-r}}{(\#G)^j}.$$

Beweis. Die Aussage wird analog zu Korollar 3.12 mit Satz 3.14 bewiesen. □

Untere Schranke für die Wahrscheinlichkeit W_3

Sei $C_F^0(l) \cong \bigoplus_{i=1}^{z} (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ die l-Sylow-Gruppe der Klassengruppe eines Funktionenkörpers F vom Geschlecht g über $\mathbb{F}_q$ und $C_i = l^{(i-1)} \cdot C_F^0(l)$. Wie wir zuvor beschrieben haben, ist die Dimension der Untervektorräume U_i von $C_F^0[l]$ durch $n_i = \sum_{j=i}^{z} \delta_j$ für $i \in \{1, \ldots, z\}$ bestimmt und es gilt $U_1 \supseteq U_2 \cdots \supseteq U_z$ sowie $U_{z+1} = \{\}$ mit $n_{i+1} = 0$. Nach Korollar 3.15 bezeichnet $W_3(n_i, j, n_{i+1}, \#C_i)$ die Wahrscheinlichkeit, dass unter j vielen zufällig gewählten Elementen aus $C_i \subseteq C_F^0(l)$ ein Erzeugendensystem von $U_i \backslash U_{i+1}$ ist. Dabei wird für folgende Überlegungen vorausgesetzt, dass $U_i \neq U_{i+1}$ gilt. Sei ε eine reelle Zahl zwischen 0 und 1. Wir fixieren $i \in \{1, \ldots, z\}$ und fragen uns ab welchem $j \in \mathbb{Z}$ mit $j \geq n_i - n_{i+1}$ die Wahrscheinlichkeit

$$W_3(n_i, j, n_{i+1}, \#C_i) \geq \varepsilon$$

ist. Kennen wir diesen Wert, dann wissen wir, wie oft zur Bestimmung eines Erzeugendensystem von $U_i \setminus U_{i+1}$ zufällig Elemente aus der Untergruppe C_i von $C_F^0(l)$ gewählt werden müssen. Zunächst werden wir die Wahrscheinlichkeit $W_3(n_i, j, n_{i+1}, \#C_i)$ für ein festes i nach unten abschätzen, um damit eine obere Schranke für die Iterationsgröße j zu erhalten. Mit Korollar 3.11 gilt

$$\begin{aligned} W_3(n_i, j, n_{i+1}, \#C_i) &= \frac{\sum_{r=n_i-n_{i+1}}^{j} \binom{j}{r} \cdot H_2(n_i, r, n_{i+1}, l) \cdot (\#C_i - l^{n_i})^{j-r}}{(\#C_i)^j} \\ &> \frac{\sum_{r=n_i}^{j} \binom{j}{r} \cdot H_1(n_i, r, l) \cdot (\#C_i - l^{n_i})^{j-r}}{(\#C_i)^j} \\ &= \frac{\sum_{r=n_i}^{j} \binom{j}{r} \cdot \prod_{k=0}^{n_i-1} (l^r - l^k) \cdot (\#C_i - l^{n_i})^{j-r}}{(\#C_i)^j}. \end{aligned}$$

Anstatt $W_3(n_i, j, n_{i+1}, \#C_i)$ betrachten wir die Folge

$$x_j(n) := \frac{\sum_{r=n}^{j} \binom{j}{r} \cdot \prod_{k=0}^{n-1} (l^r - l^k) \cdot (\#C_i - l^n)^{j-r}}{(\#C_i)^j}.$$

Zunächst werden wir $x_j(n)$ durch eine einfachere Folge $y_j(n)$ nach unten abschätzen, um darüber j zu beschränken. Anschließend bestimmen wir für spätere Laufzeitanalysen eine Schranke von j, die ausschließlich von dem Funktionenkörper $F/\mathbb{F}_q$ und der Wahrscheinlichkeit ε abhängt. Wir werden dafür vom folgenden Lemma Gebrauch machen.

Lemma 3.16. *Für die positiven ganzen Zahlen l, n und r mit $r \geq n$ und $l > 1$ gilt*

$$\prod_{k=0}^{n-1}(l^r - l^k) = l^{nr} + \sum_{a=0}^{\zeta} \lambda_a l^{(\beta_a r + \gamma_a)},$$

wobei $\zeta \leq 2^n - 1$, $\beta_a \leq n-1$ und $\gamma_a \leq \frac{n(n-1)}{2}$ sowie $\lambda_a \in \{-1, 1\}$ für alle a ist.

Beweis. Das Produkt $\prod_{k=0}^{n-1}(l^r - l^k)$ hat n Faktoren, also muss ζ kleiner gleich $2^n - 1$ sein. Die Aussagen über β_a und λ_a sowie die Summendarstellung des Produktes sind ebenso offensichtlich. Wegen $\prod_{k=0}^{n-1} l^k = l^{\frac{n(n-1)}{2}}$ gilt $\gamma_a \leq \frac{n(n-1)}{2}$ für alle a.

□

Wir schätzen die Folge $x_j(n)$ mit Hilfe von Lemma 3.16 ab, um eine überschaubarere zu erhalten. Es gilt $\prod_{k=0}^{n-1}(l^r - l^k) = 0$ für $0 \leq r < n$. Damit ist $x_j(n)$ gleich

$$\begin{aligned}
&\left(\frac{\#C_i - l^n}{\#C_i}\right)^j \sum_{r=0}^{j} \binom{j}{r} \cdot \prod_{k=0}^{n-1}(l^r - l^k) \cdot \left(\frac{1}{\#C_i - l^n}\right)^r \\
&= \left(\frac{\#C_i - l^n}{\#C_i}\right)^j \sum_{r=0}^{j} \binom{j}{r} \cdot \left(l^{nr} + \sum_{a=0}^{\zeta} \lambda_a l^{(\beta_a r + \gamma_a)}\right) \cdot \left(\frac{1}{\#C_i - l^n}\right)^r \\
&= \left(\frac{\#C_i - l^n}{\#C_i}\right)^j \left(\sum_{r=0}^{j} \binom{j}{r} \cdot \left(\frac{l^n}{\#C_i - l^n}\right)^r + \sum_{r=0}^{j} \binom{j}{r} \cdot \sum_{a=0}^{\zeta} \lambda_a l^{\gamma_a} \cdot \left(\frac{l^{\beta_a}}{\#C_i - l^n}\right)^r\right) \\
&= \left(\frac{\#C_i - l^n}{\#C_i}\right)^j \left(\left(\frac{l^n}{\#C_i - l^n} + 1\right)^j + \sum_{a=0}^{\zeta} \lambda_a l^{\gamma_a} \cdot \sum_{r=0}^{j} \binom{j}{r} \cdot \left(\frac{l^{\beta_a}}{\#C_i - l^n}\right)^r\right) \\
&= \left(\frac{\#C_i - l^n}{\#C_i}\right)^j \left(\left(\frac{l^n}{\#C_i - l^n} + 1\right)^j + \sum_{a=0}^{\zeta} \lambda_a l^{\gamma_a} \cdot \left(\frac{l^{\beta_a}}{\#C_i - l^n} + 1\right)^j\right) \\
&= \left(\frac{\#C_i - l^n}{\#C_i}\right)^j \left(\left(\frac{\#C_i}{\#C_i - l^n}\right)^j + \sum_{a=0}^{\zeta} \lambda_a l^{\gamma_a} \cdot \left(\frac{\#C_i - l^n + l^{\beta_a}}{\#C_i - l^n}\right)^j\right) \\
&= 1 + \sum_{a=0}^{\zeta} \lambda_a l^{\gamma_a} \cdot \left(\frac{\#C_i - l^n + l^{\beta_a}}{\#C_i}\right)^j \\
&> 1 - \sum_{\widetilde{a}=0}^{\widetilde{\zeta}} l^{\gamma_{\widetilde{a}}} \cdot \left(\frac{\#C_i - l^n + l^{\beta_{\widetilde{a}}}}{\#C_i}\right)^j
\end{aligned}$$

Dabei wird jetzt nur noch über solche $\widetilde{a}$ summiert, für die $\lambda_{\widetilde{a}}$ kleiner Null ist. Nach dem letzten Lemma folgt, dass $\gamma_{\widetilde{a}} \leq \frac{n(n-1)}{2}$ und $\beta_{\widetilde{a}} \leq n-1$ gilt. Da wir nur über negative Summanden summieren und ζ durch $2^n - 1$ nach oben beschränkt ist, erhalten wir $\widetilde{\zeta} \leq 2^{n-1}$. Folglich ist der letzte Ausdruck der Abschätzung größer als

$$1 - \sum_{\widetilde{a}=0}^{\widetilde{\zeta}} l^{\frac{n(n-1)}{2}} \left(\frac{\#C_i - l^n + l^{n-1}}{\#C_i}\right)^j \geq 1 - 2^{n-1} l^{\frac{n(n-1)}{2}} \left(\frac{\#C_i - l^n + l^{n-1}}{\#C_i}\right)^j$$

Damit haben wir eine einfachere Folge gefunden, die von $x_j(n)$ dominiert wird. Wir definieren

$$y_j(n) := 1 - 2^{n-1} l^{\frac{n(n-1)}{2}} \left(\frac{\#C_i - l^n + l^{n-1}}{\#C_i}\right)^j.$$

Ist $y_j(n_i)$ größer gleich $\varepsilon \in (0,1)$, dann ist es auch $x_j(n_i)$. Insbesondere folgt dann $W_3(n_i, j, n_{i+1}, \#C_i) \geq \varepsilon$.

Aus $y_j(n_i) \geq \varepsilon$ erhalten wir

$$\begin{aligned} & 1 - 2^{n_i-1} l^{\frac{n_i(n_i-1)}{2}} \left(\frac{\#C_i - l^{n_i} + l^{n_i-1}}{\#C_i}\right)^j \geq \varepsilon \\ \iff \qquad & \frac{\log(1-\varepsilon) - \log(2^{n_i-1} l^{\frac{n_i(n_i-1)}{2}})}{\log(\#C_i - l^{n_i} + l^{n_i-1}) - \log(\#C_i)} \geq j \end{aligned}$$

Wir definieren

$$A(\#C_i, n_i, \varepsilon) := \left\lceil \frac{\log(1-\varepsilon) - \log(2^{n_i-1} l^{\frac{n_i(n_i-1)}{2}})}{\log(\#C_i - l^{n_i} + l^{n_i-1}) - \log(\#C_i)} \right\rceil. \tag{3.11}$$

Für $j := A(\#C_i, n_i, \varepsilon)$ und $i \in \{1, \ldots, z\}$ haben wir

$$W_3(n_i, j, n_{i+1}, \#C_i) > x_j(n_i) > y_j(n_i) \geq \varepsilon.$$

Das heißt, wählen wir zufällig $A(\#C_i, n_i, \varepsilon)$ viele Elemente aus C_i, so ist die Wahrscheinlichkeit größer als ε, dass darunter ein Erzeugendensystem des $\mathbb{F}_l$-Vektorraumes $U_i \setminus U_{i+1}$ ist.

Unabhängige obere Schranke von j

Wir suchen eine Schranke für j, welche ausschließlich von den charakteristischen Größen des Funktionenkörpers $F/\mathbb{F}_q$ und der Wahrscheinlichkeit ε abhängt. Dazu wählen wir die folgende Vorgehensweise. Wir zeigen:

1. Es existiert genau ein $j_n \in \mathbb{Z}^{>0}$, so dass $y_j(n) \geq y_j(1)$ gilt für alle $j \geq j_n$ und $n \in \mathbb{Z}^{>1}$.

2. Die Folge j_n ist für $n \in \mathbb{Z}^{>1}$ monoton fallend.

Dann ist entweder $j \leq j_2$ oder es gilt $y_j(n_i) \geq y_j(1)$. Mit $y_j(1) \geq \varepsilon$ beziehungsweise

$$j \leq \frac{\log(1-\varepsilon)}{\log(\#C_i - l + 1) - \log(\#C_i)}$$

erhalten wir ein weitere Schranke für j. Für das Maximum j' der beiden Schranken haben wir $W_3(n_i, j', n_{i+1}, \#C_i) \geq \varepsilon$.

Zu 1.: Bestimmung von j_n

Wir berechnen den Index j_n ab dem $y_j(n) \geq y_j(1)$ ist. Aus $y_j(n) \geq y_j(1)$ folgt

$$\begin{aligned} & \left(\frac{\#C_i - l + 1}{\#C_i}\right)^j \geq 2^{n-1} l^{\frac{n(n-1)}{2}} \left(\frac{\#C_i - l^n + l^{n-1}}{\#C_i}\right)^j \\ \iff & \qquad j \geq \frac{\log(2^{n-1} l^{\frac{n(n-1)}{2}})}{\log\left(\frac{\#C_i - l + 1}{\#C_i - l^n + l^{n-1}}\right)}. \end{aligned}$$

Wir definieren den gesuchten Index durch $j_n := \left\lceil \frac{\log(2^{n-1} l^{\frac{n(n-1)}{2}})}{\log\left(\frac{\#C_i - l + 1}{\#C_i - l^n + l^{n-1}}\right)} \right\rceil$.

Zu 2.: Monotonie von j_n

Wir wollen zeigen, dass $j_n \geq j_{n+1}$ gilt für $n \in \mathbb{Z}^{>1}$. Sei dazu $B_n := \frac{\#C_i - l + 1}{\#C_i - l^n + l^{n-1}}$. Wir betrachten

$$\begin{aligned} & \frac{\log(2^{n-1} l^{\frac{n(n-1)}{2}})}{\log(B_n)} \geq \frac{\log(2^n l^{\frac{(n+1)n}{2}})}{\log(B_{n+1})} \\ \iff & \log(2^{n-1} l^{\frac{n(n-1)}{2}}) \log(B_{n+1}) \geq \log(2^n l^{\frac{(n+1)n}{2}}) \log(B_n) \\ \iff & \log(2^{n-1} l^{\frac{n(n-1)}{2}}) \log(B_{n+1}) \geq \log(2^{n-1} l^{\frac{n(n-1)}{2}} 2l^n) \log(B_n) \\ \iff & \log(2^{n-1} l^{\frac{n(n-1)}{2}}) (\log(B_{n+1}) - \log(B_n)) \geq \log(2l^n) \log(B_n) \\ \iff & \log(2^{n-1} l^{\frac{n(n-1)}{2}}) \log\left(\frac{\#C_i - l^n + l^{n-1}}{\#C_i - l^{n+1} + l^n}\right) \geq \log(2l^n) \log(B_n). \end{aligned} \tag{3.12}$$

Wir betrachten eine Fallunterscheidung für (3.12). Für den Fall $n \geq 3$ ist $2^{n-1} l^{\frac{n(n-1)}{2}}$ größer als $2l^n$, folglich müssen wir nur zeigen , dass

$$\frac{\#C_i - l^n + l^{n-1}}{\#C_i - l^{n+1} + l^n} \geq B_n = \frac{\#C_i - l + 1}{\#C_i - l^n + l^{n-1}}$$

gilt. Der letzte Ausdruck ist aber gerade äquivalent zu

$$(\#C_i - l^n + l^{n-1})^2 \geq (\#C_i - l^{n+1} + l^n)(\#C_i - l + 1) \tag{3.13}$$
$$\iff \#C_i 2(l^{n-1} - l^n) + (l^{n-1} - l^n)^2 \geq \#C_i((1-l) + (l^n - l^{n+1})) + (1-l)(l^n - l^{n+1}).$$

Nun gilt

(i) wegen $l \geq 2$, dass $2(l^{n-1} - l^n)$ größer gleich $l(l^{n-1} - l^n)$ ist, also insbesondere $\#C_i 2(l^{n-1} - l^n) \geq \#C_i((1-l) + l(l^{n-1} - l^n))$ und

(ii) für $n \geq 3$, dass $(1-l)(l^n - l^{n+1}) = (l^n - 2l^{n+1} + l^{n+2}) < l^{(n-2)}(l^n - 2l^{n+1} + l^{n+2}) = l^{2(n-1)} - 2l^{2n-1} + l^{2n} = (l^{n-1} - l^n)^2$ ist.

Also gilt (3.13) und damit ist $j_n \geq j_{n+1}$ für $n \geq 3$.

Sei nun $n = 2$. Wir wollen zeigen, dass $j_2 \geq j_3$ gilt. Mit $n = 2$ in (3.12) eingesetzt, erhalten wir

$$\begin{aligned} & \log(2l) \log\left(\frac{\#C_i - l^2 + l}{\#C_i - l^3 + l^2}\right) \geq \log(2l^2) \log\left(\frac{\#C_i - l + 1}{\#C_i - l^2 + 1}\right) \\ \iff & \frac{\#C_i - l^2 + l}{\#C_i - l^3 + l^2} \geq \left(\frac{\#C_i - l + 1}{\#C_i - l^2 + 1}\right)^2 \\ \iff & (\#C_i - l^2 + 1)^3 \geq (\#C_i - l^3 + l^2)(\#C_i - l + 1)^2 \end{aligned}$$

gilt. Wir multiplizieren diese Ungleichung aus und kommen auf

$$(\#C_i)^2 \underbrace{(l^2 - 3l - 1)}_{=:p_1(l)} + \#C_i \underbrace{(l^3 + 5l^2 - 4l - 2)}_{=:p_2(l)} - (l^5 - 3l^2 + l + 1) \geq 0.$$

Weil die Gruppe $C_i := l^{i-1} \cdot C_F^0(l)$ den $\mathbb{F}_l$-Vektorraum U_i der Dimension $n = n_i$ enthält, muss gelten, dass $\#C_i \geq l^n$ ist. Also gilt in diesem Fall $\#C_i \geq l^2$. Da $p_1(l), p_2(l) > 0$ für $l \geq 4$ sind, betrachten wir

$$\begin{aligned} & (\#C_i)^2(l^2 - 3l - 1) + \#C_i(l^3 + 5l^2 - 4l - 2) - (l^5 - 3l^2 + l + 1) \\ & \quad \geq l^4(l^2 - 3l - 1) + l^2(l^3 + 5l^2 - 4l - 2) - (l^5 - 3l^2 + l + 1) \\ & \quad = \underbrace{l^6 - 3l^5 + 4l^4 - 4l^3 + l^2 - l - 1}_{=:p_3(l)} \geq 0. \end{aligned}$$

Wegen $p_3(l) > 0$ für $l \geq 1$ gilt die letzte Behauptung und damit die Monotonie von j_n, bis auf den Fall $n = 2$ und $l = 2, 3$. Mit $n = 2$ folgt aus $y_j(2) \geq \varepsilon$ für $l = 2$ beziehungsweise $l = 3$, dass

$$j \leq \frac{\log\left(\frac{1-\varepsilon}{4}\right)}{\log\left(\frac{\#C_i - 2}{\#C_i}\right)} \text{ und } j \leq \frac{\log\left(\frac{1-\varepsilon}{6}\right)}{\log\left(\frac{\#C_i - 6}{\#C_i}\right)}$$

gilt. Für alle anderen Fälle können wir mittels 1. und 2., wie schon beschrieben, die Größe j wie folgt nach oben abschätzen:

$$j \leq \max \left\{ \overbrace{\left\lceil \frac{\log(2l)}{\log\left(\frac{\#C_i-l+1}{\#C_i-l^2+l}\right)} \right\rceil}^{=j_2}, \frac{\log(1-\varepsilon)}{\log\left(\frac{\#C_i-l+1}{\#C_i}\right)} \right\}$$

Wir sehen für $n = 2$ und $l = 2, 3$ entsprechen die Schranken für j bis auf Konstanten gleich der letzten. Deshalb müssen wir diese Fälle nicht mehr gesondert betrachten. Im Folgenden werden wir die letzte Schranke so verändern, dass sie ausschließlich von $F/\mathbb{F}_q$ und ε abhängt.
Wir betrachten die Untergruppe $C_i := l^{i-1} \cdot C_F^0(l)$ wie zu Beginn dieses Teilabschnittes. Wir wollen die Ausdrücke

$$(i) \left\lceil \frac{\log(2l)}{\log\left(\frac{\#C_i-l+1}{\#C_i-l^2+l}\right)} \right\rceil \text{ und } (ii) \frac{\log(1-\varepsilon)}{\log\left(\frac{\#C_i-l+1}{\#C_i}\right)}$$

nach oben abschätzen. Der erste Ausdruck ist maximal, wenn der Nenner minimal wird. Da wir $\frac{\#C_i-l+1}{\#C_i-l^2+l} > 1$ haben, ist der Nenner genau dann minimal, wenn $l = 2$ gilt und $\#C_i$ maximal ist. Wir betrachten $\log\left(\frac{\#C_i-1}{\#C_i-2}\right)$:
Für $x > 0$ ist $\ln(x) \geq \frac{x-1}{x}$. Zusätzlich nutzen wir aus, dass $\#C_i \leq h_F$ und nach Satz 1.18 die Klassenzahl durch $(\sqrt{q}+1)^{2g}$ beschränkt wird. Wir erhalten

$$\ln\left(\frac{\#C_i-1}{\#C_i-2}\right) = \ln\left(1+\frac{1}{\#C_i-2}\right) \geq \frac{1}{\#C_i-1} \geq \frac{1}{(\sqrt{q}+1)^{2g}-1}.$$

Wegen $l \leq h_F = \mathcal{O}(q^g)$ gilt $\log(2l) = \mathcal{O}(g\log(q))$ und damit folgt

$$\left\lceil \frac{\log(2l)}{\log\left(\frac{\#C_i-l+1}{\#C_i-l^2+l}\right)} \right\rceil = \mathcal{O}(g\log(q)q^g) = \mathcal{O}\left(gq^{g+\mathcal{O}(1)}\right).$$

Der Ausdruck (ii) ist genau dann maximal, wenn der Quotient $\frac{\#C_i-l+1}{\#C_i} < 1$ am Größten ist, also wenn $l = 2$ gilt und $\#C_i$ maximal wird. Wir benutzen wieder die Ungleichung $\#C_i \leq (\sqrt{q}+1)^{2g}$. Wegen $\ln(x) \leq x-1$ für $x > 0$ und $\log(1-\varepsilon) < 0$ folgt

$$\frac{\log(1-\varepsilon)}{\ln\left(\frac{\#C_i-1}{\#C_i}\right)} = \frac{\log(1-\varepsilon)}{\ln\left(1-\frac{1}{\#C_i}\right)} \leq \#C_i \log(1/(1-\varepsilon)) \leq (\sqrt{q}+1)^{2g}\log(1/(1-\varepsilon)).$$

Damit erhalten wir

$$\frac{\log(1-\varepsilon)}{\log\left(\frac{\#C_i-l+1}{\#C_i}\right)} = \mathcal{O}\left(\log(1/(1-\varepsilon))q^g\right).$$

In beiden Fällen ist $\mathcal{O}(\log(1/(1-\varepsilon))gq^{g+\mathcal{O}(1)})$ eine obere Schranke. Zusammenfassend gilt also

$$j = \mathcal{O}\left(\log(1/(1-\varepsilon))gq^{g+\mathcal{O}(1)}\right). \tag{3.14}$$

Bemerkung 3.17. *Die soweit angegebenen Schranken für j sind leider äußerst ungenau. Wie sich später herausstellen wird, ist gerade die letzte Schranke signifikant für die Komplexität des probabilistischen Algorithmus zur Berechnung von Klassengruppen von globalen Funktionenkörpern.*

Berechnung der Iterationsgrößen

Sei immer noch $C_F^0(l) \cong \bigoplus_{i=1}^{z}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ die l-Sylow-Gruppe der Klassengruppe eines Funktionenkörpers F vom Geschlecht g über $\mathbb{F}_q$ für den Primteiler l der Klassenzahl von F.
Wir betrachten $U_i = \{l^{(i-1)}[D] \mid [D] \in C_F^0(l) \text{ und } \mathrm{Ord}([D]) = l^i\} \cup \{[0]\}$, die Untervektorräume von $C_F^0[l]$ der Dimension n_i, für $i \in \{1,\ldots,z\}$. Wegen $U_{z+1} = \{\}$ ist $U_{i+1} \subseteq U_i$ für alle i. Da $n_{z+1} = 0$ ist, gilt nach (3.9)

$$\dim_{\mathbb{F}_l}(U_{i+1}) = n_{i+1} \le n_i \le n_{i+1} + \lfloor \frac{s_i}{i} \rfloor \text{ mit } s_i = s - \sum_{j=i+1}^{z} j\delta_j.$$

Sei $\widetilde{U}_i$ ein Untervektorraum des $\mathbb{F}_l$-Vektorraumes C_F^0/lC_F^0 der Dimension n_i sowie $\widetilde{U}_{z+1} := \{\}$. Wegen der Gleichheit von $\dim_{\mathbb{F}_l}(C_F^0/lC_F^0)$ und $\dim_{\mathbb{F}_l}(C_F^0[l])$ gilt nach Korollar 3.4, dass $\dim_{\mathbb{F}_l}(C_F^0/lC_F^0) \le \min\{2g, s-z+1\}$.

Als Zusammenfassung der vorherigen Ergebnisse beschreiben wir nun einen Algorithmus der für $\varepsilon \in (0,1)$ die Iterationsgrößen j_1, j_2 und j_3 berechnet, so dass

1. $W_2(n_{i+1} + \lfloor \frac{s_i}{i} \rfloor, j_1, n_{i+1}, l) \ge \sqrt{\varepsilon}$,
2. $W_2(\min\{2g, s-z+1\}, j_2, n_{i+1}, l) \ge \sqrt{\varepsilon}$ und
3. $W_3(n_{i+1} + \lfloor \frac{s_i}{i} \rfloor, j_3, n_{i+1}, \#C_i) \ge \sqrt{\varepsilon}$.

Diese Ausdrücke beschreiben mit $C_i = l^{(i-1)} \cdot C_F^0(l)$ und $i \in \{1,\ldots,z\}$, dass

1. unter j_1 vielen zufällig gewählten Elementen aus U_i mit Wahrscheinlichkeit größer gleich $\sqrt{\varepsilon}$ ein Erzeugendensystem von $U_i \setminus U_{i+1}$ ist,

2. unter j_2 vielen zufällig gewählten Elementen aus C_F^0/lC_F^0 mit Wahrscheinlichkeit größer gleich $\sqrt{\varepsilon}$ ein Erzeugendensystem von $(C_F^0/lC_F^0) \setminus \widetilde{U}_{i+1}$ ist und

3. unter j_3 vielen zufällig gewählten Elementen aus C_i mit Wahrscheinlichkeit größer gleich $\sqrt{\varepsilon}$ ein Erzeugendensystem von $U_i \setminus U_{i+1}$ ist.

Später werden wir für den Fall $l \neq p$ aus einem Erzeugendensystem E_i von $U_i \setminus U_{i+1}$ und einem von C_F^0/lC_F^0 mittels Algorithmus 1 aus Kapitel 2 beziehungsweise für $l = p$ mit Algorithmus 2 direkt aus E_i, eine Basis von $U_i \setminus U_{i+1}$ bestimmen. Ist eine Wahrscheinlichkeit $\varepsilon \in (0,1)$ vorgegeben sowie l teilerfremd zu p, dann bestimmen die Größen j_2 und j_3 gerade wie oft wir maximal Elemente zufällig aus C_F^0/lC_F^0 beziehungsweise C_i wählen müssen, damit Algorithmus 1 daraus mit Wahrscheinlichkeit größer gleich $\sqrt{\varepsilon} \cdot \sqrt{\varepsilon} = \varepsilon$ eine Basis von $U_i \setminus U_{i+1}$ berechnet. Haben wir schon j_1 viele Elemente in U_i gefunden, dann berechnet Algorithmus 1 ebenfalls mit Wahrscheinlichkeit größer gleich ε daraus eine Basis von $U_i \setminus U_{i+1}$. Das heißt, wir wählen höchstens j_3 viele Elemente aus C_i. Betrachten wir $l = p$ und statt ε das Quadrat davon, dann beschreiben die Größen j_1 und j_3 wie viele Elemente wir aus U_i beziehungsweise aus C_i zufällig wählen müssen, damit Algorithmus 2 mit Wahrscheinlichkeit $\geq \varepsilon$ daraus eine Basis von $U_i \setminus U_{i+1}$ berechnet. In diesem Fall spielt j_2 keine Rolle.
Wir gehen davon aus, dass die Werte δ_i für $j = i+1, \ldots, z$ schon berechnet wurden. Für $i < z$ sei $s_i := s - \sum_{i+1}^{z} j\delta_j$ und $m_i := \sum_{j=i+1}^{z} (j-(i-1))\delta_j = \log_l(\#C_{i+1})$. Im Fall $i = z$ setzen wir $s_z := s$ und $m_z := 0$.

Algorithmus 4 : Iterationsgrößen

Eingabe: $\varepsilon \in (0,1)$, Primzahl l mit $l^s \mid h_F$ und $l^{s+1} \nmid h_F$, n_{i+1}, m_i, s_i mit $s_i \geq i$ und g Geschlecht von F sowie $d(q,l)$ der Einbettungsgrad von F bezüglich l.
Ausgabe: j_1, j_2 und j_3 wie zuvor beschrieben.

1: $j_1 \leftarrow \lfloor \frac{s_i}{i} \rfloor$
2: **while** $W_2(n_{i+1} + \lfloor \frac{s_i}{i} \rfloor, j_1, n_{i+1}, l) < \sqrt{\varepsilon}$ **do**
3: $\quad j_1 \leftarrow j_1 + 1$
4: **end while**
5: **if** $d(q,l) > 1$ **then**
6: $\quad s \leftarrow 2g-1$, $z \leftarrow 0$, $j_2 \leftarrow 2g - n_{i+1}$
7: **else**
8: $\quad j_2 \leftarrow \min\{2g, s-z+1\} - n_{i+1}$
9: **end if**
10: **while** $W_2(\min\{2g, s-z+1\}, j_2, n_{i+1}, l) < \sqrt{\varepsilon}$ **do**
11: $\quad j_2 \leftarrow j_2 + 1$
12: **end while**
13: $j_3 \leftarrow \max\{A(l^{\lfloor \frac{s_i}{i} \rfloor + m_i}, n_i, \sqrt{\varepsilon}) \mid n_{i+1} < n_i \leq n_{i+1} + \lfloor \frac{s_i}{i} \rfloor \}$ {A aus in (3.11)}
14: **return** j_1, j_2, j_3

Korrektheit:
Für $s_i \geq i$, folgt aus Korollar 3.4, dass $n_{i+1} < n_{i+1} + \lfloor \frac{s_i}{i} \rfloor \leq \dim_{\mathbb{F}_l}(C_F^0[l]) \leq \min\{2g, s - z + 1\}$ gilt. Somit sind $W_2(n_{i+1} + \lfloor \frac{s_i}{i} \rfloor, j_1, n_{i+1}, l)$ und $W_2(\min\{2g, s-z+1\}, j_2, n_{i+1}, l)$ wohldefiniert. Nach Lemma 3.13 konvergieren für $j_1, j_2 \to \infty$ die Folgen $W_2(n_{i+1} + \lfloor \frac{s_i}{i} \rfloor, j_1, n_{i+1}, l)$ und $W_2(\min\{2g, s-z+1\}, j_2, n_{i+1}, l)$ gegen 1. Mit $\varepsilon < 1$, ist es auch $\sqrt{\varepsilon}$ und damit terminieren die beiden While-Schleifen nach endlichen vielen Schritten, also auch der Algorithmus. Ebenfalls nach Lemma 3.13 ist $W_2(n_i, j_1, n_{i+1}, l)$ in n_i monoton fallend. Wegen $n_i \leq n_{i+1} + \lfloor \frac{s_i}{i} \rfloor$ gilt

$$W_2(n_i, j_1, n_{i+1}, l) \geq W_2(n_{i+1} + \lfloor \frac{s_i}{i} \rfloor, j_1, n_{i+1}, l).$$

Also ist für j_1 mit $W_2(n_{i+1} + \lfloor \frac{s_i}{i} \rfloor, j_1, n_{i+1}, l) \geq \sqrt{\varepsilon}$ auch $W_2(n_i, j_1, n_{i+1}, l) \geq \sqrt{\varepsilon}$ und damit j_1 korrekt.

Wegen $\dim_{\mathbb{F}_l}(C_F^0/lC_F^0) \leq \min\{2g, s - z + 1\}$ folgt analog

$$W_2(\dim_{\mathbb{F}_l}(C_F^0/lC_F^0), j_2, n_{i+1}, l) \geq W_2(\min\{2g, s - z + 1\}, j_2, n_{i+1}, l).$$

Für j_2 mit $W_2(\min\{2g, s - z + 1\}, j_2, n_{i+1}, l) \geq \sqrt{\varepsilon}$ ist ebenfalls die Wahrscheinlichkeit $W_2(\dim_{\mathbb{F}_l}(C_F^0/lC_F^0), j_2, n_{i+1}, l)$ größer gleich $\sqrt{\varepsilon}$. Also wird j_2 im Fall $d(q, l) = 1$ korrekt berechnet. Im anschließenden Abschnitt werden wir einen Algorithmus beschreiben, der mittels der Iterationsgrößen j_1, j_2 und j_3 entsprechende Erzeugendensysteme von Untervektorräumen von $C_F^0[l]$ bestimmt und im Fall $l \neq p$ mit Algorithmus 1 aus Kapitel 2 die Anzahl der linear unabhängigen Elemente davon berechnet. Ist der Einbettungsgrad $d(q, l)$ größer 1, müssen wir, wie in Teilabschnitt 2.1.3 beschrieben, ein Erzeugendensystem von $C_{F'}^0/lC_{F'}^0$ bestimmen. Dabei bezeichnet F' eine Konstantenkörpererweiterung des Funktionenkörpers $F/\mathbb{F}_q$ vom Grad $d(q, l)$. Die Dimension von $C_{F'}^0/lC_{F'}^0$ ist größer gleich der von C_F^0/lC_F^0. Es gilt aber immer $\dim_{\mathbb{F}_l}(C_{F'}^0/lC_{F'}^0) \leq 2g$. Belegen wir s mit $2g - 1$ und z mit 0, dann ist $s - z + 1 = 2g$ und damit

$$\begin{aligned} W_2(\dim_{\mathbb{F}_l}(C_{F'}^0/lC_{F'}^0), j_2, n_{i+1}, l) & \\ \geq W_2(\min\{2g, s - z + 1\}, j_2, n_{i+1}, l) & \\ = W_2(2g, j_2, n_{i+1}, l). & \end{aligned}$$

Demnach wird auch für den Fall $d(q, l) > 1$ die Iterationsgröße j_2 richtig berechnet. Nun zur Korrektheit von j_3. Wählen wir $j_3 := A(\#C_i, n_i, \sqrt{\varepsilon})$, wie in (3.11) definiert wurde, dann ist $W_3(n_{i+1} + \lfloor \frac{s_i}{i} \rfloor, j_3, n_{i+1}, \#C_i) \geq \sqrt{\varepsilon}$. Fassen wir $A(\#C_i, n_i, \sqrt{\varepsilon})$ als Folge in $\#C_i$ auf, dann ist diese monoton wachsend. Es gilt mit $\#C_F^0(l) = l^s$ und $s = \sum_{i=1}^{z} i\delta_i$, dass

$$C_i = l^{(i-1)} \cdot C_F^0(l) \cong l^{(i-1)} \cdot \bigoplus_{j=1}^{z} (\mathbb{Z}/l^j\mathbb{Z})^{\delta_j} \cong \bigoplus_{j=i}^{z} (\mathbb{Z}/l^{(j-(i-1))}\mathbb{Z})^{\delta_j},$$

wobei dann $\log_l(\#C_i) = \sum_{j=i}^{z} (j - (i-1))\delta_j$ ist. Nach Satz 3.2 gilt $\delta_i \leq \lfloor \frac{s_i}{i} \rfloor$ und damit

$$\#C_i = l^{\sum_{j=i}^{z} (j-(i-1))\delta_j} = l^{\delta_i} \cdot l^{\sum_{j=i+1}^{z} (j-(i-1))\delta_j} = l^{\delta_i + m_i} \leq l^{\lfloor \frac{s_i}{i} \rfloor + m_i}.$$

Folglich gilt $A(\#C_i, n_i, \sqrt{\varepsilon}) \leq A(l^{\lfloor \frac{s_i}{i} \rfloor + m}, n_i, \sqrt{\varepsilon})$. Wegen $n_{i+1} < n_i \leq n_{i+1} + \lfloor \frac{s_i}{i} \rfloor$ erhalten wir

$$A(\#C_i, n_i, \sqrt{\varepsilon}) \leq \max\{A(l^{\lfloor \frac{s_i}{i} \rfloor + m_i}, n_i, \sqrt{\varepsilon}) \mid n_{i+1} < n_i \leq n_{i+1} + \lfloor \frac{s_i}{i} \rfloor \} =: j_3.$$

Damit ergibt sich die Korrektheit von Algorithmus 3.

Schranken:
Wir wollen Schranken für die Iterationsgrößen j_1, j_2 und j_3 bestimmen, mit denen wir die Komplexität späterer Algorithmen beschreiben können.

Lemma 3.18. *Für die nicht-negativen ganzen Zahlen n, i und l mit $i \geq n$ und $l > 1$ gilt*

$$\prod_{k=0}^{n-1} (l^i - l^k) \geq l^{ni} - l^{n+(n-1)i}.$$

Beweis. Wir beweisen die Behauptung durch Induktion über n. Sei $n = 1$, dann ist $\prod_{k=0}^{n-1}(l^i - l^k) = l^i - l$. Gelte die Ungleichung bis zu einem festen n (IV). Dann erhalten wir

$$\begin{aligned}
\prod_{k=0}^{n} (l^i - l^k) = \prod_{k=0}^{n-1} (l^i - l^k)(l^i - l^n) &\overset{\text{(IV)}}{\geq} (l^{ni} - l^{n+(n-1)i})(l^i - l^n) \\
&= l^{(n+1)i} - 2l^{ni+n} + l^{(n-1)i+2n} \\
&\geq l^{(n+1)i} - l^{ni+n+1} + l^{(n-1)i+2n} \\
&> l^{(n+1)i} - l^{ni+n+1}.
\end{aligned}$$

□

Damit können wir $W_2(n_i, j_1, n_{i+1}, l)$ wie folgt abschätzen. Wir wenden dazu in gleicher Reihenfolge Definition 3.12, Korollar 3.11, Definition 3.8 und Lemma 3.18 an:

$$\begin{aligned}
W_2(n_i, j_1, n_{i+1}, l) &= \frac{H_2(n_i, j_1, n_{i+1}, l)}{l^{n_i j_1}} \\
&\geq \frac{H_1(n_i, j_1, l)}{l^{n_i j_1}} \\
&= \frac{\prod_{k=0}^{n_i - 1} (l^{j_1} - l^k)}{l^{n_i j_1}} \\
&\geq \frac{(l^{n_i j_1} - l^{n_i + (n_i - 1) j_1})}{l^{n_i j_1}} \\
&= 1 - l^{n_i - j_1}.
\end{aligned}$$

Aus $W_2(n_{i+1} + \lfloor \frac{s_i}{i} \rfloor, j_1, n_{i+1}, l) \geq 1 - l^{(n_{i+1}+\lfloor \frac{s_i}{i} \rfloor - j_1)} \geq \sqrt{\varepsilon}$ folgt unmittelbar

$$j_1 \leq n_{i+1} + \lfloor \frac{s_i}{i} \rfloor + \log_{\frac{1}{l}}(1 - \sqrt{\varepsilon}).$$

Mit $W_2(\min\{2g, s-z+1\}, j_2, n_{i+1}, l) \geq 1 - l^{(\min\{2g,s-z+1\}-j_2)} \geq \sqrt{\varepsilon}$ ergibt sich analog

$$j_2 \leq \min\{2g, s-z+1\} + \log_{\frac{1}{l}}(1 - \sqrt{\varepsilon}).$$

Da $l \geq 2$ ist und $2g$ das Minimum von $2g$ und $s-z+1$ sowie den Wert $n_{i+1} + \lfloor \frac{s_i}{i} \rfloor$ nach oben beschränkt, gilt

$$j_1, j_2 \leq 2g + \log_{\frac{1}{2}}(1 - \sqrt{\varepsilon}) = 2g + \log(1/(1 - \sqrt{\varepsilon})). \tag{3.15}$$

Der folgende Satz beschreibt abschließend die Resultate dieses Teilabschnittes.

Satz 3.19. *Sei $F/\mathbb{F}_q$ einen Funktionenkörper vom Geschlecht g und $\varepsilon \in (0,1)$. Mit der Notation dieses Abschnittes erhalten wir*

$$j_1, j_2 = \mathcal{O}((g + \log(1/(1 - \sqrt{\varepsilon}))) \text{ und } j_3 = \mathcal{O}(\log(1/(1 - \sqrt{\varepsilon}))gq^{g+\mathcal{O}(1)}).$$

Beweis. Wegen (3.15) folgt die erste Aussagen. Mit (3.14) und $\sqrt{\varepsilon}$ statt ε gilt die zweite. □

Bemerkung 3.20. *Für den Fall $l = p$ können wir die obigen Schranken noch verbessern. Wie Eingangs beschrieben wurde, benötigen wir zur Berechnung einer Basis des Untervektorraumes $U_i \setminus U_{i+1}$ für $i \in \{1, \ldots, z\}$ mittels Algorithmus 2 lediglich ein Erzeugendensystem davon und nicht wie zuvor noch eines von C_F^0/lC_F^0. Haben wir ein solches Erzeugendensystem von $U_i \setminus U_{i+1}$ mit Wahrscheinlichkeit größer gleich ε, dann terminiert Algorithmus 2 mit gleicher Wahrscheinlichkeit korrekt. Folglich können wir in der obigen Betrachtung ε durch ε^2 ersetzen. Damit ergeben sich die folgenden neuen Schranken für j_1 und j_3, wenn $l = p$ ist:*

$$j_1 = \mathcal{O}(g + (\log(1/(1 - \varepsilon))) \text{ und } j_3 = \mathcal{O}(\log((1/(1 - \varepsilon))gq^{g+\mathcal{O}(1)}).$$

Der Aufwand zur Berechnung der Iterationsgrößen j_1, j_2 und j_3 durch Algorithmus 3 kann vernachlässigt werden. Wieder sei hier angemerkt, dass die Schranke von j_3 ungenau ist.
Betrachten wir die Wahrscheinlichkeiten W_2 und W_3 für den Fall, dass wir zufällig gewählte Elemente nicht zurücklegen, dann lassen sich die Schranken von j_1, j_2 und j_3 verbessern. Das würde auch eine Verbesserung der Komplexität der folgenden Algorithmen nach sich ziehen.

3.4.3 Berechnung von Basen der Untervektorräume U_i

Mit den Resultaten des vergangenen Teilabschnittes können wir nun beschreiben, wie wir probabilistisch sowohl Erzeugendensysteme der Untervektorräume von $C_F^0[l]$ als auch von C_F^0/lC_F^0 berechnen und anschließend mit den aus Kapitel 2 zur Verfügung stehenden Mitteln daraus die linear unabhängigen Elemente bestimmen.
Sei $\varepsilon \in (0,1)$. Wir betrachten $C_F^0(l) \cong \bigoplus_{i=1}^{z}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ mit $\#C_F^0(l) = l^s$ und $s = \sum_{i=1}^{z} i\delta_i$ sowie die Untervektorräume U_i von $C_F^0[l]$, wie in (3.5) definiert. Zunächst sei $l \neq p$. Anschließend betrachten wir den Fall, dass l gleich p ist.

Der Fall l ist ungleich p

Sei E_i ein Erzeugendensystem von $U_i \backslash U_{i+1}$ und erzeuge $\widetilde{E}_i$ den $\mathbb{F}_l$-Vektorraum C_F^0/lC_F^0. Mittels Algorithmus 1 aus Kapitel 2 können wir aus E_i eine Basis B_i von $U_i \setminus U_{i+1}$ bestimmen. Sind E_i und $\widetilde{E}_i$ Erzeugendensysteme jeweils mit einer Wahrscheinlichkeit größer gleich $\sqrt{\varepsilon}$, dann berechnet Algorithmus 1 mit einer Wahrscheinlichkeit größer gleich $(\sqrt{\varepsilon})^2 = \varepsilon$ eine Basis von $U_i \setminus U_{i+1}$.
Seien für $j \in \{i+1, \dots z\}$ die Werte δ_j bekannt und $s_i := s - \sum_{j=i+1}^{z} j\delta_j$ sowie $m_i := \sum_{j=i+1}^{z}(j-(i-1))\delta_j = \log_l(\#C_{i+1})$ für $i < z$. Ist $i = z$, dann setzen wir s_z gleich s und m_z gleich 0. Zusammenfassend erhalten wir den folgenden Algorithmus, der für $1 \leq i \leq z$ zunächst ein Erzeugendensystem und anschließend daraus eine Basis von $U_i \setminus U_{i+1}$ berechnet.

Algorithmus 5 : Probabilistische Basisberechnung von $U_i \setminus U_{i+1}$ für $l \neq p$

Eingabe: • $\varepsilon \in (0,1)$, Klassengruppe C_F^0 von $F/\mathbb{F}_q$, Geschlecht g von F sowie $d(q,l)$.
• Primzahl $l \neq p$ mit $h_F = l^s c$ und $\mathrm{ggT}(l,c) = 1$ sowie z mit l^z maximale Ordnung in $C_F^0(l)$.
• Für $i = z$: $B_{i+1} = \widetilde{B}_{i+1} = \{\}$ andernfalls B_{i+1} Basis von U_{i+1} und $\widetilde{B}_{i+1}$ Teilmenge linear unabhängiger Elemente aus $C_{F'}^0/lC_{F'}^0$, so dass T_l eingeschränkt auf $\mathrm{Spann}(B'_{i+1}) \times \mathrm{Spann}(\widetilde{B}_{i+1})$ nicht-ausgeartet ist, wobei F' eine Konstantenkörpererweiterung von F vom Grad $d(q,l)$ und B'_{i+1} das Bild von B_{i+1} unter eine Einbettung von C_F^0 nach $C_{F'}^0$ ist; s_i und m_i.

Ausgabe: • Basis B_i von U_i mit Basiselementen von $U_i \backslash U_{i+1}$, falls diese existieren, der Form $l^{(i-1)} \cdot [D]$ und $\mathrm{Ord}([D]) = l^i$, Menge $\widetilde{B}_i$ mit $\#B_i$ linear unabhängige Elemente in $C_{F'}^0/lC_{F'}^0$ und δ_i, jeweils mit Wahrscheinlichkeit $\geq \varepsilon$ korrekt.

$j_1, j_2, j_3 \leftarrow$ Algorithmus 4$(\varepsilon, l^s, \#B_{i+1}, s_i, m_i, g, d(q,l))$\{Iterationsgrößen\}
$E_i \leftarrow \{\}, \widetilde{E}_i \leftarrow \{\}$

---

for $k = 1, \ldots, j_3$ **do** {Bestimmung eines Erzeugendensystems E_i von $U_i \setminus U_{i+1}$}
  Wähle $[D_k]$ zufällig aus C_F^0
  if $l^i c \cdot [D_k] = [0]$ **then**
    Füge $l^{(i-1)} c \cdot [D_k]$ in E_i ein
  end if
  if $\#L_1 = j_1$ **then**
    break
  end if
end for
$F' \leftarrow$ Konstantenkörpererweiterung von F vom Grad $d(q,l)$
Wähle zufällig $\widetilde{D}_1, \ldots, \widetilde{D}_{j_2} \in C_{F'}^0$
$\widetilde{E}_i \leftarrow \{\widetilde{D}_1, \ldots, \widetilde{D}_{j_2}\}$ {Erzeugendensystem von $C_{F'}^0 / lC_{F'}^0$}
$E_i', B_{i+1}' \leftarrow$ Einbetten von E_i beziehungsweise B_i in $C_{F'}^0[l]$
$B_i', \widetilde{B}_i \leftarrow$ Algorithmus 1$(B_{i+1}', E_i', \widetilde{B}_{i+1}, \widetilde{E}_i, \#B_{i+1} + \lfloor \frac{s_i}{i} \rfloor)$ {Basisberechnung}
$B_i \leftarrow$ Urbild von B_i' in $C_F^0[l]$ unter der Einbettung
return $B_i, \widetilde{B}_i, \#B_i - \#B_{i+1}$

Bemerkung 3.21. *Ist $d(q,l)$ der Einbettungsgrad von F bezüglich l gleich 1, dann ist F' eine Konstantenkörpererweiterung von F vom Grad 1. Also gilt $F' = F$.*

Satz 3.22. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper vom Geschlecht g und l ein zu q teilerfremder Primteiler der Klassenzahl von F. Weiter sei $C_F^0(l) \cong \bigoplus_{j=1}^{z} (\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ und $\varepsilon \in (0,1)$. Dann berechnet Algorithmus 5 mit einer Wahrscheinlichkeit größer gleich ε für $i \in \{1, \ldots, z\}$ zu der eben beschriebenen Eingabe eine Basis von U_i. Die Komplexität des Algorithmus ist gleich*

$$(\log(1/(1-\sqrt{\varepsilon})))^{\mathcal{O}(1)} g^{\mathcal{O}(1)} l^{\mathcal{O}(1)} \mathcal{O}\left(q^{g+\mathcal{O}(1)}\right)$$

Die Anzahl der zu speichernden Divisorenklassen vom Grad 0 entspricht $\mathcal{O}(\log(1/(1-\sqrt{\varepsilon})) + g)$.

Beweis. Wir definieren B_{z+1} und $\widetilde{B}_{z+1}$ als die leere Menge. Im letzten Teilabschnitt haben wir gesehen, dass Algorithmus 4 mit der Eingabe $\varepsilon, l^s, n_{i+1} = \#B_{i+1}, s_i, m_i, g$ und $d(q,l)$ die Iterationsgrößen j_1, j_2 und j_3 so berechnet, dass jeweils mit Wahrscheinlichkeit größer gleich $\sqrt{\varepsilon}$

1. j_1 zufällig gewählte Elemente aus U_i zusammen mit B_{i+1} den Vektorraum U_i erzeugen,

2. j_2 zufällig gewählte Elemente aus $C_{F'}^0 / lC_{F'}^0$ zusammen mit $\widetilde{B}_{i+1}$ den Vektorraum $C_{F'}^0 / lC_{F'}^0$ erzeugen, wobei F' eine Konstantenkörpererweiterung von F vom Grad $d(q,l)$ ist und

3. unter j_3 zufällig gewählten Elementen aus C_i zusammen mit B_{i+1} ein Erzeugendensystem von U_i ist.

Wegen $h_F = l^s c$ mit $\mathrm{ggT}(c, l) = 1$ ist für eine zufällig gewählte Divisorenklasse $[D_k]$ vom Grad 0 das Element $l^{(i-1)}c[D_k] = [l^{(i-1)}cD_k]$ zufällig aus $C_i = l^{(i-1)} \cdot C_F^0(l)$ gewählt. Gilt $l^i c[D_k] = l[l^{(i-1)}cD_k] = [0]$, dann ist $\mathrm{Ord}([cD_k]) \leq l^i$ und damit $[l^{(i-1)}cD_k] \in U_i$ zufällig gewählt. Ebenso ist für ein zufällig gewähltes Element $[D] \in C_{F'}^0$ dessen Klasse $[D] + lC_{F'}^0$ zufällig aus $C_{F'}^0/lC_{F'}^0$ gewählt. Nach den Überlegungen des letzten Teilabschnittes ist damit E_i ein Erzeugendensystem von $U_i \setminus U_{i+1}$ und $\widetilde{E}_i$ eines von $C_{F'}^0/lC_{F'}^0$, jeweils mit Wahrscheinlichkeit größer gleich $\sqrt{\varepsilon}$. Durch das Einbetten von E_i und B_i in $C_{F'}^0[l]$ sind alle Voraussetzungen erfüllt, so dass Algorithmus 1, wie in Teilabschnitt 2.1.3 beschrieben, daraus eine Basis von U_i berechnet. Diesmal aber in Abhängigkeit der Wahrscheinlichkeit ε. Falls $U_i \neq U_{i+1}$ gilt, beinhaltet $B_i \setminus B_{i+1}$ ausschließlich linear unabhängige Elemente aus $C_F^0[l]$ der Form $l^{i-1}[D]$. Demnach muss $\mathrm{Ord}([D]) = l^i$ gelten für alle $l^{i-1}[D] \in B_i \setminus B_{i+1}$. Andernfalls ist B_i gleich B_{i+1}.
Nach [Böt08, S.99] ist der Aufwand zum Bestimmen einer zufälligen Divisorenklasse aus C_F^0 polynomiell in $\log(q)$. Es werden maximal j_3 Elemente zufällig aus C_F^0 gewählt, mit einer Zahl multipliziert und getestet, ob das resultierende Element gleich $[0]$ ist. Wegen Bemerkung 1.30 ist die Komplexität dafür gleich $g^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(j_3)$. Nach Satz 2.10 ist F' eine Konstantenkörpererweiterung von F vom Grad kleiner gleich $\varphi(l) = l - 1$. Da wir j_2 Elemente zufällig aus $C_{F'}^0$ wählen, ist der Aufwand dafür gleich $l^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(j_2)$. Abschließend wird Algorithmus 1 mit den Erzeugendensystemen E_i' und $\widetilde{E}_i$ gestartet, wobei $\#E_i' \leq j_1$ und $\#\widetilde{E}_i = j_2$ gilt. Wegen Satz 2.12 entspricht der Aufwand dafür $g^{\mathcal{O}(1)}l^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(j_1 j_2)$. Zusammenfassend ist die Komplexität von Algorithmus 5 gleich

$$g^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(j_3) + l^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(j_2) + g^{\mathcal{O}(1)}l^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(j_1 j_2)$$

beziehungsweise $g^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(j_3) + g^{\mathcal{O}(1)}l^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(j_1 j_2)$. Mit den Abschätzungen von j_1, j_2 und j_3 aus Satz 3.19 erhalten wir die Komplexitätsbeschreibung des Satzes. Die Aussage zum Speicheraufwand folgt mit dem selben Satz aus $\#E_i' + \#\widetilde{E}_i \leq j_1 + j_2$.

□

Bemerkung 3.23. *Ist der Einbettungsgrad $d(q, l)$ gleich 1, so zeigt man analog, dass die Komplexität von Algorithmus 5 gleich* $\log(1/(1-\sqrt{\varepsilon}))^{\mathcal{O}(1)}g^{\mathcal{O}(1)}\mathcal{O}\left(q^{g+\mathcal{O}(1)}\right)$ *ist.*

Der Fall l ist gleich p

Ist l gleich der Charakteristik von $\mathbb{F}_q$, dann berechnen wir aus einem Erzeugendensystem E_i von $U_i \setminus U_{i+1}$ mit Algorithmus 2 eine Basis. Ist E_i ein Erzeugendensystem von $U_i \setminus U_{i+1}$ mit einer Wahrscheinlichkeit größer gleich ε, dann berechnet Algorithmus 2 mit der selben Wahrscheinlichkeit eine Basis davon. Seien s_i und m_i für $i \in \{1, \ldots, z\}$

wie schon zuvor definiert. Wir erhalten folgenden Algorithmus, der für festes i ein Erzeugendensystem von $U_i \setminus U_{i+1}$ und daraus eine Basis berechnet.

Algorithmus 6 : Probabilistische Basisberechnung von $U_i \setminus U_{i+1}$ für $l = p$

Eingabe: • $\varepsilon \in (0,1)$, Klassengruppe C_F^0 von $F/\mathbb{F}_q$, Geschlecht g von F, s_i und m_i.
• $h_F = p^s c$ mit $\mathrm{ggT}(p,c) = 1$ und z mit p^z maximale Ordnung in $C_F^0(p)$.
• $\widetilde{B}$ eine Basis von $\Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$, für $i < z$ eine Basis B_{i+1} von U_{i+1} andernfalls $B_{z+1} = \{\}$.

Ausgabe: • Basis B_i von U_i, mit Basiselementen von $U_i \setminus U_{i+1}$, falls diese existieren, der Form $p^{(i-1)} \cdot [D]$ mit $\mathrm{Ord}([D]) = p^i$ sowie δ_i die Dimension von $U_i \setminus U_{i+1}$, jeweils mit Wahrscheinlichkeit $\geq \varepsilon$ korrekt.

1: $E_i \leftarrow \{\}$
2: $j_1, j_3 \leftarrow$ Algorithmus 4$(\varepsilon^2, l^s, \#B_{i+1}, s_i, m_i, g, d(q,l))$ {Iterationsgrößen}
3: **for** $k = 1, \ldots, j_3$ **do** {Bestimmung eines Erzeugendensystems E_i von $U_i \setminus U_{i+1}$}
4: Wähle D_k zufällig aus C_F^0
5: **if** $p^i c \cdot D_k = [0]$ **then**
6: Füge $p^{(i-1)} c \cdot [D_k]$ in E_i ein
7: **end if**
8: **if** $\#L_1 = j_1$ **then**
9: **break**
10: **end if**
11: **end for**
12: $B_i \leftarrow$ Algorithmus 2$(E_i, \widetilde{B}, B_{i+1}, \#B_{i+1} + \lfloor \frac{s_i}{i} \rfloor)$ {Basisberechnung}
13: **return** $B_i, \#B_i - \#B_{i+1}$

Analog zum Beweis von Satz 3.22 beweist man auch die Korrektheit von Algorithmus 6. Die einzige Besonderheit ist, dass wir in diesem Fall, wie Eingangs erwähnt, nur ein Erzeugendensystem berechnen und deshalb Algorithmus 4 mit ε^2 anstelle von ε starten. Mit Satz 2.15 kann man ebenso analog zeigen, dass die Komplexität von Algorithmus 6 gleich $g^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(j_3) + g^{\mathcal{O}(1)}(\log(q))^{\mathcal{O}(1)}\mathcal{O}(j_1)$ ist. Mit den Schranken für j_1 und j_3 aus Bemerkung 3.20 folgt abschließend, dass die Komplexität von Algorithmus 6

$$\mathcal{O}\left(\log(1/(1-\varepsilon))q^{g+\mathcal{O}(1)}\right) g^{\mathcal{O}(1)} \tag{3.16}$$

entspricht. Es müssen $\mathcal{O}(\log(1/(1-\sqrt{\varepsilon}))+g)$ viele Elemente aus C_F^0 gespeichert werden.

3.4.4 Berechnung von $C_F^0(l)$

Unser Ziel ist es, die Klassengruppe C_F^0 eines Funktionenkörpers $F/\mathbb{F}_q$ zu berechnen. Wir beschreiben in diesem Teilabschnitt wie mit Hilfe der vorhergehenden Überlegungen

die l-Sylow-Gruppe $C_F^0(l)$ ($\cong \bigoplus_{j=1}^{z}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$) der Klassengruppe für einen Primteiler l der Klassenzahl bestimmt wird. Bezeichne fortan $\mathbb{P}$ die Menge der Primzahlen.

Algorithmus 7 : Probabilistische l-Sylow-Gruppen Berechnung

Eingabe: Klassengruppe C_F^0 von $F/\mathbb{F}_q$, g Geschlecht von F, $l \in \mathbb{P}$ mit $h_F = l^s c$ und $\mathrm{ggT}(l, c) = 1$ und $\varepsilon \in (0, 1)$.

Ausgabe: Basis $B := \{l^{i-1}[D_{j_i}] \mid i \in \{1, \ldots, z\}$ und $j_i \in \{1, \ldots, \delta_i\}\}$ von $C_F^0[l]$ sowie $\bigoplus_{i=1}^{z}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i} (\cong C_F^0(l))$, beides mit hoher Wahrscheinlichkeit korrekt.

$G \leftarrow \langle 0 \rangle$ triviale Gruppe
$B, \widetilde{B} \leftarrow \{\}$
$z \leftarrow$ Algorithmus $3(C_F^0, l^s c, \varepsilon)\{l^z$ ist maximale Ordnung in $C_F^0(l)\}$
$i \leftarrow z$, $d \leftarrow d(q, l)$, $s_z \leftarrow s$, $m_z \leftarrow 0$
$\eta_z \leftarrow \min\{2g, s_z - z + 1\}\{$Obere Schranke von $\dim_{\mathbb{F}_l}(C_F^0[l])\}$
if $l = p$ **then**
  $\widetilde{B} \leftarrow$ Basis von $\Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$
  $\eta_z \leftarrow \#\widetilde{B}$
end if
if $s_z = z \cdot \eta_z$ **then**
  $B, \widetilde{B} \leftarrow$ Algorithmus $5/6(\varepsilon, C_F^0, g, d, l^s c, z, B, \widetilde{B}, s_z, m_z)\{B$ Basis von $C_F^0[l]\}$
  return $B, (\mathbb{Z}/l^z\mathbb{Z})^{\eta_z}$
end if
while true do
  HELP$\leftarrow$ **false**
  $s_i \leftarrow s - \sum_{j=i+1}^{z} j\delta_j\{l^{s_i} = \#\bigoplus_{j=1}^{i}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}\}$
  if $i > s_i$ **then**$\{$maximale Ordnung in $\bigoplus_{j=1}^{i}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ ist größer als $l^{s_i}\}$
    $\delta_j \leftarrow 0$ für $j \in \{s_i + 1, \ldots, i\}$, $i \leftarrow s_i$
    Aktualisiere Index von η_i, s_i, m_i durch neues i
  end if
  $m_i \leftarrow \sum_{j=i+1}^{z}(j - (i-1))\delta_j\{l^{m_i} = \#C_{i+1}\}$
  if $i = 1$ **then**
    $B \leftarrow$ Algorithmus $5/6(\varepsilon, C_F^0, g, d, l^s c, z, B, \widetilde{B}, s_1, m_1)\{B$ Basis von $C_F^0[l]\}$
    return $B, G \oplus (\mathbb{Z}/l\mathbb{Z})^{s_1}$
  else if $s_i = i \cdot \eta_i$ **then**
    $B \leftarrow$ Algorithmus $5/6(\varepsilon, C_F^0, g, d, l^s c, z, B, \widetilde{B}, s_i, m_i)\{B$ Basis von $C_F^0[l]\}$
    return $B, G \oplus (\mathbb{Z}/l^i\mathbb{Z})^{\eta_i}$
  end if
  if $s_i - i < \eta_i - 1$ **then**$\{\delta_i = 0$ oder $\eta_i > \dim_{\mathbb{F}_l}(C_F^0[l] \setminus U_{i+1})\}$
    TEMP$\leftarrow \eta_i$, HELP$\leftarrow$ **true**$\{$Angenommen $\delta_i \neq 0\}$
    $\eta_i \leftarrow s_i - i + 1\{\eta_i \geq \dim_{\mathbb{F}_l}(C_F^0[l] \setminus U_{i+1})\}$
  end if

if $l \neq p$ **then**
  $B, \widetilde{B}, \delta_i \leftarrow$ Algorithmus $5(\varepsilon, C_F^0, g, d, l^s c, z, B, \widetilde{B}, s_i, m_i)\{B$ Basis von $U_i\}$
else
  $B, \delta_i \leftarrow$ Algorithmus $6(\varepsilon, C_F^0, g, l^s c, z, B, \widetilde{B}, s_i, m_i)\{B$ Basis von $U_i\}$
end if
if $\delta_i = 0$ **then**
  if HELP= **true then**
    $\eta_i \leftarrow$TEMP{Annahme $\delta_i \neq 0$ war falsch, Aktualisierung von η_i}
  end if
else
  $G \leftarrow G \oplus (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}\{G = \bigoplus_{j=i}^{z}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ mit hoher Wahrscheinlichkeit $\}$
end if
$i \leftarrow i-1$ und $\eta_i \leftarrow \eta_{i+1} - \delta_{i+1}$
end while

Bemerkung 3.24. *Mit Algorithmus 5/6 ist gemeint, dass im Fall $l \neq p$ Algorithmus 5 mit der entsprechenden Eingabe gestartet wird und andernfalls Algorithmus 6. Für $l = p$ spielt der Einbettungsgrad $d = d(q, l)$ keine Rolle.*

Satz 3.25. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper vom Geschlecht g und l ein Primteiler dessen Klassenzahl. Für $\varepsilon \in (0,1)$ berechnet Algorithmus 7 mit hoher Wahrscheinlichkeit die Gruppenstruktur der l-Sylow-Gruppe der Klassengruppe von F korrekt. Die Komplexität des Algorithmus ist gleich*

$$\left(\log(1/(1-\sqrt{\varepsilon}))\right)^{\mathcal{O}(1)} g^{\mathcal{O}(1)} l^{\mathcal{O}(1)} \mathcal{O}\left(q^{g+\mathcal{O}(1)}\right)$$

und die Anzahl der zu speichernden Divisorenklassen vom Grad 0 entspricht $\mathcal{O}(\log(1/(1-\sqrt{\varepsilon})) + g)$.

Beweis. Wir betrachten den Fall $l \neq p$. Anschließend wird $l = p$ untersucht. Algorithmus 5 berechnet zur Erinnerung für die l-Sylow-Gruppe der Klassengruppe mit $C_F^0(l) \cong \bigoplus_{j=1}^{z}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ für $i \in \{1, \ldots, z\}$ den Wert δ_i sowie eine Basis $\{l^{\gamma-1}[D_{j_\gamma}] \mid \gamma \in \{i, \ldots, z\}$ und $j_\gamma \in \{1, \ldots, \delta_i\}\}$ des $\mathbb{F}_l$-Untervektorraumes U_i von $C_F^0[l]$, so dass $\langle\{[D_{j_\gamma}] \mid \gamma \in \{i, \ldots, z\}$ und $j_\gamma \in \{1, \ldots, \delta_i\}\}\rangle$ isomorph zu $\bigoplus_{j=i}^{z}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ ist. Durch $\#(l^{(i-1)} \cdot C_F^0(l)) = \#C_i \leq l^{\delta_i + m_i}$ wird die Größe m_i bestimmt. Ist $i = z$, dann gilt $\#C_i = l^{\delta_z}$ und $m_z = 0$.
In Teilabschnitt 3.4.1 haben wir bereits bewiesen, dass Algorithmus 3 in Abhängigkeit der Wahrscheinlichkeit ε den Wert z richtig bestimmt, so dass l^z die maximale Ordnung in $C_F^0(l)$ ist. Nach Lemma 3.4 gilt $\dim_{\mathbb{F}_l}(C_F^0[l]) \leq \eta_z := \min\{2g, s_z - z + 1\}$, also ist $s_z = \log_l(\#C_F^0(l)) \leq zn_z$. Bei Gleichheit haben wir $C_F^0(l) \cong (\mathbb{Z}/l^z\mathbb{Z})^{\eta_z}$ und Algorithmus 5 berechnet mit der entsprechenden Eingabe korrekt eine Basis B von $U_z = C_F^0[l]$ mit einer Wahrscheinlichkeit größer gleich ε. Ist andererseits $s_z < zn_z$ beginnt für $i = z$ die While-Schleife mit $z > 1$, denn bei Gleichheit wäre $\eta_z = \min\{2g, s_z - z + 1\} = s_z$

und damit $s_z = z\eta_z$ wie schon zuvor. Für $z = s_z$ gelte $\eta_z = \min\{2g, 1\} = 1$ und somit ebenfalls $s_z = z\eta_z$. Wegen $z < s_z$ und $\eta_z \leq s_z - z + 1$ wird direkt Algorithmus 5 gestartet, der wegen $m_z = 0$ für die entsprechende Eingabe mit Wahrscheinlichkeit $\geq \varepsilon$ den korrekten Wert δ_z und eine Basis von U_z bestimmt. Da η_z eine obere Schranke von $\dim_{\mathbb{F}_l}(C_F^0[l])$ ist, beschränkt $\eta_{z-1} := \eta_z - \delta_z$ die Dimension von $C_F^0[l] \setminus U_z$.
Wir zeigen, dass in der i-ten Schleife mit $1 \leq i < z$ der Algorithmus eine Basis von U_i sowie den Wert δ_i mit einer Wahrscheinlichkeit größer gleich ε korrekt berechnet und dass der Algorithmus terminiert. Wir betrachten die Gruppe $G := \bigoplus_{j=i+1}^{z}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ und definieren $G_i := \bigoplus_{j=1}^{i}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$, dann ist

$$C_F^0(l) \cong G \oplus G_i.$$

Seien für $j \in \{i+1, \ldots, z\}$ die Werte δ_j, j und η_j sowie eine Basis von U_{i+1} in Abhängigkeit von ε korrekt berechnet. Dabei bezeichnet η_j eine obere Schranke der Dimension von $C_F^0[l] \setminus U_{j+1}$. Wir werden entlang der Reihenfolge des Pseudocodes die Korrektheit des Algorithmus beweisen.

Haben wir $i > s_i$, muss $\delta_j = 0$ für $j \in \{s_i + 1, \ldots, i\}$ gelten und i auf δ_i gesetzt sowie η_i, s_i und m_i durch den neuen Index i aktualisiert werden. Da δ_j gleich Null gesetzt wird für $j \in \{s_i + 1, \ldots, i\}$, bleiben die Werte η_i, s_i und m_i auch für den neuen Index korrekt. Es gilt dann $\log_l(\#G_i) = s_i \geq i$.
Für i gleich 1 ist $G_i = (\mathbb{Z}/l\mathbb{Z})^{s_i}$ und damit $C_F^0(l) \cong G \oplus (\mathbb{Z}/l\mathbb{Z})^{s_i}$. Algorithmus 5 berechnet nun in Abhängigkeit von ε eine Basis von U_i.
Da nach Voraussetzung $\eta_{i+1} \geq \dim_{\mathbb{F}_l}(C_F^0[l] \setminus U_{i+2}) = n - \sum_{j=i+2}^{z}\delta_j = \sum_{j=1}^{i+1}\delta_j$ ist, beschränkt $\eta_i := \eta_{i+1} - \delta_{i+1}$ den Wert $\sum_{j=1}^{i+1}\delta_j - \delta_{i+1} = \dim_{\mathbb{F}_l}(C_F^0[l] \setminus U_{i+1})$, also die Dimension von $G_i[l]$.
Die Anzahl l^{s_i} der Elemente von G_i muss kleiner gleich als $l^{i\eta_i}$ sein. Bei Gleichheit gilt offensichtlich $G_i = (\mathbb{Z}/l^i\mathbb{Z})^{\eta_i}$ und damit $C_F^0(l) \cong G \oplus (\mathbb{Z}/l^i\mathbb{Z})^{\eta_i}$. Mit Hilfe von Algorithmus 5 bestimmen wir anschließend eine Basis von $C_F^0[l]$ in Abhängigkeit von ε.

Für $\delta_i \geq 1$ ist nach Lemma 3.3 $s_i - i \geq \dim_{\mathbb{F}_l}(C_F^0[l]) - \sum_{j=i+1}^{z}\delta_j - 1 = \dim_{\mathbb{F}_l}(G_i[l]) - 1$. Da nach Voraussetzung δ_j für $j > i$ in Abhängigkeit von ε korrekt ist, ist es auch $s_i = s - \sum_{j=i+1}^{z} j\delta_j$. Wegen $\dim_{\mathbb{F}_l}(G_i[l]) \leq \eta_i$ folgt aus $s_i - i < \eta_i - 1$, dass i oder η_i zu groß ist. Das heißt, es gilt $\delta_i = 0$, also gibt es kein Element in G_i der Ordnung l^i oder wir haben $\eta_i > \dim_{\mathbb{F}_l}(G_i[l])$. Angenommen ein Element der Ordnung l^i existiert (für $i = z$ ist die Existenz garantiert), dann ist i korrekt und wir können die Ungleichung aus Lemma 3.3 zu $\dim_{\mathbb{F}_l}(G_i[l]) \leq s_i - i + 1$ umformen und setzen $\eta_i := s_i - i + 1$. Wir merken uns den Wert der alten Schranke. Wegen $s_i \geq i$ ist $\eta_i > 0$ und wir berechnen mit entsprechender Eingabe in Abhängigkeit von ε den Wert δ_i mittels Algorithmus 5. Ergibt sich $\delta_i = 0$, dann hat G_i mit Wahrscheinlichkeit $\geq \varepsilon$ kein Element der Ordnung l^i und wir müssen die neue Schranke η_i wieder mit der alten aktualisieren.
Ist $\delta_i > 0$, dann ist die i-te zyklische Komponente von $C_F^0(l)$ bis auf Isomorphie in Abhängigkeit von ε gerade $(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$. Setzten wir $G := \bigoplus_{j=i}^{z}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ und definieren

$G_{i-1} := \bigoplus_{j=1}^{i-1}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ sowie $\eta_{i-1} := \eta_i - \delta_i$, dann berechnet Algorithmus 7 im nächsten Schritt δ_{i-1} und eine Basis von U_{i-1} in Abhängigkeit von ε korrekt.
Nach maximal z Wiederholungen ist $i = 1$ und der Algorithmus terminiert. Dann haben wir für jedes $i \in \{1, \ldots, z\}$ den Wert δ_i korrekt berechnet sowie eine Basis B von $C_F^0[l]$, jeweils in Abhängigkeit von ε. Da Algorithmus 5 in jeder Schleife eine Basis der Form $\{l^{\gamma-1}[D_{j_\gamma}] \mid \gamma \in \{i, \ldots, z\} \text{ und } j_\gamma \in \{1, \ldots, \delta_i\}\}$ von U_i bestimmt, ist B gleich $\{l^{i-1}[D_{j_i}] \mid i \in \{1, \ldots, z\} \text{ und } j_i \in \{1, \ldots, \delta_i\}\}$.
Gilt $l = p$ dann wissen wir aus Teilabschnitt 3.4.3 von der Korrektheit von Algorithmus 6. In diesem Fall beweist man die Richtigkeit von Algorithmus 7 analog wie zuvor. Wir analysieren nun die Komplexität des Algorithmus:
Einmalig berechnen wir mittels Algorithmus 3 den Wert z, so dass l^z die größte Ordnung in $C_F^0(l)$ ist. Nach Satz 1.18 gilt $\#C_F^0(l) \leq h_F \leq (\sqrt{q}+1)^{2g}$ und damit $z = \mathcal{O}(g\log(q))$.
Für den Fall $l \neq p$ wird maximal z-mal Algorithmus 5 gestartet. Mit der Komplexitätsbeschreibung von Algorithmus 3 und 5 aus Satz 3.6 beziehungsweise Satz 3.22 ergibt sich damit für Algorithmus 7 im Fall $l \neq p$ ein Gesamtaufwand von

$$g^{\mathcal{O}(1)}\log(q)^{\mathcal{O}(1)}\mathcal{O}\left(\log(1/(1-\varepsilon))\right) + \left(\log(1/(1-\sqrt{\varepsilon}))\right)^{\mathcal{O}(1)} g^{\mathcal{O}(1)} l^{\mathcal{O}(1)} \mathcal{O}\left(q^{g+\mathcal{O}(1)}\right).$$

Ist $l = p$, dann berechnen wir zusätzlich zu Algorithmus 3 einmalig eine Basis von $\Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$. Der Aufwand dafür ist polynomiell in g und $\log(q)$. Anstelle von Algorithmus 5 benutzen wir Algorithmus 6. Wegen (3.16) ergibt sich als Schranke für die Komplexität von Algorithmus 7 analog

$$g^{\mathcal{O}(1)}\log(q)^{\mathcal{O}(1)}\mathcal{O}\left(\log(1/(1-\varepsilon))\right) + \mathcal{O}\left(\log(1/(1-\varepsilon)q^{g+\mathcal{O}(1)}\right) g^{\mathcal{O}(1)}.$$

Weil $\log(1/(1-\varepsilon)) \leq \log(1/(1-\sqrt{\varepsilon}))$ ist, können wir die Anzahl der Operationen von Algorithmus 7 in beiden Fällen durch

$$\left(\log(1/(1-\sqrt{\varepsilon}))\right)^{\mathcal{O}(1)} g^{\mathcal{O}(1)} l^{\mathcal{O}(1)} \mathcal{O}\left(q^{g+\mathcal{O}(1)}\right)$$

beschreiben. Der Speicheraufwand von Algorithmus 7 entspricht offensichtlich dem von Algorithmus 5 und 6. □

Bemerkung 3.26. *Ist der Einbettungsgrad $d(q,l)$ von F bezüglich l gleich eins, dann entspricht die Komplexität von Algorithmus 7 wegen Bemerkung 3.23*

$$\left(\log(1/(1-\sqrt{\varepsilon}))\right)^{\mathcal{O}(1)} g^{\mathcal{O}(1)} \mathcal{O}\left(q^{g+\mathcal{O}(1)}\right).$$

3.4.5 Berechnung der Klassengruppe

Zusammenfassend werden wir in diesem Teilabschnitt den ersten Algorithmus zur Berechnung der Divisorenklassengruppe eines globalen Funktionenkörpers beschreiben. Sei

F ein Funktionenkörper vom Geschlecht g über dem endlichen Körper $\mathbb{F}_q$ und h_F dessen Klassenzahl. Die Divisorenklassengruppe C_F von F ist isomorph zu $C_F^0 \oplus \langle [D] \rangle$, wobei $[D]$ einen Divisor vom Grad 1 bezeichnet. Seien $l_1, \ldots, l_n$ die Primteiler von h_F, dann gilt $C_F^0 = \bigoplus_{i=1}^n C_F^0(l_i)$. Kennen wir die Gruppeninvarianten der l_i-Sylow-Gruppen der Klassengruppe für jedes i, dann können wir die Gruppeninvarianten von C_F^0 angeben. Haben wir zusätzlich ein minimales Erzeugendensystem von $C_F^0(l_i)$ für jedes i, dann konstruieren wir, wie in Bemerkung 3.1 beschrieben, daraus eines von C_F^0. Durch Hinzunahme eines Divisors vom Grad 1 erhalten wir dann C_F.
Der folgende Algorithmus berechnet in Abhängigkeit der Wahrscheinlichkeit $\varepsilon \in (0,1)$ für den Funktionenkörper F, dessen Klassenzahl bekannt ist, die Struktur der Klassengruppe. Damit meinen wir zur Erinnerung die Gruppeninvarianten $c_1, \ldots, c_n \in \mathbb{Z}^{>0}$ und Divisoren $D_1, \ldots, D_n$ vom Grad 0 mit

$$C_F^0 = \langle [D_1] \rangle \oplus \cdots \oplus \langle [D_n] \rangle \cong (\mathbb{Z}/c_1\mathbb{Z}) \oplus \cdots \oplus (\mathbb{Z}/c_n\mathbb{Z}).$$

Algorithmus 8 : Probabilistische Klassengruppenberechnung

Eingabe: Klassengruppe C_F^0 eines Funktionenkörper F vom Geschlecht g über dem endlichen Körper $\mathbb{F}_q$, h_F die Klassenzahl von F und $\varepsilon \in (0,1)$.

Ausgabe: $(\mathbb{Z}/c_1\mathbb{Z}) \oplus \cdots \oplus (\mathbb{Z}/c_n\mathbb{Z}) (\cong C_F^0)$ sowie ein minimales Erzeugendensystem von C_F^0, beides in Abhängigkeit von ε korrekt.

1: **for** $l \in \mathbb{P}$ mit $l | h_F$ **do**
2: Berechne $h_F = l^s c$ mit $\mathrm{ggT}(l, c) = 1$
3: $\widetilde{B}_l, G_l \leftarrow$ Algorithmus 7$(C_F^0, g, l^s c, \varepsilon)$
4: **end for**
5: **return** $\bigoplus_{l|h_F} G_l, \sum_{l|h_F} \widetilde{B}_l$ {gemäß der in (3.8) definierten Addition}

Bemerkung 3.27. *Eigentlich ist die Ausgabe von Algorithmus 7 eine Basis $\widetilde{B}_l$ von $C_F^0[l]$ der Form $\{l^{i-1}[D_{j_i}] \mid i \in \{1, \ldots, z\}$ und $j_i \in \{1, \ldots, \delta_i\}\}$. Wir betrachten stattdessen das minimale Erzeugendensystem $B_j := \{[D_{j_i}] \mid i \in \{1, \ldots, z\}$ und $j_i \in \{1, \ldots, \delta_i\}\}$ der Gruppe $C_F^0(l)$. Für die Komplexität des Algorithmus ist das irrelevant.*

Satz 3.28. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper vom Geschlecht g und die Klassenzahl h_F von F bekannt. Für $\varepsilon \in (0,1)$ berechnet Algorithmus 8 die Struktur der Klassengruppe von F in Abhängigkeit von ε korrekt. Dessen Komplexität ist*

$$\exp(\widetilde{O}((q \log(q))^{\frac{1}{3}}) + \left(\log(1/(1-\sqrt{\varepsilon}))\right)^{\mathcal{O}(1)} g^{\mathcal{O}(1)} \beta^{\mathcal{O}(1)} \mathcal{O}\left(q^{g+\mathcal{O}(1)}\right),$$

wobei β das Maximum aller Primteiler von h_F bezeichnet. Es müssen $\mathcal{O}(\log(1/(1-\sqrt{\varepsilon})) + g)$ Divisorenklassen vom Grad 0 gespeichert werden.

Beweis. Die Korrektheit des Algorithmus folgt unmittelbar aus der Korrektheit von Algorithmus 7 und den Überlegungen zuvor. Da die Klassenzahl h_F endlich ist, gibt es auch nur endlich viele Primteiler davon. Folglich terminiert Algorithmus 8 nach endlich vielen Schritten.
Um alle Primteiler l der Klassenzahl h_F zu bestimmen mit dem Exponenten s, so dass l^s die Klassenzahl exakt teilt, faktorisieren wir h_F. Nach [vzGG03, S.531] kann man das beispielsweise mit dem Zahlenkörpersieb in $\exp(\widetilde{\mathcal{O}}(\log(h_F))^{\frac{1}{3}})$ Rechenoperationen realisieren. Informationen zum Zahlenkörpersieb finden sich in [LL93]. Da nach Satz 1.18 die Klassenzahl durch $(\sqrt{q}+1)^{2g}$ nach oben beschränkt ist, können wir den Aufwand zum Faktorisieren der Klassenzahl mittels $\exp(\widetilde{O}((g\log(q))^{\frac{1}{3}})$ beschreiben.
Anschließend müssen wir für jeden Primteiler von h_F Algorithmus 7 aufrufen. Die Anzahl der Primteiler der Klassenzahl ist echt kleiner als $\log(h_F) \leq \log((\sqrt{q}+1)^{2g})$ und damit gleich $\mathcal{O}(g\log(q))$. Zusammenfassend entspricht der Komplexität von Algorithmus 8, wegen des im Satz 3.25 beschriebenen Aufwands von Algorithmus 7, gerade

$$\exp(\widetilde{O}((g\log(q))^{\frac{1}{3}}) + \left(\log(1/(1-\sqrt{\varepsilon}))\right)^{\mathcal{O}(1)} g^{\mathcal{O}(1)}\beta^{\mathcal{O}(1)}\mathcal{O}\left(q^{g+\mathcal{O}(1)}\right),$$

wobei β das Maximum aller Primteiler von h_F bezeichnet. Der Speicheraufwand überträgt sich von Algorithmus 7. □

Bemerkung 3.29.

(i) Die Komplexität des Zahlenkörpersiebes ist eine erwartete. Die Annahmen, die dazu führen, können in [LL93, S.80-84] nachgelesen werden.

(ii) Wollen wir unter den gleichen Voraussetzungen des letzten Satzes die Divisorenklassengruppe C_F bestimmen, müssen wir nur zusätzlich zu Algorithmus 8 einen Divisor vom Grad 1 berechnen. Benutzen wir [Hesed, Algorithm 33], dann ist der Aufwand dafür polynomiell in g und $\log(q)$. Folglich ist die Komplexität zur Berechnung von C_F mittels Algorithmus 8 die gleiche wie in Satz 3.28.

(iii) Ist für jeden Primteiler l der Klassenzahl der Einbettungsgrad $d(q,l)$ gleich 1, dann können wir wegen Bemerkung 3.26 die Komplexität von Algorithmus 8 zu Berechnung sowohl der Divisorenklassengruppe als auch der Klassengruppe durch

$$\exp(\widetilde{O}((g\log(q))^{\frac{1}{3}}) + \left(\log(1/(1-\sqrt{\varepsilon}))\right)^{\mathcal{O}(1)} g^{\mathcal{O}(1)}\mathcal{O}\left(q^{g+\mathcal{O}(1)}\right)$$

beschreiben.

3.5 Struktur von C_F^0 ohne Erzeuger

In diesem Abschnitt werden wir eine vereinfachte Version von Algorithmus 8 vorstellen, die ausschließlich die Gruppeninvarianten der Klassengruppe eines globalen Funktionenkörpers berechnet. Motivierend dafür sind zum Beispiel die globalen elliptischen

Funktionenkörper, also die globalen Funktionenkörper vom Geschlecht 1. Betrachten wir einen solchen Funktionenkörper $F/\mathbb{F}_q$, dann gilt für die l-Sylow-Gruppe der Klassengruppe C_F^0 wegen (1.1)

$$C_F^0(l) \cong (\mathbb{Z}/l^z\mathbb{Z}) \oplus (\mathbb{Z}/l^{(s-z)}\mathbb{Z}), \tag{3.17}$$

wobei l^s die Klassenzahl von F exakt teilt und l^z die maximale Ordnung in $C_F^0(l)$ bezeichnet. Kennen wir s, dann können wir mittels Algorithmus 3 in Abhängigkeit von einer Wahrscheinlichkeit ε den Wert für z ermitteln und damit die Gruppeninvarianten von $C_F^0(l)$. Sind s und z groß sowie $(s-z)$ klein, dann wird es verhältnismäßig aufwendig ein Element in $C_F^0(l)$ durch zufälliges Wählen zu finden, dessen 2. Komponente gemäß (3.17) multipliziert mit $l^{(z-s)-1}$ ungleich 0 ist. Das heißt, die Konstruktion eines Erzeugendensystems des 2-dimensionalen $\mathbb{F}_l$-Vektorraumes $C_F^0[l]$, wie in Teilabschnitt 3.4.3 beschrieben, wird aufwendig und dadurch auch Algorithmus 7 und 8.
Verzichten wir auf die Berechnung von Erzeugern der Klassengruppe, dann können wir in diesem Fall die Invarianten von $C_F^0(l)$ ausschließlich mit Algorithmus 3 bestimmen. Das Ergebnis ist mit einer Wahrscheinlichkeit größer gleich ε korrekt. Nach Satz 3.6 berechnen wir auf diese Weise die Gruppeninvarianten der l-Sylow-Gruppe der Klassengruppe eines globalen elliptischen Funktionenkörpers mit dem Aufwand

$$g^{\mathcal{O}(1)} \log(q)^{\mathcal{O}(1)} \left(\log(1/(1-\sqrt{\varepsilon})\right)^{\mathcal{O}(1)}.$$

Allgemein bestimmen wir mit Hilfe der folgenden Algorithmen die Gruppeninvarianten einer Klassengruppe eines globalen Funktionenkörpers, dessen Klassenzahl bekannt ist. Zuerst stellen wir eine vereinfachte Version von Algorithmus 7 vor und anschließend beschreiben wir einen Algorithmus zur Berechnung der gesamten Klassengruppe eines globalen Funktionenkörpers.

Algorithmus 9 : l-Sylow-Gruppen Berechnung ohne Erzeuger

Eingabe: Klassengruppe C_F^0 eines Funktionenkörpers $F/\mathbb{F}_q$, g Geschlecht von F, Primteiler l der Klassenzahl mit $h_F = l^s c$ und $\mathrm{ggT}(l,c) = 1$ sowie $\varepsilon \in (0,1)$.
Ausgabe: $\bigoplus_{i=1}^{z} (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i} (\cong C_F^0(l))$ mit hoher Wahrscheinlichkeit korrekt.

$G \leftarrow \langle 0 \rangle$ triviale Gruppe,
$B, \widetilde{B} \leftarrow \{\}$
$z \leftarrow$ Algorithmus $3(C_F^0, l^s c, \varepsilon)\{l^z$ ist maximale Ordnung in $C_F^0(l)\}$
$i \leftarrow z$, $d \leftarrow d(q,l)$, $s_z \leftarrow s$, $m_z \leftarrow 0$
$\eta_z \leftarrow \min\{2g, s_z - z + 1\}\{$Obere Schranke von $\dim_{\mathbb{F}_l}(C_F^0[l])\}$
if $l = p$ **then**
  $\widetilde{B} \leftarrow$ Basis von $\Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$
  $\eta_z \leftarrow \#\widetilde{B}$
end if

if $s_z = z \cdot \eta_z$ **then**
  return $(\mathbb{Z}/l^z\mathbb{Z})^{\eta_z}$
end if
while true do
  HELP$\leftarrow$ **false**
  $s_i \leftarrow s - \sum_{j=i+1}^{z} j\delta_j \{l^{s_i} = \#\bigoplus_{j=1}^{i}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}\}$
  if $i > s_i$ **then** {maximale Ordnung in $\bigoplus_{j=1}^{i}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ ist größer als l^{s_i}}
    $\delta_j \leftarrow 0$ für $j \in \{s_i + 1, \ldots, i\}$ und $i \leftarrow s_i$
    Aktualisiere Index von η_i, s_i, m_i durch neues i
  end if
  $m_i \leftarrow \sum_{j=i+1}^{z}(j - (i-1))\delta_j \{l^{m_i} = \#C_{i+1}\}$
  if $i = 1$ **then**
    return $G \oplus (\mathbb{Z}/l\mathbb{Z})^{s_1}$
  else if $s_i = i \cdot \eta_i$ **then**
    return $G \oplus (\mathbb{Z}/l^i\mathbb{Z})^{\eta_i}$
  end if
  if $s_i - i < \eta_i - 1$ **then** $\{\delta_i = 0$ oder $\eta_i > \dim_{\mathbb{F}_l}(C_F^0[l] \setminus U_{i+1})\}$
    TEMP$\leftarrow \eta_i$, HELP$\leftarrow$ **true**{Angenommen $\delta_i \neq 0$}
    $\eta_i \leftarrow s_i - i + 1$
  end if
  if $l \neq p$ **then**
    $B, \widetilde{B}, \delta_i \leftarrow$ Algorithmus $5(\varepsilon, C_F^0, g, d, l^s c, z, B, \widetilde{B}, s_i, m_i)\{B$ Basis von $U_i\}$
  else
    $B, \delta_i \leftarrow$ Algorithmus $6(\varepsilon, C_F^0, g, l^s c, z, B, \widetilde{B}, s_i, m_i)\{B$ Basis von $U_i\}$
  end if
  if $\delta_i = 0$ **then**
    if HELP= **true then**
      $\eta_i \leftarrow$TEMP{Annahme $\delta_i \neq 0$ war falsch, Aktualisierung von η_i}
    end if
  else
    $G \leftarrow G \oplus (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i} \{G = \bigoplus_{j=i}^{z}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ mit hoher Wahrscheinlichkeit $\}$
  end if
  $i \leftarrow i - 1$ und $\eta_i \leftarrow \eta_{i+1} - \delta_{i+1}$
end while

Bemerkung 3.30. *Interessieren wir uns ausschließlich für die Dimension der l-Torsionsgruppe der Klassengruppe eines globalen Funktionenkörpers, dann eignet sich der letzte Algorithmus besonders gut. Im Kapitel 5 werden wir eine Anwendung aus der algebraischen Geometrie diesbezüglich vorstellen.*

Algorithmus 10 : Klassengruppenberechnung ohne Erzeuger

Eingabe: Klassengruppe C_F^0 eines Funktionenkörpers $F/\mathbb{F}_q$, Geschlecht g von F, h_F, Klassenzahl h_F von F und $\varepsilon \in (0,1)$.

Ausgabe: $(\mathbb{Z}/c_1\mathbb{Z}) \oplus \cdots \oplus (\mathbb{Z}/c_n\mathbb{Z})(\cong C_F^0)$ in Abhängigkeit von ε korrekt.

1: **for** $l \in \mathbb{P}$ mit $l|h_F$ **do**
2: Berechne $h_F = l^s c$ mit $\mathrm{ggT}(l,c) = 1$
3: $G_l \leftarrow$ Algorithmus 9$(C_F^0, g, l^s c, \varepsilon)$
4: **end for**
5: **return** $\bigoplus_{l|h_F} G_l$

Da die letzten beiden Algorithmen nur Spezialfälle von Algorithmus 7 beziehungsweise 8 sind, folgt ihre Korrektheit unmittelbar. Die Komplexitätsbeschreibung überträgt sich ebenfalls. In Kapitel 4 werden wir sehen wie schnell die zuvor beschriebenen Algorithmen im praktischen Vergleich sind.

3.6 Nicht-probabilistischer Algorithmus

In den vorhergehenden Abschnitten haben wir einen probabilistischen Algorithmus beschrieben, welcher die Struktur der Klassengruppe beziehungsweise der Divisorenklassengruppe eines globalen Funktionenkörpers F berechnet.
Sei nun angenommen, dass $S = \{[D_1], \ldots, [D_f]\}$ ein Erzeugendensystem der Klassengruppe C_F^0 eines Funktionenkörpers F vom Geschlecht g über $\mathbb{F}_q$ ist. Sei dabei $q = p^k$ und p eine Primzahl sowie $k \in \mathbb{Z}^{>0}$. Wir werden im Folgenden einen nicht-probabilistischen Algorithmus vorstellen, der die Struktur von C_F^0 bestimmt. Damit meinen wir die Gruppeninvarianten sowie ein minimales Erzeugendensystem von C_F^0. Durch Hinzunahme eines Divisors vom Grad 1 erhalten wir die Divisorenklassengruppe C_F. Auch hier sei vorausgesetzt, dass die Klassenzahl h_F bekannt ist. Weiter nehmen wir an, dass die Repräsentanten der Divisorenklassen aus S den Annahmen aus Teilabschnitt 1.5.2 genügen.
Wie zuvor werden wir zunächst wegen $C_F^0 = \bigoplus_{l|h_F} C_F^0(l)$ einen Algorithmus beschreiben, der für einen Primteiler l der Klassenzahl die Struktur einer l-Sylow-Gruppe $C_F^0(l)$ von C_F^0 berechnet. Anschließend können wir darüber einen Algorithmus zur Berechnung der Klassengruppe entwickeln. Sei $C_F^0(l) \cong \bigoplus_{i=1}^{z} (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ Wir nehmen zur Beschreibung des Algorithmus die folgende Unterteilung vor:

1. Für $i \in \{1, \ldots z\}$: Konstruktion von Erzeugendensystemen E_i der $\mathbb{F}_l$-Vektorräume

$$U_i := \{l^{(i-1)}[D] \mid [D] \in C_F^0(l) \text{ und } \mathrm{Ord}([D]) = l^i\} \cup \{[0]\} \subseteq C_F^0[l]$$

aus einem Erzeugendensystem S von C_F^0 und anschließend die Berechnung einer Basis davon.

2. Berechnung von $C_F^0(l)$ aus einem Erzeugendensystem S von C_F^0.
3. Berechnung von C_F^0 aus einem Erzeugendensystem S von C_F^0 durch das Zusammenfügen vorheriger Ergebnisse.

3.6.1 Konstruktion von Erzeugendensystemen

Sei $C_F^0(l) \cong \bigoplus_{i=1}^{z}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ die l-Sylow-Gruppe von C_F^0 für einen Primteiler l der Klassenzahl und S ein Erzeugendensystem der Klassengruppe. Wir definieren

$$E_i(S) := \{l^{i-1}c[D] \mid [D] \in S \text{ und } \mathrm{Ord}(c[D]) = l^i\},$$

wobei $h_F = l^s c$ mit $\mathrm{ggT}(l, c) = 1$ gelte. Erzeugen für $i \in \{1, \ldots, z\}$ die Mengen $E_i(S)$ die $\mathbb{F}_l$-Vektorräume U_i und setzen wir $U_{z+1} := \{\}$, dann können wir Basen B_i der Form $\{l^{i-1}[D_{i_1}], \ldots, l^{i-1}[D_{i_{\delta_i}}]\}$ der Vektorräume $U_i \setminus U_{i+1}$ berechnen und somit eine Basis $B_l := \{l^{i-1}[D_{j_i}] \mid i \in \{1, \ldots, z\} \text{ und } j_i \in \{1, \ldots, \delta_i\}\}$ von $C_F^0[l]$. Folglich ist die Menge

$$\{[D_{j_i}] \mid i \in \{1, \ldots, z\} \text{ und } j_i \in \{1, \ldots, \delta_i\}\}$$

ein minimales Erzeugendensystem von $C_F^0(l)$. Im Allgemeinen erzeugen die Mengen $E_i(S)$ aber nicht die Vektorräume $U_i \setminus U_{i+1}$.
Im Folgenden beschreiben wir ein Verfahren, dass für $i \in \{1, \ldots, z\}$, beginnend mit $i = z$, Mengen S_i aus S erzeugt mit der Eigenschaft

$$\mathrm{Spann}_{\mathbb{F}_l}(E_i(S_i)) = U_i \setminus U_{i+1}.$$

Für die anschließenden Überlegungen benötigen wir die nachfolgenden Definitionen. Sei G eine abelsche Gruppe, die isomorph zu $\bigoplus_{i=1}^{z}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ ist. Für ein Erzeugendensystem S von G definieren wir die Mengen

(i) $E_i(S) := \{l^{(i-1)}s \mid s \in S \text{ und } \mathrm{Ord}(s) = l^i\}$ und

(ii) $U_i := \{l^{(i-1)}g \mid g \in G \text{ und } \mathrm{Ord}(g) = l^i\} \cup \{0\}$.

Sei $\{b_1, \ldots, b_n\} \subset G$, so dass die Elemente von $B := \{l^{e_1-1}b_1, \ldots, l^{e_n-1}b_n\}$ linear unabhängig in $G[l]$ sind. Gelte $i \leq e_j$ für $j \in \{1, \ldots, n\}$ und $l^{i-1}s \in \mathrm{Spann}(B)$, dann gibt es ganze Zahlen $0 \leq \lambda_1, \ldots, \lambda_n \leq l - 1$ mit

$$l^{i-1}s = \sum_{j=1}^{n} \lambda_j l^{e_j-1}b_j \iff s = \sum_{j=1}^{n} \lambda_j l^{e_j-i}b_j + R_B(s),$$

wobei die Ordnung von $R_B(s) \in G$ echt kleiner als l^i ist. Wir definieren

$$R_B(s) := s - \sum_{j=1}^{n} \lambda_j l^{e_j-i}b_j \tag{3.18}$$

und nennen es den Rest von s bezüglich B. Beschreibe $G_{a,b}$ den zu $\bigoplus_{i=a}^{b}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ korrespondierenden Teil von G, wobei für $a > b$ die Gruppe $G_{a,b}$ der trivialen Gruppe entspricht.

Definition und Satz 3.31. *Gelte die Notation von eben und sei S ein Erzeugendensystem der Gruppe G. Dann ist $E_z(S)$ ein Erzeugendensystem des $\mathbb{F}_l$-Vektorraumes U_z. Für eine Basis $B := \{l^{z-1}b_1, \ldots, l^{z-1}b_n\}$ von U_z definieren wir die Menge*

$$S_B(S) := \{s \in S \mid \mathrm{Ord}(s) < l^z\} \cup \{R_B(s) \mid s \in S \text{ und } \mathrm{Ord}(s) = l^z\}.$$

In diesem Fall erzeugt $\{lb_1, \ldots, lb_n\} \cup S_B(S)$ die Gruppe

$$G' := \langle lb_1\rangle \oplus \cdots \oplus \langle lb_n\rangle \oplus G_{1,z-1}.$$

Beweis. Sei $S' := \{s \in S \mid \mathrm{Ord}(s) = l^z\}$. Da z maximal ist, gilt $G_{z,z} = \langle S'\rangle$ und wegen $E_z(S') \subseteq E_z(S)$ folgt die erste Behauptung.
Sei $g \in G' \subseteq \langle S\rangle$ beliebig. Wir können g mit ganzen Koeffizienten schreiben als

$$g = \sum_i \lambda_i s_i + \sum_j \beta_j \tilde{s}_j$$

mit $s_i, \tilde{s}_j \in S$ und $\mathrm{Ord}(s_i) = l^z$ sowie $\mathrm{Ord}(\tilde{s}_j) < l^z$. Das heißt, es gilt $l^{z-1}s_i \in \mathrm{Spann}(B)$ für jedes i. Nach (3.18) können wir s_i schreiben als $s_i = \sum_k \gamma_k b_k + R_B(s_i)$ mit $\mathrm{Ord}(R_B(s_i)) < l^z$ und ganzen Koeffizienten γ_k. Folglich gibt es ganze Zahlen α_i, η_k und β_j mit

$$g = \sum_i \alpha_i b_i + \sum_k \eta_k R_B(s_k) + \sum_j \beta_j \tilde{s}_j.$$

Die Ordnung von g ist kleiner als l^z sowie $\mathrm{Ord}(\sum_k \eta_k R_B(s_k))$ und $\mathrm{Ord}(\sum_j \beta_j \tilde{s}_j)$. Also gilt $\mathrm{Ord}(\sum_i \alpha_i b_i) < l^z$. Weil $\langle b_1\rangle \oplus \cdots \oplus \langle b_n\rangle \cong (\mathbb{Z}/l^z\mathbb{Z})^n$ ist, muss l die Koeffizienten α_i teilen. Das heißt, wir haben $\alpha_i = \tilde{\alpha}_i l$ für alle i und

$$g = \sum_i \tilde{\alpha}_i l b_i + \sum_k \eta_k R_B(s_k) + \sum_j \beta_j \tilde{s}_j \in \langle \{lb_1, \ldots, lb_n\} \cup S_B(S)\rangle .$$

□

Korollar und Definition 3.32. *Seien S ein Erzeugendensystem der abelschen Gruppe G mit $G \cong \bigoplus_{i=1}^{z}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ und $B_j := \{l^{j-1}b_{j_1}, \ldots, l^{j-1}b_{j_{n_j}}\}$ Basen der Vektorräume U_j für $j \in \{i+1 \ldots, z\}$ gegeben. Wir definieren*

$$S_i := \begin{cases} S_{B_{i+1}}(S_{B_{i+2}}(\ldots(S_{B_z}(S))\ldots)) & \text{für } i < z \\ S & \text{für } i = z. \end{cases}$$

Dann beinhaltet die Menge $E_i(S_i)$ ein Erzeugendensystem von $U_i \setminus U_{i+1}$ beziehungsweise von U_z für $i = z$.

Beweis. Ist $i = z$, dann folgt die Behauptung unmittelbar aus Satz 3.31. Sei $i \in \{1, \dots z-1\}$ beliebig und $j \in \{i+1 \dots, z\}$. Wir betrachten die Basis B_j von U_j und definieren die Untergruppe

$$G_{j-1} := \langle lb_{j_1} \rangle \oplus \cdots \oplus \langle lb_{j_{n_j}} \rangle \oplus G_{1,j-1} \cong (\mathbb{Z}/l^{j-1}\mathbb{Z})^{n_j} \oplus \bigoplus_{k=1}^{j-1} (\mathbb{Z}/l^k\mathbb{Z})^{\delta_k}$$

von G. Mit Satz 3.31 folgt induktiv, dass die Mengen $\widetilde{S}_{j-1} := \{lb_{j_1}, \dots, lb_{j_{n_j}}\} \cup S_{j-1}$ Erzeugendensysteme der Gruppen G_{j-1} sind. Dann ist insbesondere $\widetilde{S}_i$ ein Erzeugendensystem von G_i. Weiter gilt nach Satz 3.31, dass $E_i(\widetilde{S}_i) = \{l^{i-1}s \mid s \in \widetilde{S}_i\}$ ein Erzeugendensystem von U_i beinhaltet. Da $B_{i+1} \subseteq E_i(\widetilde{S}_i)$ eine Basis von $U_{i+1} \subseteq U_i$ ist, beinhaltet $E_i(S_i)$ ein Erzeugendensystem von $U_i \setminus U_{i+1}$. □

Die Vorgehensweise zur Berechnung eines minimalen Erzeugendensystems der Gruppe G ist die folgende:

Sei S ein Erzeugendensystem der Gruppe $G \cong \bigoplus_{i=1}^{z} (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$. Zuerst berechnen wir eine Basis B_z aus dem Erzeugendensystem $E_z(S)$ von U_z und bilden $S_{z-1} = S_{B_z}(S)$. Nach Korollar 3.32 beinhaltet diese Menge eine Basis $\widetilde{B}_{z-1}$ von $U_{z-1} \setminus U_z$. Wir definieren $B_{z-1} := B_z \cup \widetilde{B}_{z-1}$ und bilden damit S_{z-2}. Anschließend bestimmen wir aus $E_{z-2}(S_{z-2})$ eine Basis $\widetilde{B}_{z-2}$ von $U_{z-2} \setminus U_{z-1}$ und setzen B_{z-2} gleich $B_{z-1} \cup \widetilde{B}_{z-2}$. Nach maximal z Schritten erhalten wir für $i \in \{1, \dots, z-1\}$ Basen $\widetilde{B}_i$ und B_z für $i = z$, so dass

$$B := B_z \cup \widetilde{B}_{z-1} \cup \cdots \cup \widetilde{B}_1 = \{l^{i-i} b_{j_i} \mid i \in \{1, \dots, z\} \text{ und } j_i \in \{1, \dots, \delta_i\}\}$$

eine Basis von $G[l]$ ist und $\{b_{j_i} \mid i \in \{1, \dots, z\}$ und $j_i \in \{1, \dots, \delta_i\}\}$ ein minimales Erzeugendensystem der Gruppe G.

Berechnung von S_i

Wir betrachten nun wieder für den Funktionenkörper $F/\mathbb{F}_q$ mit $q = p^k$ die Klassengruppe C_F^0 mit einem Erzeugendensystem S. Sei für den Primteiler l der Klassenzahl $C_F^0(l)$ die l-Sylow-Gruppe von C_F^0. Ist $C_F^0(l)$ isomorph zu $\bigoplus_{i=1}^{z} (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$, dann beinhalten die Mengen $E_i(S_i)$ für $i \in \{1, \dots, z\}$, wie eben definiert, Erzeugendensysteme der $\mathbb{F}_l$-Vektorräume $U_i \setminus U_{i+1}$, dabei gilt $U_{z+1} = \{\}$. Wie zuletzt beschrieben, können wir darüber ein minimales Erzeugendensystem von $C_F^0(l)$ ermitteln und damit die Struktur von C_F^0. Zur Berechnung von S_i mit $i \in \{1, \dots, z-1\}$ müssen wir $R_{B_{i+1}}([D])$ bestimmen für alle $[D] \in S_{i+1}$ der Ordnung l^{i+1}, wobei B_{i+1} eine Basis von U_{i+1} bezeichnet. Wir stellen nun eine Möglichkeit vor die Elemente $R_{B_{i+1}}([D])$ zu berechnen. Zunächst betrachten wir den Fall, dass $l \neq p$ ist und anschließend, dass $l = p$ gilt. Sei in beiden Fällen $B := \{l^i[D_1], \dots, l^i[D_n]\}$ eine Menge linear unabhängiger Elemente aus $C_F^0[l]$. Wir wollen $R_B([D])$ für $l^i[D] \in \text{Spann}(B)$ berechnen. Wir nehmen zunächst an, dass

der Einbettungsgrad $d(q,l)$ gleich 1 ist. Den Fall $d(q,l) > 1$ verschieben wir auf später. Es sei nochmals darauf hingewiesen, dass wir $\{0,\ldots,l-1\}$ als Vertretersystem des endlichen Körpers $\mathbb{F}_l$ auffassen und damit $\lambda \in \mathbb{F}_l$ auch als ganze Zahl interpretieren können.

Primteiler l ist ungleich p

Sei $E := \{[E_1],\ldots,[E_n]\}$ eine Menge von Repräsentanten linear unabhängiger Elemente aus C_F^0/lC_F^0, so dass die Tate-Lichtenbaum Paarung T_l eingeschränkt auf die Menge $\mathrm{Spann}(B) \times \mathrm{Spann}(E)$ nicht-ausgeartet ist. Nach (3.18) gilt

$$R_B([D]) = [D] - \sum_{j=1}^{n} \lambda_j [D_j]$$

mit $\lambda_j \in \mathbb{F}_l$. Also müssen wir um $R_B([D])$ zu bestimmen, lediglich die λ_j berechnen. Äquivalent dazu können wir die letzte Gleichung schreiben als

$$\sum_{j=1}^{n} \lambda_j l^i [D_j] = l^i [D]. \tag{3.19}$$

Für $\kappa \in \{1,\ldots,n\}$ erhalten wir

$$\sum_{j=1}^{n} \lambda_j T_l(l^i[D_j],[E_\kappa]) = T_l(l^i[D],[E_\kappa])$$

und bekommen somit ein lineares Gleichungssystem für die Koeffizienten λ_j.

Primteiler l ist gleich p

Indem wir, wie in Abschnitt 2.2 beschrieben, die Isomorphie der $\mathbb{F}_p$-Vektorräume $C_F^0[p]$ und $\Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$ ausnutzen, berechnen wir für $l^i[D]$ und für die Elemente aus B die Koordinatenvektoren $v_D, v_{D_1}, \ldots, v_{D_n}$ in $\mathbb{F}_p^m$ mit $m := \dim_{\mathbb{F}_p}(C_F^0[p])$. Die Gleichung (3.19) entspricht nun

$$v_D = \sum_{j=1}^{n} \lambda_j v_{D_j}.$$

In diesem Fall lösen wir also auch ein lineares Gleichungssystem um $R_B([D])$ zu berechnen.

Bemerkung 3.33. *In unserem Fall ist $B = B_{i+1}$ eine Basis von U_{i+1} der Form $\{l^{\gamma-1}[D_{j_\gamma}] \mid \gamma \in \{i+1,\ldots,z\}$ und $j_\gamma \in \{1,\ldots,\delta_i\}\}$. Indem wir $\widetilde{D}_{j_\gamma} := l^{\gamma-(i+1)} D_{j_\gamma}$ setzen, erhalten wir $B_{i+1} = \{l^i[\widetilde{D}_{j_\gamma}] \mid \gamma \in \{i+1,\ldots,z\}$ und $j_\gamma \in \{1,\ldots,\delta_i\}\}$. In diesem Sinn werden wir im Folgenden $R_{B_{i+1}}([D])$ für $l^i[D]$ aus U_{i+1} berechnen.*

Algorithmus

Sei $F/\mathbb{F}_q$ ein Funktionenkörper vom Geschlecht g und $q = p^k$ sowie S ein Erzeugendensystem von C_F^0. Gelte $C_F^0(l) \cong \bigoplus_{i=1}^{z}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$ für einen Primteiler l der Klassenzahl. Für $i \in \{1,\ldots,z\}$ berechnet der folgende Algorithmus die Divisoren $D_1,\ldots,D_{\delta_i}$ vom Grad 0, so dass $\langle[D_1]\rangle \oplus \cdots \oplus \langle[D_{\delta_i}]\rangle = C_F^0(l)_{i,i}$ ist. Für einen Divisor D vom Grad 0 mit $\mathrm{Ord}([D]) = l^m$ definieren wir $e([D]) := m$. Sei der Einbettungsgrad $d(q,l)$ immer noch 1. Wir schreiben wieder, statt der Klasse $[D] + lC_F^0 \in C_F^0/lC_F^0$, der Einfachheit halber den Repräsentanten $[D]$ hin, wir meinen aber eigentlich dessen Klassen.

Algorithmus 11 : Basisberechnung von U_i

Eingabe:

- $B_{i+1} := \{l^{e([D_{\alpha_i}])-1}[D_{\alpha_i}] \mid i \in \{1,\ldots,n\}\}$ mit $\langle[D_{\alpha_1}]\rangle \oplus \cdots \oplus \langle[D_{\alpha_n}]\rangle = C_F^0(l)_{i+1,z}$ für $i = z$ sei $B_{z+1} := \{\}$.
- Für $l \neq p$ sei $\widetilde{E}_{i+1} := \{[E_1],\ldots,[E_n]\}$ eine Menge linear unabhängiger Elemente in C_F^0/lC_F^0, so dass T_l eingeschränkt auf $\mathrm{Spann}(B_{i+1}) \times \mathrm{Spann}(\widetilde{E}_{i+1})$ nicht-ausgeartet ist (für $i = z$ sei $\widetilde{E}_{i+1} := \{\}$) und für $l = p$ sei $\widetilde{E}_{i+1}$ eine Basis des $\mathbb{F}_p$-Vektorraumes $\Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$.
- S_i Erzeugendensystem von $l^{e([D_{\alpha_1}])-i}[D_{\alpha_1}] \oplus \cdots \oplus l^{e([D_{\alpha_1}])-i}[D_{\alpha_1}] \oplus C_F^0(l)_{1,i}$.
- S Erzeugendensystem von $C_F^0(l)$ sowie n_i eine obere Schranke von δ_i.

Ausgabe:

- $B_i := B_{i+1} \cup \{l^{i-1}[D_1],\ldots,l^{i-1}[D_{\delta_i}]\}$ mit $\langle[D_1]\rangle \oplus \cdots \oplus \langle[D_{\delta_i}]\rangle = C_F^0(l)_{i,i}$,
- $\widetilde{E}_i$ Menge von linear unabhängigen Elementen aus C_F^0/lC_F^0, so dass T_l eingeschränkt auf $\mathrm{Spann}(B_i) \times \mathrm{Spann}(\widetilde{E}_i)$ nicht-ausgeartet ist, für $l = p$ gilt $\widetilde{E}_i = \widetilde{E}_{i+1}$.
- S_{i-1} Erzeugendensystem von $\langle\gamma_1[D_{\alpha_1}]\rangle \oplus \cdots \oplus \langle\gamma_n[D_{\alpha_n}]\rangle \oplus \langle l[D_1]\rangle \oplus \cdots \oplus \langle l[D_{\delta_i}]\rangle \oplus C_F^0(l)_{1,i-1}$ mit $\gamma_j := l^{e([D_{\alpha_j}])-(i+1)}$ für $j \in \{1,\ldots,n\}$.

1: $E_i(S_i) \leftarrow \{l^{i-1}[D] \mid [D] \in S_i$ und $\mathrm{Ord}([D]) = l^i\}$ {Erzeugendensystem von $U_i \setminus U_{i+1}$}
2: $B_i, \widetilde{E}_i \leftarrow$ Algorithmus $1/2(B_{i+1}, E_i(S_i), \widetilde{E}_{i+1}, S, n_i)$ {B_i ist Basis von U_i}
3: $S_{i-1} \leftarrow \{[D] \in S_i \mid \mathrm{Ord}([D]) < l^i\} \cup \{R_{B_i}([D]) \mid [D] \in S_i$ und $\mathrm{Ord}([D]) = l^i\}$
4: **if** $l = p$ **then**
5: $\quad \widetilde{E}_i \leftarrow \widetilde{E}_{i+1}$
6: **end if**
7: **return** $B_i, \widetilde{E}_i, S_{i-1}$

Bemerkung 3.34. *Mit Algorithmus 1/2 ist gemeint, dass für $l \neq p$ Algorithmus 1 mit der entsprechenden Eingabe gestartet wird und andernfalls Algorithmus 2. Für $l = p$ spielt das Erzeugendensystem S keine Rolle.*

Korrektheit und Komplexität:
Da S ein Erzeugendensystem von C_F^0 ist, erzeugen die Elemente $[D] + lC_F^0$ mit $[D] \in S$ die Gruppe C_F^0/lC_F^0. Weil Algorithmus 1 mit den Repräsentanten der Elemente von C_F^0/lC_F^0 rechnet, ergibt die obige Eingabe für Algorithmus 1 Sinn. Die Korrektheit des

Algorithmus 11 folgt aus den Überlegungen zuvor. Wir analysieren die Komplexität. Im Beweis von Satz 3.6 wurde gezeigt, dass der Aufwand zur Berechnung der Ordnung eines Elements $[D]$ aus $C_F^0(l)$ polynomiell in g und $\log(q)$ ist. Folglich ist die Anzahl der Rechenoperationen um die Menge $E_i(S_i)$ zu bestimmen polynomiell in g und $\log(q)$ sowie linear in $\#S_i$. Wegen $d(q,l)=1$ ist nach Satz 2.7 die Komplexität zur Berechnung von B_i und $\widetilde{E}_i$ im Fall $l \neq p$ durch Algorithmus 1 gleich $g^{\mathcal{O}(1)}\log(q)^{\mathcal{O}(1)}\mathcal{O}(\#S\#E_i(S_i))$. Zur Bestimmung von S_{i-1} berechnen wir maximal $\#S_i$-mal $R_{B_i}([D])$. Für $l \neq p$ entspricht der Aufwand dafür wegen $\dim_{\mathbb{F}_l}(C_F^0[l]) \leq 2g$ dem höchstens $((2g)^2+2g)$-maligen Auswerten der Tate-Lichtenbaum Paarung T_l und dem Lösen eines quadratischen linearen Gleichungssystems über $\mathbb{F}_l$ mit höchstens $2g$ Gleichungen. Im Beweis von Satz 2.7 haben wir gezeigt, dass der Aufwand zum Auswerten von T_l polynomiell in g und $\log(q)$ ist, wenn wir annehmen, dass die Repräsentanten der Divisorenklassen den Voraussetzungen aus Teilabschnitt 1.5.2 genügen. Das lineare Gleichungssytem kann in $\mathcal{O}(g^3)$ Rechenoperationen gelöst werden. Letztendlich ist die Komplexität zur Berechnung von S_{i-1} gleich $g^{\mathcal{O}(1)}\log(q)^{\mathcal{O}(1)}\mathcal{O}(\#S_i)$. Der Fall $l=p$ ist in jedem Teilschritt von Algorithmus 11 weniger aufwendig, somit spielt er bei dieser Analyse keine Rolle. Nehmen wir an, dass $\#S_i = \mathcal{O}(\#S)$ gilt, dann ist die Anzahl der von Algorithmus 11 insgesamt benötigten Rechenoperationen, wegen $\#E_i(S_i) = \mathcal{O}(\#S_i)$, gleich

$$\mathcal{O}(\#S^2)\log(q)^{\mathcal{O}(1)}g^{\mathcal{O}(1)}. \tag{3.20}$$

Bemerkung 3.35. *Wir werden später zeigen, dass in unserem Fall $\#S_{i+1} = \mathcal{O}(\#S)$ ist. Dabei bezeichnet dann S ein Erzeugendensystem von C_F^0. Ist $\#S$ groß, dann ist es auch der Aufwand von Algorithmus 11, insbesondere im Fall $l \neq p$. Alternativ können wir, wie in Teilabschnitt 3.4.2 beschrieben, abhängig von $\varepsilon \in (0,1)$ ein Erzeugendensystem E von C_F^0/lC_F^0 bestimmen und Algorithmus 1 mit E statt S starten. Damit erhalten wir dann eine probabilistische Variante von Algorithmus 11, dessen Komplexität wegen Satz 3.19 gleich $\mathcal{O}(\log(1/(1-\sqrt{\varepsilon}))\#S)\log(q)^{\mathcal{O}(1)}g^{\mathcal{O}(1)}$ ist.*

3.6.2 Berechnung von $C_F^0(l)$

Sei F ein Funktionenkörper vom Geschlecht g über dem endlichen Körper $\mathbb{F}_q$ mit $q=p^k$. Wir können nun als Resultat des letzten Teilabschnittes einen Algorithmus beschreiben, der für einen Primteiler l der Klassenzahl h_F und ein Erzeugendensystem S der Klassengruppe C_F^0 die Struktur der l-Sylow-Gruppe $C_F^0(l)$ berechnet.

Algorithmus 12 : l-Sylow-Gruppen Berechnung

Eingabe: $S := \{[D_1],\ldots,[D_f]\}$ mit $\langle S\rangle = C_F^0$, g Geschlecht von F, l Primteiler von h_F mit $d(q,l)=1$ und $h_F = l^s c$ sowie $\mathrm{ggT}(l,c)=1$.

Ausgabe: $B := \{l^{i-1}[D_{j_i}] \mid i \in \{1,\ldots,z\}$ und $j_i \in \{1,\ldots,\delta_i\}\}$ Basis von $C_F^0[l]$ und $\bigoplus_{i=1}^{z}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i} \cong C_F^0(l)$.

$G \leftarrow \langle 0 \rangle$ triviale Gruppe
$S_z \leftarrow \{c[D_i] \mid i \in \{1, \ldots, f\}\}$
$B_{z+1} \leftarrow \{\}$, $E_{z+1} \leftarrow \{\}$
$z \leftarrow \max\{\log_l(\mathrm{Ord}([D])) \mid [D] \in S_z\}\{l^z$ ist maximale Ordnung in $C_F^0(l)\}$
$\eta_z \leftarrow \min\{2g, s_z - z + 1\}\{$Obere Schranke von $\dim_{\mathbb{F}_l}(C_F^0[l])\}$
if $l = p$ **then**
  $\widetilde{E}_{z+1} \leftarrow$ Basis von $\Omega^0(F/\mathbb{F}_q) \cap \Omega^{\log}(F/\mathbb{F}_q)$
  $\eta_z \leftarrow \#\widetilde{E}_{z+1}$
end if
if $s_z = z \cdot \eta_z$ **then**
  $B_z \leftarrow$Algorithmus 11$(B_{z+1}, \widetilde{E}_{z+1}, S_z, S_z, \eta_z)\{B$ Basis von $C_F^0[l]\}$
  return $B_z, (\mathbb{Z}/l^z\mathbb{Z})^{\eta_z}$
end if
while true do
  HELP$\leftarrow$ **false**
  $s_i \leftarrow s - \sum_{j=i+1}^{z} j\delta_j\{l^{s_i} = \#C_F^0(l)_{1,i}\}$
  if $i > s_i$ **then** {maximale Ordnung in $\bigoplus_{j=1}^{i}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}$ ist größer als $l^{s_i}\}$
    $\delta_j \leftarrow 0$ für $j \in \{s_i + 1, \ldots, i\}$, $i \leftarrow s_i$
    Aktualisiere Index von η_i, s_i, B_{i+1}, $\widetilde{E}_{i+1}$, S_i durch neues i
  end if
  if $i = 1$ **then**
    $B_1 \leftarrow$Algorithmus 11$(B_2, \widetilde{E}_2, S_1, S_z, \#B_2 + s_1)\{B$ Basis von $C_F^0[l]\}$
    return $B_1, G \oplus (\mathbb{Z}/l\mathbb{Z})^{s_1}$
  else if $s_i = i \cdot \eta_i$ **then**
    $B_i \leftarrow$Algorithmus 11$(B_{i+1}, \widetilde{E}_{i+1}, S_i, S_z, \#B_{i+1} + \lfloor \frac{s_i}{i} \rfloor)\{B$ Basis von $C_F^0[l]\}$
    return $B_i, G \oplus (\mathbb{Z}/l^i\mathbb{Z})^{\eta_i}$
  end if
  if $s_i - i < \eta_i - 1$ **then** $\{\delta_i = 0$ oder $\eta_i > \dim_{\mathbb{F}_l}(C_F^0[l] \setminus U_{i+1})\}$
    TEMP$\leftarrow \eta_i$, HELP$\leftarrow$ **true**{Angenommen $\delta_i \neq 0\}$
    $\eta_i \leftarrow s_i - i + 1$
  end if
  $B_i, \widetilde{E}_i, S_{i-1} \leftarrow$Algorithmus 11$(B_{i+1}, \widetilde{E}_{i+1}, S_i, S_z, \#B_{i+1} + \lfloor \frac{s_i}{i} \rfloor)\{B$ Basis von $U_i\}$
  if $\#B_i = \#B_{i+1}$ **then**
    if HELP= **true then**
      $\eta_i \leftarrow$TEMP{Annahme $\delta_i \neq 0$ war falsch, Aktualisierung von $\eta_i\}$
    end if
  else
    $G \leftarrow G \oplus (\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}\{G = \bigoplus_{j=i}^{z}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j}\}$
  end if
  $i \leftarrow i - 1$ und $\eta_i \leftarrow \eta_{i+1} - \delta_{i+1}$
end while

Satz 3.36. *Sei $F/\mathbb{F}_q$ ein Funktionenkörper vom Geschlecht g, dessen Klassenzahl bekannt und l ein Primteiler davon. Erzeugt S die Klassengruppe C_F^0, dann berechnet Algorithmus 12 die Struktur von $C_F^0(l)$ und benötigt dafür*

$$\mathcal{O}(\#S^2)\log(q)^{\mathcal{O}(1)}g^{\mathcal{O}(1)}$$

Rechenoperationen. Der Algorithmus speichert $\mathcal{O}(\#S)$ Elemente aus C_F^0.

Beweis. Wir schreiben $h_F = l^s c$ mit $\mathrm{ggT}(l,c) = 1$. Sei $C_F^0(l) \cong \bigoplus_{i=1}^{z}(\mathbb{Z}/l^i\mathbb{Z})^{\delta_i}$. Da S ein Erzeugendensystem von C_F^0 ist, erzeugt $cS := \{c[D] \mid [D] \in S\}$ die Gruppe $C_F^0(l)$. Folglich muss es ein Element der Ordnung l^z in cS geben. Da Algorithmus 12 im Grunde aus Algorithmus 7 entsteht, indem wir Algorithmus 5/6 durch Algorithmus 11 austauschen und sowohl Algorithmus 7 als auch Algorithmus 11 korrekt terminieren, zeigt man die Korrektheit von Algorithmus 12 analog zum Beweis von Satz 3.25.
Für $i \in \{1,\ldots,z\}$ bezeichnet S_i ein Erzeugendensystem der Untergruppe G_{i-1} von $C_F^0(l)$, die isomorph zu $\bigoplus_{j=1}^{i-1}(\mathbb{Z}/l^j\mathbb{Z})^{\delta_j} \oplus (\mathbb{Z}/l^{i-1}\mathbb{Z})^{n_i}$ ist mit $n_i := \sum_{j=i}^{z}\delta_j$. Definieren wir $S_z := cS$, dann berechnet Algorithmus 11 beginnend mit $i = z$ mittels S_i das Erzeugendensystem S_{i-1} mit $\#S_{i-1} \leq \#S_i$. Folglich gilt $\#S_i = \mathcal{O}(\#S)$ für jedes i. Höchstens s-mal wird Algorithmus 11 gestartet. Wegen $s \leq \log_l(h_F) = \mathcal{O}(g\log(q))$ und (3.20) ist die Komplexität von Algorithmus 12 gleich

$$\mathcal{O}(\#S^2)\log(q)^{\mathcal{O}(1)}g^{\mathcal{O}(1)}.$$

Da wir für jedes i die Menge S_i und S abspeichern müssen, ist die Anzahl der zu speichernden Divisorenklassen vom Grad 0 gleich $\mathcal{O}(\#S)$. □

Bemerkung 3.37. *Benutzen wir alternativ die probabilistische Variante von Algorithmus 11 aus Bemerkung 3.35, dann erhalten wir eine probabilistische Version von Algorithmus 12. Für $\varepsilon \in (0,1)$ beschreibt*

$$\mathcal{O}(\log(1/(1-\sqrt{\varepsilon}))\#S)\log(q)^{\mathcal{O}(1)}g^{\mathcal{O}(1)}$$

dessen Komplexität und $\mathcal{O}(\#S)$ dessen Speicheraufwand, gemessen in Elementen aus C_F^0.

Der Fall $d(q,l) > 1$

Wenn der Einbettungsgrad von $F/\mathbb{F}_q$ bezüglich eines Primteilers l der Klassenzahl ungleich 1 ist, müssen wir, wie in Teilabschnitt 2.1.3 erklärt, alle Rechnungen, welche die Tate-Lichtenbaum Paarung verwenden, statt in C_F^0 in $C_{F'}^0$ durchführen, wobei F' eine Konstantenkörpererweiterung von F vom Grad $d(q,l)$ bezeichnet. Dann ist der Aufwand jedes Schrittes von Algorithmus 12 polynomiell in $\log(q^{d(q,l)})$, also polynomiell in $d(q,l)$. Nach Satz 2.10 gilt $d(q,l) \leq l-1$. Der Primteiler l ist kleiner gleich h_F und wegen $h_F \leq (\sqrt{q}+1)^{2g}$ haben wir $l = \mathcal{O}(q^g)$. Das bedeutet die Komplexität von Algorithmus

12 ist für $d(q,l) > 1$ polynomiell in q^g. Des Weiteren benötigen wir ein Erzeugendensystem S' der Klassengruppe $C^0_{F'}$ der Konstantenkörpererweiterung F' von F vom Grad $d(q,l)$ für jeden Primteiler l der Klassenzahl mit $d(q,l) > 1$.
Zusammenfassend läßt sich im Allgemeinen mit den eben beschriebenen Methoden im Fall $d(q,l) > 1$ die Klassengruppe nicht mehr effizient berechnen. Ist l klein und kann man Erzeugendensysteme von Klassengruppe von Funktionenkörpern mit großem Konstantenkörper effizient berechnen, so ist es möglich $C^0_F(l)$ für $d(q,l) > 1$ mit einer ähnlichen Komplexität wie in Satz 3.36 zu bestimmen. Wir gehen wegen der eben erwähnten Gründe nicht weiter auf diesen Fall ein.

3.6.3 Berechnung der Klassengruppe

Wir fügen nun die Ergebnisse der letzten Teilabschnitte zusammen. Sei $F/\mathbb{F}_q$ ein Funktionenkörper und C^0_F dessen Klassengruppe. Der folgende Algorithmus berechnet aus einem Erzeugendensystem S der Klassengruppe ihre Struktur. Damit meinen wir die Gruppeninvarianten $c_1, \ldots, c_n \in \mathbb{Z}^{>0}$ und Divisoren $D_1, \ldots, D_n$ vom Grad 0 mit

$$C^0_F = \langle [D_1] \rangle \oplus \cdots \oplus \langle [D_n] \rangle \cong (\mathbb{Z}/c_1\mathbb{Z}) \oplus \cdots \oplus (\mathbb{Z}/c_n\mathbb{Z}).$$

Algorithmus 13 : Klassengruppenberechnung

Eingabe: S Erzeugendensystem von C^0_F, h_F die Klassenzahl von F, so dass $d(q,l) = 1$ für alle Primteiler l von h_F gilt und g das Geschlecht von F.
Ausgabe: Die Struktur von C^0_F.

1: **for** $l \in \mathbb{P}$ mit $l | h_F$ **do**
2: Berechne $h_F = l^s c$ mit $\mathrm{ggT}(l,c) = 1$
3: $\widetilde{B}_l, G_l \leftarrow$ Algorithmus 12$(C^0_F, g, l^s c, S)$
4: **end for**
5: **return** $\bigoplus_{l|h_F} G_l, \sum_{l|h_F} \widetilde{B}_l$ {gemäß der in (3.8) definierten Addition}

Bemerkung 3.38. *Auch hier ist die Ausgabe von Algorithmus 12 eigentlich eine Basis $\widetilde{B}_l$ von $C^0_F[l]$ der Form $\{l^{i-1}[D_{j_i}] \mid i \in \{1,\ldots,z\}$ und $j_i \in \{1,\ldots,\delta_i\}\}$. Wir betrachten stattdessen $\{[D_{j_i}] \mid i \in \{1,\ldots,z\}$ und $j_i \in \{1,\ldots,\delta_i\}\}$ das minimale Erzeugendensystem von $C^0_F(l)$.*

Satz 3.39. *Sei F ein Funktionenkörper vom Geschlecht g über $\mathbb{F}_q$ und dessen Klassenzahl bekannt. Gelte $d(q,l) = 1$ für jeden Primteiler l der Klassenzahl, dann berechnet Algorithmus 13 die Struktur der Klassengruppe von F. Dessen Komplexität entspricht*

$$\exp(\widetilde{O}((g\log(q))^{\frac{1}{3}}) + \mathcal{O}(\#S^2)\log(q)^{\mathcal{O}(1)} g^{\mathcal{O}(1)}$$

und die Anzahl der zu speichernden Elemente aus C^0_F ist gleich $\mathcal{O}(\#S)$.

Beweis. Da die Korrektheit von Algorithmus 12 schon nachgewiesen wurde, beweist man die Korrektheit von Algorithmus 13 analog zum Beweis von Satz 3.28. Ebenso analog folgt wegen Satz 3.36 die Komplexitätsbeschreibung des Satzes und die Aussage über den Speicheraufwand. □

Bemerkung 3.40. *Verwenden wir in Algorithmus 13, statt Algorithmus 12, dessen probabilistische Version aus Bemerkung 3.37, dann erhalten wir einen Algorithmus, der in Abhängigkeit einer Wahrscheinlichkeit $\varepsilon \in (0,1)$, die Struktur von C_F^0 aus einem Erzeugendensystem S berechnet. Dessen Komplexität entspricht*

$$\exp(\widetilde{O}((g\log(q))^{\frac{1}{3}}) + \mathcal{O}(\log(1/(1-\sqrt{\varepsilon}))\#S)\log(q)^{\mathcal{O}(1)}g^{\mathcal{O}(1)}$$

und die Anzahl der zu speichernden Elemente aus C_F^0 ist gleich $\mathcal{O}(\#S)$.

3.7 Erzeugendensysteme der Klassengruppe

Um die Klassengruppe eines Funktionenkörpers F vom Geschlecht g über dem endlichen Körper $\mathbb{F}_q$ zu berechnen, benötigen wir für die im letzten Abschnitt beschriebenen Algorithmen sowie für [Tes98, Algorithm 2.1] und [BS05, Algorithm 2] ein Erzeugendensystem S. Wir stellen kurz zwei Möglichkeiten vor, um ein Erzeugendensystem S der Klassengruppe zu bestimmen:

Variante 1

Nach [Hes99, S.45-46] besitzt die Klassengruppe C_F^0 von F ein Erzeugendensystem $\{[D_1],\dots[D_m]\}$ mit $m \leq 2g$, so dass die Grade aller Stellen, die in den Trägern der Repräsentanten $D_1,\dots,D_m$ vorkommen, durch

$$d := \max\{\lceil 2\log_q(4g-2)\rceil, \lceil 2\log_q(2g)\rceil + 1\}$$

nach oben beschränkt sind. Durch die Wahl eines geeigneten Divisors A vom Grad 1 von F erzeugt die Menge

$$S := \{[\mathrm{P} - \deg(\mathrm{P})A] \mid \mathrm{P} \in \mathbb{P}_F \text{ und } \deg(\mathrm{P}) \leq d\}$$

die Klassengruppe C_F^0 von F. Berechnen wir mittels [Hesed, Algorithm 33] den Divisor A, dann ist $h(\mathrm{P} - \deg(\mathrm{P})A) = \mathcal{O}((\log_q(g))^2)$, $\mathrm{supp}(\mathrm{P} - \deg(\mathrm{P})A) = \mathcal{O}(1)$ und das Maximum der Grade, der in S vorkommenden Stellen, gleich $\mathcal{O}(\log_q(g))$. Das rechtfertigt die Annahme aus Teilabschnitt 1.5.2.
Nach [Sti93, S.170] ist die Anzahl der Stellen vom Grad r eines Funktionenkörpers vom Geschlecht g über einem endlichen Körper mit q Elementen kleiner gleich

$$2gq^{\frac{r}{2}} + q^r + 1.$$

Für $\sqrt{q} \geq g$ gilt $2gq^{\frac{r}{2}} + q^r + 1 = \mathcal{O}(q^r)$. Folglich ist die Anzahl der Stellen von F vom Grad kleiner gleich d durch

$$\left(\sum_{i=1}^{d} 2gq^{\frac{i}{2}} + q^i + 1 \right) = \mathcal{O}(q^d)$$

nach oben beschränkt, also gilt $\#S = \mathcal{O}(q^{\log_q(g)})$.

Bemerkung 3.41. *Sei der Einbettungsgrad $d(q,l)$ gleich 1 für jeden Primteiler der Klassenzahl. Geben wir S, wie eben definiert, als Erzeugendensystem der Klassengruppe C_F^0 vor, dann berechnet Algorithmus 13 daraus die Struktur von C_F^0. Wegen Satz 3.39 ist die Anzahl der dafür benötigten Rechenoperationen gleich*

$$\exp(\widetilde{O}((g\log(q))^{\frac{1}{3}}) + \mathcal{O}(q^{2\log_q(g)})\log(q)^{\mathcal{O}(1)}g^{\mathcal{O}(1)}.$$

Dabei müssen $\mathcal{O}(q^{2\log_q(g)})$ Elemente aus C_F^0 gespeichert werden.

Variante 2

Als Nächstes sei $\varepsilon \in (0,1)$ und $[D_1], \ldots, [D_k]$ zufällig gewählte Divisorenklassen vom Grad 0 sowie $S_k := \{[D_1], \ldots, [D_k]\}$. Nach [ER65, Theorem 2] erzeugt S_k für $k \geq B(h_F, \varepsilon)$ (wie auf S.33 definiert) die abelsche Gruppe C_F^0 mit Wahrscheinlichkeit größer gleich ε. Wählen wir k als $\lceil B(h_F,\varepsilon)\rceil$, dann ist $\#S_k = \mathcal{O}(B(h_F,\varepsilon)) = \mathcal{O}(\log(h_F) + \log(1/(1-\varepsilon)))$. Wegen Satz 1.18 ist die Klassenzahl gleich $\mathcal{O}(q^g)$ und damit

$$\#S_k = \mathcal{O}(g\log(q) + \log(1/(1-\varepsilon)).$$

Bemerkung 3.42. *Sei der Einbettungsgrad $d(q,l)$ gleich 1 für jeden Primteiler der Klassenzahl und $\varepsilon \in (0,1)$. Geben wir S, wie eben definiert, als Erzeugendensystem der Klassengruppe C_F^0 von F vor, dann berechnet Algorithmus 13 daraus die Struktur von C_F^0 mit einer Wahrscheinlichkeit größer gleich ε korrekt. Wegen Satz 3.39 ist die Anzahl der dafür benötigten Rechenoperationen gleich*

$$\exp(\widetilde{O}((g\log(q))^{\frac{1}{3}}) + \mathcal{O}((\log(1/(1-\varepsilon)))^2)\log(q)^{\mathcal{O}(1)}g^{\mathcal{O}(1)}$$

und die Anzahl der zu speichernden Elemente aus C_F^0 gleich $\mathcal{O}(g\log(q)+\log(1/(1-\varepsilon)))$.

3.8 Vergleich der Algorithmen

In diesem Abschnitt diskutieren wir kurz die Komplexität von Algorithmus 8 und Algorithmus 13 im Kontext der Konkurrenzfähigkeit zu den bestehenden Algorithmen aus Abschnitt 3.1. Sei dazu $F/\mathbb{F}_q$ ein Funktionenkörper vom Geschlecht g und C_F^0 dessen Klassengruppe sowie S ein Erzeugendensystem von C_F^0. Da wir die Komplexität

der Algorithmen im Worst-Case miteinander vergleichen wollen, also den maximalen Aufwand abhängig von einem fest gewählten Funktionenkörper $F/\mathbb{F}_q$ und dessen Geschlecht, spielen die Average-Case Komplexitäten von Algorithmus [Tes98, Algorithm 2.1] und [Hes99, Algorithmus 5.5], die auf bestimmten Annahmen beruhen, in diesem Diskurs keine Rolle.
Die neuentwickelten Algorithmen müssen zur Berechnung der Klassengruppe zuerst die Klassenzahl von F faktorisieren. Die erwartete Komplexität dafür ist, wenn wir das Zahlenkörpersieb verwenden, wegen $h_F = \mathcal{O}(q^g)$ gleich $\exp(\widetilde{O}((g\log(q))^{\frac{1}{3}})$. Im Allgemeinen ist die Komplexität der schnellsten Faktorisierungsalgorithmen zur Faktorisierung von h_F in der eben beschriebenen Größenordnung. Ist die Klassenzahl groß, dann ist dies auch der Aufwand zum Faktorisieren. Das heißt, wenn die Ordnung der Klassengruppe zu groß wird, scheitern die Algorithmen bereits an der Faktorisierung dieser. Deswegen bieten im Allgemeinen die neuentwickelten Algorithmen keine Verbesserung der schon bekannten asymptotischen Schranken für die Komplexität zur Berechnung der Klassengruppe eines globalen Funktionenkörpers.
Wir nehmen nun an, dass die Zerlegung der Klassenzahl in Primfaktoren bekannt ist oder dass h_F schnell faktorisiert werden kann. Nach Satz 3.28 ist dann der Aufwand von Algorithmus 8 polynomiell im größten Primteiler l der Klassenzahl mit $d(q,l) > 1$. Hat l die Größenordnung von $h_F = \mathcal{O}(q^g)$, dann ist die Komplexität von Algorithmus 8 polynomiell in q^g und es ergibt keinen Sinn dessen Aufwand mit dem der anderen Algorithmen zu vergleichen. Wir betrachten also ausschließlich den Fall, dass $d(q,l)$ für jeden Primteiler l der Klassenzahl gleich eins ist.
Selbst unter diesen Annahmen ist nach Bemerkung 3.29(iii) der Aufwand von Algorithmus 8 exponentiell in $g+\mathcal{O}(1)$. Asymptotisch ist damit dessen Komplexität noch größer als die von [BS05, Algorithm 2], welche nach (3.2) „nur“ exponentiell in g ist. Im Gegensatz dazu, ist der Aufwand von Algorithmus 13 in diesem Fall polynomiell in g und $\log(q)$ sowie linear in $\#S$. Das heißt, in dieser speziellen Situation bietet Algorithmus 13 eine erhebliche Verbesserung der asymptotischen Schranken für die Komplexität der Berechnung der Klassengruppe eines globalen Funktionenkörpers.
Im anschließenden Kapitel werden wir die praktischen Laufzeiten einiger der eben genannten Algorithmen vergleichen. Wir werden sehen, dass Algorithmus 8 und besonders dessen vereinfachte Version Algorithmus 10 sowie Algorithmus 13 in vielen Fällen im Gegensatz zu den anderen Algorithmen noch Ergebnisse liefern. Erklärend dafür ist deren geringer Speicheraufwand (gemessen in Divisorenklassen vom Grad 0). Dieser ist für Algorithmus 8 und 10 mit $\varepsilon \in (0,1)$ logarithmisch in $1/(1-\sqrt{\varepsilon})$ und linear in g beziehungsweise für Algorithmus 13 linear in $\#S$. Besonders die schon bekannten und etablierten Verfahren aus [Hes99] und [BS05] müssen $\mathcal{O}(gq^{\log_q(g)})$ beziehungsweise $\mathcal{O}(\#S(\sqrt{q})^g)$ Elemente der Klassengruppe speichern. Dieser erhebliche Unterschied wird sich im praktischen Vergleich widerspiegeln.
An dieser Stelle wollen wir nochmals erwähnen, dass die verhältnismäßig schlechte Komplexität von Algorithmus 8 eine Folge der ungenauen Abschätzungen der Wahrscheinlichkeiten aus Teilabschnitt 3.4.2 ist. Des Weiteren haben wir zwischen dem Aufwand zum Auswerten der Tate-Lichtenbaum Paarung und dem zum Rechnen in C_F^0 nicht un-

terschieden. Besonders Algorithmus 13 verwendet oft die Tate-Lichtenbaum Paarung. Dagegen benutzt [Tes98, Algorithm 2.1] ausschließlich die Gruppenoperationen in C_F^0.

3.9 Berechnung des Isomorphismus und des Diskreten Logarithmus

Berechnung des Isomorphismus

In den bisherigen Abschnitten dieses Kapitels haben wir Algorithmen beschrieben, die Divisoren $D_1, \ldots, D_n$ vom Grad Null eines Funktionenkörper $F/\mathbb{F}_q$ vom Geschlecht g und positive ganze Zahlen $c_1, \ldots, c_n$ berechnen, so dass

$$C_F^0 = \langle [D_1] \rangle \oplus \cdots \oplus \langle [D_n] \rangle \overset{\psi}{\cong} \mathbb{Z}/c_1\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/c_n\mathbb{Z} := G$$

gilt. Im Folgenden beschreiben wir eine Möglichkeit mit der Tate-Lichtenbaum Paarung den Isomorphismus

$$\psi : \langle [D_1] \rangle \oplus \cdots \oplus \langle [D_n] \rangle \longrightarrow G$$

zu berechnen. Seien $g_1, \ldots, g_n$ die Erzeuger der zyklischen Komponenten von G. Da der Isomorphismus ψ durch $\psi([D_i]) = g_i$ mit $i \in \{1, \ldots, n\}$ eindeutig bestimmt ist, müssen wir, um für $[D] \in C_F^0$ das Element $\psi([D])$ zu berechnen, $\lambda_1, \ldots, \lambda_n \in \mathbb{Z}$ finden, so dass

$$[D] = \sum_{i=1}^{n} \lambda_i [D_i] \tag{3.21}$$

gilt. Seien für einen Primteiler l von h_F die ganzen Zahlen $s(l)$ und $c(l)$, so dass $h_F = l^{s(l)}c(l)$ mit $\mathrm{ggT}(l, c(l)) = 1$ gilt. Wir betrachten $c(l)[D]$ und $c(l)[D_i]$ für $i \in \{1, \ldots, n\}$. Für alle diese Elemente, die ungleich Null sind, definieren wir $[E_l] := (\mathrm{Ord}([D])/l)[D]$, $[E_{l_i}] := (\mathrm{Ord}([D])/l)[D_i]$ und damit die Menge $B_l := \{[E_{l_1}], \ldots [E_{l_n}]\}$. Wir beschreiben nun wie man $\lambda_{l_1}, \ldots, \lambda_{l_n} \in \mathbb{Z}$ bestimmt, damit

$$[E_l] = \sum_{i=1}^{n} \lambda_{l_i} [E_{l_i}] \tag{3.22}$$

gilt. Haben wir für jeden Primteiler l der Klassenzahl die Werte $\lambda_{l_1}, \ldots, \lambda_{l_n} \in \mathbb{Z}$ berechnet, so dass (3.22) gilt, dann lösen $\lambda_i := \sum_{l|h_F} \lambda_{l_i}$ mit $i \in \{1, \ldots, n\}$ die Gleichung (3.21), da C_F^0 die direkte Summe ihrer l-Sylow-Gruppen ist.

Wie nehmen an, dass $c(l)[D_i] \neq [0]$ für alle i ist. Gilt $c(l)[D_i] = [0]$ für ein i, dann setzen wir das entsprechende λ_{l_i} auf Null. Für $c(l)[D] = [0]$ setzen wir alle $\lambda_{l_i} = 0$. Aufgrund

der letzten Festlegung ist die Menge B_l eine Basis von $C_F^0[l]$. Wir berechnen, wie in (3.18) definiert, die ganzen Zahlen $\beta_{1_1}, \ldots, \beta_{1_n}$, so dass

$$R_1 := R_{B_l}([E_l]) = [E_l] - \sum_{i=1}^{n} \beta_{1_i}[E_{l_i}]$$

gilt. Dabei ist $\mathrm{Ord}(R_1) < \mathrm{Ord}([E_l])$. Als nächstes berechnen wir $\beta_{2_1}, \ldots, \beta_{2_n} \in \mathbb{Z}$ mit

$$R_2 := R_{B_l}(R_1) = R_1 - \sum_{i=1}^{n} \beta_{2_i}[E_{l_i}],$$

wobei $\mathrm{Ord}(R_2) < \mathrm{Ord}(R_1)$ ist. Spätestens nach $s(l) - 1$ Schritten erhalten wir $R_{s(l)-1}$. Wegen $1 \leq \mathrm{Ord}(R_{s(l)-1}) < \frac{\mathrm{Ord}([E_l])}{l^{s(l)-1}} \leq l$ ist $\mathrm{Ord}(R_{s(l)-1}) = 1$ und damit $R_{s(l)-1}$ gleich der Null in C_F^0. Folglich gilt

$$[E_l] = \sum_{j=1}^{s(l)-1} \sum_{i=1}^{n} \beta_{ij}[E_{l_i}] = \sum_{i=1}^{n} \sum_{j=1}^{s(l)-1} \beta_{ij}[E_{l_i}].$$

Definieren wir $\lambda_{l_i} := \sum_{j=1}^{s(l)-1} \beta_{ij}$ für $i \in \{1, \ldots, n\}$, so erfüllen diese λ_{l_i} die Gleichung (3.22).
Im Teilabschnitt 3.6.1 wurde für $l \neq p$ und $d(q, l) = 1$ beziehungsweise $l = p$ beschrieben wie $\beta_{i_1}, \ldots, \beta_{i_n}$ für $i \in \{1, \ldots, s(l) - 1\}$ mittels der Tate-Lichtenbaum Paarung berechnet werden können. Der Aufwand dafür ist für jedes i wegen $n \leq 2g$ polynomiell in g und $\log(q)$. Da sowohl die Anzahl der Primteiler von h_F als auch $s(l)$ durch $\log((\sqrt{q}+1)^{2g})$ abgeschätzt werden kann, ist der Aufwand zur Berechnung von $\psi([D])$ ebenfalls polynomiell in g und $\log(q)$.
Ist für einen Primteiler l der Klassenzahl der Einbettungsgrad $d(q, l) > 1$, dann müssen wir, wie in Teilabschnitt 2.1.3 beschrieben, $[D], [D_1], \ldots, [D_n]$ in $C_{F'}^0$ einbetten, wobei F' eine Konstantenkörpererweiterung von F vom Grad $d(q, l)$ bezeichnet. Dann können wir, genauso wie eben beschrieben, $\lambda_1, \ldots, \lambda_n \in \mathbb{Z}$ berechnen, so dass $[D'] = \sum_{i=1}^{n} \lambda_i [D_i']$ gilt. Hierbei bezeichne $[D_i']$ und $[D']$ das Bild von $[D_i]$ beziehungsweise $[D]$ unter einer Einbettung von C_F^0 nach $C_{F'}^0$. Berechnen wir auf diese Weise $\psi([D])$, dann ist die Komplexität polynomiell in g und $\zeta \log(q)$, wobei ζ das Maximum aller Primteiler von h_F entspricht.

Diskreter Logarithmus

Seien $[D_1]$ und $[D_2]$ Elemente der Klassengruppe. Wir wollen den diskreten Logarithmus von $[D_2]$ zur Basis $[D_1]$ berechnen. Damit meinen wir die ganze Zahl d zwischen 0 und $\mathrm{Ord}([D_1])$, so dass

$$d[D_1] = [D_2]$$

gilt. Ist die Struktur der Klassengruppe C_F^0 bekannt, dann können wir mittels des eben beschriebenen Verfahrens zur Berechnung des Isomorphismus ψ, die letzte Gleichung, statt in C_F^0, in $\mathbb{Z}/c_1\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/c_n\mathbb{Z}$ betrachten. Aufgrund der einfachen Arithmetik können wir das Problem $d\psi([D_1]) = \psi([D_2])$ dort effizient lösen.

Kapitel 4

Beispiele

In diesem Kapitel betrachten wir einige Beispiele zur Berechnung von Klassengruppen von globalen hyperelliptischen Funktionenkörpern. Die Algorithmen 8, 10 und 13 wurden mittels des Computeralgebrasystems MAGMA [BCP97] implementiert. Sowohl [Hes99, Algorithmus 5.5] als auch [Tes98, Algorithm 2.1] stehen in MAGMA zur Verfügung. Dabei ist der letzte Algorithmus in einer zu [Tes98] veränderten Version implementiert. Die Eingabe dieser Version ist die Klassengruppe C_F^0 sowie die Klassenzahl h_F eines globalen hyperelliptischen Funktionenkörpers F. Davon ausgehend wird ein Erzeugendensystem S in Abhängigkeit einer festen Wahrscheinlichkeit erzeugt. Daraus berechnet [Tes98, Algorithm 2.1] eine Untergruppe G von C_F^0. Gilt $\#G = h_F$, dann ist G selbst die Klassengruppe und es wird terminiert. Andernfalls wird eine Menge $S' \subseteq C_F^0$ wie S erzeugt mit der Hoffnung, dass $G' := \langle S \cup S' \rangle$ die Klassengruppe ist. Bei Mißerfolg wird der letzte Schritt solange wiederholt bis ein Erzeugendensystem der Klassengruppe gefunden und daraus ihre Struktur bestimmt wurde.

Für die Beispiele entspricht:

- F einem globalen Funktionenkörper,
- g dem Geschlecht von F,
- q der Anzahl an Elementen des Konstantenkörpers $\mathbb{F}_q$ von F,
- h_F der Klassenzahl von F,
- C_F^0 der Klassengruppe von F und
- S einem Erzeugendensystem der Klassengruppe.

Wir betrachten jeweils als Beispiel einen hyperelliptischen Funktionenkörper definiert durch die Gleichung $y^2 - f(x) = 0$ mit $f(x) \in \mathbb{F}_q[x]$ und vergrößern sukzessiv den Konstantenkörper. Dabei vergleichen wir die Laufzeiten der Algorithmen 8, 10, 13, [Tes98,

Algorithm 2.1] und [Hes99, Algorithmus 5.5]. Sowohl Algorithmus 8 als auch Algorithmus 10 werden abhängig von der Wahrscheinlichkeit $\varepsilon := 0.7$ die folgenden Beispiele berechnen. Alle Algorithmen, bis auf Algorithmus 10, geben auch ein minimales Erzeugendensystem der Klassengruppe aus. Die Berechnungen wurden auf einem Intel Core2Duo (64-Bit) System mit 2-GHz-Taktung durchgeführt. Falls einer der Algorithmen nach 3000 Sekunden kein Ergebnis ausgibt, deklarieren wir es mit „∞“ und brechen die Rechnung ab. Stoppt ein Algorithmus vorher ohne Ergebnisse wegen einer zu hohen Speicherauslastung, dann schreiben wir kurz „Sp. voll“. Ist der Einbettungsgrad $d(q, l)$ bezüglich eines Primteilers l der Klassenzahl größer eins, dann ist Algorithmus 13 nicht definiert. Diesen Fall kennzeichnen wir durch „$-$“.

Beispiel 1

Wie betrachten den elliptischen Funktionenkörper F mit der definierenden Gleichung $y^2 + 82x^3 + x = 0$ über dem Körper $\mathbb{F}_q$ mit q Elementen und $\varepsilon = 0.7$.

$q = 83^2$	$\#S = 31$
$C_F^0 \cong (\mathbb{Z}/84\mathbb{Z})^2$	
$h_F = 2^4 \cdot 3^2 \cdot 7^2$	
Algorithmus 8	0.7s
Algorithmus 10	0.04s
Algorithmus 13	0.8s
Algorithmus [Hes99, Algorithmus 5.5]	∞
Algorithmus [Tes98, Algorithm 2.1]	0.0s

$q = 83^4$	$\#S = 47$
$C_F^0 \cong (\mathbb{Z}/6888\mathbb{Z})^2$	
$h_F = 2^6 \cdot 3^2 \cdot 7^2 \cdot 41^2$	
Algorithmus 8	0.9s
Algorithmus 10	0.04s
Algorithmus 13	7.6s
Algorithmus [Hes99, Algorithmus 5.5]	∞
Algorithmus [Tes98, Algorithm 2.1]	0.01s

$q = 83^8$	$\#S = 76$
$C_F^0 \cong (\mathbb{Z}/47458320\mathbb{Z})^2$	
$h_F = 2^8 \cdot 3^2 \cdot 5^2 \cdot 7^2 \cdot 13^2 \cdot 41^2 \cdot 53^2$	
Algorithmus 8	2.6s
Algorithmus 10	0.19s
Algorithmus 13	21.1
Algorithmus [Hes99, Algorithmus 5.5]	SP. voll
Algorithmus [Tes98, Algorithm 2.1]	0.05s

$q = 83^{24}$ $\#S = 184$ $C_F^0 \cong (\mathbb{Z}/106890007738661124410160\mathbb{Z})^2$ $h_F = 2^8 \cdot 3^4 \cdot 5^2 \cdot 7^2 \cdot 13^2 \cdot 19^2 \cdot 41^2 \cdot 53^2 \cdot 367^2 \cdot 2269^2 \cdot 47451433^2$	
Algorithmus 8	46.1s
Algorithmus 10	2.2s
Algorithmus 13	285.3
Algorithmus [Hes99, Algorithmus 5.5]	SP. voll
Algorithmus [Tes98, Algorithm 2.1]	∞

$q = 83^{48}$ $\#S = 340$ $C_F^0 \cong (\mathbb{Z}/1142547375437103506328021703713734673016004592 0\mathbb{Z})^2$ $h_F = 2^{10} \cdot 3^4 \cdot 5^2 \cdot 7^2 \cdot 13^2 \cdot 17^2 \cdot 19^2 \cdot 41^2 \cdot 53^2 \cdot 73^2 \cdot 367^2 \cdot 2269^2 \cdot 19121^2 \cdot 4745143\backslash$ $3^2 \cdot 22522921846807 21^2$	
Algorithmus 8	2910s
Algorithmus 10	15.51s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	Sp. voll

$q = 83^{96}$ $\#S = 650$ $C_F^0 \cong (\mathbb{Z}/a\mathbb{Z})^2$ $a := 13054145051182135527078136482865615137841702893489810130131529337 78\backslash$ $18768621823991356828738240$ $h_F = 2^{12} \cdot 3^4 \cdot 5^2 \cdot 7^2 \cdot 13^2 \cdot 17^2 \cdot 19^2 \cdot 41^2 \cdot 53^2 \cdot 73^2 \cdot 367^2 \cdot 1297^2 \cdot 2269^2 \cdot 10289^2\backslash$ $\cdot 16001^2 \cdot 19121^2 \cdot 41281^2 \cdot 67153^2 \cdot 295153^2 \cdot 6840289^2 \cdot 47451433^2 \cdot 4780206650257^2$ $\cdot 2252922184680721^2$	
Algorithmus 8	∞
Algorithmus 10	252.6s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	Sp. voll

Beispiel 2

Wie betrachten den hyperelliptischen Funktionenkörper F mit der definierenden Gleichung $y^2+6x^5+5 = 0$ über dem Körper $\mathbb{F}_q$ mit q Elementen und $\varepsilon = 0.7$. Das Geschlecht von F ist $g = 2$.

$q = 7^4$	$\#S = 44$
$C_F^0 \cong (\mathbb{Z}/50\mathbb{Z})^4$	
$h_F = 2^4 \cdot 5^8$	
Algorithmus 8	1.2s
Algorithmus 10	0.03s
Algorithmus 13	4.2s
Algorithmus [Hes99, Algorithmus 5.5]	682s
Algorithmus [Tes98, Algorithm 2.1]	0.1s

$q = 7^7$	$\#S = 57$
$C_F^0 \cong (\mathbb{Z}/117650\mathbb{Z})^2$	
$h_F = 2^2 \cdot 5^4 \cdot 13^2 \cdot 181^2$	
Algorithmus 8	2.2s
Algorithmus 10	1.7s
Algorithmus 13	–[1]
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	0.02s

$q = 7^8$	$\#S = 69$
$C_F^0 \cong (\mathbb{Z}/2400\mathbb{Z})^4$	
$h_F = 2^{20} \cdot 3^4 \cdot 5^8$	
Algorithmus 8	6.2s
Algorithmus 10	0.12s
Algorithmus 13	20.8s
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	10.9s

$q = 7^{12}$	$\#S = 94$
$C_F^0 \cong (\mathbb{Z}/117650\mathbb{Z})^4$	
$h_F = 2^4 \cdot 5^8 \cdot 13^4 \cdot 181^4$	
Algorithmus 8	9.8s
Algorithmus 10	0.17s
Algorithmus 13	8.2
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	∞

[1] $d(7^7, 5) = 4$, $d(7^7, 13) = d(7^6, 181) = 12$

$q = 7^{16}$	$\#S = 118$
$C_F^0 \cong (\mathbb{Z}/5764800\mathbb{Z})^4$	
$h_F = 2^{24} \cdot 3^4 \cdot 5^8 \cdot 1201^4$	
Algorithmus 8	20.5s
Algorithmus 10	0.34s
Algorithmus 13	48.5s
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	Sp. voll

$q = 7^{24}$	$\#S = 165$
$C_F^0 \cong (\mathbb{Z}/13841287200\mathbb{Z})^4$	
$h_F = 2^{20} \cdot 3^8 \cdot 5^8 \cdot 13^4 \cdot 19^4 \cdot 43^4 \cdot 181^4$	
Algorithmus 8	46.0s
Algorithmus 10	0.99s
Algorithmus 13	34.9s
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	Sp. voll

$q = 7^{32}$	$\#S = 211$
$C_F^0 \cong (\mathbb{Z}/33232930569600\mathbb{Z})^4$	
$h_F = 2^{28} \cdot 3^4 \cdot 5^8 \cdot 17^4 \cdot 1201^4 \cdot 169553^4$	
Algorithmus 8	165.0s
Algorithmus 10	2.3s
Algorithmus 13	82.4s
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	Sp. voll

$q = 7^{64}$	$\#S = 12$
$C_F^0 \cong (\mathbb{Z}/1104427674243920646305299200\mathbb{Z})^4$	
$h_F = 2^{32} \cdot 3^4 \cdot 5^8 \cdot 17^4 \cdot 353^4 \cdot 1201^4 \cdot 1695534 \cdot 470721396174$	
Algorithmus 8	395.8
Algorithmus 10	21.6s
Algorithmus 13	435,3s
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	Sp. voll

$q = 7^{128}$ $\#S = 757$ $C_F^0 \cong (\mathbb{Z}/12197604876358357001385738625629718207556152941312384 00\mathbb{Z})^4$ $h_F = 2^{36} \cdot 3^4 \cdot 5^8 \cdot 17^4 \cdot 353^4 \cdot 1201^4 \cdot 169553^4 \cdot 7699649^4 \cdot 134818753^4 \cdot 4707213967^4 \setminus$ $\cdot 531968664833^4$	
Algorithmus 8	∞
Algorithmus 10	175s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	Sp. voll

Beispiel 3

Wie betrachten den hyperelliptischen Funktionenkörper F mit der definierenden Gleichung $y^2 + 6x^7 + 2x^4 + 6x + 3 = 0$ über dem Körper $\mathbb{F}_q$ mit q Elementen und $\varepsilon = 0.7$. Das Geschlecht von F ist $g = 3$.

$q = 7^2$ $\#S = 37$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/5278\mathbb{Z}$ $h_F = 2^6 \cdot 7 \cdot 13 \cdot 29$	
Algorithmus 8	2.0s
Algorithmus 10	0.6s
Algorithmus 13	5.8s
Algorithmus [Hes99, Algorithmus 5.5]	0.8s
Algorithmus [Tes98, Algorithm 2.1]	0.02s

$q = 7^{11}$ $\#S = 121$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/193274963793726823316680 7906\mathbb{Z}$ $h_F = 2^3 \cdot 7 \cdot 13 \cdot 463 \cdot 22936292667710206170541$	
Algorithmus 8	38.8s
Algorithmus 10	2.8s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	8.1s

$q = 7^{16}$ $\#S = 165$ $C_F^0 \cong \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/20176\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$ $a := 27758191360621930064018316972272$ $h_F = 2^{24} \cdot 7 \cdot 13^2 \cdot 29 \cdot 41 \cdot 97^2 \cdot 248177 \cdot 267217 \cdot 464897 \cdot 5361593$	
Algorithmus 8	168s [2]
Algorithmus 10	2.5s
Algorithmus 13	–
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	109s

[2] $d(7^{16}, 97) = 6$, d.h. es wird in $\mathbb{F}_{7^{16 \cdot 6}}$ gerechnet.

$q = 7^{24}$ $\#S = 234$
$C_F^0 \cong \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/104\mathbb{Z} \oplus \mathbb{Z}/104\mathbb{Z} \oplus \mathbb{Z}/104\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$
$a := 976739809955813440633640899535160444980210780009295 28$
$h_F = 2^{18} \cdot 7 \cdot 13^4 \cdot 29 \cdot 41 \cdot 97 \cdot 4297 \cdot 4327 \cdot 172411 \cdot 359101 \cdot 5361593 \cdot 18848042099\backslash$
3087170081

Algorithmus 8	133s
Algorithmus 10	3.9s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	7.7s

$q = 7^{25}$ $\#S = 7$
$C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$
$a := 602966258011612348430884178591122725745071764689941526993339142$
$h_F = 2^3 \cdot 7 \cdot 13 \cdot 251 \cdot 1021 \cdot 6282361 \cdot 205778155332434745742992942817417764853\backslash$
5904739951

Algorithmus 8	63.9s
Algorithmus 10	3.2s
Algorithmus 13	67s
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	0.4s

$q = 7^{32}$ $\#S = 303$
$C_F^0 \cong \mathbb{Z}/32\mathbb{Z} \oplus \mathbb{Z}/32\mathbb{Z} \oplus \mathbb{Z}/32\mathbb{Z} \oplus \mathbb{Z}/32\mathbb{Z} \oplus \mathbb{Z}/40352\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$
$a := 31838079525832570748161956966685048591515314694320707639383128477 58\backslash$
7936
$h_F = 2^{30} \cdot 7 \cdot 13^2 \cdot 29 \cdot 41 \cdot 97^2 \cdot 248177 \cdot 267217 \cdot 464897 \cdot 5361593 \cdot$
$573489805445293061869955176503631215169$

Algorithmus 8	594s [3]
Algorithmus 10	14.3s
Algorithmus 13	–
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	Sp. voll

[3] $d(7^{32}, 97) = d(7^{32}, 13) = 3$, d.h. es wird in $\mathbb{F}_{7^{32 \cdot 3}}$ gerechnet.

$q = 7^{48}$ $\#S = 440$ $C_F^0 \cong \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/208\mathbb{Z} \oplus \mathbb{Z}/208\mathbb{Z} \oplus \mathbb{Z}/20176\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$ $a := 221266699392787568585575388398658052734796023786211931508607122628547869305017917616812991592867225948117405168$ $h_F = 2^{24} \cdot 7 \cdot 13^4 \cdot 29 \cdot 4 \cdot 97^2 \cdot 4297 \cdot 4327 \cdot 172411 \cdot 248177 \cdot 267217 \cdot 359101 \cdot 46489 \cdot 7 \cdot 984241 \cdot 5361593 \cdot 28585969 \cdot 188480420993087170081 \cdot 130578129806260814233 \cdot 4018209$	
Algorithmus 8,	834s
Algorithmus 10	14.5s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	Sp. voll

Beispiel 4

Wie betrachten den hyperelliptischen Funktionenkörper F mit der definierenden Gleichung $y^2 + 12x^7 + 10x^6 + 9x^4 + 10x^3 + 12x + 9 = 0$ über dem Körper $\mathbb{F}_q$ mit q Elementen und $\varepsilon = 0.7$. Das Geschlecht von F ist $g = 3$.

$q = 13$ $\#S = 29$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/1298\mathbb{Z}$ $h_F = 2^2 \cdot 11 \cdot 59$	
Algorithmus 8	0.6s
Algorithmus 10	0.06s
Algorithmus 13	1.75s
Algorithmus [Hes99, Algorithmus 5.5]	0.23s
Algorithmus [Tes98, Algorithm 2.1]	0.01s

$q = 13^2$ $\#S = 43$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/593186\mathbb{Z}$ $h_F = 2^4 \cdot 11 \cdot 59 \cdot 457$	
Algorithmus 8	1.5s
Algorithmus 10	0.08s
Algorithmus 13	12.2s
Algorithmus [Hes99, Algorithmus 5.5]	2.98s
Algorithmus [Tes98, Algorithm 2.1]	0.01s

$q = 13^4$ $\#S = 69$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/180086524112\mathbb{Z}$ $h_F = 2^{11} \cdot 11 \cdot 59 \cdot 137 \cdot 277 \cdot 457$	
Algorithmus 8	6.8s
Algorithmus 10	3.8s
Algorithmus 13	14.3s
Algorithmus [Hes99, Algorithmus 5.5]	∞
Algorithmus [Tes98, Algorithm 2.1]	0.03s

$q = 13^8$ $\#S = 117$ $C_F^0 \cong \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/331243064697382050917 44\mathbb{Z}$ $h_F = 2^{19} \cdot 11 \cdot 59 \cdot 137 \cdot 277 \cdot 457 \cdot 7577 \cdot 12137753$	
Algorithmus 8	72s
Algorithmus 10	48s
Algorithmus 13	934s
Algorithmus [Hes99, Algorithmus 5.5]	∞
Algorithmus [Tes98, Algorithm 2.1]	9.4s

$q = 13^{12}$ $\#S = 163$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$ $a := 98798573910318443640581328269773235792$ $h_F = 2^{11} \cdot 11 \cdot 13 \cdot 31 \cdot 43 \cdot 59 \cdot 61 \cdot 137 \cdot 277 \cdot 457 \cdot 2887 \cdot 9433 \cdot 100999 \cdot 188691541$	
Algorithmus 8	111s
Algorithmus 10	41s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	0.4s

$q = 13^{24}$ $\#S = 300$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/7312\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$ $a := 213592120192273190190412069952656451866852255820551923744878596903709485 12$ $h_F = 2^{19} \cdot 11 \cdot 13 \cdot 31 \cdot 43 \cdot 59 \cdot 61 \cdot 137 \cdot 277 \cdot 457^2 \cdot 2017 \cdot 2887 \cdot 7577 \cdot 9433 \cdot 10099 \backslash$ $\cdot 12137753 \cdot 188691541 \cdot 582724226737385280409$	
Algorithmus 8	∞ [4]
Algorithmus 10	439s
Algorithmus 13	–
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	23.7s

[4] $d(13^{24}, 457) = 19$, d.h. es wird in $\mathbb{F}_{13^{24 \cdot 19}}$ gerechnet.

$q = 13^{31}$ $\#S = 379$ $C_F^0 \cong Z/5\mathbb{Z} \oplus \mathbb{Z}/10\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$ $a := 790245006416955643124442853284135066614106451703564205895496370900 41\backslash$ $873390893052638495359154051033961 0$ $h_F = 2^2 \cdot 5^3 \cdot 11 \cdot 59 \cdot 12176348326917652436432093270941988699755107114076490\backslash$ $075431377055476405761308636770184184769499388 9$	
Algorithmus 8	∞[5]
Algorithmus 10	31.8s
Algorithmus 13	–
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	3.7s

$q = 13^{36}$ $\#S = 435$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$ $a := 158005454505381857906588465057635310428998471711903671915009114836 17\backslash$ $290618587621297828523074715189981564643713769637328$ $h_F = 2^{11} \cdot 11 \cdot 13 \cdot 31 \cdot 43 \cdot 59 \cdot 61 \cdot 109 \cdot 137 \cdot 277 \cdot 457 \cdot 2887 \cdot 9433 \cdot 3672 \cdot 55009 \cdot 10\backslash$ $0999 \cdot 188691541 \cdot 2880471338623117 \cdot 119497341962142160309 \cdot 2110205995580370\backslash$ 123743007754274653	
Algorithmus 8	978s
Algorithmus 10	500s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	5.8s

Beispiel 5

Wie betrachten den hyperelliptischen Funktionenkörper F mit der definierenden Gleichung $y^2 + 4x^9 + 3x^3 + 4x^2 + 2 = 0$ über dem Körper $\mathbb{F}_q$ mit q Elementen und $\varepsilon = 0.7$. Das Geschlecht von F ist $g = 4$.

$q = 5^2$ $\#S = 39$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/6\mathbb{Z} \oplus \mathbb{Z}/20178\mathbb{Z}$ $h_F = 2^4 \cdot 3^3 \cdot 19 \cdot 59$	
Algorithmus 8	2.1s
Algorithmus 10	0.28s
Algorithmus 13	16.5s
Algorithmus [Hes99, Algorithmus 5.5]	0.7s
Algorithmus [Tes98, Algorithm 2.1]	0.02s

[5] $d(13^{31}, 5) = 4$, d.h. es wird in $\mathbb{F}_{13^{31 \cdot 4}}$ gerechnet.

$q = 5^4$	$\#S = 61$
$C_F^0 \cong \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/12\mathbb{Z} \oplus \mathbb{Z}/735649524\mathbb{Z}$	
$h_F = 2^8 \cdot 3^3 \cdot 19 \cdot 59 \cdot 18229$	
Algorithmus 8	5.3s
Algorithmus 10	4.3s
Algorithmus 13	45.8
Algorithmus [Hes99, Algorithmus 5.5]	30.7s
Algorithmus [Tes98, Algorithm 2.1]	0.02s

$q = 5^{12}$	$\#S = 94$
$C_F^0 \cong \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/360\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$	
$a := 240941198447806147243748604 0$	
$h_F = 2^{18} \cdot 3^5 \cdot 5^2 \cdot 19 \cdot 43 \cdot 59 \cdot 673 \cdot 18229 \cdot 30763 \cdot 122632801$	
Algorithmus 8	143s
Algorithmus 10	46.8s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	∞
Algorithmus [Tes98, Algorithm 2.1]	6.1s

$q = 5^{16}$	$\#S = 179$
$C_F^0 \cong \mathbb{Z}/336\mathbb{Z} \oplus \mathbb{Z}/336\mathbb{Z} \oplus \mathbb{Z}/336\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$	
$a := 142910547640160215181307569994346806 24$	
$h_F = 2^{16} \cdot 3^5 \cdot 7^4 \cdot 19 \cdot 59 \cdot 18229 \cdot 340481 \cdot 407713 \cdot 23490473 \cdot 212762993$	
Algorithmus 8	164s
Algorithmus 10	94s
Algorithmus 13	-[6]
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	2.3s

$q = 5^{29}$	$\#S = 303$
$C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$	
$a := 60185310762354400029294707349177441178720309304560200685689967942 91\backslash$	
29478393879622	
$h_F = 2^2 \cdot 3^2 \cdot 19 \cdot 72708167453 \cdot 102311898907 \cdot 22469589503035124242684507 01 \cdot \backslash$	
$10528340421415403882030538237 1$	
Algorithmus 8	394,5s
Algorithmus 10	163.6s
Algorithmus 13	371.2s
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	4.5s

[6] $d(5^{16}, 7) = 3$

$q = 5^{32}$ $\#S = 331$ $C_F^0 \cong \mathbb{Z}/672\mathbb{Z} \oplus \mathbb{Z}/672\mathbb{Z} \oplus \mathbb{Z}/672\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$ $a := 96839544017792478633558619531608695580396065851102161244617437307 4\backslash$ 352347631350368 $h_F = 2^{20} \cdot 3^5 \cdot 7^4 \cdot 19 \cdot 59 \cdot 18229 \cdot 26209 \cdot 340481 \cdot 407713 \cdot 486817 \cdot 23490473 \cdot 212\backslash$ $762993 \cdot 1500634433 \cdot 176956843462474221833900 9$	
Algorithmus 8	484,7s
Algorithmus 10	320,8s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	7.8s

Beispiel 6

Wie betrachten den hyperelliptischen Funktionenkörper F mit der definierenden Gleichung $y^2+16x^9+13x^3+4x^2+13 = 0$ über dem Körper $\mathbb{F}_q$ mit q Elementen und $\varepsilon = 0.7$. Das Geschlecht von F ist $g = 4$.

$q = 17$ $\#S = 36$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/42376\mathbb{Z}$ $h_F = 2^4 \cdot 5297$	
Algorithmus 8	0.6s
Algorithmus 10	0.2s
Algorithmus 13	0.8s
Algorithmus [Hes99, Algorithmus 5.5]	1.3s
Algorithmus [Tes98, Algorithm 2.1]	0.0s

$q = 17^2$ $\#S = 56$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/222940136\mathbb{Z}$ $h_F = 2^8 \cdot 5261 \cdot 5297$	
Algorithmus 8	3.8s
Algorithmus 10	2.7s
Algorithmus 13	11.1s
Algorithmus [Hes99, Algorithmus 5.5]	11.3s
Algorithmus [Tes98, Algorithm 2.1]	0.02s

$q = 17^4$	$\#S = 92$
$C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/32\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$	
$a := 5876303367996832$	
$h_F = 2^{18} \cdot 277 \cdot 5261 \cdot 5297 \cdot 23789$	

Algorithmus 8	44.2s
Algorithmus 10	19.4s
Algorithmus 13	31.7s
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	2.7s

$q = 17^8$	$\#S = 161$
$C_F^0 \cong \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/192\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$	
$a := 752780481769684817792235159681984$	
$h_F = 2^{26} \cdot 3^2 \cdot 277 \cdot 953 \cdot 5261 \cdot 5297 \cdot 23789 \cdot 1501681 \cdot 14919089$	

Algorithmus 8	478.4s
Algorithmus 10	132.1s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	Sp. voll

$q = 17^{16}$	$\#S = 295$
$C_F^0 \cong \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/384\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$	
$a := 139251483519157526659659545636963284308782355953500147203709083825\backslash$ 8928	
$h_F = 2^{34} \cdot 3^2 \cdot 277 \cdot 953 \cdot 5261 \cdot 5297 \cdot 23789 \cdot 318881 \cdot 377761 \cdot 1501681 \cdot 14919089 \cdot \backslash$ 49361633 $\cdot$ 197443297 $\cdot$ 7878142081	

Algorithmus 8	1680.9s
Algorithmus 10	662.75s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	Sp. voll

Beispiel 7

Wie betrachten den hyperelliptischen Funktionenkörper F mit der definierenden Gleichung $y^2 + 2x^{13} + 2 = 0$ über dem Körper $\mathbb{F}_q$ mit q Elementen und $\varepsilon = 0.7$. Das Geschlecht von F ist $g = 6$.

$q = 3$ $\#S = 26$ $C_F^0 \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/78\mathbb{Z}$ $h_F = 2^4 \cdot 3 \cdot 13$	
Algorithmus 8	1.4s
Algorithmus 10	0.2s
Algorithmus 13	6.4s
Algorithmus [Hes99, Algorithmus 5.5]	0.8s
Algorithmus [Tes98, Algorithm 2.1]	0.0s

$q = 3^2$ $\#S = 39$ $C_F^0 \cong \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/8268\mathbb{Z}$ $h_F = 2^8 \cdot 3 \cdot 13 \cdot 53$	
Algorithmus 8	3.4s
Algorithmus 10	3.4s
Algorithmus 13	23.6s
Algorithmus [Hes99, Algorithmus 5.5]	4.4s
Algorithmus [Tes98, Algorithm 2.1]	0.02s

$q = 3^4$ $\#S = 62$ $C_F^0 \cong \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z} \oplus \mathbb{Z}/552914232\mathbb{Z}$ $h_F = 2^{12} \cdot 3 \cdot 13 \cdot 29 \cdot 53 \cdot 1153$	
Algorithmus 8	5.2s
Algorithmus 10	10.1s
Algorithmus 13	51.5s
Algorithmus [Hes99, Algorithmus 5.5]	∞
Algorithmus [Tes98, Algorithm 2.1]	0.06s

$q = 3^8$ $\#S = 103$ $C_F^0 \cong \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/16\mathbb{Z} \oplus \mathbb{Z}/1947432065479877152\mathbb{Z}$ $h_F = 2^{16} \cdot 3 \cdot 13 \cdot 29 \cdot 53 \cdot 113 \cdot 1153 \cdot 155846161$	
Algorithmus 8	11.5s
Algorithmus 10	10.3s
Algorithmus 13	170.6
Algorithmus [Hes99, Algorithmus 5.5]	∞
Algorithmus [Tes98, Algorithm 2.1]	0.14s

$q = 3^{16}$	$\#S = 183$
$C_F^0 \cong \mathbb{Z}/32\mathbb{Z} \oplus \mathbb{Z}/32\mathbb{Z} \oplus \mathbb{Z}/32\mathbb{Z} \oplus \mathbb{Z}/a\mathbb{Z}$	
$a := 194173750035177922011833360867960175542496$	
$h_F = 2^{20} \cdot 3 \cdot 13 \cdot 29 \cdot 53 \cdot 113 \cdot 1153 \cdot 29291921 \cdot 155846161 \cdot 170196392961169$	

Algorithmus 8	157.5s
Algorithmus 10	150.9s
Algorithmus 13	∞
Algorithmus [Hes99, Algorithmus 5.5]	Sp. voll
Algorithmus [Tes98, Algorithm 2.1]	11.8s

Kapitel 5

Ausblick

In diesem Kapitel wollen wir eine Anwendung der in dieser Arbeit vorgestellten Methoden beschreiben sowie einen kurzen Ausblick geben, welcher Problemstellung der Autor sich in Zukunft widmen wird.

Wie in der Einleitung erwähnt wurde, ist es möglich die Berechnung von Klassengruppen von globalen Funktionenkörpern zu nutzen um Endomorphismenringe von Jacobischen Varietäten von glatten projektiven algebraischen Kurven über endlichen Körpern zu bestimmen. Eine Definition dazu findet sich in [CF06, S.78]. Wir gehen im Folgenden auf den Spezialfall der Jacobischen Varietät einer glatten hyperelliptischen Kurve über dem endlichen Körper $\mathbb{F}_q$ ein, dabei sei angemerkt, dass alle Überlegungen und Definitionen auch in einem allgemeineren Kontext gültig bleiben. Eine solche Kurve C ist eine projektive Varietät über $\mathbb{F}_q$ der Dimension 1, dessen affiner Teil durch die Gleichung $y^2 = f(x)$ beschrieben wird. Dabei ist $f \in \mathbb{F}_q[x]$ normiert, quadratfrei und $\deg(f)$ sowie $q = p^k$ koprim zu 2. Allgemeine Definitionen und Erläuterungen findet der Leser in [Har77] beziehungsweise [CF06]. Der Quotientenkörper $K(C)$ des Restklassenringes

$$\mathbb{F}_q[x][y]/(y^2 - f(x))\mathbb{F}_q[x][y]$$

ist ein hyperelliptischer Funktionenkörper. Das Geschlecht g des Funktionenkörper definiert das Geschlecht der Kurve. Nach [HS00, S.134-135] existiert zu C eine abelsche Varietät $\mathcal{J}(C)$, genannt Jacobische Varietät von C, dessen Dimension dem Geschlecht g der Kurve entspricht und mit der Eigenschaft $\mathcal{J}(C) \cong C^0_{K(C)}$. Dabei sei letzte Isomorphie aufgefaßt als eine zwischen abelschen Gruppen.

Ein Endomorphismus von $\mathcal{J}(C)$ ist ein Morphismus $\alpha : \mathcal{J}(C) \longrightarrow \mathcal{J}(C)$, der zusätzlich ein Gruppenendomorphismus ist. Der Begriff Morphismus zwischen abelschen Varietäten wird in [Har77, S.14] erklärt. Ein Beispiel eines Endomorphismus von $\mathcal{J}(C)$ ist der sogenannte Frobeniusendomorphismus ϕ. Er ist dadurch definiert, dass er die Koordinaten eines Punktes $P \in \mathcal{J}(C)$ in die p-te Potenz setzt. Hat das charakteristische Polynom des Endomorphismus ϕ nur einfache Nullstellen, dann ist die Menge aller Endomorphismen von $\mathcal{J}(C)$ nach [Tat66, S.140] bis auf Isomorphie eine Ordnung $\mathcal{O}$ in

einem Zahlenkörper $\mathbb{K}$, der folgende Eigenschaften erfüllt:

1. $\mathbb{K}$ ist eine total imaginäre quadratische Erweiterung einer total reellen Erweiterung von $\mathbb{Q}$ und
2. $[\mathbb{K} : \mathbb{Q}] = 2g$, dabei beschreibt g das Geschlecht der Kurve C.

Es existiert ein primitives Element π mit $\mathbb{K} = \mathbb{Q}(\pi)$, welches zum Frobeniusendomorphismus ϕ korrespondiert. Bezeichne $\mathcal{O}_{\mathbb{Q}(\pi)}$ die Maximalordnung von $\mathbb{Q}(\pi)$ und $\overline{\pi}$ die komplex Konjugierte von π. Der Einfachheit halber setzen wir voraus, dass p, die Charakteristik von $\mathbb{F}_q$, nicht $[\mathcal{O}_{\mathbb{Q}(\pi)} : \mathbb{Z}[\pi, \overline{\pi}]]$ teilt. Sei $\{\alpha_1, \ldots, \alpha_n\}$ ein Erzeugendensystem der $\mathbb{Z}$-Algebra $\mathcal{O}_{\mathbb{Q}(\pi)}$ und $n_i \in \mathbb{Z}^{>0}$ minimal, so dass $n_i\alpha_i \in \mathbb{Z}[\pi]$ gilt und beschreibe $n_i = \prod_j l_{ij}^{d_{ij}}$ die Primfaktorzerlegung von n_i. Für jedes (i, j) sei $k_{ij} \in \mathbb{Z}^{>0}$ durch $\pi^{k_{ij}} - 1 \in l_{ij}\mathcal{O}_{\mathbb{Q}(\pi)}$ bestimmt, dann erzeugt die Menge

$$\left\{ \frac{n_i}{l_{ij}^{d_{ij}}}\alpha_i \;\middle|\; l_{ij}^2 \mid [\mathcal{O}_{\mathbb{Q}(\pi)} : \mathbb{Z}[\pi]] \right\} \cup \left\{ \frac{\pi^{k_{ij}} - 1}{l_{ij}} \;\middle|\; l_{ij}^2 \nmid [\mathcal{O}_{\mathbb{Q}(\pi)} : \mathbb{Z}[\pi]] \text{ und } l_{i,j} \neq p \right\}$$

die Maximalordnung $\mathcal{O}_{\mathbb{Q}(\pi)}$ über $\mathbb{Z}[\pi, \bar{\pi}]$. Detailliertere Informationen dazu findet der Leser in [FL07].

Um die zum Endomorphismenring korrespondierende Ordnung $\mathcal{O}$ zu berechnen, muss lediglich getestet werden, welche Erzeuger der Maximalordnung zu Endomorphismen von $\mathcal{J}(C)$ korrespondieren. Sei $l := l_{ij}$ ungleich p. Nach [Lau04, S.12] korrespondiert das Element $(\pi^{k_{ij}} - 1)/l$ genau dann zu einem Endomorphismus von C, wenn $\dim_{\mathbb{F}_l}(\mathcal{J}(\mathbb{F}_{p^{k_{ij}}})[l]) = 2g$ ist, wobei $\mathcal{J}(\mathbb{F}_{p^{k_{ij}}})$ die Jacobische Varietät der Kurve C über dem Körper $\mathbb{F}_{p^{k_{ij}}}$ bezeichnet. Hierbei fassen wir die l-Torsionsgruppe der Jacobischen Varietät, wie in Abschnitt 1.1 beschrieben, als $\mathbb{F}_l$-Vektorraum auf.
Betrachten wir $\beta_i := n_i\alpha_i/l^d$ mit $d := d_{ij}$, dann gibt es wegen $l^d\beta_i \in \mathbb{Z}[\pi]$ für β_i die Darstellung

$$\beta_i = \frac{\sum_{k=0}^{2g} a_k\pi^k}{l^d}$$

mit Koeffizienten $a_k \in \mathbb{Z}$. Nach [Lau04, S.11] ist β_i genau dann in $\mathcal{O}$ enthalten, wenn der zu $T := \sum_{k=0}^{2g} a_k\pi^k$ korrespondierende Endomorphismus ϕ_T auf $\mathcal{J}(\mathbb{F}_{p^\kappa})[l^d]$ trivial operiert. Dabei sei κ so gwählt, dass $\mathrm{Rang}(\mathcal{J}(\mathbb{F}_{p^\kappa})[l^d]) = 2g$ gilt. Da π dem effizient berechenbaren Frobeniusendomorphismus der Kurve C entspricht, können wir ϕ_T ebenfalls effizient in $P \in \mathcal{J}(C)[l^d]$ auswerten.

Mittels der Isomorphie der abelschen Gruppen $\mathcal{J}(C)$ und $C^0_{K(C)}$ sind wir in der Lage mit den Algorithmen der vorliegenden Arbeit die eben beschriebenen Tests zu realisieren und somit den Endomorphismenring der Jacobischen Varietät $\mathcal{J}(C)$ zu bestimmen.

Eine der weiterführenden Aufgaben, mit denen sich der Autor befasst, wird die genaue Analyse des zuvor grob beschriebenen Verfahrens zur Berechnung von Endomorphismenringen sein. Es wird versucht werden Methoden zu formulieren, die Gleiches auch für beliebige abelsche Varietäten über endlichen Körpern leisten.

Hyperelliptische Kurven vom Geschlecht 1 heißen elliptische Kurven. Die Jacobische Varietät einer elliptischen Kurve E ist isomorph zur Kurve, also ist E selber eine abelsche Gruppe. Eine elliptische Kurve E über dem endlichen Körper mit Charakteristik p für die $E[p] \cong 0$ gilt, nennen wir supersingulär. Nach [Was03, S.303] ist der Endomorphismenring einer solchen supersingulären Kurve isomorph zu einer nicht-kommutativen Ordnung in einer Quaternionen Algebra. Je allgemeiner die zu betrachtenden abelschen Varietäten werden, desto abstrakter wird auch die Struktur derer Endomorphismenringe.

Ausgehend von den Methoden aus [Koh96] zur Berechnung von Endomorphismenringen von supersingulären elliptischen Kurven über endlichen Körpern, wird versucht werden für allgemeinere Situationen Ähnliches zu entwickeln.

Literaturverzeichnis

[BCP97] Bosma, W., J. Cannon und C. Playoust: *The Magma algebra system I: The user language.* J. Symbolic Comp., 24 3/4:235–265, 1997.

[Böt08] Böttle, M.: *Berechnung von Zetafunktionen algebraischer Kurven über endlichen Köroern.* Diplomarbeit, Technische Universität Berlin, 2008.

[BS05] Buchmann, J. und A. Schmidt: *Computing the structure of a Finite Abelian Group.* Mathematics Of Computation, 74:2017–2026, 2005.

[Buc03] Buchmann, J.: *Einführung in die Kryptographie 3. erweit. Aufl.* Springer-Verlag, Berlin-Heidelberg-New York, 2003.

[CF06] Cohen, H. und G. Frey: *Handbook of Elliptic and Hyperelliptic Curve Cryptography.* Chapman & Hall, London, 2006.

[Coh91] Cohn, P. M.: *Algebraic Numbers and Algebraic Functions.* Chapman & Hall, London, 1991.

[ER65] Erdös, P. und A. Renyi: *Probabilistic methods in group theory.* Journal d'Analyse Mathematique, 14:127–138, 1965.

[FL07] Freeman, D. und K. Lauter: *Computing endomorphism rings of Jacobians of genus 2 curves over finite fields.* 2007. http://arxiv.org/abs/math/0701305.

[Fra07] Fraatz, R.: *On the computation of Integral Closures of Cyclic Extensions of Function Fields.* LMS J. Comput. Math., Seiten 141–160, 2007.

[Har77] Hartshorne, R.: *Algebraic Geometry.* Springer-Verlag, Berlin-Heidelberg-New York, 1977.

[Hes99] Hess, F.: *Zur Divisorenklassengruppenberechnung in globalen Funktionenkörpern.* Dissertation, Technische Universität Berlin, 1999.

[Hes02] Hess, F.: *Computing Riemann-Roch spaces in algebraic function fields and related topics.* J. Symbolic Comp., 33:425–445, 2002.

[Hes04] Hess, F.: *A Note on the Tate Pairing of Curves over Finite Fields.* Archives of Mathematics, 82:28–32, 2004.

[Hesed] Hess, F.: *Computing relation in divisor class groups of algebraic curves over finite fields.* J. Symbolic Comp., submitted.

[HS00] Hindry, M. und J. H. Silverman: *Diophantine Geometry.* Springer-Verlag, Berlin-Heidelberg-New York, 2000.

[Koh96] Kohel, D.: *Endomorphism rings of elliptic curves over finite fields.* PhD Thesis, University of Carlifornia at Berkeley, 1996.

[Lan73] Lang, S.: *Elliptic Functions.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1973.

[Lau04] Lauter, K. Eisenträger & K.: *A CRT algorithm for constructing genus 2 curves over finite fields.* 2004. http://arxiv.org/abs/math/0405305v2.

[LL93] Lenstra, A. und H. Lenstra: *The Development of the Number Field Sieve.* Springer-Verlag, Berlin-Heidelberg-New York, 1993.

[Neu92] Neukirch, J.: *Algebraische Zahlentheorie.* Springer-Verlag, Berlin-Heidelberg-New York, 1992.

[SD06] Salvador, V. und G. Daniel: *Topics in the theory of algebraic function fields.* Birkhäuser Verlag, Berlin-Heidelberg-New York, 2006.

[Sti93] Stichtenoth, H.: *Algebraic Function Fields and Codes.* Springer-Verlag, Berlin-Heidelberg-New York, 1993.

[Tat66] Tate, J.: *Endomorphisms of abelian varieties over finite fields.* Inventiones Mathematicae, Seiten 134–144, 1966.

[Tes98] Teske, E.: *A space efficient algorithm for Group structure computation.* Mathematics Of Computation, 67:1637–1663, 1998.

[vzGG03] Gathen, J. von zur und J. Gerhard: *Modern Computer Algebra, Second Edition.* Cambridge University Press, Cambridge, 2003.

[Was03] Washington, L. C.: *Elliptic Curves.* Chapman & Hall, London, 2003.