Attributbasierte Verschlüsselung mittels Gittermethoden - Mathematische Grundlagen, Verfahren und Sicherheitsbeweise

Fakultät für Elektotechnik, Informatik und Mathematik
Universität Paderborn

Bachelorarbeit

vorgelegt bei
Prof. Dr. Johannes Blömer
und
Prof. Dr. Peter Bürgisser

Kathlén Kohn

Paderborn, 25. Februar 2013

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass ich die gesamte Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen als Hilfsmittel verwendet habe. Weiterhin sind alle Stellen, die aus den Quellen entnommen wurden, als solche gekennzeichnet. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Paderborn, den 25. Februar 2013

Kathlén Kohn

Inhaltsverzeichnis

1. Einleitung

Attributbasierte Verschlüsselung liefert eine flexible Lösungsmöglichkeit zur Realisierung komplexer Zugriffsstrukturen mit Kryptografie. Dabei können zum Beispiel die Zugriffsrechte eines Teilnehmers beim Erstellen seines geheimen Schlüssels verwendet werden, so dass er mit diesem Schlüssel genau solche verschlüsselten Nachrichten entschlüsseln kann, für welche er befugt ist. Dahingegen wird bei herkömmlichen asymmetrischen Verschlüsselungsverfahren eine Nachricht mit dem öffentlichen Schlüssel eines Teilnehmers verschlüsselt, so dass dieser Teilnehmer die Nachricht mit seinem geheimen Schlüssel wieder entschlüsseln kann. Nachrichten, die mit anderen öffentlichen Schlüsseln verschlüsselt wurden, kann er nicht entschlüsseln. Somit können mit solchen Verschlüsselungsverfahren nur eingeschränkte Zugriffsrechte realisiert werden. Soll eine Nachricht mehreren Teilnehmern zugänglich sein, so muss die Nachricht für jeden Teilnehmer einzeln mit seinem öffentlichen Schlüssel verschlüsselt werden oder die Teilnehmer verwenden ein gemeinsames Schlüsselpaar aus öffentlichem und geheimem Schlüssel. Dann existieren aber entweder viele Verschlüsselungen derselben Nachricht oder es muss im schlimmsten Fall für alle möglichen Teilmengen von Teilnehmern ein Schlüsselpaar erstellt werden. Das Konzept der Attributbasierten Verschlüsselung löst diese Probleme.

Zunächst unabhängig von Attributbasierter Verschlüsselung werden seit mehr als zehn Jahren kryptografische Primitiven, wie zum Beispiel Einwegfunktionen, digitale Signaturen oder Verschlüsselungsverfahren, untersucht, deren Sicherheit auf schweren Gitterproblemen basiert. Ein Gitter ist eine diskrete, additive Untergruppe des $\mathbb{R}^m$. Ein typisches Gitterproblem ist das Finden eines kürzesten Vektors in einem gegebenen Gitter, der nicht der Nullvektor ist. Seine Entscheidungsvariante ist unter randomisierten Reduktionen NP-schwer [Ajt98]. In den letzten Jahren wurden mehrere Verschlüsselungsverfahren entwickelt, bei denen das Brechen des Verfahrens mindestens so schwer wie das Lösen der Worst Case Instanz eines schweren Gitterproblems ist. Weiterhin wird vermutet, dass solche Gitterprobleme auch auf Quantencomputern schwer sind. Diese Verschlüsselungsverfahren genügen somit starken Sicherheitsanforderungen und haben

daher eine wichtige Rolle in der modernen Kryptografie eingenommen.

In dieser Arbeit wird ein Verschlüsselungsverfahren vorgestellt, welches einen Spezialfall Attributbasierter Verschlüsselung realisiert und mit Gittern arbeitet. Seine Sicherheit basiert auf einem Problem, das stark mit schweren Gitterproblemen zusammenhängt. Dieses Verfahren wurde zuerst in [ABV+12] beschrieben. Allerdings werden dort sowohl seine Korrektheit als auch seine Sicherheit nur unvollständig und fehlerhaft bewiesen. Beides wird in dieser Arbeit korrigiert. Außerdem wird das Verfahren mit einem anderen Verfahren verglichen, welches dieselben Zugriffsrechte realisieren kann, aber mit bilinearen Abbildungen anstelle von Gittern arbeitet.

Auf die Verfahren wird erst im zweiten Teil dieser Arbeit eingegangen. Die Grundlagen für das Verschlüsselungsverfahren mit Gittern werden im ersten Teil „Attributbasierte Verschlüsselung mittels Gittermethoden - Mathematische Grundlagen“ geschaffen. Dieser Teil ist die Bachelorarbeit im Fach Mathematik. Zu Beginn wird die Verwendung von Gittern in der Kryptografie in Kapitel 2 „Verschlüsselung mittels Gittermethoden“ motiviert. In Kapitel 3 „Gitter und ihre grundlegenden Eigenschaften“ werden Gitter formal definiert und wichtige mit Gittern zusammenhängende Begriffe eingeführt. Danach werden im Kapitel 4 „Komplexität von Gitterproblemen“ einige Probleme auf Gittern sowie deren Zusammenhang untereinander beschrieben. Dabei wird insbesondere das Problem vorgestellt, welches als Sicherheitsannahme für das Verschlüsselungsverfahren verwendet wird, sowie die NP-Vollständigkeit seiner Entscheidungsvariante gezeigt. Zudem wird eine diskrete Gaußsche Verteilung auf Gittern definiert, nach der Gittervektoren gesampelt werden sollen. Es wird erläutert, dass das Sampeln desto schwieriger ist, je kleiner der verwendete Gaußsche Parameter ist. In Kapitel 5 „Diskrete Gaußsche Verteilung auf Gittern“ wird die Verteilung, nach der gesampelt werden soll, genauer untersucht. Ab welchem hinreichend großen Gaußschen Parameter effizient gesampelt werden kann, wird in Kapitel 6 „Algorithmen zum Sampeln von Vektoren“ angegeben. Der Algorithmus, der dieses effiziente Sampeln realisiert, wird im zweiten Teil dieser Arbeit für das Verschlüsselungsverfahren benötigt. Der Beweis der Aussage ist sehr technisch und wird daher in dieser Arbeit nicht aufgeführt. Dafür wird ein simpler Algorithmus präsentiert, der bei deutlich größerem Parameter effizient Gittervektoren sampelt. Zum Schluss des ersten Teils werden weitere Algorithmen auf Gittern vorgestellt, die im Verschlüsselungsverfahren benutzt werden.

Zu Beginn des zweiten Teils „Attributbasierte Verschlüsselung mittels Gittermethoden - Verfahren und Sicherheitsbeweise“, der die Bachelorarbeit im Fach Informatik

darstellt, werden die Vorteile und das Prinzip der Attributbasierten Verschlüsselung in Kapitel 7 „Attributbasierte und Fuzzy Identitätsbasierte Verschlüsselung“ erklärt. Die beiden vorgestellten Verfahren sind Fuzzy Identitätsbasierte Verschlüsselungsverfahren. Dies ist ein Spezialfall von Attributbasierter Verschlüsselung, der ebenfalls in Kapitel 7 erläutert wird. Die Verschlüsselungsverfahren werden zusammen mit ihrem jeweiligen Korrektheitsbeweis in Kapitel 8 „Konkrete Fuzzy Identitätsbasierte Verschlüsselungsverfahren“ angegeben. In Kapitel 9 „Sicherheitsannahmen und -beweise“ wird zunächst das gemeinsame Sicherheitsmodell beider Verfahren erörtert. Danach werden die Sicherheitsannahmen beider Verfahren verglichen. Die Sicherheit des Verfahrens mit Gittern beruht auf dem in Kapitel 4 vorgestellten, mit Gittern zusammenhängenden Problem. In Kapitel 9 wird erklärt, warum diese Sicherheitsannahme besser begründbar ist als die des Verfahrens mit bilinearen Abbildungen. Am Ende dieses Kapitels wird die Sicherheit der beiden Verfahren unter ihren Sicherheitsannahmen gezeigt. Zum Abschluss der Arbeit wird in Kapitel 10 „Vergleich der Effizienz und Erweiterbarkeit“ erläutert, dass das Verfahren mit Gittern deutlich ineffizienter ist und dass es sich wesentlich schwerer zu einem Attributbasierten Verschlüsselungsverfahren erweitern lässt als das Verfahren mit bilinearen Abbildungen.

Diese Arbeit liefert demnach eine Einführung in Gitter und insbesondere in Gaußsche Verteilungen auf Gittern. Es wird gezeigt, dass das Verschlüsselungsverfahren mit Gittern große Sicherheit gewährleistet, aber dafür einige Abstriche in der Effizienz gegenüber dem Verfahren mit bilinearen Abbildungen machen muss.

1.1. Notation

In beiden Teilen der Arbeit wird folgende gemeinsame Notation verwendet.

- Die Menge der natürlichen Zahlen ohne Null wird mit $\mathbb{N}$ bezeichnet. $\mathbb{N}_0$ bezeichnet die Menge der natürlichen Zahlen mit Null.
- Für $p, k \in \mathbb{N}$ mit p Primzahl bezeichnet $\mathbb{F}_{p^k}$ den Körper mit p^k Elementen. Ist $k = 1$, so wird dafür auch $\mathbb{Z}_p$ geschrieben.
- Ist R ein Ring, so bezeichnet $R^{m \times n}$ die Menge der $m \times n$-Matrizen über R. Für $A \in R^{m \times n}$ wird die Schreibweise $A = (a_{i,j})_{\substack{1 \leq i \leq m, \\ 1 \leq j \leq n}}$ benutzt. $R^{m \times 1}$ wird kurz als R^m geschrieben.

- Ist R ein Ring und $A \in R^{m \times n}$, so bezeichnet A^T die transponierte Matrix von A.
- Ist $\mathbb{K}$ ein Körper und $A \in \mathbb{K}^{m \times n}$, so bezeichnet $\mathrm{rk}\,(A)$ den Rang von A.
- Ist R ein Ring mit Eins, so bezeichnet $I_m \in R^{m \times m}$ die Einheitsmatrix.
- Für $v := (v_1, \ldots, v_m)^T \in \mathbb{R}^m$ sei $\|v\| := \sqrt{\sum_{i=1}^{m} v_i^2}$.
- Für eine Teilmenge $M \subseteq \mathbb{R}^m$ und $\lambda \in \mathbb{R}$ sowie $a \in \mathbb{R}^m$ sei $\lambda M := \{\lambda x \mid x \in M\}$ und $M + a := \{x + a \mid x \in M\}$.
- Für eine Teilmenge $M \subseteq \mathbb{R}^m$ und einen Vektor $v \in \mathbb{R}^m$ definiere den Abstand von v zu M als $\mathrm{dist}\,(v, M) := \inf\{\|v - x\| \mid x \in M\}$.
- Ist $M \subseteq \mathbb{R}^m$ eine abzählbare Menge sowie $f : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$, dann ist $f\,(M) := \sum_{x \in M} f\,(x)$.
- Ist $M := \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^m$, so ist $\|M\| := \max\{\|v_1\|, \ldots, \|v_n\|\}$. Ist $A \in \mathbb{R}^{m \times n}$ mit den Spaltenvektoren $v_1, \ldots, v_n$, so ist analog $\|A\| := \max\{\|v_1\|, \ldots, \|v_n\|\}$.
- Für $x \in \mathbb{R}$ definiere $\lfloor x \rfloor := \max\{z \in \mathbb{Z} \mid z \leq x\}$, $\lceil x \rceil := \min\{z \in \mathbb{Z} \mid z \geq x\}$ sowie

$$\lfloor x \rceil := \begin{cases} \lfloor x \rfloor & \text{, falls } x - \lfloor x \rfloor < \frac{1}{2} \\ \lceil x \rceil & \text{, falls } x - \lfloor x \rfloor \geq \frac{1}{2} \end{cases} .$$

  Für einen Vektor $v := (v_1, \ldots, v_m)^T \in \mathbb{R}^m$ sei $\lfloor v \rfloor := (\lfloor v_1 \rfloor, \ldots, \lfloor v_m \rfloor)^T$. $\lceil v \rceil$ und $\lfloor v \rceil$ seien analog definiert.
- Für $a \in \mathbb{R}$ und $p \in \mathbb{R}_{>0}$ definiere $a \mod p := a - \left\lfloor \frac{a}{p} \right\rfloor p \in [0, p)$. Für einen Vektor $b := (b_1, \ldots, b_m)^T \in \mathbb{R}^m$ sei $b \mod p := (b_1 \mod p, \ldots, b_m \mod p)^T$.
- Sind $q \in \mathbb{N}$ eine Primzahl, $a \in \mathbb{Z}_q$, $b_1 \in \mathbb{Z}$, $b_2 \in \mathbb{Z} \setminus q\mathbb{Z}$ und $b := \frac{b_1}{b_2} \in \mathbb{Q}$, so werden $a + b = b + a$ und $ab = ba$ als Elemente in $\mathbb{Z}_q$ aufgefasst, das heißt

$$\begin{aligned} b + a := a + b &:= a + (b_1 + q\mathbb{Z})\,(b_2 + q\mathbb{Z})^{-1} \in \mathbb{Z}_q, \\ ba := ab &:= a\,(b_1 + q\mathbb{Z})\,(b_2 + q\mathbb{Z})^{-1} \in \mathbb{Z}_q. \end{aligned}$$

- Sei $q \in \mathbb{N}$ eine Primzahl. Zur Vereinfachung der Notation wird $\mathbb{Z}_q$ mit seinem Repräsentantensystem $\{0, \ldots, q-1\}$ identifiziert, in welchem alle Berechnungen modulo q ausgeführt werden.

- Für $v, w \in \mathbb{R}^m$ sei $\langle v, w \rangle := v^T w$.

- Ist $\mathbb{K}$ ein Körper und $M \subseteq \mathbb{K}^m$, so ist

$$\operatorname{span}(M) := \left\{ \sum_{i=1}^{n} \lambda_i v_i \;\middle|\; \lambda_i \in \mathbb{K}, v_i \in M, n \in \mathbb{N} \right\}.$$

  Sind $v_1, \ldots, v_n \in \mathbb{K}^m$, dann ist $\operatorname{span}(v_1, \ldots, v_n) := \operatorname{span}(\{v_1, \ldots, v_n\})$.

- Für einen Vektorraum V über einem Körper $\mathbb{K}$ bezeichne $\dim(V)$ seine Dimension.

- Der Zweierlogarithmus von $x \in \mathbb{R}_{>0}$ wird mit $\log(x)$ bezeichnet, der natürliche Logarithmus von x mit $\ln(x)$.

- Für $f, g : \mathbb{N} \to \mathbb{R}$ wird zusätzlich zu den herkömmlichen Landau-Symbolen $f \in \tilde{O}(g(n))$ geschrieben, falls $f \in O\left(g(n) (\log n)^k\right)$ für ein $k \in \mathbb{N}_0$ ist.

- Für $r \in \mathbb{R}_{>0}$ und $c \in \mathbb{R}^m$ definiere $\mathcal{B}_r(c) := \{x \in \mathbb{R}^m \mid \|x - c\| < r\}$ sowie $\overline{\mathcal{B}}_r(c) := \{x \in \mathbb{R}^m \mid \|x - c\| \leq r\}$.

- Sei $X \subseteq \mathbb{R}^m$ eine Untermannigfaltigkeit der Dimension $n \leq m$ und $M \subseteq X$ messbar. Dann bezeichnet $\operatorname{vol}_n(M)$ das n-dimensionale Volumen von M, das heißt

$$\operatorname{vol}_n(M) := \int_X \chi_M(x)\, dx$$

  mit

$$\chi_M(x) := \begin{cases} 1 & \text{, falls } x \in M \\ 0 & \text{, falls } x \in X \setminus M \end{cases}.$$

  In dieser Arbeit wird das Volumen von n-dimensionalen Parallelepipeden und von (Vereinigungen von) m-dimensionalen Kugeln im $\mathbb{R}^m$ betrachtet, welche messbare Teilmengen von Untermannigfaltigkeiten der entsprechenden Dimension sind. Für eine Einführung in dieses Themengebiet siehe zum Beispiel [Kö04], insbesondere Kapitel 11.

- Für $\sigma \in \mathbb{R}_{>0}$ und $\mu \in \mathbb{R}$ bezeichne $\mathcal{N}(\mu, \sigma)$ die Normalverteilung mit Standardabweichung σ um μ.

- Ist $\mathcal{D}$ eine Wahrscheinlichkeitsverteilung auf einer Grundmenge X, so bezeichnet $x \sim \mathcal{D}$ ein zufällig $\mathcal{D}$-verteiltes $x \in X$. Weiterhin bezeichnet $y \sim \mathcal{D}^m$ ein $y = (y_1, \ldots, y_m)^T \in X^m$, wobei $y_1, \ldots, y_m$ unabhängig zufällig $\mathcal{D}$-verteilt sind.

- Ist $\mathcal{A}$ ein Algorithmus, der bei mehreren Durchläufen mit derselben Eingabe verschiedene Ausgaben liefern kann, so wird $\mathcal{A}$ als probabilistisch bezeichnet. Ansonsten heißt $\mathcal{A}$ deterministisch.

- Ist $\mathcal{A}$ ein probabilistischer Algorithmus und Y eine Eingabe für $\mathcal{A}$, dann bezeichnet $X \leftarrow \mathcal{A}(Y)$ eine Ausgabe des Algorithmus, die nach der Verteilung von $\mathcal{A}$ bei Eingabe Y verteilt ist. Bei einem deterministischem Algorithmus wird $X = \mathcal{A}(Y)$ geschrieben.

- Bei diskreten bzw. kontinuierlichen Wahrscheinlichkeitsverteilungen wird die Verteilung häufig genauso bezeichnet wie ihre Wahrscheinlichkeitsfunktion bzw. Wahrscheinlichkeitsdichte.

- Ist G eine Gruppe mit neutralem Element $e \in G$, so definiere $\hat{G} := G \setminus \{e\}$.

- Werden n Elemente $a_1, \ldots, a_n$ aufgezählt, so wird immer davon ausgegangen, dass $n \in \mathbb{N}$ ist.

Teil I.

Attributbasierte Verschlüsselung mittels Gittermethoden - Mathematische Grundlagen

2. Verschlüsselung mittels Gittermethoden

Der erste Teil dieser Arbeit beschäftigt sich ausschließlich mit Gittern. Bevor diese formal betrachtet werden, wird in diesem Kapitel auf die Rolle von Gittern in der modernen Kryptografie eingegangen. Ausführungen dazu lassen sich zum Beispiel in Kapitel 8 aus [MG02] finden.

Bei einem sicheren Verschlüsselungsverfahren soll kein Angreifer in Polynomialzeit das Verfahren mit einer signifikanten Wahrscheinlichkeit brechen können, wenn die Parameter und verwendeten Schlüssel des Systems zufällig gewählt sind. Damit ist dieses Konzept der Sicherheit ein Average Case Konzept. Das Analogon im Worst Case würde für kryptografische Zwecke auch nicht ausreichen, da dann von einem sicheren Verfahren nur gefordert werden würde, dass es keinen Polynomialzeitangreifer gibt, der das Verfahren für alle Parameter- und Schlüsselwahlen brechen kann. Demnach sind in der modernen Kryptografie Probleme gesucht, die im Average Case schwer sind.

1996 zeigte M. Ajtai für gewisse Gitterprobleme folgenden Zusammenhang zwischen ihrer Komplexität im Worst Case sowie im Average Case: Wenn es keinen probabilistischen Polynomialzeitalgorithmus gibt, der das eine Problem mit einem polynomiellen Approximationsfaktor auf jedem Gitter löst, so gibt es auch keinen probabilistischen Polynomialzeitalgorithmus, der das andere Problem exakt auf einem zufälligen Gitter mit Wahrscheinlichkeit mindestens $\frac{1}{2}$ löst [Ajt04]. Weiterhin weiste er nach, dass dann eine Einwegfunktion existiert. Eine Einwegfunktion ist eine Funktion, die leicht zu berechnen ist, aber bei einem zufälligen Element des Wertebereichs ist es schwer, ein Element aus dessen Urbild zu finden. Somit erfüllt M. Ajtai's Einwegfunktion, dass die effiziente Urbildberechnung unter der Funktion im Average Case bereits die effiziente, polynomielle Approximation des einen betrachteten Gitterproblems im Worst Case impliziert. Durch diesen Zusammenhang sind Gitter für die moderne Kryptografie interessant geworden.

Seit 1996 gab es mehrere Verbesserungen von M. Ajtai's Aussage, zum Beispiel in [MR07]. Zudem wurden mit diesen Aussagen einige Verschlüsselungsverfahren entwickelt, deren Sicherheit unter der Annahme gezeigt werden kann, dass kein Polynomialzeitalgo-

rithmus für ein Gitterproblem im Worst Case existiert. Beispiele für solche oder ähnliche Verfahren sind in Kapitel 8 von [MG02] sowie in Abschnitt 8.2 dieser Arbeit zu finden.

Ein weiterer Vorteil für die Nutzung von Gitterproblemen als Sicherheitsannahme von Verschlüsselungsverfahren ist, dass bei den verwendeten schweren Gitterproblemen Quantenalgorithmen bisher keine signifikanten Laufzeitverbesserungen gegenüber klassischen Algorithmen liefern [MR09]. Deshalb sind diese Verfahren auch ein wichtiges Forschungsthema in der Post-Quantum Kryptografie.

3. Gitter und ihre grundlegenden Eigenschaften

Nachdem nun die Betrachtung von Gittern in der Kryptografie motiviert wurde, werden Gitter und weitere mit Gittern zusammenhängende Begriffe formal definiert sowie einige Aussagen gezeigt, die für diese Arbeit nützlich sind.

3.1. Gitter

Definition 3.1.1. *Seien $b_1, \ldots, b_n \in \mathbb{R}^m$ linear unabhängig. Dann ist*

$$\mathcal{L}(b_1, \ldots, b_n) := \left\{ \sum_{i=1}^{n} z_i b_i \;\middle|\; z_1, \ldots, z_n \in \mathbb{Z} \right\}$$

ein Gitter *mit der* Basis $(b_1, \ldots, b_n)$*. Es bezeichne $B \in \mathbb{R}^{m \times n}$ die Matrix, deren Spalten die Vektoren $b_1, \ldots, b_n$ sind. Dann definiere $\mathcal{L}(B) := \mathcal{L}(b_1, \ldots, b_n)$. Ferner sei m die* Dimension *sowie n der* Rang *des Gitters. Das Gitter habe* vollen Rang*, falls $n = m$ ist.*

Um Gitter besser zu verstehen, kann zunächst überlegt werden, dass eine Teilmenge des $\mathbb{R}^m$ genau dann ein Gitter ist, wenn sie eine nicht-triviale, diskrete Untergruppe von $(\mathbb{R}^m, +)$ ist. Diese Aussage soll im Folgenden gezeigt werden. Dafür wird vorher der Begriff der diskreten Untergruppe von $(\mathbb{R}^m, +)$ definiert sowie eine einfache Aussage für solche Untergruppen gezeigt.

Definition 3.1.2. *Sei $(\Lambda, +)$ eine Untergruppe von $(\mathbb{R}^m, +)$. Dann heißt $(\Lambda, +)$* diskret*, falls es ein $\epsilon \in \mathbb{R}_{>0}$ gibt, so dass für alle $x, y \in \Lambda$ mit $x \neq y$ gilt, dass $\|x - y\| \geq \epsilon$ ist.*

Lemma 3.1.3. *Seien $(\Lambda, +)$ eine diskrete Untergruppe von $(\mathbb{R}^m, +)$ sowie $M \subseteq \mathbb{R}^m$ eine beschränkte Teilmenge. Dann enthält $M \cap \Lambda$ endlich viele Elemente.*

Beweis. Da $(\Lambda, +)$ eine diskrete Untergruppe von $(\mathbb{R}^m, +)$ ist, gibt es ein $\epsilon \in \mathbb{R}_{>0}$, so dass für alle $x, y \in \Lambda$ mit $x \neq y$ gilt, dass $\|x - y\| \geq \epsilon$ ist. Deshalb sind für alle $x, y \in \Lambda$

mit $x \neq y$ $\mathcal{B}_{\frac{\epsilon}{2}}(x)$ und $\mathcal{B}_{\frac{\epsilon}{2}}(y)$ disjunkt. Weil $M \subseteq \mathbb{R}^m$ beschränkt ist, existieren außerdem $a \in \mathbb{R}^m$ und $r \in \mathbb{R}_{>0}$ mit $M \subseteq \mathcal{B}_r(a)$. Für alle $x \in M \cap \Lambda$ ist dann $\mathcal{B}_{\frac{\epsilon}{2}}(x) \subseteq \mathcal{B}_{r+\epsilon}(a)$. Es gilt

$$\sum_{x \in M \cap \Lambda} \mathrm{vol}_m\left(\mathcal{B}_{\frac{\epsilon}{2}}(x)\right) = \mathrm{vol}_m\left(\bigcup_{x \in M \cap \Lambda} \mathcal{B}_{\frac{\epsilon}{2}}(x)\right) \leq \mathrm{vol}_m\left(\mathcal{B}_{r+\epsilon}(a)\right) < \infty.$$

Daher hat $M \cap \Lambda$ endlich viele Elemente. □

Nun kann eine Richtung der gewünschten Aussage gezeigt werden.

Lemma 3.1.4. *Sei $(\Lambda, +)$ eine diskrete Untergruppe von $(\mathbb{R}^m, +)$ mit $\Lambda \neq \{0\}$. Dann existieren linear unabhängige $b_1, \ldots, b_n \in \mathbb{R}^m$ mit $\Lambda = \mathcal{L}(b_1, \ldots, b_n)$.*

Beweis. Sei $V := \mathrm{span}(\Lambda)$. Dann ist V ein Vektorraum der Dimension $n \in \{1, \ldots, m\}$. Daher existieren linear unabhängige $a_1, \ldots, a_n \in \Lambda$ mit $V = \mathrm{span}(a_1, \ldots, a_n)$. Für alle $i \in \{1, \ldots, n\}$ sei ferner $V_i := \mathrm{span}(a_1, \ldots, a_i)$. Nun wird folgende Behauptung gezeigt:

$$\text{Es existieren linear unabhängige } b_1, \ldots, b_n \in \Lambda, \text{ so dass für alle } i \in \{1, \ldots, n\}$$
$$\Lambda \cap V_i = \mathcal{L}(b_1, \ldots, b_i) \text{ ist.}$$

Aus dieser Behauptung folgt direkt das Lemma, da dann

$$\Lambda = \Lambda \cap V = \Lambda \cap V_n = \mathcal{L}(b_1, \ldots, b_n)$$

gilt. Deshalb wird die Behauptung jetzt durch induktive Konstruktion der $b_1, \ldots, b_n$ bewiesen.

- Induktionsanfang ($i = 1$): Wähle $\lambda_1 \in (0, 1]$ minimal, so dass $\lambda_1 a_1 \in \Lambda$ ist. Diese Wahl kann getroffen werden, da $\mathcal{B}_{\|a_1\|}(0) \cap \Lambda$ ansonsten unendlich viele Elemente enthalten müsste. Dies steht aber im Widerspruch zu Lemma 3.1.3. Setze dann $b_1 := \lambda_1 a_1$. Damit ist $b_1 \in \Lambda \setminus \{0\}$ und somit linear unabhängig. Als Nächstes wird noch die Gleichheit $\Lambda \cap V_1 = \mathcal{L}(b_1)$ gezeigt.
  - $\mathcal{L}(b_1) \subseteq \Lambda \cap V_1$: Sei $x \in \mathcal{L}(b_1)$. Dann existiert ein $z_1 \in \mathbb{Z}$ mit $x = z_1 b_1$. Da $(\Lambda, +)$ eine Untergruppe von $(\mathbb{R}^m, +)$ ist, folgt $x \in \Lambda$. Außerdem ist $x \in V_1$, weil $b_1 = \lambda_1 a_1 \in V_1$ ist. Daher ist $x \in \Lambda \cap V_1$.

– $\Lambda \cap V_1 \subseteq \mathcal{L}(b_1)$: Sei $x \in \Lambda \cap V_1$. Dann existiert ein $r_1 \in \mathbb{R}$ mit $x = r_1 a_1$. Daraus folgt, dass $x = \frac{r_1}{\lambda_1} b_1$ ist. Setze nun $\tilde{x} := \left\lfloor \frac{r_1}{\lambda_1} \right\rfloor b_1 \in \Lambda$. Damit ist ebenfalls

$$\left(\frac{r_1}{\lambda_1} - \left\lfloor \frac{r_1}{\lambda_1} \right\rfloor \right) \lambda_1 a_1 = \left(\frac{r_1}{\lambda_1} - \left\lfloor \frac{r_1}{\lambda_1} \right\rfloor \right) b_1 = x - \tilde{x} \in \Lambda.$$

Weil aber $\left(\frac{r_1}{\lambda_1} - \left\lfloor \frac{r_1}{\lambda_1} \right\rfloor \right) \in [0, 1)$ ist, folgt aus der minimalen Wahl von λ_1, dass schon $\left(\frac{r_1}{\lambda_1} - \left\lfloor \frac{r_1}{\lambda_1} \right\rfloor \right) = 0$ gilt. Deshalb ist $\frac{r_1}{\lambda_1} \in \mathbb{Z}$ und es folgt $x \in \mathcal{L}(b_1)$.

- Induktionsschritt $(i \mapsto i+1)$: Sei dafür $i \in \{1, \ldots, n-1\}$, so dass $b_1, \ldots, b_i \in \Lambda$ bereits konstruiert sind. Es soll also gelten, dass $b_1, \ldots, b_i$ linear unabhängig sind und dass $\Lambda \cap V_j = \mathcal{L}(b_1, \ldots, b_j)$ für alle $j \in \{1, \ldots, i\}$ ist. Da V_i ein i-dimensionaler Vektorraum ist, gilt $V_i = \operatorname{span}(b_1, \ldots b_i)$. Außerdem ist $a_{i+1} \in \Lambda \cap V_{i+1} \setminus V_i$. Setze nun

$$\mathcal{P} := \left\{ \sum_{j=1}^{i} s_j b_j + s_{i+1} a_{i+1} \;\middle|\; s_1, \ldots, s_{i+1} \in [0, 1] \right\}.$$

Aus Lemma 3.1.3 folgt, dass $\mathcal{P} \cap \Lambda$ endlich viele Elemente enthält. Weil ferner $a_{i+1} \in \mathcal{P} \cap \Lambda \setminus V_i$ ist, gilt $\mathcal{P} \cap \Lambda \setminus V_i \neq \emptyset$. Deswegen kann ein $b_{i+1} \in \mathcal{P} \cap \Lambda \setminus V_i$ mit minimalem Abstand zu V_i gewählt werden. Da $b_{i+1} \notin V_i$ ist, sind $b_1, \ldots, b_{i+1}$ linear unabhängig und es gilt $V_{i+1} = \operatorname{span}(b_1, \ldots, b_{i+1})$. Zuletzt ist noch die Gleichheit $\Lambda \cap V_{i+1} = \mathcal{L}(b_1, \ldots, b_{i+1})$ zu zeigen. Dafür definiere zunächst

$$\tilde{\mathcal{P}} := \left\{ \sum_{j=1}^{i+1} s_j b_j \;\middle|\; s_1, \ldots, s_{i+1} \in [0, 1] \right\}$$

und zeige folgende Zwischenbehauptung.

– Zwischenbehauptung. Für alle $v \in \tilde{\mathcal{P}} \cap \Lambda$ existieren $s_1, \ldots, s_{i+1} \in \{0, 1\}$ mit $v = \sum_{j=1}^{i+1} s_j b_j$.

Beweis. Sei $v \in \tilde{\mathcal{P}} \cap \Lambda$. Dann existieren $s_1, \ldots, s_{i+1} \in [0, 1]$ mit $v = \sum_{j=1}^{i+1} s_j b_j$. Weil außerdem $b_{i+1} \in \mathcal{P}$ ist, existieren $t_1, \ldots, t_{i+1} \in [0, 1]$, so dass

$b_{i+1} = \sum_{j=1}^{i} t_j b_j + t_{i+1} a_{i+1}$ ist. Deshalb gilt

$$v = \sum_{j=1}^{i} (s_j + s_{i+1} t_j)\, b_j + s_{i+1} t_{i+1} a_{i+1}.$$

Da für alle $j \in \{1, \ldots, i\}$ gilt, dass $(s_j + s_{i+1} t_j) \in [0, 2]$ ist, und da zudem $s_{i+1} t_{i+1} \in [0, 1]$ ist, existieren $c_1, \ldots, c_i \in \{0, 1\}$ mit

$$\tilde{v} := v - \sum_{j=1}^{i} c_j b_j = \sum_{j=1}^{i} (s_j - c_j)\, b_j + s_{i+1} b_{i+1} \in \mathcal{P} \cap \Lambda.$$

Falls $v \in V_i$ ist, dann ist $v \in \Lambda \cap V_i = \mathcal{L}(b_1, \ldots, b_i)$. Deshalb muss bereits $s_{i+1} = 0$ sein sowie $s_1, \ldots, s_i \in \{0, 1\}$ gelten. Sei also nun ohne Beschränkung der Allgemeinheit $v \notin V_i$. Dann muss $s_{i+1} \neq 0$ sein. Außerdem ist ebenfalls $\tilde{v} \notin V_i$. Damit folgt aus der minimalen Wahl von b_{i+1}, dass

$$\begin{aligned}
\operatorname{dist}(b_{i+1}, V_i) &\leq \operatorname{dist}(\tilde{v}, V_i) = \inf\{\|\tilde{v} - w\| \mid w \in V_i\} \\
&= \inf\left\{ \left\| \tilde{v} - \sum_{j=1}^{i} \lambda_j b_j \right\| \;\middle|\; \lambda_1, \ldots, \lambda_i \in \mathbb{R} \right\} \\
&= \inf\left\{ \left\| \sum_{j=1}^{i} (s_j - c_j - \lambda_j)\, b_j + s_{i+1} b_{i+1} \right\| \;\middle|\; \lambda_1, \ldots, \lambda_i \in \mathbb{R} \right\} \\
&= s_{i+1} \inf\left\{ \left\| \sum_{j=1}^{i} \frac{s_j - c_j - \lambda_j}{s_{i+1}} b_j + b_{i+1} \right\| \;\middle|\; \lambda_1, \ldots, \lambda_i \in \mathbb{R} \right\} \\
&= s_{i+1} \inf\left\{ \left\| b_{i+1} - \sum_{j=1}^{i} \frac{c_j + \lambda_j - s_j}{s_{i+1}} b_j \right\| \;\middle|\; \lambda_1, \ldots, \lambda_i \in \mathbb{R} \right\} \\
&= s_{i+1} \inf\left\{ \left\| b_{i+1} - \sum_{j=1}^{i} \tilde{\lambda}_j b_j \right\| \;\middle|\; \tilde{\lambda}_1, \ldots, \tilde{\lambda}_i \in \mathbb{R} \right\} \\
&= s_{i+1} \inf\{\|b_{i+1} - w\| \mid w \in V_i\} = s_{i+1} \operatorname{dist}(b_{i+1}, V_i)
\end{aligned}$$

gilt. Deswegen muss schon $s_{i+1} = 1$ sein. Dadurch ist aber

$$v - b_{i+1} = \sum_{j=1}^{i} s_j b_j \in \Lambda \cap V_i = \mathcal{L}(b_1, \ldots, b_i)$$

und es folgt $s_1, \ldots, s_i \in \{0,1\}$. □

Nun wird die gewünschte Gleichheit $\Lambda \cap V_{i+1} = \mathcal{L}(b_1, \ldots, b_{i+1})$ gezeigt.

- $\mathcal{L}(b_1, \ldots, b_{i+1}) \subseteq \Lambda \cap V_{i+1}$: Sei $x \in \mathcal{L}(b_1, \ldots, b_{i+1})$. Dann existieren $z_1, \ldots, z_{i+1} \in \mathbb{Z}$ mit $x = \sum_{j=1}^{i+1} z_j b_j$. Weil $b_1, \ldots, b_{i+1} \in \Lambda \cap V_{i+1}$ sind, ist auch $x \in \Lambda \cap V_{i+1}$.
- $\Lambda \cap V_{i+1} \subseteq \mathcal{L}(b_1, \ldots, b_{i+1})$: Sei $x \in \Lambda \cap V_{i+1}$. Dann existieren $r_1, \ldots, r_{i+1} \in \mathbb{R}$ mit $x = \sum_{j=1}^{i+1} r_j b_j$. Setze $\tilde{x} := \sum_{j=1}^{i+1} \lfloor r_j \rfloor b_j \in \Lambda$. Damit ist ebenfalls $\sum_{j=1}^{i+1} (r_j - \lfloor r_j \rfloor) b_j = x - \tilde{x} \in \Lambda$. Weil aber für alle $j \in \{1, \ldots, i+1\}$ gilt, dass $(r_j - \lfloor r_j \rfloor) \in [0,1)$ ist, gilt zudem $\sum_{j=1}^{i+1} (r_j - \lfloor r_j \rfloor) b_j \in \tilde{\mathcal{P}}$. Aus der Zwischenbehauptung folgt nun, dass für alle $j \in \{1, \ldots, i+1\}$ bereits $(r_j - \lfloor r_j \rfloor) \in \{0,1\}$, also $(r_j - \lfloor r_j \rfloor) = 0$ ist. Daher sind $r_1, \ldots, r_{i+1} \in \mathbb{Z}$ und es folgt $x \in \mathcal{L}(b_1, \ldots, b_{i+1})$.

□

Beispiel 3.1.5. *Sei $q \in \mathbb{N}$ eine Primzahl und $A \in \mathbb{Z}_q^{n \times m}$. Dann ist $\Lambda_q^\perp(A) := \{e \in \mathbb{Z}^m \mid Ae = 0\}$ ein Gitter mit vollem Rang.*

Beweis. Sind $a, b \in \Lambda_q^\perp(A)$, so ist $A(a-b) = Aa - Ab = 0$ und deswegen ist ebenfalls $a - b \in \Lambda_q^\perp(A)$. Daher ist $\left(\Lambda_q^\perp(A), +\right)$ eine Untergruppe von $(\mathbb{R}^m, +)$. Außerdem ist $\left(\Lambda_q^\perp(A), +\right)$ offensichtlich diskret, weil $\Lambda_q^\perp(A) \subseteq \mathbb{Z}^m$ ist. Wegen $q\mathbb{Z}^m \subseteq \Lambda_q^\perp(A)$ ist $\Lambda_q^\perp(A) \neq \{0\}$ und damit folgt aus Lemma 3.1.4, dass $\Lambda_q^\perp(A)$ ein Gitter ist.

Zudem enthält $\Lambda_q^\perp(A)$ wegen $q\mathbb{Z}^m \subseteq \Lambda_q^\perp(A)$ die m linear unabhängigen Spalten der Matrix $qI_m \in \mathbb{Z}^{m \times m}$. Damit muss jede Basis für $\Lambda_q^\perp(A)$ aus m Vektoren bestehen. $\Lambda_q^\perp(A)$ ist also ein Gitter mit vollem Rang. □

Bevor gezeigt werden kann, dass jedes Gitter eine nicht-triviale, diskrete Untergruppe von $(\mathbb{R}^m, +)$ ist, wird zunächst das Gram-Schmidtsche Orthogonalisierungsverfahren wiederholt.

3.2. Gram-Schmidtsche Orthogonalisierung

Definition 3.2.1. *Seien $b_1, \dots, b_n \in \mathbb{R}^m$ linear unabhängig. Dann definiere für alle $i \in \{1, \dots, n\}$*

$$\tilde{b}_i := b_i - \sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b}_j \rangle}{\langle \tilde{b}_j, \tilde{b}_j \rangle} \tilde{b}_j \in \mathbb{R}^m.$$

$\tilde{b}_1, \dots, \tilde{b}_n$ wird als die Gram-Schmidtsche Orthogonalisierung *von $b_1, \dots, b_n$ bezeichnet. Für $B := \{b_1, \dots, b_n\}$ sei $\tilde{B} := \{\tilde{b}_1, \dots, \tilde{b}_n\}$. Ist $T \in \mathbb{R}^{m \times n}$ die Matrix, deren Spalten die Vektoren $b_1, \dots, b_n$ sind, so sei außerdem $\tilde{T} \in \mathbb{R}^{m \times n}$ die Matrix, deren Spalten die Vektoren $\tilde{b}_1, \dots, \tilde{b}_n$ sind.*

Bekannte Eigenschaften der Gram-Schmidtschen Orthogonalisierung sind:

1. Für alle $i, j \in \{1, \dots, n\}$ mit $i \neq j$ ist $\langle \tilde{b}_i, \tilde{b}_j \rangle = 0$.
2. Für alle $i \in \{1, \dots, n\}$ ist $\text{span}\,(b_1, \dots, b_i) = \text{span}\left(\tilde{b}_1, \dots, \tilde{b}_i\right)$.
3. Für alle $i \in \{1, \dots, n\}$ ist $\sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b}_j \rangle}{\langle \tilde{b}_j, \tilde{b}_j \rangle} \tilde{b}_j$ die orthogonale Projektion von b_i auf $\text{span}\left(\tilde{b}_1, \dots, \tilde{b}_{i-1}\right)$.
4. Die Reihenfolge der Vektoren $b_1, \dots, b_n$ beeinflusst das Ergebnis $\tilde{b}_1, \dots, \tilde{b}_n$.

Zwei weitere nützliche Lemmata zur Gram-Schmidtschen Orthogonalisierung werden nun mit Beweis angeführt.

Lemma 3.2.2. *Seien $b_1, \dots, b_n \in \mathbb{R}^m$ linear unabhängig mit Gram-Schmidtscher Orthogonalisierung $\tilde{b}_1, \dots, \tilde{b}_n \in \mathbb{R}^m$. Ferner sei $B := \{b_1, \dots, b_n\}$. Dann ist $\|\tilde{B}\| \leq \|B\|$.*

Beweis. Für alle $i \in \{1, \dots, n\}$ gilt

$$\begin{aligned}
\|\tilde{b}_i\|^2 = \langle \tilde{b}_i, \tilde{b}_i \rangle &= \left\langle b_i - \sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b_j} \rangle}{\langle \tilde{b}_j, \tilde{b}_j \rangle} \tilde{b}_j, b_i - \sum_{k=1}^{i-1} \frac{\langle b_i, \tilde{b_k} \rangle}{\langle \tilde{b}_k, \tilde{b}_k \rangle} \tilde{b}_k \right\rangle \\
&= \left\langle b_i, b_i - \sum_{k=1}^{i-1} \frac{\langle b_i, \tilde{b_k} \rangle}{\langle \tilde{b}_k, \tilde{b}_k \rangle} \tilde{b}_k \right\rangle - \sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b_j} \rangle}{\langle \tilde{b}_j, \tilde{b}_j \rangle} \left\langle \tilde{b}_j, b_i - \sum_{k=1}^{i-1} \frac{\langle b_i, \tilde{b_k} \rangle}{\langle \tilde{b}_k, \tilde{b}_k \rangle} \tilde{b}_k \right\rangle \\
&= \langle b_i, b_i \rangle - \sum_{k=1}^{i-1} \frac{\langle b_i, \tilde{b_k} \rangle}{\langle \tilde{b}_k, \tilde{b}_k \rangle} \langle b_i, \tilde{b}_k \rangle - \sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b_j} \rangle}{\langle \tilde{b}_j, \tilde{b}_j \rangle} \langle \tilde{b}_j, b_i \rangle + \sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b_j} \rangle}{\langle \tilde{b}_j, \tilde{b}_j \rangle} \sum_{k=1}^{i-1} \frac{\langle b_i, \tilde{b_k} \rangle}{\langle \tilde{b}_k, \tilde{b}_k \rangle} \langle \tilde{b}_j, \tilde{b}_k \rangle
\end{aligned}$$

$$= \|b_i\|^2 - 2\sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b_j}\rangle}{\langle \tilde{b}_j, \tilde{b}_j\rangle}\langle b_i, \tilde{b}_j\rangle + \sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b_j}\rangle}{\langle \tilde{b}_j, \tilde{b}_j\rangle}\frac{\langle b_i, \tilde{b_j}\rangle}{\langle \tilde{b}_j, \tilde{b}_j\rangle}\langle \tilde{b}_j, \tilde{b}_j\rangle$$
$$= \|b_i\|^2 - \sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b_j}\rangle^2}{\|\tilde{b}_j\|^2} \leq \|b_i\|^2.$$

Daher gilt für alle $i \in \{1, \ldots, n\}$, dass $\|\tilde{b}_i\| \leq \|b_i\|$ ist, und damit folgt $\|\tilde{B}\| \leq \|B\|$. □

Folgendes Lemma stellt einen interessanten Zusammenhang zwischen der Gram-Schmidtschen Orthogonalisierung einer Gitterbasis und der Länge von Gittervektoren dar und kann als Theorem 1.1 in [MG02] gefunden werden.

Lemma 3.2.3. *Seien $b_1, \ldots, b_n \in \mathbb{R}^m$ linear unabhängig mit Gram-Schmidtscher Orthogonalisierung $\tilde{b}_1, \ldots, \tilde{b}_n \in \mathbb{R}^m$. Dann ist für alle $x \in \mathcal{L}(b_1, \ldots, b_n) \setminus \{0\}$*

$$\|x\| \geq \min\left\{\|\tilde{b}_i\| \mid i \in \{1, \ldots, n\}\right\}.$$

Beweis. Sei $x \in \mathcal{L}(b_1, \ldots, b_n) \setminus \{0\}$. Dann existieren $z_1, \ldots, z_n \in \mathbb{Z}$ mit $x = \sum_{i=1}^{n} z_i b_i$. Sei $k \in \{1, \ldots, n\}$ maximal, so dass $z_k \neq 0$ ist. Dann ist $x = \sum_{i=1}^{k} z_i b_i$. Für alle $i \in \{1, \ldots, k\}$ gilt außerdem

$$\langle b_i, \tilde{b}_k\rangle = \left\langle \tilde{b}_i + \sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b_j}\rangle}{\langle \tilde{b}_j, \tilde{b}_j\rangle}\tilde{b}_j, \tilde{b}_k \right\rangle = \langle \tilde{b}_i, \tilde{b}_k\rangle + \sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b_j}\rangle}{\langle \tilde{b}_j, \tilde{b}_j\rangle}\langle \tilde{b}_j, \tilde{b}_k\rangle$$
$$= \begin{cases} 0 & \text{, falls } i < k \\ \langle \tilde{b}_k, \tilde{b}_k\rangle & \text{, falls } i = k \end{cases}.$$

Daraus folgt mithilfe der Cauchy-Schwarzschen Ungleichung

$$\|x\|\|\tilde{b}_k\| \geq |\langle x, \tilde{b}_k\rangle| = \left|\left\langle \sum_{i=1}^{k} z_i b_i, \tilde{b}_k \right\rangle\right| = \left|\sum_{i=1}^{k} z_i \langle b_i, \tilde{b}_k\rangle\right| = |z_k\langle \tilde{b}_k, \tilde{b}_k\rangle| = |z_k|\|\tilde{b}_k\|^2 \geq \|\tilde{b}_k\|^2.$$

Da $\tilde{b}_1, \ldots, \tilde{b}_n$ linear unabhängig sind, gilt nun, dass

$$\|x\| \geq \|\tilde{b}_k\| \geq \min\left\{\|\tilde{b}_i\| \mid i \in \{1, \ldots, n\}\right\}$$

ist. □

Jetzt kann die Umkehrung von Lemma 3.1.4 gezeigt werden.

Lemma 3.2.4. *Seien $b_1, \ldots, b_n \in \mathbb{R}^m$ linear unabhängig. Dann ist $(\mathcal{L}(b_1, \ldots, b_n), +)$ eine diskrete Untergruppe von $(\mathbb{R}^m, +)$ mit $\mathcal{L}(b_1, \ldots, b_n) \neq \{0\}$.*

Beweis. Zunächst ist $\mathcal{L}(b_1, \ldots, b_n) \neq \{0\}$ klar, da $0 \neq b_1 \in \mathcal{L}(b_1, \ldots, b_n)$ ist. Seien nun $x, y \in \mathcal{L}(b_1, \ldots, b_n)$. Dann existieren $r_1, \ldots, r_n, s_1, \ldots, s_n \in \mathbb{Z}$ mit $x = \sum_{i=1}^{n} r_i b_i$ und $y = \sum_{i=1}^{n} s_i b_i$. Deshalb ist $x - y = \sum_{i=1}^{n} (r_i - s_i) b_i \in \mathcal{L}(b_1, \ldots, b_n)$. Damit ist $(\mathcal{L}(b_1, \ldots, b_n), +)$ eine Untergruppe von $(\mathbb{R}^m, +)$.

Sei weiter $\tilde{b}_1, \ldots, \tilde{b}_n$ die Gram-Schmidtsche Orthogonalisierung von $b_1, \ldots, b_n$. Setze $\epsilon := \min\left\{\|\tilde{b}_i\| \mid i \in \{1, \ldots, n\}\right\}$. Dann ist $\epsilon > 0$, da $\tilde{b}_1, \ldots, \tilde{b}_n$ linear unabhängig sind. Falls $x \neq y$ ist, so folgt aus Lemma 3.2.3, dass $\|x - y\| \geq \epsilon$ ist. Deshalb ist $(\mathcal{L}(b_1, \ldots, b_n), +)$ eine diskrete Untergruppe von $(\mathbb{R}^m, +)$. □

3.3. Fundamentalparallelepiped und Determinante

Definition 3.3.1. *Seien $b_1, \ldots, b_n \in \mathbb{R}^m$ linear unabhängig. Dann ist*

$$\mathcal{P}(b_1, \ldots, b_n) := \left\{ \sum_{i=1}^{n} r_i b_i \;\middle|\; r_1, \ldots, r_n \in [0, 1) \right\}$$

das Fundamentalparallelepiped *zur Basis $(b_1, \ldots, b_n)$. Es bezeichne $B \in \mathbb{R}^{m \times n}$ die Matrix, deren Spalten die Vektoren $b_1, \ldots, b_n$ sind. Dann definiere $\mathcal{P}(B) := \mathcal{P}(b_1, \ldots, b_n)$.*

Zunächst lässt sich feststellen, dass der durch $b_1, \ldots, b_n$ aufgespannte Vektorraum durch affine Translationen von $\mathcal{P}(b_1, \ldots, b_n)$ an jedem Vektor aus $\mathcal{L}(b_1, \ldots, b_n)$ partitioniert wird. Dies wird in folgendem Lemma bewiesen.

Lemma 3.3.2. *Seien $b_1, \ldots, b_n \in \mathbb{R}^m$ linear unabhängig und $V := \operatorname{span}(b_1, \ldots, b_n)$. Dann ist $\{x + \mathcal{P}(b_1, \ldots, b_n) \mid x \in \mathcal{L}(b_1, \ldots, b_n)\}$ eine Partition von V.*

Beweis. Es ist klar, dass für alle $x \in \mathcal{L}(b_1, \ldots, b_n)$ gilt, dass $x + \mathcal{P}(b_1, \ldots, b_n) \subseteq V$ ist. Sei nun $y \in V$. Dann existieren $r_1, \ldots, r_n \in \mathbb{R}$ mit $y = \sum_{i=1}^{n} r_i b_i$. Setze $\tilde{y} := \sum_{i=1}^{n} \lfloor r_i \rfloor b_i$. Damit ist $\tilde{y} \in \mathcal{L}(b_1, \ldots, b_n)$ und $y - \tilde{y} = \sum_{i=1}^{n} (r_i - \lfloor r_i \rfloor) b_i \in \mathcal{P}(b_1, \ldots, b_n)$. Deswegen folgt, dass $y = \tilde{y} + (y - \tilde{y}) \in \tilde{y} + \mathcal{P}(b_1, \ldots, b_n)$ ist. Somit gilt $V = \bigcup_{x \in \mathcal{L}(b_1, \ldots, b_n)} x + \mathcal{P}(b_1, \ldots, b_n)$.

Seien $x, \tilde{x} \in \mathcal{L}(b_1, \ldots, b_n)$. Daher existieren $z_1, \ldots, z_n, \tilde{z}_1, \ldots, \tilde{z}_n \in \mathbb{Z}$ mit $x = \sum_{i=1}^{n} z_i b_i$ und $\tilde{x} = \sum_{i=1}^{n} \tilde{z}_i b_i$. Angenommen es gibt ein $y \in (x + \mathcal{P}(b_1, \ldots, b_n)) \cap (\tilde{x} + \mathcal{P}(b_1, \ldots, b_n))$.

Dann existieren $r_1, \ldots, r_n \in \mathbb{R}$ mit $y = \sum_{i=1}^{n} r_i b_i$. Außerdem gilt für alle $i \in \{1, \ldots, n\}$, dass $r_i \in [z_i, z_i + 1)$ und $r_i \in [\tilde{z}_i, \tilde{z}_i + 1)$ ist. Für alle $i \in \{1, \ldots, n\}$ folgt damit $z_i = \lfloor r_i \rfloor = \tilde{z}_i$ und deshalb auch $x = \tilde{x}$. Die Elemente aus $\{x + \mathcal{P}(b_1, \ldots, b_n) \mid x \in \mathcal{L}(b_1, \ldots, b_n)\}$ sind also paarweise disjunkt und daher ist alles gezeigt. □

Dieses Lemma impliziert sofort, dass es für jedes $y \in V$ genau ein $x \in \mathcal{L}(b_1, \ldots, b_n)$ mit $y \in x + \mathcal{P}(b_1, \ldots, b_n)$ gibt. Damit ist aber auch $y - x \in \mathcal{P}(b_1, \ldots, b_n)$ eindeutig bestimmt.

Definition 3.3.3. *Seien $b_1, \ldots, b_n \in \mathbb{R}^m$ linear unabhängig und $V := \operatorname{span}(b_1, \ldots, b_n)$. Für $y \in V$ sei $y \mod \mathcal{L}(b_1, \ldots, b_n)$ der eindeutige Vektor $x \in \mathcal{P}(b_1, \ldots, b_n)$, so dass $y - x \in \mathcal{L}(b_1, \ldots, b_n)$ ist.*

Ist $y = \sum_{i=1}^{n} r_i b_i$ für $r_1, \ldots, r_n \in \mathbb{R}$, so ist $y \mod \mathcal{L}(b_1, \ldots, b_n) = \sum_{i=1}^{n} (r_i - \lfloor r_i \rfloor) b_i$, da $\sum_{i=1}^{n} (r_i - \lfloor r_i \rfloor) b_i \in \mathcal{P}(b_1, \ldots, b_n)$ und $y - \sum_{i=1}^{n} (r_i - \lfloor r_i \rfloor) b_i = \sum_{i=1}^{n} \lfloor r_i \rfloor b_i \in \mathcal{L}(b_1, \ldots, b_n)$ ist. Außerdem kann die Länge von $y \mod \mathcal{L}(b_1, \ldots, b_n)$ nach oben beschränkt werden.

Lemma 3.3.4. *Seien $b_1, \ldots, b_n \in \mathbb{R}^m$ linear unabhängig und $x \in \mathcal{P}(b_1, \ldots, b_n)$. Dann ist $\|x\| < \sum_{i=1}^{n} \|b_i\|$.*

Beweis. Da $x \in \mathcal{P}(b_1, \ldots, b_n)$ ist, existieren $r_1, \ldots, r_n \in [0, 1)$ mit $x = \sum_{i=1}^{n} r_i b_i$. Deswegen ist

$$\|x\| = \left\| \sum_{i=1}^{n} r_i b_i \right\| \leq \sum_{i=1}^{n} \|r_i b_i\| = \sum_{i=1}^{n} r_i \|b_i\| < \sum_{i=1}^{n} \|b_i\|.$$

□

Als Nächstes werden zwei Möglichkeiten angegeben, wie das Volumen von $\mathcal{P}(b_1, \ldots, b_n)$ aus $b_1, \ldots, b_n$ berechnet werden kann.

Lemma 3.3.5. *Seien $b_1, \ldots, b_n \in \mathbb{R}^m$ linear unabhängig mit Gram-Schmidtscher Orthogonalisierung $\tilde{b}_1, \ldots, \tilde{b}_n \in \mathbb{R}^m$. Dann ist $\operatorname{vol}_n(\mathcal{P}(b_1, \ldots, b_n)) = \prod_{i=1}^{n} \|\tilde{b}_i\|$.*

Beweis. Der Beweis erfolgt durch Induktion nach n.

- Induktionsanfang ($n = 1$): Es gilt $\operatorname{vol}_1(\mathcal{P}(b_1)) = \|b_1\| = \|\tilde{b}_1\|$.

- Induktionsschritt ($n \mapsto n+1$): Dafür sei $n \in \{1, \ldots, m-1\}$. Dann ist

$$\mathrm{vol}_{n+1}\left(\mathcal{P}\left(b_1, \ldots, b_{n+1}\right)\right) = \mathrm{vol}_n\left(\mathcal{P}\left(b_1, \ldots, b_n\right)\right) \mathrm{dist}\left(b_{n+1}, \mathrm{span}\left(b_1, \ldots, b_n\right)\right).$$

  Aus der zweiten und dritten in Abschnitt 3.2 genannten Eigenschaft der Gram-Schmidtschen Orthogonalisierung folgt, dass

$$\mathrm{dist}\left(b_{n+1}, \mathrm{span}\left(b_1, \ldots, b_n\right)\right) = \left\| b_{n+1} - \sum_{j=1}^{n} \frac{\langle b_{n+1}, \tilde{b_j}\rangle}{\langle \tilde{b}_j, \tilde{b}_j\rangle} \tilde{b}_j \right\| = \|\tilde{b}_{n+1}\|$$

  ist. Daher ergibt sich mit der Induktionsvoraussetzung

$$\begin{aligned} \mathrm{vol}_{n+1}\left(\mathcal{P}\left(b_1, \ldots, b_{n+1}\right)\right) &= \mathrm{vol}_n\left(\mathcal{P}\left(b_1, \ldots, b_n\right)\right) \|\tilde{b}_{n+1}\| \\ &= \left(\prod_{i=1}^{n} \|\tilde{b}_i\|\right) \|\tilde{b}_{n+1}\| = \prod_{i=1}^{n+1} \|\tilde{b}_i\|. \end{aligned}$$

□

Lemma 3.3.6. *Seien $b_1, \ldots, b_n \in \mathbb{R}^m$ linear unabhängig und $B \in \mathbb{R}^{m \times n}$ bezeichne die Matrix, deren Spalten die Vektoren $b_1, \ldots, b_n$ sind. Dann ist* $\mathrm{vol}_n\left(\mathcal{P}\left(B\right)\right) = \sqrt{\det\left(B^T B\right)}$.

Beweis. Es bezeichne $\tilde{b}_1, \ldots, \tilde{b}_n \in \mathbb{R}^m$ die Gram-Schmidtsche Orthogonalisierung von $b_1, \ldots, b_n$. Setze dann für alle $i \in \{1, \ldots, n\}$ $u_i := \frac{\tilde{b}_i}{\|\tilde{b}_i\|}$. Damit gilt für alle $i \in \{1, \ldots, n\}$

$$b_i = \sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b_j}\rangle}{\langle \tilde{b}_j, \tilde{b}_j\rangle} \tilde{b}_j + \tilde{b}_i = \sum_{j=1}^{i-1} \frac{\langle b_i, \tilde{b_j}\rangle}{\langle \tilde{b}_j, \tilde{b}_j\rangle} \|\tilde{b}_j\| u_j + \|\tilde{b}_i\| u_i.$$

Sei nun $Q \in \mathbb{R}^{m \times n}$ die Matrix, deren Spalten die Vektoren $u_1, \ldots, u_n$ sind. Ferner sei $R = (r_{j,i})_{1 \le j,i \le n} \in \mathbb{R}^{n \times n}$ die obere Dreiecksmatrix mit

$$r_{j,i} = \begin{cases} \frac{\langle b_i, \tilde{b_j}\rangle}{\langle \tilde{b}_j, \tilde{b}_j\rangle} \|\tilde{b}_j\| & \text{, falls } j < i \\ \|\tilde{b}_i\| & \text{, falls } j = i \\ 0 & \text{, falls } j > i \end{cases}.$$

Dann ist für alle $i \in \{1, \ldots, n\}$ $b_i = \sum_{j=1}^{n} r_{j,i} u_j$ und daher gilt $B = QR$. Nach Definition

von Q ist $Q^TQ = I_n$. Daraus folgt

$$\begin{aligned}\det\left(B^TB\right) &= \det\left((QR)^T(QR)\right) = \det\left(R^TQ^TQR\right)\\ &= \det\left(R^TR\right) = \det\left(R^T\right)\det(R) = \det(R)^2 = \left(\prod_{i=1}^{n}\|\tilde{b}_i\|\right)^2\end{aligned}$$

und mit Lemma 3.3.5 ergibt sich

$$\mathrm{vol}_n(\mathcal{P}(B)) = \prod_{i=1}^{n}\|\tilde{b}_i\| = \sqrt{\det(B^TB)}.$$

□

Für ein Gitter kann es mehrere Basen geben, die dieses Gitter erzeugen. Bei all diesen Basen ist aber das Volumen des zugehörigen Fundamentalparallelepipeds gleich. Damit ist dieses Volumen eine Gitterinvariante, die als Determinante bezeichnet werden wird. Um dies zu beweisen, wird vorher betrachtet, wann zwei Basen dasselbe Gitter erzeugen. Diese Aussage kann auch bei [Reg04] als Lemma 3 in Lecture 1: „Introduction" gefunden werden.

Lemma 3.3.7. *Seien $B, C \in \mathbb{R}^{m\times n}$ mit jeweils linear unabhängigen Spalten. Dann gilt $\mathcal{L}(B) = \mathcal{L}(C)$ genau dann, wenn es ein $U \in \mathbb{Z}^{n\times n}$ mit $|\det(U)| = 1$ und $C = BU$ gibt.*

Beweis. Es bezeichnen $b_1, \ldots, b_n \in \mathbb{R}^m$ die Spalten von B und $c_1, \ldots, c_n \in \mathbb{R}^m$ die Spalten von C.

Sei zunächst $\mathcal{L}(B) = \mathcal{L}(C)$. Dann ist für alle $i \in \{1, \ldots, n\}$ zum einen $b_i \in \mathcal{L}(C)$ und zum anderen $c_i \in \mathcal{L}(B)$. Deswegen existieren $U_1, U_2 \in \mathbb{Z}^{n\times n}$ mit $B = CU_1$ und $C = BU_2$. Daher folgt $B = CU_1 = BU_2U_1$ und somit gilt, dass

$$\begin{aligned}\det\left(B^TB\right) &= \det\left((BU_2U_1)^T(BU_2U_1)\right) = \det\left((U_2U_1)^TB^TB(U_2U_1)\right)\\ &= \det\left((U_2U_1)^T\right)\det\left(B^TB\right)\det(U_2U_1)\\ &= \det\left(B^TB\right)(\det(U_2U_1))^2\end{aligned}$$

ist. Weil aber $\det\left(B^TB\right) > 0$ ist, folgt, dass $(\det(U_2U_1))^2 = 1$ und damit $|\det(U_2)||\det(U_1)| = 1$ ist. Da $\det(U_1) \in \mathbb{Z}$ und $\det(U_2) \in \mathbb{Z}$ sind, gilt schon $|\det(U_1)| = |\det(U_2)| = 1$. Dies zeigt eine Richtung der gewünschten Aussage.

Nun existiere ein $U \in \mathbb{Z}^{n\times n}$ mit $|\det(U)| = 1$ und $C = BU$. Für alle $i \in \{1,\dots,n\}$ ist dann $c_i \in \mathcal{L}(B)$. Daraus folgt $\mathcal{L}(C) \subseteq \mathcal{L}(B)$. Außerdem ist $B = CU^{-1}$ und es gilt $|\det(U^{-1})| = \left|\frac{1}{\det(U)}\right| = 1$. Da $U^{-1} = \frac{1}{\det(U)}\operatorname{adj}(U)$ ist, wobei $\operatorname{adj}(U) \in \mathbb{Z}^{n\times n}$ die Adjunkte von U bezeichnet, ist $U^{-1} \in \mathbb{Z}^{n\times n}$. Für alle $i \in \{1,\dots,n\}$ ist daher $b_i \in \mathcal{L}(C)$ und es gilt $\mathcal{L}(B) \subseteq \mathcal{L}(C)$. Insgesamt folgt also $\mathcal{L}(B) = \mathcal{L}(C)$. □

Jetzt kann leicht nachgerechnet werden, dass das Volumen des Fundamentalparallelepipeds bei verschiedenen Basen für dasselbe Gitter invariant bleibt.

Lemma 3.3.8. *Seien $B, C \in \mathbb{R}^{m\times n}$ mit jeweils linear unabhängigen Spalten, so dass $\mathcal{L}(B) = \mathcal{L}(C)$ gilt. Dann ist* $\operatorname{vol}_n(\mathcal{P}(B)) = \operatorname{vol}_n(\mathcal{P}(C))$.

Beweis. Wegen Lemma 3.3.7 existiert ein $U \in \mathbb{Z}^{n\times n}$ mit $|\det(U)| = 1$ und $C = BU$. Deshalb gilt

$$\begin{aligned}\det\left(C^TC\right) &= \det\left((BU)^T(BU)\right) = \det\left(U^TB^TBU\right)\\ &= \det\left(U^T\right)\det\left(B^TB\right)\det(U)\\ &= \det\left(B^TB\right)(\det(U))^2 = \det\left(B^TB\right).\end{aligned}$$

Aus Lemma 3.3.6 folgt somit $\operatorname{vol}_n(\mathcal{P}(B)) = \sqrt{\det(B^TB)} = \sqrt{\det(C^TC)} = \operatorname{vol}_n(\mathcal{P}(C))$. □

Damit wird folgende Definition sinnvoll.

Definition 3.3.9. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter mit Rang n. Dann ist*

$$\det(\Lambda) := \operatorname{vol}_n(\mathcal{P}(b_1,\dots,b_n))$$

die Determinante *von Λ, wobei $(b_1,\dots,b_n)$ eine beliebige Basis von Λ ist.*

3.4. Duales Gitter

Definition 3.4.1. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter. Dann ist*

$$\Lambda^* := \{y \in \operatorname{span}(\Lambda) \mid \forall x \in \Lambda : \langle x, y\rangle \in \mathbb{Z}\}$$

das duale Gitter *zu Λ.*

Das nächste Lemma zeigt insbesondere, dass Λ^* tatsächlich ein Gitter ist, wobei der Rang von Λ^* gleich dem Rang von Λ ist. Die folgenden Rechnungen sind auch bei [Reg04] in Lecture 8: „Dual Lattices" aufgeführt.

Lemma 3.4.2. *Sei $B \in \mathbb{R}^{m\times n}$ mit linear unabhängigen Spalten. Für $D := B\left(B^TB\right)^{-1}$ gilt dann $\mathcal{L}(B)^* = \mathcal{L}(D)$.*

Beweis. Beachte zunächst, dass die Spalten von $D \in \mathbb{R}^{m\times n}$ linear unabhängig sind, da ansonsten wegen $B = D\left(B^TB\right)$ die Spalten von B nicht linear unabhängig sein könnten. Im Folgenden bezeichnen $b_1, \ldots, b_n \in \mathbb{R}^m$ die Spalten von B und $d_1, \ldots, d_n \in \mathbb{R}^m$ die Spalten von D. Nun wird die Gleichheit $\mathcal{L}(B)^* = \mathcal{L}(D)$ gezeigt.

- $\mathcal{L}(B)^* \subseteq \mathcal{L}(D)$: Wegen $D = B\left(B^TB\right)^{-1}$ sind $d_1, \ldots, d_n \in \operatorname{span}(b_1, \ldots, b_n)$. Deshalb ist $\operatorname{span}(d_1, \ldots, d_n) \subseteq \operatorname{span}(b_1, \ldots, b_n)$. Analog folgt aus $B = D\left(B^TB\right)$, dass $b_1, \ldots, b_n \in \operatorname{span}(d_1, \ldots, d_n)$ und damit $\operatorname{span}(b_1, \ldots, b_n) \subseteq \operatorname{span}(d_1, \ldots, d_n)$ ist. Für jedes $y \in \mathcal{L}(B)^*$ gilt somit

$$y \in \operatorname{span}(\mathcal{L}(B)) = \operatorname{span}(b_1, \ldots, b_n) = \operatorname{span}(d_1, \ldots, d_n).$$

  Daher gibt es $r_1, \ldots, r_n \in \mathbb{R}$ mit $y = \sum_{i=1}^{n} r_i d_i$. Wegen $B^TD = B^TB\left(B^TB\right)^{-1} = I_n$ gilt für alle $i \in \{1, \ldots, n\}$

$$r_i = \sum_{j=1}^{n} r_j \langle b_i, d_j\rangle = \left\langle b_i, \sum_{j=1}^{n} r_j d_j \right\rangle = \langle b_i, y\rangle \in \mathbb{Z}.$$

  Deswegen ist $y \in \mathcal{L}(D)$.

- $\mathcal{L}(D) \subseteq \mathcal{L}(B)^*$: Sei $x \in \mathcal{L}(B)$. Dann existieren $a_1, \ldots, a_n \in \mathbb{Z}$ mit $x = \sum_{i=1}^{n} a_i b_i$. Für alle $i \in \{1, \ldots, n\}$ gilt dann

$$\langle x, d_i\rangle = \left\langle \sum_{j=1}^{n} a_j b_j, d_i \right\rangle = \sum_{j=1}^{n} a_j \langle b_j, d_i\rangle = a_i \in \mathbb{Z}.$$

  Somit sind $d_1, \ldots, d_n \in \mathcal{L}(B)^*$. Da für alle $c_1, \ldots, c_n \in \mathbb{Z}$ gilt, dass $\left\langle x, \sum_{i=1}^{n} c_i d_i \right\rangle = \sum_{i=1}^{n} c_i \langle x, d_i\rangle \in \mathbb{Z}$ ist, folgt $\mathcal{L}(D) \subseteq \mathcal{L}(B)^*$.

□

Zum Abschluss dieses Abschnitts werden noch zwei einfache Eigenschaften dualer Gitter nachgerechnet.

Lemma 3.4.3. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter. Dann gilt $(\Lambda^*)^* = \Lambda$, d.h. Λ ist das duale Gitter zu Λ^*.*

Beweis. Sei $B \in \mathbb{R}^{m \times n}$ mit linear unabhängigen Spalten, so dass $\Lambda = \mathcal{L}(B)$ ist. Setze $D := B\left(B^T B\right)^{-1}$ sowie $C := D\left(D^T D\right)^{-1}$. Aus Lemma 3.4.2 folgt $\Lambda^* = \mathcal{L}(B)^* = \mathcal{L}(D)$ und damit $(\Lambda^*)^* = \mathcal{L}(D)^* = \mathcal{L}(C)$. Wegen

$$\begin{aligned} D^T D &= \left(B\left(B^T B\right)^{-1}\right)^T \left(B\left(B^T B\right)^{-1}\right) \\ &= \left(\left(B^T B\right)^{-1}\right)^T B^T B \left(B^T B\right)^{-1} \\ &= \left(\left(B^T B\right)^{-1}\right)^T = \left(\left(B^T B\right)^T\right)^{-1} = \left(B^T B\right)^{-1} \end{aligned}$$

gilt, dass

$$C = D\left(D^T D\right)^{-1} = D\left(B^T B\right) = \left(B\left(B^T B\right)^{-1}\right)\left(B^T B\right) = B$$

ist. Deshalb folgt $(\Lambda^*)^* = \mathcal{L}(B) = \Lambda$. □

Lemma 3.4.4. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter. Dann ist* $\det(\Lambda^*) = \frac{1}{\det(\Lambda)}$.

Beweis. Sei $B \in \mathbb{R}^{m \times n}$ mit linear unabhängigen Spalten, so dass $\Lambda = \mathcal{L}(B)$ ist. Setze $D := B\left(B^T B\right)^{-1}$. Aus Lemma 3.4.2 folgt $\Lambda^* = \mathcal{L}(B)^* = \mathcal{L}(D)$. Nun ergibt sich wegen $D^T D = \left(B^T B\right)^{-1}$ aus Definition 3.3.9 sowie aus Lemma 3.3.6, dass

$$\begin{aligned} \det(\Lambda^*) &= \mathrm{vol}_n(\mathcal{P}(D)) = \sqrt{\det(D^T D)} = \sqrt{\det\left(\left(B^T B\right)^{-1}\right)} \\ &= \sqrt{\frac{1}{\det(B^T B)}} = \frac{1}{\mathrm{vol}_n(\mathcal{P}(B))} = \frac{1}{\det(\Lambda)} \end{aligned}$$

ist. □

3.5. Sukzessive Minima

In diesem Abschnitt werden für alle $i \in \{1, \ldots, n\}$ kürzeste Mengen von i linear unabhängigen Gittervektoren betrachtet.

Definition 3.5.1. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter mit Rang n. Für $i \in \{1, \ldots, n\}$ ist*

$$\lambda_i(\Lambda) := \inf\left\{ r \;\middle|\; \dim\left(\operatorname{span}\left(\Lambda \cap \overline{\mathcal{B}}_r(0)\right)\right) \geq i \right\}$$

das i-te sukzessive Minimum *von Λ.*

Das folgende Lemma zeigt, dass es in jedem Gitter Λ tatsächlich i linear unabhängige Gittervektoren gibt, deren maximale Länge gleich $\lambda_i(\Lambda)$ ist. Außerdem zeigt der Beweis, dass immer nur ein Gittervektor sukzessive zu einer solchen Menge von linear unabhängigen Gittervektoren hinzugewählt werden muss, um die nächstgrößere Menge von linear unabhängigen Gittervektoren zu erhalten, deren maximale Länge gleich einem sukzessiven Minimum ist.

Lemma 3.5.2. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter mit Rang n. Dann existieren linear unabhängige $v_1, \ldots, v_n \in \Lambda$, so dass $\|v_1\| \leq \|v_2\| \leq \ldots \leq \|v_n\|$ ist und für alle $i \in \{1, \ldots, n\}$ $\lambda_i(\Lambda) = \|v_i\|$ gilt.*

Beweis. Für $i \in \{1, \ldots, n\}$ werden die Vektoren $v_i \in \mathbb{R}^m$ induktiv konstruiert.

- Induktionsanfang ($i = 1$): Angenommen es würde kein kürzester Vektor in $\Lambda \setminus \{0\}$ existieren. Für ein beliebiges $v \in \Lambda \setminus \{0\}$ existierten dann unendlich viele Elemente in $\mathcal{B}_{\|v\|}(0) \cap \Lambda$. Dies kann aber wegen Lemma 3.1.3 und Lemma 3.2.4 nicht sein. Sei also $v_1 \in \Lambda \setminus \{0\}$ solch ein kürzester Gittervektor. Dann ist v_1 linear unabhängig und es gilt $\|v_1\| = \lambda_1(\Lambda)$, weil $\dim\left(\operatorname{span}\left(\Lambda \cap \overline{\mathcal{B}}_{\|v_1\|}(0)\right)\right) \geq 1$ und für alle $r \in (0, \|v_1\|)$ $\dim\left(\operatorname{span}\left(\Lambda \cap \overline{\mathcal{B}}_r(0)\right)\right) = 0$ ist.

- Induktionsschritt ($i \mapsto i + 1$): Sei dafür $i \in \{1, \ldots, n-1\}$, so dass $v_1, \ldots, v_i \in \Lambda$ bereits konstruiert sind. Dann sind $v_1, \ldots, v_i$ linear unabhängig und es gilt $\|v_1\| \leq \|v_2\| \leq \ldots \leq \|v_i\|$. Außerdem ist für alle $j \in \{1, \ldots, i\}$ $\lambda_j(\Lambda) = \|v_j\|$. Angenommen es existierte kein kürzester Vektor in $\Lambda \setminus \operatorname{span}(v_1, \ldots, v_i)$. Für ein beliebiges $v \in \Lambda \setminus \operatorname{span}(v_1, \ldots, v_i)$ existierten dann unendlich viele Elemente in $\mathcal{B}_{\|v\|}(0) \cap \Lambda$. Wie bereits im Induktionsanfang festgestellt wurde, kann dies aber nicht passieren. Wähle nun $v_{i+1} \in \Lambda \setminus \operatorname{span}(v_1, \ldots, v_i)$ mit minimaler Norm $\|v_{i+1}\|$. Dann sind $v_1, \ldots, v_{i+1}$ linear unabhängig.

  Angenommen es gölte $\|v_{i+1}\| < \|v_i\|$. Dann existierte ein minimales $j \in \{1, \ldots, i\}$ mit $\|v_{i+1}\| < \|v_j\|$. Falls $j = 1$ wäre, so gölte $\|v_{i+1}\| < \|v_1\| = \lambda_1(\Lambda)$, was nach der

Wahl von v_1 nicht sein kann. Falls $j > 1$ wäre, so wäre

$$\|v_1\| \leq \ldots \leq \|v_{j-1}\| \leq \|v_{i+1}\| < \|v_j\|$$

und damit $\|\{v_1, \ldots, v_{j-1}, v_{i+1}\}\| = \|v_{i+1}\| < \|v_j\| = \lambda_j(\Lambda)$. Dies kann aber nicht passieren, weil aufgrund der linearen Unabhängigkeit von $v_1, \ldots, v_{j-1}, v_{i+1}$ $\dim\left(\operatorname{span}\left(\Lambda \cap \overline{\mathcal{B}}_{\|v_{i+1}\|}(0)\right)\right) \geq j$ ist. Es gilt also $\|v_1\| \leq \ldots \leq \|v_i\| \leq \|v_{i+1}\|$.

Weil $v_1, \ldots, v_{i+1}$ linear unabhängig sind, ist $\dim\left(\operatorname{span}\left(\Lambda \cap \overline{\mathcal{B}}_{\|v_{i+1}\|}(0)\right)\right) \geq i+1$. Dies impliziert $\|v_{i+1}\| \geq \lambda_{i+1}(\Lambda)$. Angenommen es wäre $\|v_{i+1}\| > \lambda_{i+1}(\Lambda)$. Dann gäbe es linear unabhängige $\tilde{v}_1, \ldots, \tilde{v}_{i+1} \in \Lambda$ mit

$$\lambda_{i+1}(\Lambda) \leq \|\{\tilde{v}_1, \ldots, \tilde{v}_{i+1}\}\| < \|v_{i+1}\|.$$

Weiterhin existierte ein $\tilde{v} \in \{\tilde{v}_1, \ldots, \tilde{v}_{i+1}\}$, so dass $\tilde{v} \in \Lambda \setminus \operatorname{span}(v_1, \ldots, v_i)$ ist. Dann wäre $\|\tilde{v}\| \leq \|\{\tilde{v}_1, \ldots, \tilde{v}_{i+1}\}\| < \|v_{i+1}\|$. Dies stellt einen Widerspruch zur Wahl von v_{i+1} dar. Insgesamt folgt also $\|v_{i+1}\| = \lambda_{i+1}(\Lambda)$.

□

Das letzte Lemma dieses Kapitels, welches bei [Reg04] als Claim 5 in Lecture 8: „Dual Lattices" gefunden werden kann, zeigt einen Zusammenhang zwischen den sukzessiven Minima eines Gitters und denen seines dualen Gitters.

Lemma 3.5.3. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter mit Rang n. Dann ist $\frac{1}{\lambda_1(\Lambda^*)} \leq \lambda_n(\Lambda)$.*

Beweis. Sei $v \in \Lambda^*$ mit $\|v\| = \lambda_1(\Lambda^*)$. Seien ferner $v_1, \ldots, v_n \in \Lambda$ linear unabhängig mit $\|\{v_1, \ldots, v_n\}\| = \lambda_n(\Lambda)$. All diese Vektoren existieren nach Lemma 3.5.2. Dann gibt es ein $i \in \{1, \ldots, n\}$ mit $\langle v_i, v\rangle \in \mathbb{Z} \setminus \{0\}$, da $v \in \operatorname{span}(\Lambda) = \operatorname{span}(v_1, \ldots, v_n)$ ist. Mit der Cauchy-Schwarzschen Ungleichung folgt $\|v_i\|\|v\| \geq |\langle v_i, v\rangle| \geq 1$. Daher gilt $\lambda_n(\Lambda) \geq \|v_i\| \geq \frac{1}{\|v\|} = \frac{1}{\lambda_1(\Lambda^*)}$. □

4. Komplexität von Gitterproblemen

Nachdem nun Gitter und sukzessive Minima eingeführt wurden, stellt sich die Frage, wie gut sich Letztere berechnen lassen. Diese Fragestellung und weitere mit Gittern zusammenhängende Probleme werden in diesem Kapitel behandelt.

4.1. Finden kurzer Gittervektoren

Definition 4.1.1. *Beim* Shortest Vector Problem (SVP) *ist ein* $B \in \mathbb{Z}^{m \times n}$ *mit linear unabhängigen Spalten gegeben. Das Ziel ist es, ein* $v \in \mathcal{L}(B)$ *mit* $\|v\| = \lambda_1(\mathcal{L}(B))$ *zu finden.*

Diese Problemformulierung ist die Suchvariante von *SVP*. Weitere Varianten sind die Optimierungs- sowie die Entscheidungsvariante.

Definition 4.1.2.

- *Beim* Optimierungs-SVP *ist ein* $B \in \mathbb{Z}^{m \times n}$ *mit linear unabhängigen Spalten gegeben. Das Ziel ist es,* $\lambda_1(\mathcal{L}(B))$ *zu finden.*
- *Beim* Entscheidungs-SVP *ist ein* $B \in \mathbb{Z}^{m \times n}$ *mit linear unabhängigen Spalten sowie ein* $r \in \mathbb{Q}_{>0}$ *gegeben. Das Ziel ist es, zu entscheiden, ob* $\lambda_1(\mathcal{L}(B)) \leq r$ *ist.*

1998 zeigte M. Ajtai [Ajt98], dass *Entscheidungs-SVP* unter randomisierten Reduktionen NP-schwer ist. Das bedeutet, dass es eine probabilistische Turingmaschine, also eine Turingmaschine, die ihre Übergänge nach einer Wahrscheinlichkeitsverteilung wählt, gibt, welche jedes Problem in NP in Polynomialzeit auf Instanzen von *Entscheidungs-SVP* reduziert. Dies liefert ein sehr starkes Indiz für die Schwere von *Entscheidungs-SVP*. Zudem haben die besten bekannten Algorithmen für dieses Problem exponentielle Laufzeit. Es ist allerdings ein offenes Problem, dass *Entscheidungs-SVP* unter deterministischen Reduktionen NP-schwer ist.

Wegen der Schwere von *SVP* werden seit vielen Jahren auch die folgenden Approximationsvarianten von *SVP* untersucht, wobei $\gamma \geq 1$ den Approximationsfaktor darstellt.

Definition 4.1.3.

- *Beim* approximativen Shortest Vector Problem SVP_γ *ist ein* $B \in \mathbb{Z}^{m \times n}$ *mit linear unabhängigen Spalten gegeben. Das Ziel ist es, ein* $v \in \mathcal{L}(B) \setminus \{0\}$ *mit* $\|v\| \leq \gamma \lambda_1(\mathcal{L}(B))$ *zu finden.*
- *Beim* Optimierungs-SVP_γ *ist ein* $B \in \mathbb{Z}^{m \times n}$ *mit linear unabhängigen Spalten gegeben. Das Ziel ist es, ein* $d \in [\lambda_1(\mathcal{L}(B)), \gamma\lambda_1(\mathcal{L}(B))]$ *zu finden.*

Die approximative Variante von *Entscheidungs-SVP* ist ein Promise Problem. Solch ein Problem ist ein Paar $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ mit $\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}} \subseteq \{0,1\}^*$ und $\Pi_{\mathrm{YES}} \cap \Pi_{\mathrm{NO}} = \emptyset$, wobei Π_{YES} die Ja-Instanzen des Problems und Π_{NO} die Nein-Instanzen beinhaltet. Ein Algorithmus löst ein Promise Problem, falls er bei Eingabe einer Probleminstanz $I \in \Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$ richtig entscheidet, ob $I \in \Pi_{\mathrm{YES}}$ oder $I \in \Pi_{\mathrm{NO}}$ ist. Ist die Probleminstanz $I \in \{0,1\}^* \setminus (\Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}})$, so ist die Ausgabe des Algorithmus nicht spezifiziert [MG02]. Nun kann die approximative Variante von *Entscheidungs-SVP* formuliert werden.

Definition 4.1.4. *Beim* GapSVP_γ *ist ein* $B \in \mathbb{Z}^{m \times n}$ *mit linear unabhängigen Spalten sowie ein* $r \in \mathbb{Q}_{>0}$ *gegeben.* GapSVP_γ *ist ein Promise Problem, wobei für die Ja-Instanzen* $\lambda_1(\mathcal{L}(B)) \leq r$ *und für die Nein-Instanzen* $\lambda_1(\mathcal{L}(B)) > \gamma r$ *gilt.*

Es lässt sich feststellen, dass die in Definition 4.1.3 und Definition 4.1.4 eingeführten Probleme für $\gamma = 1$ gerade die Probleme aus Definition 4.1.1 und Definition 4.1.2 sind.

Für Promise Probleme kann die Definition der Klasse NP mittels NP-Zertifikaten auf natürliche Weise verallgemeinert werden. Ein Promise Problem ist in NP, falls es für alle Ja-Instanzen ein NP-Zertifikat gibt und falls es für alle Nein-Instanzen kein NP-Zertifikat gibt. Für alle anderen Instanzen kann, aber muss es kein NP-Zertifikat geben.

Auch der Begriff der Reduktion kann auf Promise Probleme erweitert werden. Damit kann gezeigt werden, dass $GapSVP_\gamma$ für jeden konstanten Approximationsfaktor γ unter randomisierten Reduktionen NP-schwer ist [Kho05]. Unter stärkeren Annahmen kann zudem gezeigt werden, dass es keinen Polynomialzeitalgorithmus für $GapSVP_\gamma$ bis zu einem Approximationsfaktor der Größe $2^{(\log n)^{1-\epsilon}}$ gibt, wobei $\epsilon \in \mathbb{R}_{>0}$ beliebig ist [HR07]. Es ist aber noch ein offenes Problem, solche Aussagen unter deterministischen Reduktionen ohne zusätzliche Annahmen zu zeigen. Trotzdem sind diese Resultate ein Indiz dafür, dass *SVP* nicht in polynomieller Zeit mit einem konstanten Faktor oder sogar einem Faktor der Größe $2^{(\log n)^{1-\epsilon}}$ mit $\epsilon \in \mathbb{R}_{>0}$ approximiert werden kann.

Ein sehr ähnliches Problem zu *SVP* ist das *Closest Vector Problem.*

Definition 4.1.5. *Beim* Closest Vector Problem (CVP) *ist ein* $B \in \mathbb{Z}^{m \times n}$ *mit linear unabhängigen Spalten sowie ein* $t \in \mathbb{Z}^m$ *gegeben. Das Ziel ist es, ein* $v \in \mathcal{L}(B)$ *mit* $\|v - t\| = \operatorname{dist}(t, \mathcal{L}(B))$ *zu finden.*

Analog zu den Varianten von *SVP* lassen sich für *CVP* auch eine Optimierungs- und eine Entscheidungsvariante definieren. Für diese Varianten gibt es dann jeweils wieder eine approximative Variante. Als Beispiel wird nun die approximative Variante von *Entscheidungs-CVP* angeführt. Der Approximationsfaktor ist $\gamma \geq 1$.

Definition 4.1.6. *Beim* GapCVP_γ *ist ein* $B \in \mathbb{Z}^{m \times n}$ *mit linear unabhängigen Spalten, ein* $t \in \mathbb{Z}^m$ *sowie ein* $r \in \mathbb{Q}_{>0}$ *gegeben.* GapCVP_γ *ist ein Promise Problem, wobei für die Ja-Instanzen* $\operatorname{dist}(t, \mathcal{L}(B)) \leq r$ *und für die Nein-Instanzen* $\operatorname{dist}(t, \mathcal{L}(B)) > \gamma r$ *gilt.*

Es ist bekannt, dass *Entscheidungs-CVP* NP-vollständig ist. Außerdem ist $GapCVP_\gamma$ NP-schwer für $\gamma \in 2^{O\left(\frac{\log n}{\log \log n}\right)}$. Letzterer Ausdruck enthält jede polylogarithmische Funktion in n, aber keine polynomielle Funktion in n. Bisher erreichen zudem (probabilistische) Polynomialzeitalgorithmen für CVP_γ bzw. SVP_γ im Worst Case nur in n subexponentielle Approximationsfaktoren $\gamma \in 2^{O\left(\frac{n \log \log n}{\log n}\right)}$. Dies gilt auch für $GapCVP_\gamma$ und $GapSVP_\gamma$. Diese Resultate lassen sich in den Kapiteln 2 und 3 von [MG02] finden.

Folgendes Problem kann als Erweiterung von *SVP* angesehen werden.

Definition 4.1.7. *Beim* Shortest Independent Vectors Problem (SIVP) *ist ein* $B \in \mathbb{Z}^{m \times n}$ *mit linear unabhängigen Spalten gegeben. Das Ziel ist es, linear unabhängige* $v_1, \ldots, v_n \in \mathcal{L}(B)$ *mit* $\|\{v_1, \ldots, v_n\}\| = \lambda_n(\mathcal{L}(B))$ *zu finden.*

Auch für dieses Problem lassen sich eine Optimierungs- und eine Entscheidungsvariante sowie approximative Varianten formulieren. In dieser Arbeit wird die approximative Suchvariante noch häufiger auftreten, für welche die besten Polynomialzeitalgorithmen ebenfalls Approximationsfaktoren $\gamma \in 2^{O\left(\frac{n \log \log n}{\log n}\right)}$ erreichen [MG02].

Definition 4.1.8. *Beim* approximativen Shortest Independent Vectors Problem SIVP_γ *ist ein* $B \in \mathbb{Z}^{m \times n}$ *mit linear unabhängigen Spalten gegeben. Das Ziel ist es, linear unabhängige* $v_1, \ldots, v_n \in \mathcal{L}(B)$ *mit* $\|\{v_1, \ldots, v_n\}\| \leq \gamma \lambda_n(\mathcal{L}(B))$ *zu finden.*

4.2. Sampeln von Gittervektoren

Ein Gitterproblem etwas anderer Art ist das *Discrete Gaussian Sampling Problem*. Dabei soll ein Gittervektor nach einer bestimmten Wahrscheinlichkeitsverteilung gesampelt

werden. Diese Verteilung wird genauer im nächsten Kapitel untersucht und hier nur definiert. Zuvor werden aber zwei grundlegende Begriffe eingeführt, die in diesem Abschnitt und dem Rest der Arbeit sehr häufig vorkommen werden.

Definition 4.2.1. *Eine Funktion $f : \mathbb{N} \to \mathbb{R}$ heißt* vernachlässigbar*, falls es für jedes Polynom $p \in \mathbb{R}[X] \setminus \{0\}$ ein $n_0 \in \mathbb{N}$ gibt, so dass für alle $n > n_0$ gilt, dass $|f(n)| < \frac{1}{|p(n)|}$ ist.*

Im weiteren Verlauf dieser Arbeit wird der Ausdruck „bis auf eine in n vernachlässigbare Wahrscheinlichkeit“ verwendet, um eine Wahrscheinlichkeit $1 - \epsilon$ zu bezeichnen, wobei $\epsilon = \epsilon(n) \in [0, 1]$ als Funktion in n vernachlässigbar ist.

Definition 4.2.2.

- *Die* statistische Distanz *zweier Wahrscheinlichkeitsverteilungen $\mathcal{D}$ und $\tilde{\mathcal{D}}$ auf derselben abzählbaren Grundmenge X ist definiert als*

$$\Delta\left(\mathcal{D}, \tilde{\mathcal{D}}\right) := \frac{1}{2} \sum_{x \in X} \left| \mathcal{D}(x) - \tilde{\mathcal{D}}(x) \right|.$$

- *Zwei Familien von Wahrscheinlichkeitsverteilungen $(\mathcal{D}_i)_{i \in \mathbb{N}}$ und $\left(\tilde{\mathcal{D}}_i\right)_{i \in \mathbb{N}}$, wobei für alle $i \in \mathbb{N}$ $\mathcal{D}_i$ und $\tilde{\mathcal{D}}_i$ auf derselben abzählbaren Grundmenge X_i definiert sind, heißen* statistisch nah*, falls $\Delta\left(\mathcal{D}_i, \tilde{\mathcal{D}}_i\right)$ vernachlässigbar in i ist.*

In kurzer Form bezeichne „$\mathcal{D}$ und $\tilde{\mathcal{D}}$ sind statistisch nah (in i)“ den Sachverhalt aus obiger Definition, falls die entsprechenden Familien von Wahrscheinlichkeitsverteilungen klar sind.

Nun wird eine diskrete Gaußsche Verteilung auf einem beliebigen Gitter im $\mathbb{R}^m$ definiert.

Definition 4.2.3. *Sei $c \in \mathbb{R}^m$ sowie $\sigma \in \mathbb{R}_{>0}$. Dann definiere*

$$\begin{aligned} \rho_{\sigma,c} : \mathbb{R}^m &\longrightarrow (0, 1], \\ x &\longmapsto e^{-\pi \frac{\|x-c\|^2}{\sigma^2}}. \end{aligned}$$

Definition 4.2.4. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter, $t, c \in \mathbb{R}^m$ sowie $\sigma \in \mathbb{R}_{>0}$. Dann ist*

$$\begin{aligned} \mathcal{D}_{\Lambda+t,\sigma,c} : (\Lambda + t) &\longrightarrow (0,1)\,, \\ x &\longmapsto \frac{\rho_{\sigma,c}(x)}{\rho_{\sigma,c}(\Lambda + t)} \end{aligned}$$

die Wahrscheinlichkeitsfunktion einer Gaußschen Verteilung auf $\Lambda + t$.

Nach dieser Verteilung sollen nun Gittervektoren gesampelt werden. Die Schwierigkeit des Sampelns hängt von der Größe des Parameters σ ab. Deswegen sei nun $\varphi : \{\Lambda \mid \Lambda \subseteq \mathbb{R}^m \text{ ist ein Gitter}\} \to \mathbb{R}_{\geq 0}$ eine Funktion, die durch $\varphi(\Lambda)$ eine untere Schranke für den Parameter σ beim Sampeln von Vektoren aus dem Gitter Λ angibt.

Definition 4.2.5. *Beim* Discrete Gaussian Sampling Problem DGS_φ *ist ein $B \in \mathbb{Z}^{m \times n}$ mit linear unabhängigen Spalten sowie ein $\sigma > \varphi(\mathcal{L}(B))$ gegeben. Das Ziel ist es, ein $v \in \mathcal{L}(B)$ nach der Verteilung $\mathcal{D}_{\mathcal{L}(B),\sigma,0}$ auszugeben.*

Im Kapitel 6 wird ein Algorithmus vorgestellt, der in Polynomialzeit für ein Gitter $\mathcal{L}(B) \subseteq \mathbb{R}^m$ mit vollem Rang Gittervektoren nach einer Verteilung ausgibt, die statistisch nah zu $\mathcal{D}_{\mathcal{L}(B),\sigma,0}$ (in m) ist, wenn $\sigma \geq 8^m \|B\|$ ist. Die so ausgegebenen Gittervektoren können demnach als zufällig $\mathcal{D}_{\mathcal{L}(B),\sigma,0}$-verteilt betrachtet werden. Außerdem wird dort angegeben, dass sich solche Vektoren auch für deutlich kleinere untere Schranken für σ sampeln lassen. Je kleiner diese Schranken werden, desto schwieriger wird allerdings das Sampeln. So stellt O. Regev im Abschnitt 3.3 in [Reg09] einen Zusammenhang zwischen DGS_φ und $GapSVP_\gamma$ bzw. $SIVP_\gamma$ her, der ein Indiz dafür liefert, dass DGS_φ für kleine $\varphi(\mathcal{L}(B))$ schwer ist. Dieser Zusammenhang wird im nun folgenden Abschnitt genauer erläutert.

4.3. Das *Learning with Errors Problem*

Dieser Abschnitt orientiert sich stark an der Arbeit [Reg09] von O. Regev und beschäftigt sich mit einem Problem, das auf den ersten Blick scheinbar nicht mit Gittern zusammenhängt. Zunächst werden dafür folgende Verteilungen definiert.

Definition 4.3.1. *Sei $q \in \mathbb{N}$ eine Primzahl und $s \in \mathbb{Z}_q^n$.*

- *Ist $\chi : \mathbb{Z}_q \to [0,1]$ eine Wahrscheinlichkeitsfunktion einer Verteilung auf $\mathbb{Z}_q$, so ist $\mathcal{A}_{s,\chi}$ die Wahrscheinlichkeitsverteilung auf $\mathbb{Z}_q^n \times \mathbb{Z}_q$, die dadurch erhalten wird,*

dass ein $a \in \mathbb{Z}_q^n$ zufällig gleichverteilt und ein $e \in \mathbb{Z}_q$ davon unabhängig zufällig χ-verteilt gewählt werden und dann $\left(a, a^T s + e\right)$ ausgegeben wird.

- *Ist $\phi : [0,1) \to [0,1]$ eine Wahrscheinlichkeitsdichte einer Verteilung auf $[0,1)$, so ist $\mathcal{A}_{s,\phi}$ die Wahrscheinlichkeitsverteilung auf $\mathbb{Z}_q^n \times [0,1)$, die dadurch erhalten wird, dass ein $a \in \mathbb{Z}_q^n$ zufällig gleichverteilt und ein $e \in [0,1)$ davon unabhängig zufällig ϕ-verteilt gewählt werden und dann $\left(a, \left(\frac{a^T s}{q} + e\right) \mod 1\right)$ ausgegeben wird, wobei $\frac{a^T s}{q}$ als Element in $\mathbb{Q}$ betrachtet wird.*

- *U bezeichne die diskrete Gleichverteilung auf $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

Mit diesen Verteilungen kann nun das *Learning with Errors Problem* definiert werden. Dafür sei $q \in \mathbb{N}$ eine Primzahl, $\chi : \mathbb{Z}_q \to [0,1]$ eine Wahrscheinlichkeitsfunktion einer Verteilung auf $\mathbb{Z}_q$ und $\phi : [0,1) \to [0,1]$ eine Wahrscheinlichkeitsdichte einer Verteilung auf $[0,1)$.

Definition 4.3.2.

- *Beim* diskreten Learning with Errors Problem $\mathrm{LWE}_{q,\chi}$ *sind beliebig viele Samples von $\mathcal{A}_{s,\chi}$ für ein festes, unbekanntes $s \in \mathbb{Z}_q^n$ gegeben. Ein Algorithmus $\mathcal{A}$ löst $LWE_{q,\chi}$, wenn er für jedes $s \in \mathbb{Z}_q^n$ den Vektor s bis auf eine in n vernachlässigbare Wahrscheinlichkeit über die zufälligen Samples und die zufälligen Wahlen von $\mathcal{A}$ ausgibt.*

- *Beim* kontinuierlichen Learning with Errors Problem $\mathrm{LWE}_{q,\phi}$ *sind beliebig viele Samples von $\mathcal{A}_{s,\phi}$ für ein festes, unbekanntes $s \in \mathbb{Z}_q^n$ gegeben. Ein Algorithmus $\mathcal{A}$ löst $LWE_{q,\phi}$, wenn er für jedes $s \in \mathbb{Z}_q^n$ den Vektor s bis auf eine in n vernachlässigbare Wahrscheinlichkeit über die zufälligen Samples und die zufälligen Wahlen von $\mathcal{A}$ ausgibt.*

Die Schwierigkeit beider soeben definierter Probleme hängt sehr stark von χ bzw. ϕ ab. Würde beim Erstellen der Samples gar kein Fehlerterm e hinzuaddiert werden, so ließen sich die Probleme effizient mit dem Gaußschen Eliminationsverfahren lösen. Wenn χ die diskrete Gleichverteilung auf $\mathbb{Z}_q$ ist, dann lässt sich s gar nicht rekonstruieren. Ist χ eine diskrete Gaußsche Verteilung auf $\mathbb{Z}_q$ um 0, so ist $LWE_{q,\chi}$ schon bei relativ kleiner Standardabweichung schwer. Dies wird nun genauer formuliert.

Dazu wird zunächst eine konkrete Verteilung auf $[0,1)$ definiert.

Definition 4.3.3. *Sei $\alpha \in (0,1)$. Dann ist Ψ_α die Wahrscheinlichkeitsverteilung auf $[0,1)$, die dadurch erhalten wird, dass ein $x \in \mathbb{R}$ zufällig $\mathcal{N}\left(0, \frac{\alpha}{\sqrt{2\pi}}\right)$-verteilt gewählt und $x \mod 1$ ausgegeben wird.*

O. Regev zeigt in Kapitel 3 aus [Reg09], dass für eine Primzahl $q = q(n) \in \mathbb{N}$ und ein $\alpha = \alpha(n) \in (0,1)$ mit $\alpha q > 2\sqrt{n}$ ein Algorithmus mit Polynomialzeit in n, der LWE_{q,Ψ_α} löst, schon einen Quantenalgorithmus mit Polynomialzeit für $DGS_{\varphi(\Lambda)}$ auf n-dimensionalen Gittern mit vollem Rang liefert, wobei $\varphi(\Lambda)$ relativ klein ist. In Abschnitt 3.3 zeigt er dann weiter für Gitter mit vollem Rang, dass für dieses $\varphi(\Lambda)$ ein Polynomialzeitalgorithmus für $DGS_{\varphi(\Lambda)}$ bereits einen Polynomialzeitalgorithmus für $GapSVP_\gamma$ und einen Polynomialzeitalgorithmus für $SIVP_\gamma$ mit $\gamma \in \tilde{O}\left(\frac{n}{\alpha}\right)$ impliziert. Für ein α, so dass $\frac{1}{\alpha}$ polynomiell in n ist, ließe sich somit ein Quantenalgorithmus mit Polynomialzeit für $GapSVP_\gamma$ finden, wobei der Approximationsfaktor γ polynomiell in n ist. In Abschnitt 4.1 wurde aber festgehalten, dass mit Polynomialzeitalgorithmen bisher nur Approximationsfaktoren erreicht werden können, die subexponentiell in n sind. Dies gilt auch für Quantenalgorithmen. Da es eine weit verbreitete Annahme ist, dass auch Quantencomputer in Polynomialzeit keine wesentlich besseren Approximationsfaktoren liefern werden, ist die Reduktion von O. Regev ein Indiz dafür, dass sowohl $DGS_{\varphi(\Lambda)}$ mit diesem relativ kleinen $\varphi(\Lambda)$ als auch LWE_{q,Ψ_α} mit obigen Parametern schwer sind.

Ferner untersucht O. Regev in Kapitel 4 aus [Reg09] Varianten des *Learning with Errors Problem*. Dort zeigt er einen Zusammenhang zwischen $LWE_{q,\chi}$ und $LWE_{q,\phi}$, für den folgende Definition notwendig ist.

Definition 4.3.4. *Sei $q \in \mathbb{N}$ eine Primzahl und $\phi : [0,1) \to [0,1]$ eine Wahrscheinlichkeitsdichte einer Verteilung auf $[0,1)$. Dann ist die* Diskretisierung von ϕ auf $\mathbb{Z}_q$ *die Wahrscheinlichkeitsfunktion $\overline{\phi} : \mathbb{Z}_q \to [0,1]$ der Verteilung auf $\mathbb{Z}_q$, die erhalten wird, indem ein $x \in [0,1)$ zufällig ϕ-verteilt gewählt und dann $\lfloor qx \rceil \mod q$ ausgegeben wird.*

Somit ist $\overline{\Psi}_\alpha$ eine diskrete Gaußsche Verteilung auf $\mathbb{Z}_q$ um 0. Da O. Regev in Lemma 4.3 zeigt, dass für jede Primzahl $q \in \mathbb{N}$ und jede Wahrscheinlichkeitsdichte $\phi : [0,1) \to [0,1]$ gilt, dass ein Polynomialzeitalgorithmus für $LWE_{q,\overline{\phi}}$ schon einen Polynomialzeitalgorithmus für $LWE_{q,\phi}$ liefert, impliziert ein Polynomialzeitalgorithmus für $LWE_{q,\overline{\Psi}_\alpha}$ mit $\alpha \in (0,1)$ und $\alpha q > 2\sqrt{n}$ bereits einen Quantenalgorithmus für $GapSVP_\gamma$ und einen Quantenalgorithmus für $SIVP_\gamma$, wobei beide Algorithmen Polynomialzeit haben und $\gamma \in \tilde{O}\left(\frac{n}{\alpha}\right)$ ist.

O. Regev zeigt aber auch, dass es schon ausreicht, die Verteilung $\mathcal{A}_{s,\chi}$ von der diskreten Gleichverteilung U auf $\mathbb{Z}_q^n \times \mathbb{Z}_q$ zu unterscheiden. Der Begriff des Unterscheiders lässt sich mithilfe von Orakeln definieren.

Definition 4.3.5. *Sei $\mathcal{D}$ eine beliebige Wahrscheinlichkeitsverteilung auf einer Grundmenge X. Ein* Orakel $\mathcal{O}_{\mathcal{D}}$ *gibt bei einer Anfrage ein $x \in X$ nach der Verteilung $\mathcal{D}$ aus.*

Definition 4.3.6.

- *Der* Vorteil *eines Algorithmus $\mathcal{A}$, der nur akzeptieren oder ablehnen kann und Zugriff auf ein Orakel $\mathcal{O}$ hat, beim Unterscheiden zwischen zwei Wahrscheinlichkeitsverteilungen $\mathcal{D}$ und $\tilde{\mathcal{D}}$ auf derselben Grundmenge ist definiert als*

$$\Upsilon_{\mathcal{A}}\left(\mathcal{D}, \tilde{\mathcal{D}}\right) := \frac{1}{2}\left|\Pr\left(\mathcal{A} \text{ akzeptiert } \mid \mathcal{O} = \mathcal{O}_{\mathcal{D}}\right) - \Pr\left(\mathcal{A} \text{ akzeptiert } \mid \mathcal{O} = \mathcal{O}_{\tilde{\mathcal{D}}}\right)\right|,$$

  wobei die Wahrscheinlichkeiten über die zufälligen Wahlen von $\mathcal{A}$ sowie die Ausgaben des Orakels sind.

- *Ein Algorithmus $\mathcal{A}$ mit Zugriff auf ein Orakel $\mathcal{O}$ ist ein* Unterscheider *zwischen zwei Familien von Wahrscheinlichkeitsverteilungen $(\mathcal{D}_i)_{i\in\mathbb{N}}$ und $\left(\tilde{\mathcal{D}}_i\right)_{i\in\mathbb{N}}$, wobei für alle $i \in \mathbb{N}$ $\mathcal{D}_i$ und $\tilde{\mathcal{D}}_i$ auf derselben Grundmenge definiert sind, falls er nur akzeptieren oder ablehnen kann und $\Upsilon_{\mathcal{A}}\left(\mathcal{D}_i, \tilde{\mathcal{D}}_i\right)$ nicht vernachlässigbar in i ist.*

Im Folgenden wird der Begriff des Unterscheiders zwischen zwei Wahrscheinlichkeitsverteilungen auf $\mathbb{Z}_q^n \times \mathbb{Z}_q$ für einen Unterscheider zwischen den entsprechenden, durch n indizierten Familien verwendet.

O. Regev zeigt in Lemma 4.1 sowie im Beweis von Lemma 4.2, dass ein Unterscheider zwischen $\mathcal{A}_{s,\chi}$ und U mit Polynomialzeit in n für eine in n nicht vernachlässigbare Teilmenge aller $s \in \mathbb{Z}_q^n$ bereits einen Algorithmus für $LWE_{q,\chi}$ mit Laufzeit $qp(n)$ liefert, wobei $p(n)$ eine in n polynomielle Funktion ist. Wäre q polynomiell in n, so ließe sich somit ein Polynomialzeitalgorithmus für $LWE_{q,\chi}$ erhalten.

Nun können alle Überlegungen von O. Regev in einem Theorem zusammengefasst werden. Dafür wird zunächst eine Variante des *Learning with Errors Problem* definiert, die in der Form auch im zweiten Teil dieser Arbeit wichtig sein wird. Sei wieder $q \in \mathbb{N}$ eine Primzahl und $\chi : \mathbb{Z}_q \to [0,1]$ eine Wahrscheinlichkeitsfunktion einer Verteilung auf $\mathbb{Z}_q$.

Definition 4.3.7. *Beim* Unterscheidungs-LWE$_{q,\chi}$ *ist ein Orakel* $\mathcal{O}$ *gegeben, welches entweder von der Form* $\mathcal{O}_{\mathcal{A}_{s,\chi}}$ *für ein zufällig gleichverteiltes, unbekanntes* $s \in \mathbb{Z}_q^n$ *oder von der Form* $\mathcal{O}_U$ *ist.*

- *Der* Vorteil *eines Algorithmus* $\mathcal{A}$*, der nur akzeptieren oder ablehnen kann, bei* Unterscheidungs-LWE$_{q,\chi}$ *ist definiert als*

$$\frac{1}{2}\left|\Pr\left(\mathcal{A} \text{ akzeptiert } \mid \mathcal{O} = \mathcal{O}_{\mathcal{A}_{s,\chi}}\right) - \Pr\left(\mathcal{A} \text{ akzeptiert } \mid \mathcal{O} = \mathcal{O}_U\right)\right|,$$

  wobei die Wahrscheinlichkeiten über die zufällige Wahl von s*, die zufälligen Wahlen von* $\mathcal{A}$ *sowie die Ausgaben des Orakels sind.*

- $\mathcal{A}$ löst Unterscheidungs-LWE$_{q,\chi}$*, falls sein Vorteil nicht vernachlässigbar in* n *ist.*

Gibt es nun einen Polynomialzeitalgorithmus, der *Unterscheidungs-LWE*$_{q,\chi}$ löst, so unterscheidet er zwischen $\mathcal{A}_{s,\chi}$ und U für eine in n nicht vernachlässigbare Teilmenge aller $s \in \mathbb{Z}_q^n$. Deshalb gibt es einen Algorithmus für *LWE*$_{q,\chi}$ mit Laufzeit $qp(n)$, wobei $p(n)$ eine in n polynomielle Funktion ist. Wird davon ausgegangen, dass die Eingabegrößen für *GapSVP*$_\gamma$ und *SIVP*$_\gamma$ polynomiell im Gitterrang n sind, und ist $\chi = \overline{\Psi}_\alpha$ für ein $\alpha \in (0,1)$ mit $\alpha q > 2\sqrt{n}$, so gibt es dann zudem einen Quantenalgorithmus für *GapSVP*$_\gamma$ mit Laufzeit $qp_1(n)$ sowie einen Quantenalgorithmus für *SIVP*$_\gamma$ mit Laufzeit $qp_2(n)$, wobei $p_1(n)$ und $p_2(n)$ in n polynomielle Funktionen sind und $\gamma \in \tilde{O}\left(\frac{n}{\alpha}\right)$ ist. Dies wird nun als Theorem formuliert.

Theorem 4.3.8. *Sei* $q = q(n) \in \mathbb{N}$ *eine Primzahl und* $\alpha = \alpha(n) \in (0,1)$ *mit* $\alpha q > 2\sqrt{n}$*. Gibt es dann einen Algorithmus mit Polynomialzeit in* n*, der* Unterscheidungs-LWE$_{q,\overline{\Psi}_\alpha}$ *löst, so gibt es für* n*-dimensionale Gitter mit vollem Rang einen Quantenalgorithmus mit Laufzeit* $qp_1(n)$*, der* GapSVP$_\gamma$ *bei in* n *polynomieller Eingabegröße löst, sowie einen Quantenalgorithmus mit Laufzeit* $qp_2(n)$*, der* SIVP$_\gamma$ *bei in* n *polynomieller Eingabegröße löst, wobei* $p_1(n)$ *und* $p_2(n)$ *polynomielle Funktionen in* n *sind und* $\gamma \in \tilde{O}\left(\frac{n}{\alpha}\right)$ *ist.*

An dieser Stelle sei bemerkt, dass das Theorem für *GapSVP*$_\gamma$ bei in n exponentiellem q auch für klassische Algorithmen statt Quantenalgorithmen gezeigt wurde [Pei09]. Im Jahr 2012 wurde dies auch für allgemeines q gezeigt, aber es liegt dazu noch keine Veröffentlichung vor.

Wie bereits oben festgestellt wurde, liefert dieses Theorem ein Indiz dafür, dass *Unterscheidungs-LWE*$_{q,\overline{\Psi}_\alpha}$ schwer ist, wenn zum Beispiel $\frac{1}{\alpha}$ und q polynomiell in n

sind. Außerdem gibt es – je nach Ausprägung der Parameter n, q und α – bisher nur Algorithmen mit leicht subexponentieller bzw. exponentieller Zeit, die $LWE_{q,\overline{\Psi}_\alpha}$ lösen [Reg10, AG11].

Unabhängig von diesen Resultaten kann zudem gezeigt werden, dass das *Learning with Errors Problem* in seiner Entscheidungsvariante NP-vollständig ist. Dies soll nun am Ende dieses Kapitels bewiesen werden.

4.4. NP-Vollständigkeit des *Learning with Errors Problem*

Für den Rest des Abschnittes sei $q \in \mathbb{N}$ eine Primzahl. Bei $LWE_{q,\chi}$ sind beliebig viele Samples von $\mathcal{A}_{s,\chi}$ gegeben. Werden genau $m \geq n$ Samples erlaubt, so muss das unbekannte $s \in \mathbb{Z}_q^n$ aus $As + e$ rekonstruiert werden, wobei $A \in \mathbb{Z}_q^{m \times n}$ zufällig gleichverteilt gewählt wird und bekannt ist und $e \sim \chi^m$ nicht bekannt ist. Damit s überhaupt rekonstruiert werden kann, muss der additive Fehlervektor $e \in \mathbb{Z}_q^m$ hinreichend klein sein. Wegen der zyklischen Struktur von $\mathbb{Z}_q$ eignet sich eine gewöhnliche p-Norm für die Bestimmung der Länge des Vektors e nicht. Stattdessen definiere für $p \in \mathbb{N}$

$$\begin{aligned} \|\cdot\|_{q,p} : \mathbb{Z}_q^m &\longrightarrow \mathbb{R}_{\geq 0}, \\ e = (e_1, \ldots, e_m)^T &\longmapsto \left(\sum_{i=1}^{m} (\min\{e_i, q - e_i\})^p \right)^{\frac{1}{p}}. \end{aligned}$$

Somit ist beim *Learning with Errors Problem* bei gegebenem $A \in \mathbb{Z}_q^{m \times n}$ und $y \in \mathbb{Z}_q^m$ von der Form $y = As + e$ ein $s' \in \mathbb{Z}_q^n$ gesucht, so dass $\|y - As'\|_{q,p}$ für ein festes $p \in \mathbb{N}$ minimal ist. Das zugehörige Entscheidungsproblem kann als Sprache wie folgt formuliert werden.

Definition 4.4.1.

$$\mathrm{LWE}_{q,p} := \left\{ \langle A, y, w \rangle \;\middle|\; \begin{array}{l} A \in \mathbb{Z}_q^{m \times n}, y \in \mathbb{Z}_q^m, w \in \mathbb{N}_0 \text{ und es existiert ein } s \in \mathbb{Z}_q^n \\ \text{mit } \|y - As\|_{q,p}^p \leq w. \end{array} \right\}.$$

Um die NP-Vollständigkeit von $LWE_{q,p}$ zu zeigen, wird folgende Sprache betrachtet, deren NP-Vollständigkeit bereits 1972 von R. M. Karp gezeigt wurde [Kar72].

Definition 4.4.2.

$$\text{3DimensionalMatching} := \left\{ \langle T, U \rangle \;\middle|\; \begin{array}{l} T = \{t_1, \ldots, t_n\}, U \subseteq T^3 \textit{ und es existiert ein} \\ W \subseteq U \textit{ mit } |W| = |T| \textit{ und keine zwei} \\ \textit{Elemente von } W \textit{ stimmen in einer ihrer} \\ \textit{Komponenten überein.} \end{array} \right\}.$$

Ähnlich wie in [BMVT78] kann damit gezeigt werden, dass folgende Sprache ebenfalls NP-vollständig ist.

Definition 4.4.3.

$$\text{CosetWeigths}_{q,p} := \left\{ \langle A, y, w \rangle \;\middle|\; \begin{array}{l} A \in \mathbb{Z}_q^{m \times n}, y \in \mathbb{Z}_q^m, w \in \mathbb{N}_0 \textit{ und es existiert ein} \\ x \in \mathbb{Z}_q^n \textit{ mit } Ax = y \textit{ und } \|x\|_{q,p}^p \leq w. \end{array} \right\}.$$

Lemma 4.4.4. $\text{CosetWeigths}_{q,p}$ *ist NP-vollständig.*

Beweis. *CosetWeigths*$_{q,p}$ ist offensichtlich in NP, da bei gegebenem $A \in \mathbb{Z}_q^{m \times n}$, $y \in \mathbb{Z}_q^m$, $w \in \mathbb{N}_0$ und $x \in \mathbb{Z}_q^n$ in polynomieller Zeit verifiziert werden kann, ob $Ax = y$ und $\|x\|_{q,p}^p \leq w$ ist. Zu zeigen ist daher noch, dass *3DimensionalMatching* polynomiell auf *CosetWeigths*$_{q,p}$ reduzierbar ist. Dafür seien $T = \{t_1, \ldots, t_n\}$ und $U \subseteq T^3$ gegeben. Dann sei $A \in \mathbb{Z}_q^{3|T| \times |U|}$ die Matrix, die U codiert, indem jede Spalte von A zu einem Element aus U gehört und innerhalb der Zeilen $1, \ldots, |T|$ sowie $|T|+1, \ldots, 2|T|$ und auch $2|T|+1, \ldots, 3|T|$ jeweils genau eine 1 und sonst nur 0 enthält. Damit gibt die erste 1 an, welches Element aus T an erster Stelle in dem Element aus U vorkommt. Analog geben die zweite bzw. dritte 1 an, welches Element aus T an zweiter bzw. dritter Stelle in dem Element aus U vorkommt. Ferner seien $y := (1, \ldots, 1)^T \in \mathbb{Z}_q^{3|T|}$ sowie $w := |T|$. Da A, y und w effizient berechenbar sind, ist nur noch zu zeigen, dass $\langle T, U \rangle \in$ *3Dimensional-Matching* genau dann ist, wenn $\langle A, y, w \rangle \in$ *CosetWeigths*$_{q,p}$ ist.

- Behauptung 1. Wenn $\langle T, U \rangle \in$ *3DimensionalMatching* ist, dann ist $\langle A, y, w \rangle \in$ *CosetWeigths*$_{q,p}$.

  Beweis. Da $\langle T, U \rangle \in$ *3DimensionalMatching* ist, gibt es ein $W \subseteq U$, so dass $|W| = |T|$ ist und keine zwei Elemente von W in einer ihrer Komponenten übereinstimmen. Sei nun $x \in \mathbb{Z}_q^{|U|}$, wobei für alle $i \in \{1, \ldots, |U|\}$ gilt, dass $x_i = 1$ genau dann ist, wenn W das i-te Element aus U enthält; ansonsten sei $x_i = 0$.

Nach Definition von A und y gilt dann $Ax = (1, \ldots, 1)^T = y$. Außerdem ist $\|x\|_{q,p}^p = |W|1^p = w$. Damit ist $\langle A, y, w\rangle \in \mathit{CosetWeigths}_{q,p}$. □

- Behauptung 2. Wenn $\langle A, y, w\rangle \in \mathit{CosetWeigths}_{q,p}$ ist, dann ist $\langle T, U\rangle \in \mathit{3DimensionalMatching}$.

  Beweis. Weil $\langle A, y, w\rangle \in \mathit{CosetWeigths}_{q,p}$ ist, existiert ein $x \in \mathbb{Z}_q^{|U|}$ mit $Ax = y = (1, \ldots, 1)^T$ und $\|x\|_{q,p}^p \leq w = |T|$. Daher muss es für alle $i \in \{1, \ldots, 3|T|\}$ ein $j_i \in \{1, \ldots, |U|\}$ geben, so dass $a_{i,j_i} = 1$ und $x_{j_i} \neq 0$ ist. Da aber jede Spalte von A in den ersten $|T|$ Zeilen genau eine 1 enthält, muss für alle $i, i' \in \{1, \ldots, |T|\}$ mit $i \neq i'$ gelten, dass $j_i \neq j_{i'}$ ist. Daher enthält x mindestens $|T|$ Einträge ungleich 0. Da aber $\|x\|_{q,p}^p \leq |T|$ ist, enthält x höchstens $|T|$ Einträge ungleich 0. Somit enthält x genau $|T|$ Einträge ungleich 0. Für $W \subseteq U$, wobei W das j-te Element von U genau dann enthält, wenn $x_j \neq 0$ ist, gilt dann $|W| = |T|$.

  Außerdem stimmen keine zwei Elemente von W in einer ihrer Komponenten überein. Denn angenommen es gäbe $u_j, u_l \in W$ mit $j \neq l$, so dass u_j und u_l in einer ihrer Komponenten übereinstimmen, wobei ohne Beschränkung der Allgemeinheit dies die erste Komponente sei, dann gölte nach Definition von W, dass $x_j \neq 0$ und $x_l \neq 0$ ist. Ferner gäbe es genau ein $i \in \{1, \ldots, |T|\}$ mit $a_{i,j} = 1$ und $a_{i,l} = 1$. Deshalb gölte für alle $k \in \{1, \ldots, |T|\} \setminus \{i\}$, dass sowohl $a_{k,j} = 0$ als auch $a_{k,l} = 0$ ist. Wie aber bereits oben festgestellt wurde, gäbe es für alle diese k ein $j_k \in \{1, \ldots, |U|\}$ mit $a_{k,j_k} = 1$ und $x_{j_k} \neq 0$. Für alle $k \in \{1, \ldots, |T|\} \setminus \{i\}$ wäre somit $j \neq j_k$ und $l \neq j_k$. Da zudem alle j_k paarweise verschieden wären und $x_{j_k} \neq 0$ sowie $x_j \neq 0$ und $x_l \neq 0$ gölten, hätte x mindestens $|T| + 1$ Einträge ungleich 0. Dies ist aber ein Widerspruch, da x genau $|T|$ Einträge ungleich 0 besitzt. Damit stimmen keine zwei Elemente von W in einer ihrer Komponenten überein und somit ist $\langle T, U\rangle \in \mathit{3DimensionalMatching}$. □

□

In [BMVT78] wurde zudem erwähnt, dass $\mathit{CosetWeigths}_{q,p}$ NP-vollständig bleibt, auch wenn für die Matrix A vorausgesetzt wird, dass sie vollen Zeilenrang hat. Um dies zu zeigen, wird zunächst eine Abänderung der Sprache $\mathit{CosetWeigths}_{q,p}$ definiert.

Definition 4.4.5.

$$\text{CWSurjective}_{q,p} := \left\{ \langle A, y, w\rangle \;\middle|\; \begin{array}{l} A \in \mathbb{Z}_q^{m\times n} \textit{ mit vollem Zeilenrang, } y \in \mathbb{Z}_q^m, w \in \mathbb{N}_0 \\ \textit{und es existiert ein } x \in \mathbb{Z}_q^n \textit{ mit } Ax = y \textit{ und} \\ \|x\|_{q,p}^p \leq w. \end{array} \right\}.$$

Lemma 4.4.6. $\text{CWSurjective}_{q,p}$ *ist NP-vollständig.*

Beweis. Bei gegebenem $A \in \mathbb{Z}_q^{m\times n}$, $y \in \mathbb{Z}_q^m$, $w \in \mathbb{N}_0$ und $x \in \mathbb{Z}_q^n$ kann in polynomieller Zeit verifiziert werden, ob $Ax = y$ und $\|x\|_{q,p}^p \leq w$ ist sowie A vollen Zeilenrang hat. Deswegen ist $CWSurjective_{q,p}$ in NP. Jetzt ist zu zeigen, dass $CosetWeights_{q,p}$ polynomiell auf $CWSurjective_{q,p}$ reduzierbar ist. Dafür seien $A \in \mathbb{Z}_q^{m\times n}, y \in \mathbb{Z}_q^m$ und $w \in \mathbb{N}_0$ gegeben. Falls es kein $t \in \mathbb{Z}_q^n$ gibt mit $At = y$, so setze $A' := I_2 \in \mathbb{Z}_q^{2\times 2}, y' := (1,1)^T \in \mathbb{Z}_q^2$ und $w' := 1$. Falls es ein $t \in \mathbb{Z}_q^n$ gibt mit $At = y$ und $A \neq 0$, so sei $w' := w$ und A' wird durch das Streichen linear abhängiger Zeilen aus A und y' durch Streichen der gleichen Zeilen aus y erhalten. Falls $A = 0$ und $y = 0$ ist, so setze $A' := I_2 \in \mathbb{Z}_q^{2\times 2}, y' := (1,0)^T \in \mathbb{Z}_q^2$ und $w' := 1$. Da effizient bestimmt werden kann, ob es ein $t \in \mathbb{Z}_q^n$ mit $At = y$ gibt und in jedem Fall A', y' und w' effizient berechenbar sind, ist nun noch zu zeigen, dass $\langle A, y, w\rangle \in CosetWeights_{q,p}$ genau dann ist, wenn $\langle A', y', w'\rangle \in CWSurjective_{q,p}$ ist.

- Behauptung 1. Wenn $\langle A, y, w\rangle \in CosetWeights_{q,p}$ ist, dann ist $\langle A', y', w'\rangle \in CWSurjective_{q,p}$.

  Beweis. Da $\langle A, y, w\rangle \in CosetWeights_{q,p}$ ist, gibt es ein $x \in \mathbb{Z}_q^n$ mit $Ax = y$ und $\|x\|_{q,p}^p \leq w$.

  Falls $A = 0$ ist, ist auch $y = 0$ und deshalb sind $A' = I_2, y' = (1,0)^T$ und $w' = 1$. Somit hat A' vollen Zeilenrang und für $x' := y'$ gilt $A'x' = y'$ sowie $\|x'\|_{q,p}^p = 1 = w'$. Daher ist $\langle A', y', w'\rangle \in CWSurjective_{q,p}$.

  Falls $A \neq 0$ ist, so werden aus A linear abhängige Zeilen gestrichen, um A' zu erhalten. A' hat also vollen Zeilenrang. Nach Definition von A' und y' gilt zudem $A'x = y'$. Daher ist $\langle A', y', w'\rangle = \langle A', y', w\rangle \in CWSurjective_{q,p}$. □

- Behauptung 2. Wenn $\langle A, y, w\rangle \notin CosetWeights_{q,p}$ ist, dann ist $\langle A', y', w'\rangle \notin CWSurjective_{q,p}$.

Beweis. Falls es kein $t \in \mathbb{Z}_q^n$ mit $At = y$ gibt, so gilt $A' = I_2, y' = (1,1)^T$ und $w' = 1$. Da $A'x = y'$ nur für $x = y'$ gilt und $\|y'\|_{q,p}^p = 2 > w'$ ist, ist $\langle A', y', w' \rangle \notin \mathit{CWSurjective}_{q,p}$.

Falls es ein $t \in \mathbb{Z}_q^n$ mit $At = y$ gibt, so muss für alle $x \in \mathbb{Z}_q^n$ mit $Ax = y$ gelten, dass $\|x\|_{q,p}^p > w$ ist. Weil aber A' durch das Streichen linear abhängiger Zeilen aus A und y' durch das Streichen der gleichen Zeilen aus y entsteht, gilt $A'x = y'$ genau dann, wenn $Ax = y$ ist. Deshalb gilt für alle $x \in \mathbb{Z}_q^n$ mit $A'x = y'$, dass $\|x\|_{q,p}^p > w = w'$ ist, und damit ist $\langle A', y', w' \rangle \notin \mathit{CWSurjective}_{q,p}$. □

□

Nun kann gezeigt werden, dass die Entscheidungsvariante vom *Learning with Errors Problem* NP-vollständig ist.

Lemma 4.4.7. $\mathrm{LWE}_{q,p}$ *ist NP-vollständig.*

Beweis. $\mathit{LWE}_{q,p}$ ist in NP, weil bei gegebenem $A \in \mathbb{Z}_q^{m \times n}, y \in \mathbb{Z}_q^m, w \in \mathbb{N}_0$ und $s \in \mathbb{Z}_q^n$ in polynomieller Zeit verifiziert werden kann, ob $\|y - As\|_{q,p}^p \leq w$ ist. Daher muss nun gezeigt werden, dass $\mathit{CWSurjective}_{q,p}$ polynomiell auf $\mathit{LWE}_{q,p}$ reduzierbar ist. Dafür seien $A \in \mathbb{Z}_q^{m \times n}$ mit vollem Zeilenrang, $y \in \mathbb{Z}_q^m$ sowie $w \in \mathbb{N}_0$ gegeben. Wegen des vollen Zeilenrangs gibt es ein $y' \in \mathbb{Z}_q^n$ mit $y = Ay'$. Dieses lässt sich außerdem effizient berechnen. Falls $n > m$ ist, so sei $A' \in \mathbb{Z}_q^{n \times (n-m)}$ eine Matrix, deren Spalten eine Basis vom Kern von A bilden. Solch eine Basis lässt sich ebenfalls effizient finden. Falls $n = m$ ist, so sei $A' := 0 \in \mathbb{Z}_q^{n \times 1}$. Jetzt wird gezeigt, dass $\langle A, y, w \rangle \in \mathit{CWSurjective}_{q,p}$ genau dann ist, wenn $\langle A', y', w \rangle \in \mathit{LWE}_{q,p}$ ist.

- Behauptung 1. Wenn $\langle A, y, w \rangle \in \mathit{CWSurjective}_{q,p}$ ist, dann ist $\langle A', y', w \rangle \in \mathit{LWE}_{q,p}$.

  Beweis. Da $\langle A, y, w \rangle \in \mathit{CWSurjective}_{q,p}$ ist, existiert ein $x \in \mathbb{Z}_q^n$ mit $Ax = y$ und $\|x\|_{q,p}^p \leq w$. Daher ist $0 = y - Ax = Ay' - Ax = A(y' - x)$.

  Falls $n > m$ ist, so existiert ein $s \in \mathbb{Z}_q^{n-m}$, so dass $A's = y' - x$ ist. Dann gilt $\|y' - A's\|_{q,p}^p = \|x\|_{q,p}^p \leq w$ und daher ist $\langle A', y', w \rangle \in \mathit{LWE}_{q,p}$.

  Falls $n = m$ ist, so gilt $y' = x$. Für jedes $s \in \mathbb{Z}_q$ gilt deswegen $\|y' - A's\|_{q,p}^p = \|y'\|_{q,p}^p = \|x\|_{q,p}^p \leq w$. Damit ist $\langle A', y', w \rangle \in \mathit{LWE}_{q,p}$. □

- Behauptung 2. Wenn $\langle A', y', w \rangle \in \mathit{LWE}_{q,p}$ ist, dann ist $\langle A, y, w \rangle \in \mathit{CWSurjective}_{q,p}$.

Beweis. Falls $n > m$ ist, so gibt es wegen $\langle A', y', w\rangle \in LWE_{q,p}$ ein $s \in \mathbb{Z}_q^{n-m}$, so dass $\|y' - A's\|_{q,p}^p \leq w$ ist. Dann gilt $A\,(y' - A's) = Ay' - AA's = y - 0 = y$. Also ist $\langle A, y, w\rangle \in CWSurjective_{q,p}$.

Falls $n = m$ ist, so gibt es ebenfalls wegen $\langle A', y', w\rangle \in LWE_{q,p}$ ein $s \in \mathbb{Z}_q$, so dass $\|y' - A's\|_{q,p}^p \leq w$ ist. Wegen $A' = 0$ ist daher $\|y'\|_{q,p}^p \leq w$ und aus $Ay' = y$ folgt, dass $\langle A, y, w\rangle \in CWSurjective_{q,p}$ ist. □

□

5. Diskrete Gaußsche Verteilung auf Gittern

In diesem Kapitel wird die im letzten Kapitel definierte diskrete Gaußsche Verteilung auf Gittern untersucht. Zudem werden einige damit verwandte Begriffe eingeführt.

5.1. Gaußsche Verteilungen

Mithilfe der in Definition 4.2.3 eingeführten Funktion lässt sich sowohl eine kontinuierliche Wahrscheinlichkeitsverteilung auf $\mathbb{R}^m$ als auch eine diskrete Wahrscheinlichkeitsverteilung auf einem Gitter im $\mathbb{R}^m$ definieren. Letztere Verteilung wurde in Definition 4.2.4 angegeben. Um einzusehen, dass diese Definition sinnvoll ist und wie die kontinuierliche Verteilung definiert werden kann, wird zunächst $\int\limits_{\mathbb{R}^m} \rho_{\sigma,c}(x)\, dx = \sigma^m$ nachgerechnet. Dabei werden zwei Aussagen der Integralrechnung verwendet, die im Folgenden ohne Beweis wiederholt werden und zum Beispiel in Abschnitt 8.5 aus [Kö04] nachgelesen werden können.

Lemma 5.1.1. *Es gilt* $\int\limits_{-\infty}^{\infty} e^{-u^2}\, du = \sqrt{\pi}$.

Satz 5.1.2. *(Satz von Fubini-Tonelli)*
Sei $f : \mathbb{R}^{m+n} \to \mathbb{C}$ *stetig. Wenn* $\int\limits_{\mathbb{R}^m} \left(\int\limits_{\mathbb{R}^n} |f(x,y)|\, dy \right) dx < \infty$ *ist, so gilt*

$$\int\limits_{\mathbb{R}^{m+n}} f(z)\, dz = \int\limits_{\mathbb{R}^m} \left(\int\limits_{\mathbb{R}^n} f(x,y)\, dy \right) dx = \int\limits_{\mathbb{R}^n} \left(\int\limits_{\mathbb{R}^m} f(x,y)\, dx \right) dy.$$

Nun kann das Gewünschte nachgerechnet werden.

Lemma 5.1.3. *Sei* $c \in \mathbb{R}^m$ *sowie* $\sigma \in \mathbb{R}_{>0}$. *Dann ist* $\int\limits_{\mathbb{R}^m} \rho_{\sigma,c}(x)\, dx = \sigma^m$.

Beweis. Die Aussage wird mit Induktion nach m bewiesen. Deswegen sei für diesen Beweis die Funktion aus Definition 4.2.3 mit $\rho_{\sigma,c}^{(m)}$ bezeichnet, um die Dimension des Definitionsbereichs zu verdeutlichen.

- Induktionsanfang ($m = 1$): Aus Lemma 5.1.1 ergibt sich mittels mehrfacher Substitution

$$\begin{aligned}\int_{\mathbb{R}} \rho_{\sigma,c}^{(1)}(x)\, dx &= \int_{-\infty}^{\infty} e^{-\pi\frac{(x-c)^2}{\sigma^2}}\, dx = \int_{-\infty}^{\infty} e^{-\pi\frac{y^2}{\sigma^2}}\, dy \\ &= \int_{-\infty}^{\infty} e^{-\left(\frac{\sqrt{\pi}y}{\sigma}\right)^2}\, dy = \int_{-\infty}^{\infty} e^{-u^2}\frac{\sigma}{\sqrt{\pi}}\, du \\ &= \frac{\sigma}{\sqrt{\pi}}\int_{-\infty}^{\infty} e^{-u^2}\, du = \frac{\sigma}{\sqrt{\pi}}\sqrt{\pi} = \sigma.\end{aligned}$$

- Induktionsschritt ($m \mapsto m+1$): Es sei $c = (c_1, \ldots, c_{m+1})^T$. Dann definiere $\tilde{c} := (c_1, \ldots, c_m)^T$. Für jedes $x = (x_1, \ldots, x_{m+1})^T \in \mathbb{R}^{m+1}$ sei analog $\tilde{x} := (x_1, \ldots, x_m)^T$. Da $\rho_{\sigma,c}^{(m+1)} : \mathbb{R}^{m+1} \to \mathbb{R}$ stetig ist sowie für alle $x \in \mathbb{R}^{m+1}$ $\left|\rho_{\sigma,c}^{(m+1)}(x)\right| = \rho_{\sigma,c}^{(m+1)}(x)$ gilt, folgt aus dem Satz von Fubini-Tonelli und aus der Induktionsvoraussetzung, dass

$$\begin{aligned}\int_{\mathbb{R}^{m+1}} \rho_{\sigma,c}^{(m+1)}(x)\, dx &= \int_{\mathbb{R}^{m+1}} e^{-\pi\frac{\|x-c\|^2}{\sigma^2}}\, dx \\ &= \int_{\mathbb{R}^{m+1}} e^{-\pi\frac{(x_1-c_1)^2}{\sigma^2}} \cdots e^{-\pi\frac{(x_{m+1}-c_{m+1})^2}{\sigma^2}}\, dx \\ &= \int_{\mathbb{R}^{m+1}} e^{-\pi\frac{\|\tilde{x}-\tilde{c}\|^2}{\sigma^2}} e^{-\pi\frac{(x_{m+1}-c_{m+1})^2}{\sigma^2}}\, dx \\ &= \int_{\mathbb{R}^{m+1}} \rho_{\sigma,\tilde{c}}^{(m)}(\tilde{x})\, \rho_{\sigma,c_{m+1}}^{(1)}(x_{m+1})\, dx \\ &= \int_{\mathbb{R}^m} \left(\int_{\mathbb{R}} \rho_{\sigma,\tilde{c}}^{(m)}(\tilde{x})\, \rho_{\sigma,c_{m+1}}^{(1)}(x_{m+1})\, dx_{m+1}\right) d\tilde{x} \\ &= \int_{\mathbb{R}^m} \rho_{\sigma,\tilde{c}}^{(m)}(\tilde{x}) \left(\int_{\mathbb{R}} \rho_{\sigma,c_{m+1}}^{(1)}(x_{m+1})\, dx_{m+1}\right) d\tilde{x} \\ &= \int_{\mathbb{R}^m} \rho_{\sigma,\tilde{c}}^{(m)}(\tilde{x})\, \sigma\, d\tilde{x} = \sigma \int_{\mathbb{R}^m} \rho_{\sigma,\tilde{c}}^{(m)}(\tilde{x})\, d\tilde{x} = \sigma\sigma^m = \sigma^{m+1}\end{aligned}$$

ist.

□

Mit diesem Lemma kann analog zu Definition 4.2.4 folgende Gaußsche Verteilung auf $\mathbb{R}^m$ definiert werden.

Definition 5.1.4. *Sei $c \in \mathbb{R}^m$ sowie $\sigma \in \mathbb{R}_{>0}$. Dann ist*

$$\begin{aligned} \mathcal{D}_{\sigma,c} : \mathbb{R}^m &\longrightarrow \mathbb{R}_{>0}, \\ x &\longmapsto \frac{\rho_{\sigma,c}(x)}{\sigma^m} \end{aligned}$$

die Wahrscheinlichkeitsdichte einer Gaußschen Verteilung auf $\mathbb{R}^m$.

5.2. Fourier-Transformation

Als Nächstes wird die Fourier-Transformation von $\rho_{\sigma,c}$ bestimmt. Solch eine Transformation kann wie folgt definiert werden.

Definition 5.2.1. *Sei $f : \mathbb{R}^m \to \mathbb{R}$ messbar und es gelte $\int\limits_{\mathbb{R}^m} |f(x)|\, dx < \infty$. Dann ist*

$$\begin{aligned} \hat{f} : \mathbb{R}^m &\longrightarrow \mathbb{C}, \\ y &\longmapsto \int\limits_{\mathbb{R}^m} f(x)\, e^{-2\pi i \langle x,y \rangle}\, dx \end{aligned}$$

die Fourier-Transformation *von f.*

Zur Bestimmung der Fourier-Transformation von $\rho_{\sigma,c}$ wird ein Spezialfall des cauchyschen Integralsatzes sowie die sogenannte Standardabschätzung benötigt. Beide Aussagen können zum Beispiel in Kapitel 2 aus [Fri08] gefunden werden.

Satz 5.2.2. *Sei $f : \mathbb{C} \to \mathbb{C}$ holomorph, $[a,b] \subseteq \mathbb{R}$ ein kompaktes Intervall und $\gamma : [a,b] \to \mathbb{C}$ eine stetige und stückweise stetig differenzierbare Abbildung mit $\gamma(a) = \gamma(b)$. Dann gilt*

$$\int\limits_{\gamma} f(z)\, dz := \int\limits_{a}^{b} f(\gamma(t))\, \gamma'(t)\, dt = 0.$$

Lemma 5.2.3. *(Standardabschätzung)*
Sei $f : \mathbb{C} \to \mathbb{C}$ stetig, $[a,b] \subseteq \mathbb{R}$ ein kompaktes Intervall und $\gamma : [a,b] \to \mathbb{C}$ eine stetige und stückweise stetig differenzierbare Abbildung. Falls es ein $M \in \mathbb{R}_{>0}$ mit $|f(z)| \leq M$ für alle $z \in \gamma([a,b])$ gibt, so gilt

$$\left| \int_\gamma f(z)\, dz \right| \leq M \int_a^b |\gamma'(t)|\, dt.$$

Damit kann nun die Fourier-Transformation von $\rho_{\sigma,c}$ bestimmt werden.

Lemma 5.2.4. *Sei $c \in \mathbb{R}^m$ sowie $\sigma \in \mathbb{R}_{>0}$. Dann gilt für alle $y \in \mathbb{R}^m$*

$$\hat{\rho}_{\sigma,c}(y) = \sigma^m \rho_{\frac{1}{\sigma},0}(y)\, e^{-2\pi i \langle c,y \rangle}.$$

Beweis. Stelle zunächst fest, dass $\rho_{\sigma,c}$ die Bedingungen aus Definition 5.2.1 erfüllt. Dies folgt aus Lemma 5.1.3, weil $\rho_{\sigma,c}$ als stetige Funktion messbar ist und für alle $x \in \mathbb{R}^m$ $|\rho_{\sigma,c}(x)| = \rho_{\sigma,c}(x)$ gilt. Die Aussage wird nun mit Induktion nach m gezeigt. Deshalb sei die Funktion aus Definition 4.2.3 wieder mit $\rho^{(m)}_{\sigma,c}$ bezeichnet. Weiter sei $y \in \mathbb{R}^m$.

- Induktionsanfang ($m = 1$): Mit einer Substitution ergibt sich

$$\begin{aligned}
\hat{\rho}^{(1)}_{\sigma,c}(y) &= \int_{-\infty}^{\infty} e^{-\pi \frac{(x-c)^2}{\sigma^2}} e^{-2\pi i x y}\, dx = \int_{-\infty}^{\infty} e^{-\pi \frac{v^2}{\sigma^2}} e^{-2\pi i (v+c) y}\, dv \\
&= e^{-2\pi i c y} \int_{-\infty}^{\infty} e^{-\pi \frac{v^2}{\sigma^2}} e^{-2\pi i v y}\, dv = e^{-2\pi i c y} \int_{-\infty}^{\infty} e^{-\pi \left(\frac{v^2}{\sigma^2} + 2 i v y \right)}\, dv \\
&= e^{-2\pi i c y} \int_{-\infty}^{\infty} e^{-\pi \left(\frac{v}{\sigma} + i y \sigma \right)^2} e^{-\pi y^2 \sigma^2}\, dv \\
&= e^{-2\pi i c y} \rho^{(1)}_{\frac{1}{\sigma},0}(y) \int_{-\infty}^{\infty} e^{-\pi \left(\frac{v}{\sigma} + i y \sigma \right)^2}\, dv.
\end{aligned}$$

  Dabei gilt für ein kompaktes Intervall $[a,b] \subseteq \mathbb{R}$, dass

$$\int_a^b e^{-\pi \left(\frac{v}{\sigma} + i y \sigma \right)^2}\, dv = \sigma \int_{\gamma_{[a,b]}} e^{-\pi t^2}\, dt$$

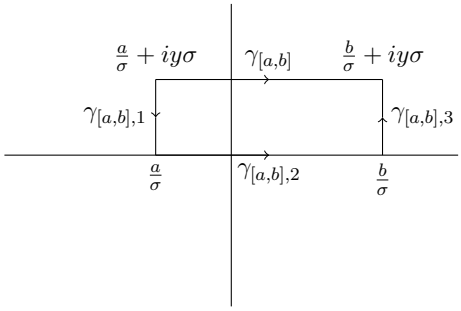
  für $\gamma_{[a,b]} : [a,b] \to \mathbb{C}, t \mapsto \frac{t}{\sigma} + i y \sigma$ ist. Weil $f : \mathbb{C} \to \mathbb{C}, t \mapsto e^{-\pi t^2}$ holomorph ist,

ergibt sich im Fall $y > 0$ als direkte Folgerung von Satz 5.2.2

$$\int\limits_{\gamma_{[a,b]}} e^{-\pi t^2}\,dt = \int\limits_{\gamma_{[a,b],1}} e^{-\pi t^2}\,dt + \int\limits_{\gamma_{[a,b],2}} e^{-\pi t^2}\,dt + \int\limits_{\gamma_{[a,b],3}} e^{-\pi t^2}\,dt$$

mit

$$\begin{aligned}
\gamma_{[a,b],1} : [0, y\sigma] &\longrightarrow \mathbb{C}, \\
t &\longmapsto \frac{a}{\sigma} + i\,(y\sigma - t)\,, \\
\gamma_{[a,b],2} : \left[\frac{a}{\sigma}, \frac{b}{\sigma}\right] &\longrightarrow \mathbb{C}, \\
t &\longmapsto t, \\
\gamma_{[a,b],3} : [0, y\sigma] &\longrightarrow \mathbb{C}, \\
t &\longmapsto \frac{b}{\sigma} + it.
\end{aligned}$$



Die Standardabschätzung liefert nun

$$0 \leq \lim_{b\to\infty} \left| \int\limits_{\gamma_{[a,b],3}} e^{-\pi t^2}\,dt \right| \leq \lim_{b\to\infty} e^{-\pi \frac{b^2}{\sigma^2}} e^{\pi y^2 \sigma^2} \cdot y\sigma = 0.$$

Deswegen gilt $\lim\limits_{b\to\infty} \int\limits_{\gamma_{[a,b],3}} e^{-\pi t^2}\,dt = 0$ und mit einer analogen Abschätzung ebenfalls $\lim\limits_{a\to-\infty} \int\limits_{\gamma_{[a,b],1}} e^{-\pi t^2}\,dt = 0$. Deshalb ist

$$\begin{aligned}
\int\limits_{-\infty}^{\infty} e^{-\pi\left(\frac{v}{\sigma}+iy\sigma\right)^2}\,dv &= \lim_{a\to-\infty} \int\limits_{a}^{0} e^{-\pi\left(\frac{v}{\sigma}+iy\sigma\right)^2}\,dv + \lim_{b\to\infty} \int\limits_{0}^{b} e^{-\pi\left(\frac{v}{\sigma}+iy\sigma\right)^2}\,dv \\
&= \lim_{a\to-\infty} \sigma \int\limits_{\gamma_{[a,0]}} e^{-\pi t^2}\,dt + \lim_{b\to\infty} \sigma \int\limits_{\gamma_{[0,b]}} e^{-\pi t^2}\,dt \\
&= \lim_{a\to-\infty} \sigma \int\limits_{\gamma_{[a,0],2}} e^{-\pi t^2}\,dt + \sigma \int\limits_{\gamma_{[a,0],3}} e^{-\pi t^2}\,dt + \sigma \int\limits_{\gamma_{[0,b],1}} e^{-\pi t^2}\,dt \\
&\quad + \lim_{b\to\infty} \sigma \int\limits_{\gamma_{[0,b],2}} e^{-\pi t^2}\,dt
\end{aligned}$$

$$
\begin{aligned}
&= \lim_{a\to-\infty} \sigma \int_{\frac{a}{\sigma}}^{0} e^{-\pi z^2}\, dz + \sigma \int_{0}^{y\sigma} e^{\pi z^2} i\, dz + \sigma \int_{0}^{y\sigma} e^{\pi(y\sigma - z)^2}\, (-i)\, dz \\
&\quad + \lim_{b\to\infty} \sigma \int_{0}^{\frac{b}{\sigma}} e^{-\pi z^2}\, dz \\
&= \lim_{a\to-\infty} \sigma \int_{\frac{a}{\sigma}}^{0} e^{-\pi z^2}\, dz + \lim_{b\to\infty} \sigma \int_{0}^{\frac{b}{\sigma}} e^{-\pi z^2}\, dz \\
&= \sigma \int_{-\infty}^{\infty} e^{-\pi z^2}\, dz
\end{aligned}
$$

Für $y \leq 0$ kann diese Gleichheit ebenfalls gezeigt werden. Damit und mit einer weiteren Substitution sowie mit Lemma 5.1.1 folgt

$$
\begin{aligned}
\hat{\rho}^{(1)}_{\sigma,c}(y) &= e^{-2\pi i c y} \rho^{(1)}_{\frac{1}{\sigma},0}(y)\, \sigma \int_{-\infty}^{\infty} e^{-\pi z^2}\, dz \\
&= e^{-2\pi i c y} \rho^{(1)}_{\frac{1}{\sigma},0}(y)\, \sigma \int_{-\infty}^{\infty} e^{-\left(\sqrt{\pi} z\right)^2}\, dz \\
&= e^{-2\pi i c y} \rho^{(1)}_{\frac{1}{\sigma},0}(y)\, \sigma \int_{-\infty}^{\infty} e^{-u^2} \frac{1}{\sqrt{\pi}}\, du \\
&= e^{-2\pi i c y} \rho^{(1)}_{\frac{1}{\sigma},0}(y)\, \sigma \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-u^2}\, du \\
&= e^{-2\pi i c y} \rho^{(1)}_{\frac{1}{\sigma},0}(y)\, \sigma \frac{1}{\sqrt{\pi}} \sqrt{\pi} = \sigma \rho^{(1)}_{\frac{1}{\sigma},0}(y)\, e^{-2\pi i c y}.
\end{aligned}
$$

- Induktionsschritt ($m \mapsto m+1$): Es sei $c = (c_1, \ldots, c_{m+1})^T$ und $y = (y_1, \ldots, y_{m+1})^T$. Dann definiere $\tilde{c} := (c_1, \ldots, c_m)^T$ sowie $\tilde{y} := (y_1, \ldots, y_m)^T$. Für jedes $x = (x_1, \ldots, x_{m+1})^T \in \mathbb{R}^{m+1}$ sei analog $\tilde{x} := (x_1, \ldots, x_m)^T$. Wegen Lemma 5.1.3 lässt sich der Satz von Fubini-Tonelli anwenden und zusammen mit der Induktionsvoraussetzung folgt

$$
\hat{\rho}^{(m+1)}_{\sigma,c}(y) = \int_{\mathbb{R}^{m+1}} e^{-\pi \frac{\|x-c\|^2}{\sigma^2}} e^{-2\pi i \langle x, y\rangle}\, dx
$$

$$
\begin{aligned}
&= \int\limits_{\mathbb{R}^{m+1}} \left(\prod_{j=1}^{m+1} e^{-\pi \frac{\left(x_j - c_j\right)^2}{\sigma^2}} \right) \left(\prod_{j=1}^{m+1} e^{-2\pi i x_j y_j} \right) dx \\
&= \int\limits_{\mathbb{R}^{m+1}} e^{-\pi \frac{\|\tilde{x}-\tilde{c}\|^2}{\sigma^2}} e^{-2\pi i \langle \tilde{x}, \tilde{y} \rangle} e^{-\pi \frac{\left(x_{m+1}-c_{m+1}\right)^2}{\sigma^2}} e^{-2\pi i x_{m+1} y_{m+1}} \, dx \\
&= \int\limits_{\mathbb{R}^m} \left(\int\limits_{\mathbb{R}} e^{-\pi \frac{\|\tilde{x}-\tilde{c}\|^2}{\sigma^2}} e^{-2\pi i \langle \tilde{x}, \tilde{y} \rangle} e^{-\pi \frac{\left(x_{m+1}-c_{m+1}\right)^2}{\sigma^2}} e^{-2\pi i x_{m+1} y_{m+1}} \, dx_{m+1} \right) d\tilde{x} \\
&= \int\limits_{\mathbb{R}^m} e^{-\pi \frac{\|\tilde{x}-\tilde{c}\|^2}{\sigma^2}} e^{-2\pi i \langle \tilde{x}, \tilde{y} \rangle} \left(\int\limits_{\mathbb{R}} e^{-\pi \frac{\left(x_{m+1}-c_{m+1}\right)^2}{\sigma^2}} e^{-2\pi i x_{m+1} y_{m+1}} \, dx_{m+1} \right) d\tilde{x} \\
&= \int\limits_{\mathbb{R}^m} e^{-\pi \frac{\|\tilde{x}-\tilde{c}\|^2}{\sigma^2}} e^{-2\pi i \langle \tilde{x}, \tilde{y} \rangle} \hat{\rho}^{(1)}_{\sigma, c_{m+1}} \left(y_{m+1}\right) d\tilde{x} \\
&= \hat{\rho}^{(1)}_{\sigma, c_{m+1}} \left(y_{m+1}\right) \int\limits_{\mathbb{R}^m} e^{-\pi \frac{\|\tilde{x}-\tilde{c}\|^2}{\sigma^2}} e^{-2\pi i \langle \tilde{x}, \tilde{y} \rangle} \, d\tilde{x} \\
&= \hat{\rho}^{(1)}_{\sigma, c_{m+1}} \left(y_{m+1}\right) \hat{\rho}^{(m)}_{\sigma, \tilde{c}} \left(\tilde{y}\right) \\
&= \left(\sigma \rho^{(1)}_{\frac{1}{\sigma}, 0} \left(y_{m+1}\right) e^{-2\pi i c_{m+1} y_{m+1}} \right) \left(\sigma^m \rho^{(m)}_{\frac{1}{\sigma}, 0} \left(\tilde{y}\right) e^{-2\pi i \langle \tilde{c}, \tilde{y} \rangle} \right) \\
&= \sigma^{m+1} e^{-\pi \sigma^2 y_{m+1}^2} e^{-\pi \sigma^2 \left(y_1^2 + \ldots + y_m^2\right)} e^{-2\pi i c_{m+1} y_{m+1}} e^{-2\pi i (c_1 y_1 + \ldots + c_m y_m)} \\
&= \sigma^{m+1} \rho^{(m+1)}_{\frac{1}{\sigma}, 0} \left(y\right) e^{-2\pi i \langle c, y \rangle}.
\end{aligned}
$$

□

Mit dieser Rechnung kann nun eine nützliche Gleichung in Lemma 5.2.7 gezeigt werden. Zuvor muss dafür bewiesen werden, dass $\rho_{\sigma,c}$ eine Schwartzfunktion ist.

Definition 5.2.5. *Eine Funktion* $f : \mathbb{R}^m \to \mathbb{C}, x := (x_1, \ldots, x_m)^T \mapsto f(x)$ *heißt* Schwartzfunktion*, falls sie beliebig oft differenzierbar ist und für alle* $\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m \in \mathbb{N}_0$ *gilt, dass*

$$
\sup_{y \in \mathbb{R}^m} \left| y_1^{\alpha_1} \cdots y_m^{\alpha_m} \frac{\partial^{\beta_1 + \ldots + \beta_m} f}{\partial x_1^{\beta_1} \cdots \partial x_m^{\beta_m}} (y) \right| < \infty
$$

ist.

Lemma 5.2.6. *Sei* $c \in \mathbb{R}^m$ *sowie* $\sigma \in \mathbb{R}_{>0}$*. Dann ist* $\rho_{\sigma,c}$ *eine Schwartzfunktion.*

Beweis. Es ist klar, dass $\rho_{\sigma,c}(x) = e^{\frac{-\pi}{\sigma^2}\left((x_1-c_1)^2 + \ldots + (x_m-c_m)^2\right)}$ beliebig oft differenzierbar ist. Für die zweite Forderung obiger Definition wird mittels Induktion nach $\beta_1 + \ldots + \beta_m$

gezeigt, dass es für alle $\beta := (\beta_1, \ldots, \beta_m)^T \in \mathbb{N}_0^m$ ein Polynom $p_\beta \in \mathbb{R}[x_1, \ldots, x_m]$ gibt, so dass $\frac{\partial^{\beta_1+\ldots+\beta_m}\rho_{\sigma,c}}{\partial x_1^{\beta_1}\cdots\partial x_m^{\beta_m}} \equiv p_\beta \rho_{\sigma,c}$ ist.

- Induktionsanfang ($\beta_1 + \ldots + \beta_m = 0$): In diesem Fall gilt $\frac{\partial^{\beta_1+\ldots+\beta_m}\rho_{\sigma,c}}{\partial x_1^{\beta_1}\cdots\partial x_m^{\beta_m}} = \rho_{\sigma,c}$ und daher ist $p_0 \equiv 1$.

- Induktionsschritt ($\beta_1 + \ldots + \beta_m > 0$): Es gibt in diesem Fall ein $i \in \{1, \ldots, m\}$, so dass $\beta_i > 0$ ist. Für alle $j \in \{1, \ldots, m\}$ definiere nun $\tilde{\beta}_j := \begin{cases} \beta_j & , \text{falls } j \neq i \\ \beta_i - 1 & , \text{falls } j = i \end{cases}$ und $\tilde{\beta} := \left(\tilde{\beta}_1, \ldots, \tilde{\beta}_m\right)^T$. Weil $\tilde{\beta}_1 + \ldots + \tilde{\beta}_m = \beta_1 + \ldots + \beta_m - 1$ ist, gibt es nach Induktionsvoraussetzung ein Polynom $p_{\tilde{\beta}} \in \mathbb{R}[x_1, \ldots, x_m]$, so dass $\frac{\partial^{\tilde{\beta}_1+\ldots+\tilde{\beta}_m}\rho_{\sigma,c}}{\partial x_1^{\tilde{\beta}_1}\cdots\partial x_m^{\tilde{\beta}_m}} \equiv p_{\tilde{\beta}}\rho_{\sigma,c}$ ist. Daher folgt

$$\begin{aligned}
\frac{\partial^{\beta_1+\ldots+\beta_m}\rho_{\sigma,c}}{\partial x_1^{\beta_1}\cdots\partial x_m^{\beta_m}} &= \frac{\partial}{\partial_{x_i}}\left(\frac{\partial^{\tilde{\beta}_1+\ldots+\tilde{\beta}_m}\rho_{\sigma,c}}{\partial x_1^{\tilde{\beta}_1}\cdots\partial x_m^{\tilde{\beta}_m}}\right) \equiv \frac{\partial}{\partial_{x_i}}\left(p_{\tilde{\beta}}\rho_{\sigma,c}\right) \\
&= \left(\frac{\partial}{\partial_{x_i}}p_{\tilde{\beta}}\right)\rho_{\sigma,c} + p_{\tilde{\beta}}\left(\frac{\partial}{\partial_{x_i}}\rho_{\sigma,c}\right) \\
&\equiv \left(\frac{\partial}{\partial_{x_i}}p_{\tilde{\beta}}\right)\rho_{\sigma,c} + p_{\tilde{\beta}}\left(p_i\rho_{\sigma,c}\right),
\end{aligned}$$

  wobei $p_i \in \mathbb{R}[x_1, \ldots, x_m]$ mit $p_i(x) := \frac{-2\pi}{\sigma^2}(x_i - c_i)$ ist. Demnach ist $p_\beta \equiv \left(\frac{\partial}{\partial_{x_i}}p_{\tilde{\beta}}\right) + p_{\tilde{\beta}}p_i \in \mathbb{R}[x_1, \ldots, x_m]$.

Für alle $\alpha := (\alpha_1, \ldots, \alpha_m)^T \in \mathbb{N}_0^m$ sei nun $P_\alpha \in \mathbb{R}[x_1, \ldots, x_m]$ mit $P_\alpha(x) := x_1^{\alpha_1}\cdots x_m^{\alpha_m}$. Dann gilt für alle $\alpha, \beta \in \mathbb{N}_0^m$

$$\sup_{y\in\mathbb{R}^m}\left|y_1^{\alpha_1}\cdots y_m^{\alpha_m}\frac{\partial^{\beta_1+\ldots+\beta_m}\rho_{\sigma,c}}{\partial x_1^{\beta_1}\cdots\partial x_m^{\beta_m}}(y)\right| = \sup_{y\in\mathbb{R}^m}\left|P_\alpha(y)\,p_\beta(y)\,\rho_{\sigma,c}(y)\right| < \infty,$$

weil $\rho_{\sigma,c}$ schneller fällt als jedes Polynom wächst. Damit ist $\rho_{\sigma,c}$ eine Schwartzfunktion. □

Als Nächstes wird die bereits erwähnte nützliche Gleichung gezeigt, die in diesem und dem nächsten Kapitel noch häufig Verwendung finden wird.

Lemma 5.2.7. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter mit vollem Rang. Ferner sei $c \in \mathbb{R}^m$ und $\sigma \in \mathbb{R}_{>0}$.*

Dann gilt

$$\rho_{\sigma,c}(\Lambda) = \det(\Lambda^*)\,\sigma^m \sum_{y\in\Lambda^*} \rho_{\frac{1}{\sigma},0}(y)\, e^{-2\pi i\langle c,y\rangle}.$$

Beweis. Nach der Poissonschen Summenformel, wie sie in Theorem C.1 aus [Lap08] aufgeführt ist, gilt für alle Schwartzfunktionen $f : \mathbb{R}^m \to \mathbb{R}$, dass $\sum_{x\in\Lambda} f(x) = \det(\Lambda^*) \sum_{y\in\Lambda^*} \hat{f}(y)$ ist. Wegen Lemma 5.2.6 und Lemma 5.2.4 folgt dann

$$\begin{aligned}
\rho_{\sigma,c}(\Lambda) &= \det(\Lambda^*) \sum_{y\in\Lambda^*} \hat{\rho}_{\sigma,c}(y) \\
&= \det(\Lambda^*) \sum_{y\in\Lambda^*} \sigma^m \rho_{\frac{1}{\sigma},0}(y)\, e^{-2\pi i\langle c,y\rangle} \\
&= \det(\Lambda^*)\,\sigma^m \sum_{y\in\Lambda^*} \rho_{\frac{1}{\sigma},0}(y)\, e^{-2\pi i\langle c,y\rangle}.
\end{aligned}$$

□

Diese Gleichung wird jetzt im Beweis des folgenden Lemmas benutzt, welches in abgeschwächter Form als Lemma 7 in Lecture 11: „Transference Theorems" bei [Reg04] aufgeführt wird.

Lemma 5.2.8. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter mit vollem Rang, $c \in \mathbb{R}^m$ sowie $\sigma \in \mathbb{R}_{>0}$. Dann gilt*

$$\rho_{\sigma,0}\left((\Lambda - c) \setminus \mathcal{B}_{\sigma\sqrt{m}}(0)\right) < 4^{-m}\rho_{\sigma,0}(\Lambda).$$

Beweis. Als Erstes lässt sich mit Lemma 5.2.7 feststellen, dass

$$\begin{aligned}
\rho_{2,0}(\Lambda - c) = |\rho_{2,c}(\Lambda)| &= \left|\det(\Lambda^*)\, 2^m \sum_{y\in\Lambda^*} \rho_{\frac{1}{2},0}(y)\, e^{-2\pi i\langle c,y\rangle}\right| \\
&\leq \det(\Lambda^*)\, 2^m \sum_{y\in\Lambda^*} \left|\rho_{\frac{1}{2},0}(y)\, e^{-2\pi i\langle c,y\rangle}\right| \\
&= \det(\Lambda^*)\, 2^m \sum_{y\in\Lambda^*} \rho_{\frac{1}{2},0}(y) \\
&\leq \det(\Lambda^*)\, 2^m \sum_{y\in\Lambda^*} \rho_{1,0}(y) = 2^m \rho_{1,0}(\Lambda)
\end{aligned}$$

ist. Zudem gilt

$$\begin{aligned}
\rho_{2,0}(\Lambda - c) \geq \rho_{2,0}\left((\Lambda - c) \setminus \mathcal{B}_{\sqrt{m}}(0)\right) &= \sum_{y \in \Lambda - c, \|y\| \geq \sqrt{m}} e^{-\frac{\pi}{4}\|y\|^2} \\
&= \sum_{y \in \Lambda - c, \|y\| \geq \sqrt{m}} e^{-\pi\|y\|^2} e^{\frac{3\pi}{4}\|y\|^2} \\
&\geq e^{\frac{3\pi m}{4}} \sum_{y \in \Lambda - c, \|y\| \geq \sqrt{m}} e^{-\pi\|y\|^2} \\
&= e^{\frac{3\pi m}{4}} \rho_{1,0}\left((\Lambda - c) \setminus \mathcal{B}_{\sqrt{m}}(0)\right) \\
&> 8^m \rho_{1,0}\left((\Lambda - c) \setminus \mathcal{B}_{\sqrt{m}}(0)\right).
\end{aligned}$$

Damit gilt nun

$$\rho_{1,0}\left((\Lambda - c) \setminus \mathcal{B}_{\sqrt{m}}(0)\right) < 8^{-m} \rho_{2,0}(\Lambda - c) \leq 8^{-m} 2^m \rho_{1,0}(\Lambda) = 4^{-m} \rho_{1,0}(\Lambda).$$

Weiter folgt daraus, dass

$$\begin{aligned}
\rho_{\sigma,0}\left((\Lambda - c) \setminus \mathcal{B}_{\sigma\sqrt{m}}(0)\right) &= \rho_{1,0}\left(\frac{1}{\sigma}\left((\Lambda - c) \setminus \mathcal{B}_{\sigma\sqrt{m}}(0)\right)\right) \\
&= \rho_{1,0}\left(\left(\frac{1}{\sigma}(\Lambda - c)\right) \setminus \mathcal{B}_{\sqrt{m}}(0)\right) \\
&= \rho_{1,0}\left(\left(\frac{1}{\sigma}\Lambda - \frac{1}{\sigma}c\right) \setminus \mathcal{B}_{\sqrt{m}}(0)\right) \\
&< 4^{-m} \rho_{1,0}\left(\frac{1}{\sigma}\Lambda\right) = 4^{-m} \rho_{\sigma,0}(\Lambda)
\end{aligned}$$

ist. □

5.3. Glättungsparameter

Der Glättungsparameter wurde zuerst in [MR07] definiert. Dort werden auch zwei Aussagen (Lemma 3.2 und Lemma 4.4) über diesen Parameter bewiesen, die im Folgenden als Lemma 5.3.2 und Lemma 5.3.3 aufgeführt werden.

Definition 5.3.1. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter sowie $\epsilon \in \mathbb{R}_{>0}$. Dann ist*

$$\eta_\epsilon(\Lambda) := \min\left\{\sigma > 0 \;\middle|\; \rho_{\frac{1}{\sigma},0}(\Lambda^* \setminus \{0\}) \leq \epsilon\right\}$$

der Glättungsparameter *von* Λ *bezüglich* ϵ*.*

Um zu sehen, dass $\eta_\epsilon(\Lambda)$ wohldefiniert ist, betrachte

$$\begin{aligned}\varphi_\Lambda : \mathbb{R}_{>0} &\longrightarrow \mathbb{R}_{>0},\\ \sigma &\longmapsto \rho_{\frac{1}{\sigma},0}(\Lambda^* \setminus \{0\}) = \sum_{y\in\Lambda^*\setminus\{0\}} e^{-\pi\sigma^2\|y\|^2}.\end{aligned}$$

φ_Λ ist stetig und streng monoton fallend. Außerdem gilt $\lim_{\sigma\to 0}\varphi_\Lambda(\sigma) = \infty$ sowie $\lim_{\sigma\to\infty}\varphi_\Lambda(\sigma) = 0$. Deshalb ist φ_Λ bijektiv und damit $\eta_\epsilon(\Lambda)$ wohldefiniert.

Lemma 5.3.2. *Sei* $\Lambda \subseteq \mathbb{R}^m$ *ein Gitter mit vollem Rang. Ferner sei* $\epsilon := 2^{-m}$*. Dann ist* $\eta_\epsilon(\Lambda) \leq \frac{\sqrt{m}}{\lambda_1(\Lambda^*)}$*.*

Beweis. Sei $\sigma := \frac{\sqrt{m}}{\lambda_1(\Lambda^*)}$. Dann gilt $\lambda_1(\sigma\Lambda^*) = \sigma\lambda_1(\Lambda^*) = \sqrt{m}$. Daher folgt mit Lemma 5.2.8, dass

$$\begin{aligned}\rho_{1,0}((\sigma\Lambda^*)\setminus\{0\}) &= \rho_{1,0}\left((\sigma\Lambda^*)\setminus\mathcal{B}_{\sqrt{m}}(0)\right)\\ &< 4^{-m}\rho_{1,0}(\sigma\Lambda^*)\\ &= 4^{-m}\left(1+\rho_{1,0}((\sigma\Lambda^*)\setminus\{0\})\right) = 4^{-m} + 4^{-m}\rho_{1,0}((\sigma\Lambda^*)\setminus\{0\})\end{aligned}$$

ist. Demnach ist $\rho_{1,0}((\sigma\Lambda^*)\setminus\{0\})(1-4^{-m}) < 4^{-m}$, woraus

$$\rho_{\frac{1}{\sigma},0}(\Lambda^*\setminus\{0\}) = \rho_{1,0}(\sigma(\Lambda^*\setminus\{0\})) = \rho_{1,0}((\sigma\Lambda^*)\setminus\{0\}) < \frac{4^{-m}}{1-4^{-m}} \leq \frac{4}{3}4^{-m} < \epsilon$$

folgt. Nach Definition 5.3.1 gilt nun $\eta_\epsilon(\Lambda) \leq \sigma = \frac{\sqrt{m}}{\lambda_1(\Lambda^*)}$. □

Lemma 5.3.3. *Sei* $\Lambda \subseteq \mathbb{R}^m$ *ein Gitter mit vollem Rang,* $c \in \mathbb{R}^m$*,* $\epsilon \in (0,1)$ *und* $\sigma \geq \eta_\epsilon(\Lambda)$*. Dann gilt*

$$\Pr\left(\|x-c\| > \sigma\sqrt{m} \mid x \sim \mathcal{D}_{\Lambda,\sigma,c}\right) \leq \frac{1+\epsilon}{1-\epsilon}2^{-m}.$$

Beweis. Nach Lemma 5.2.7 gilt

$$\begin{aligned}\rho_{\sigma,c}(\Lambda) &= \det(\Lambda^*)\,\sigma^m \sum_{y\in\Lambda^*} \rho_{\frac{1}{\sigma},0}(y)\, e^{-2\pi i\langle c,y\rangle}\\ &= \det(\Lambda^*)\,\sigma^m\left(1 + \sum_{y\in\Lambda^*\setminus\{0\}} \rho_{\frac{1}{\sigma},0}(y)\, e^{-2\pi i\langle c,y\rangle}\right),\end{aligned}$$

wobei wegen $\sigma \geq \eta_\epsilon(\Lambda)$ und Definition 5.3.1

$$\left| \sum_{y \in \Lambda^* \setminus \{0\}} \rho_{\frac{1}{\sigma},0}(y)\, e^{-2\pi i \langle c,y \rangle} \right| \leq \sum_{y \in \Lambda^* \setminus \{0\}} \rho_{\frac{1}{\sigma},0}(y) = \rho_{\frac{1}{\sigma},0}(\Lambda^* \setminus \{0\}) \leq \epsilon$$

ist. Zusammen mit Lemma 5.2.8 ergibt sich

$$\begin{aligned}
\Pr\left(\|x-c\| > \sigma\sqrt{m} \mid x \sim \mathcal{D}_{\Lambda,\sigma,c}\right) &\leq \Pr\left(\|x-c\| \geq \sigma\sqrt{m} \mid x \sim \mathcal{D}_{\Lambda,\sigma,c}\right) \\
&= \frac{\rho_{\sigma,c}\left(\Lambda \setminus \mathcal{B}_{\sigma\sqrt{m}}(c)\right)}{\rho_{\sigma,c}(\Lambda)} \\
&= \frac{\rho_{\sigma,0}\left((\Lambda - c) \setminus \mathcal{B}_{\sigma\sqrt{m}}(0)\right)}{\rho_{\sigma,c}(\Lambda)} \\
&< 4^{-m} \frac{\rho_{\sigma,0}(\Lambda)}{\rho_{\sigma,c}(\Lambda)} \\
&\leq 4^{-m} \frac{\det(\Lambda^*)\,\sigma^m\,(1+\epsilon)}{\det(\Lambda^*)\,\sigma^m\,(1-\epsilon)} \leq 2^{-m} \frac{1+\epsilon}{1-\epsilon}.
\end{aligned}$$

□

Im zweiten Teil dieser Arbeit wird zudem Lemma 3.1 aus [GPV08] verwendet. Deshalb wird dieses Lemma im Folgenden formuliert. Der Beweis wird hier weggelassen, lässt sich aber ebenfalls in [GPV08] finden.

Lemma 5.3.4. *Sei $\Lambda \subseteq \mathbb{R}^m$ ein Gitter mit vollem Rang und $g \in \omega\left(\sqrt{\log m}\right)$ mit $g(m) > 0$. Dann gibt es ein $\epsilon = \epsilon(m) \in \mathbb{R}_{>0}$, welches als Funktion in m vernachlässigbar ist, mit*

$$\eta_\epsilon(\Lambda) \leq g(m) \min\{\|\tilde{B}\| \mid B \in \mathbb{R}^{m \times m} \text{ invertierbar mit } \Lambda = \mathcal{L}(B)\}.$$

6. Algorithmen zum Sampeln von Vektoren

Im Kapitel 4 wurde bereits das *Discrete Gaussian Sampling Problem* eingeführt. Bei diesem Problem soll ein Vektor aus einem Gitter $\Lambda \subseteq \mathbb{R}^m$ nach der Verteilung $\mathcal{D}_{\Lambda,\sigma,0}$ ausgegeben werden, wobei $\sigma \in \mathbb{R}_{>0}$ nach unten beschränkt ist. Ist diese untere Schranke sehr klein, so wird vermutet, dass das *Discrete Gaussian Sampling Problem* schwer ist. In diesem Kapitel werden aber genügend große Schranken betrachtet, so dass das Sampeln der Vektoren in Polynomialzeit möglich ist.

6.1. Sampeln von Gittervektoren

Zuerst lässt sich zeigen, dass für ein genügend großes $\sigma \in \mathbb{R}_{>0}$ das Sampeln von Gittervektoren nach $\mathcal{D}_{\Lambda,\sigma,c}$ erreicht werden kann, indem nach der kontinuierlichen Verteilung $\mathcal{D}_{\sigma,c}$ gesampelt und dann gerundet wird. Diese Idee wurde bereits im Beweis von Lemma 3.2 aus [Reg09] verwendet.

Theorem 6.1.1. *Es existiert ein probabilistischer Polynomialzeitalgorithmus, der bei Eingabe eines Gitters $\Lambda \subseteq \mathbb{R}^m$ mit vollem Rang, einer invertierbaren Matrix $B \in \mathbb{R}^{m\times m}$ mit $\Lambda = \mathcal{L}(B)$, einem $\sigma \in \mathbb{R}_{>0}$ mit $\sigma \geq 8^m \|B\|$ sowie einem $c \in \mathbb{R}^m$ ein $x \in \Lambda$ nach einer Verteilung ausgibt, die statistisch nah zu $\mathcal{D}_{\Lambda,\sigma,c}$ (in m) ist.*

Beweis. Der Algorithmus funktioniert folgendermaßen.

1. Wähle $y \in \mathbb{R}^m$ zufällig $\mathcal{D}_{\sigma,c}$-verteilt.
2. Gib $x := y - (y \mod \Lambda) \in \Lambda$ aus.

Effizienz. Zunächst gilt für $y = (y_1, \ldots, y_m)^T$ und $c = (c_1, \ldots, c_m)^T$, dass

$$\begin{aligned}\mathcal{D}_{\sigma,c}(y) &= \frac{\rho_{\sigma,c}(y)}{\sigma^m} = \frac{1}{\sigma^m} e^{-\pi \frac{\|y-c\|^2}{\sigma^2}} \\ &= \frac{1}{\sigma^m} e^{-\frac{\pi}{\sigma^2}\left((y_1-c_1)^2+\ldots+(y_m-c_m)^2\right)}\end{aligned}$$

$$= \prod_{i=1}^{m} \frac{1}{\sigma} e^{-\pi \frac{(y_i - c_i)^2}{\sigma^2}} = \prod_{i=1}^{m} \frac{\rho_{\sigma,c_i}(y_i)}{\sigma} = \prod_{i=1}^{m} \mathcal{D}_{\sigma,c_i}(y_i)$$

ist. Deshalb lässt sich $y \sim \mathcal{D}_{\sigma,c}$ effizient wählen, indem jede Koordinate y_i einzeln nach $\mathcal{D}_{\sigma,c_i}$ gewählt wird. Ferner lässt sich dann die Ausgabe x effizient berechnen, weil

$$x = y - (y \mod \Lambda) = y - B\left(B^{-1}y \mod 1\right) = y - B\left(B^{-1}y - \lfloor B^{-1}y \rfloor\right) = B\lfloor B^{-1}y \rfloor$$

ist.

Korrektheit. Es ist klar, dass die Ausgabe x ein Gittervektor ist, weil $x = B\lfloor B^{-1}y \rfloor$ und $\lfloor B^{-1}y \rfloor \in \mathbb{Z}^m$ gilt. Zu zeigen bleibt daher noch, dass die Verteilung von x statistisch nah zu $\mathcal{D}_{\Lambda,\sigma,c}$ ist. Nach Konstruktion des Algorithmus ist

$$\begin{aligned} \mathcal{D} : \Lambda &\longrightarrow \mathbb{R}_{>0}, \\ \tilde{x} &\longmapsto \int_{\tilde{x}+\mathcal{P}(B)} \mathcal{D}_{\sigma,c}(\tilde{y})\, d\tilde{y} \end{aligned}$$

die Wahrscheinlichkeitsfunktion der Verteilung der Algorithmusausgabe x auf Λ. Zunächst werden $\mathcal{D}_{\Lambda,\sigma,c}(\tilde{x})$ und $\mathcal{D}(\tilde{x})$ gegen $\mathcal{D}_{\sigma,c}(\tilde{x})$ abgeschätzt.

Sei dafür im Folgenden $\epsilon := \epsilon(m) := 2^{-m}$. Mit Lemma 5.3.2, Lemma 3.5.3 und Definition 3.5.1 folgt

$$\eta_\epsilon(\Lambda) \leq \frac{\sqrt{m}}{\lambda_1(\Lambda^*)} \leq \sqrt{m}\lambda_m(\Lambda) \leq \sqrt{m}\|B\| \leq 8^m\|B\| \leq \sigma.$$

Deswegen gilt

$$\left| \sum_{\tilde{y} \in \Lambda^* \setminus \{0\}} \rho_{\frac{1}{\sigma},0}(\tilde{y})\, e^{-2\pi i \langle c, \tilde{y} \rangle} \right| \leq \sum_{\tilde{y} \in \Lambda^* \setminus \{0\}} \rho_{\frac{1}{\sigma},0}(\tilde{y}) = \rho_{\frac{1}{\sigma},0}(\Lambda^* \setminus \{0\}) \leq \epsilon$$

sowie wegen Lemma 5.3.3

$$\Pr\left(\|\tilde{x} - c\| > \sigma\sqrt{m} \mid \tilde{x} \sim \mathcal{D}_{\Lambda,\sigma,c}\right) \leq \frac{1+\epsilon}{1-\epsilon} 2^{-m}.$$

Bis auf eine in m vernachlässigbare Wahrscheinlichkeit gilt demnach $\|\tilde{x} - c\| \leq \sigma\sqrt{m}$, falls $\tilde{x} \in \Lambda$ zufällig $\mathcal{D}_{\Lambda,\sigma,c}$-verteilt gewählt ist. Im Folgenden werden daher zunächst Vektoren betrachtet, deren Abstand zu c höchstens $\sigma\sqrt{m}$ beträgt.

- Behauptung 1. Für alle $\tilde{x} \in \Lambda$ gilt $\mathcal{D}_{\Lambda,\sigma,c}(\tilde{x}) \leq \frac{\det(\Lambda)}{1-\epsilon}\mathcal{D}_{\sigma,c}(\tilde{x})$.

  Beweis. Aus Lemma 5.2.7 folgt

$$\begin{aligned}\mathcal{D}_{\Lambda,\sigma,c}(\tilde{x}) = \frac{\rho_{\sigma,c}(\tilde{x})}{\rho_{\sigma,c}(\Lambda)} &= \frac{\rho_{\sigma,c}(\tilde{x})}{\det(\Lambda^*)\,\sigma^m\left(1+\sum_{\tilde{y}\in\Lambda^*\setminus\{0\}}\rho_{\frac{1}{\sigma},0}(\tilde{y})\,e^{-2\pi i\langle\tilde{y},c\rangle}\right)} \\ &\leq \frac{\det(\Lambda)\,\rho_{\sigma,c}(\tilde{x})}{\sigma^m(1-\epsilon)} = \frac{\det(\Lambda)}{(1-\epsilon)}\mathcal{D}_{\sigma,c}(\tilde{x}).\end{aligned}$$

  □

- Behauptung 2. Für alle $\tilde{x} \in \Lambda$ gilt $\mathcal{D}_{\Lambda,\sigma,c}(\tilde{x}) \geq \frac{\det(\Lambda)}{1+\epsilon}\mathcal{D}_{\sigma,c}(\tilde{x})$.

  Beweis. Wieder aus Lemma 5.2.7 folgt

$$\begin{aligned}\mathcal{D}_{\Lambda,\sigma,c}(\tilde{x}) = \frac{\rho_{\sigma,c}(\tilde{x})}{\rho_{\sigma,c}(\Lambda)} &= \frac{\rho_{\sigma,c}(\tilde{x})}{\det(\Lambda^*)\,\sigma^m\left(1+\sum_{\tilde{y}\in\Lambda^*\setminus\{0\}}\rho_{\frac{1}{\sigma},0}(\tilde{y})\,e^{-2\pi i\langle\tilde{y},c\rangle}\right)} \\ &\geq \frac{\det(\Lambda)\,\rho_{\sigma,c}(\tilde{x})}{\sigma^m(1+\epsilon)} = \frac{\det(\Lambda)}{1+\epsilon}\mathcal{D}_{\sigma,c}(\tilde{x}).\end{aligned}$$

  □

Um auch $\mathcal{D}(\tilde{x})$ gegen $\mathcal{D}_{\sigma,c}(\tilde{x})$ abzuschätzen, wird zuvor folgende Hilfsaussage gezeigt, die in ähnlicher Form ebenfalls als Claim 2.1 in [Reg09] zu finden ist.

- Behauptung 3. Für alle $\tilde{x}, \tilde{y} \in \mathbb{R}^m$ und $a, d \in \mathbb{R}_{>0}$ mit $\|\tilde{y}-\tilde{x}\| \leq d$ und $\|\tilde{x}-c\| \leq a$ gilt $\rho_{\sigma,c}(\tilde{y}) \geq \rho_{\sigma,c}(\tilde{x})\left(1-\frac{\pi}{\sigma^2}(2da+d^2)\right)$.

  Beweis. Zunächst gilt für alle $z \in \mathbb{R}$, dass $e^z \geq 1+z$ ist. Weiterhin ist

$$\|\tilde{y}-c\| = \|\tilde{y}-\tilde{x}+\tilde{x}-c\| \leq \|\tilde{y}-\tilde{x}\| + \|\tilde{x}-c\| \leq d + \|\tilde{x}-c\| \leq d+a$$

  und insgesamt ergibt sich demnach

$$\begin{aligned}\rho_{\sigma,c}(\tilde{y}) &= e^{-\frac{\pi}{\sigma^2}\|\tilde{y}-c\|^2} \\ &\geq e^{-\frac{\pi}{\sigma^2}(\|\tilde{x}-c\|+d)^2} = e^{-\frac{\pi}{\sigma^2}\left(\|\tilde{x}-c\|^2+2d\|\tilde{x}-c\|+d^2\right)} = \rho_{\sigma,c}(\tilde{x})\,e^{-\frac{\pi}{\sigma^2}\left(2d\|\tilde{x}-c\|+d^2\right)}\end{aligned}$$

$$\geq \rho_{\sigma,c}(\tilde{x})\, e^{-\frac{\pi}{\sigma^2}\left(2da+d^2\right)} \geq \rho_{\sigma,c}(\tilde{x}) \left(1 - \frac{\pi}{\sigma^2}\left(2da + d^2\right)\right).$$

□

Im Folgenden sei $\nu := \nu(m) := \pi \cdot \left(\frac{2m\sqrt{m}}{8^m} + \frac{3m^2}{8^{2m}}\right)$.

- Behauptung 4. Für alle $\tilde{x} \in \Lambda$ mit $\|\tilde{x} - c\| \leq \sigma\sqrt{m}$ gilt $\mathcal{D}(\tilde{x}) \geq \det(\Lambda)(1-\nu)\,\mathcal{D}_{\sigma,c}(\tilde{x})$.

  Beweis. Es bezeichnen $b_1, \ldots, b_m \in \mathbb{R}^m$ die Spalten von B. Für $\tilde{y} \in \tilde{x} + \mathcal{P}(B)$ gilt nach Lemma 3.3.4, dass $\|\tilde{y} - \tilde{x}\| = \|\tilde{y} \mod \mathcal{L}(B)\| \leq \sum_{i=1}^{m} \|b_i\| \leq m\|B\|$ ist, und daher folgt mit Behauptung 3

$$\begin{aligned}
\rho_{\sigma,c}(\tilde{y}) &\geq \rho_{\sigma,c}(\tilde{x}) \left(1 - \frac{\pi}{\sigma^2}\left(2m\|B\|\sigma\sqrt{m} + m^2\|B\|^2\right)\right) \\
&\geq \rho_{\sigma,c}(\tilde{x}) \left(1 - \pi\left(\frac{2m\sqrt{m}\|B\|}{8^m\|B\|} + \frac{m^2\|B\|^2}{8^{2m}\|B\|^2}\right)\right) \\
&= \rho_{\sigma,c}(\tilde{x}) \left(1 - \pi\left(\frac{2m\sqrt{m}}{8^m} + \frac{m^2}{8^{2m}}\right)\right) \\
&\geq \rho_{\sigma,c}(\tilde{x})\,(1-\nu).
\end{aligned}$$

  Damit ergibt sich

$$\begin{aligned}
\mathcal{D}(\tilde{x}) &= \int_{\tilde{x}+\mathcal{P}(B)} \mathcal{D}_{\sigma,c}(\tilde{y})\, d\tilde{y} = \int_{\tilde{x}+\mathcal{P}(B)} \frac{\rho_{\sigma,c}(\tilde{y})}{\sigma^m}\, d\tilde{y} \\
&\geq \int_{\tilde{x}+\mathcal{P}(B)} \frac{\rho_{\sigma,c}(\tilde{x})\,(1-\nu)}{\sigma^m}\, d\tilde{y} = \frac{\det(\Lambda)\,\rho_{\sigma,c}(\tilde{x})\,(1-\nu)}{\sigma^m} \\
&= \det(\Lambda)\,(1-\nu)\,\mathcal{D}_{\sigma,c}(\tilde{x}).
\end{aligned}$$

□

- Behauptung 5. Für alle $\tilde{x} \in \Lambda$ mit $\|\tilde{x} - c\| \leq \sigma\sqrt{m}$ gilt $\mathcal{D}(\tilde{x}) \leq \frac{\det(\Lambda)}{(1-\nu)}\mathcal{D}_{\sigma,c}(\tilde{x})$.

  Beweis. Wie bereits festgestellt wurde, gilt für $\tilde{y} \in \tilde{x} + \mathcal{P}(B)$, dass $\|\tilde{y} - \tilde{x}\| \leq m\|B\|$ und $\|\tilde{y} - c\| \leq m\|B\| + \sigma\sqrt{m}$ ist. Wieder mit Behauptung 3 folgt

$$\rho_{\sigma,c}(\tilde{x}) \geq \rho_{\sigma,c}(\tilde{y}) \left(1 - \frac{\pi}{\sigma^2}\left(2m\|B\|\left(m\|B\| + \sigma\sqrt{m}\right) + m^2\|B\|^2\right)\right)$$

$$\geq \rho_{\sigma,c}(\tilde{y})\left(1-\pi\left(\frac{2m\sqrt{m}\|B\|}{8^m\|B\|}+\frac{3m^2\|B\|^2}{8^{2m}\|B\|^2}\right)\right)$$
$$=\rho_{\sigma,c}(\tilde{y})\left(1-\pi\left(\frac{2m\sqrt{m}}{8^m}+\frac{3m^2}{8^{2m}}\right)\right)=\rho_{\sigma,c}(\tilde{y})(1-\nu).$$

Außerdem ist $1-\nu(m)>0$ für alle $m\geq 1$. Dazu betrachte die Ableitung $\nu'(m)=\pi\left(\frac{\sqrt{m}}{8^m}(3-2\ln(8)m)+\frac{6m}{8^{2m}}(1-\ln(8)m)\right)$. Da für alle $m\geq 1$ sowohl $2\ln(8)m\geq 2\ln(8)>3$ als auch $\ln(8)m\geq\ln(8)>1$ gelten, ist $\nu'(m)<0$ für $m\geq 1$. Für $m\geq 1$ ist daher $\nu(m)$ streng monoton fallend und deshalb ist $1-\nu(m)$ streng monoton wachsend. Weil $1-\nu(1)>0$ ist, gilt $1-\nu(m)>0$ für alle $m\geq 1$. Somit folgt $\rho_{\sigma,c}(\tilde{y})\leq\frac{\rho_{\sigma,c}(\tilde{x})}{1-\nu}$ und daher

$$\mathcal{D}(\tilde{x})=\int\limits_{\tilde{x}+\mathcal{P}(B)}\frac{\rho_{\sigma,c}(\tilde{y})}{\sigma^m}d\tilde{y}$$
$$\leq\int\limits_{\tilde{x}+\mathcal{P}(B)}\frac{\rho_{\sigma,c}(\tilde{x})}{\sigma^m(1-\nu)}d\tilde{y}=\frac{\det(\Lambda)\rho_{\sigma,c}(\tilde{x})}{\sigma^m(1-\nu)}=\frac{\det(\Lambda)}{(1-\nu)}\mathcal{D}_{\sigma,c}(\tilde{x}).$$

□

Mit diesen Abschätzungen kann nun $\left|\mathcal{D}_{\Lambda,\sigma,c}(\tilde{x})-\mathcal{D}(\tilde{x})\right|$ untersucht werden. Ist $\tilde{x}\in\Lambda$ so, dass $\|\tilde{x}-c\|\leq\sigma\sqrt{m}$ und $\mathcal{D}_{\Lambda,\sigma,c}(\tilde{x})\geq\mathcal{D}(\tilde{x})$ gilt, dann folgt aus den Behauptungen 1 und 4, dass

$$\mathcal{D}_{\Lambda,\sigma,c}(\tilde{x})-\mathcal{D}(\tilde{x})\leq\det(\Lambda)\mathcal{D}_{\sigma,c}(\tilde{x})\left(\frac{1}{1-\epsilon}-(1-\nu)\right)=\det(\Lambda)\mathcal{D}_{\sigma,c}(\tilde{x})\frac{\epsilon+\nu-\epsilon\nu}{1-\epsilon}$$

ist. Gilt für ein $\tilde{x}\in\Lambda$ mit $\|\tilde{x}-c\|\leq\sigma\sqrt{m}$ aber $\mathcal{D}_{\Lambda,\sigma,c}(\tilde{x})<\mathcal{D}(\tilde{x})$, so folgt aus den Behauptungen 2 und 5, dass

$$\mathcal{D}(\tilde{x})-\mathcal{D}_{\Lambda,\sigma,c}(\tilde{x})\leq\det(\Lambda)\mathcal{D}_{\sigma,c}(\tilde{x})\left(\frac{1}{1-\nu}-\frac{1}{1+\epsilon}\right)=\det(\Lambda)\mathcal{D}_{\sigma,c}(\tilde{x})\frac{\epsilon+\nu}{1+\epsilon-\nu-\epsilon\nu}$$

ist. Deshalb gilt für alle $\tilde{x}\in\Lambda$ mit $\|\tilde{x}-c\|\leq\sigma\sqrt{m}$, dass

$$\left|D_{\Lambda,\sigma,c}(\tilde{x})-D(\tilde{x})\right|\leq\det(\Lambda)\mathcal{D}_{\sigma,c}(\tilde{x})\mu$$

mit $\mu := \max\left\{\frac{\epsilon+\nu-\epsilon\nu}{1-\epsilon}, \frac{\epsilon+\nu}{1+\epsilon-\nu-\epsilon\nu}\right\}$ ist. Hiermit ergibt sich

$$\begin{aligned}
&\sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|\leq\sigma\sqrt{m}} \left|\mathcal{D}_{\Lambda,\sigma,c}\left(\tilde{x}\right) - \mathcal{D}\left(\tilde{x}\right)\right| \\
&\leq \sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|\leq\sigma\sqrt{m}} \det\left(\Lambda\right)\mathcal{D}_{\sigma,c}\left(\tilde{x}\right)\mu \\
&\leq \sum_{\tilde{x}\in\Lambda} \det\left(\Lambda\right)\mathcal{D}_{\sigma,c}\left(\tilde{x}\right)\mu = \frac{\det\left(\Lambda\right)\rho_{\sigma,c}\left(\Lambda\right)\mu}{\sigma^m} \\
&= \frac{\det\left(\Lambda\right)\mu}{\sigma^m}\det\left(\Lambda^*\right)\sigma^m\left(1+\sum_{\tilde{y}\in\Lambda^*\setminus\{0\}}\rho_{\frac{1}{\sigma},0}\left(\tilde{y}\right)e^{-2\pi i\langle\tilde{y},c\rangle}\right) \\
&\leq \mu\left(1+\epsilon\right).
\end{aligned}$$

Außerdem ist

$$\sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|>\sigma\sqrt{m}} \mathcal{D}_{\Lambda,\sigma,c}\left(\tilde{x}\right) = \Pr\left(\|\tilde{x}-c\| > \sigma\sqrt{m} \mid \tilde{x}\sim\mathcal{D}_{\Lambda,\sigma,c}\right) \leq \frac{1+\epsilon}{1-\epsilon}2^{-m}$$

und damit

$$\begin{aligned}
\sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|>\sigma\sqrt{m}} \mathcal{D}\left(\tilde{x}\right) &= 1 - \sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|\leq\sigma\sqrt{m}} \mathcal{D}\left(\tilde{x}\right) \\
&\leq 1 + \sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|\leq\sigma\sqrt{m}} \left|\mathcal{D}_{\Lambda,\sigma,c}\left(\tilde{x}\right) - \mathcal{D}\left(\tilde{x}\right)\right| - \sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|\leq\sigma\sqrt{m}} \mathcal{D}_{\Lambda,\sigma,c}\left(\tilde{x}\right) \\
&\leq 1 + \mu\left(1+\epsilon\right) - 1 + \sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|>\sigma\sqrt{m}} \mathcal{D}_{\Lambda,\sigma,c}\left(\tilde{x}\right) \\
&\leq \mu\left(1+\epsilon\right) + \frac{1+\epsilon}{1-\epsilon}2^{-m}.
\end{aligned}$$

Insgesamt folgt

$$\begin{aligned}
&\Delta\left(\mathcal{D}_{\Lambda,\sigma,c},\mathcal{D}\right) \\
&= \frac{1}{2}\sum_{\tilde{x}\in\Lambda}\left|\mathcal{D}_{\Lambda,\sigma,c}\left(\tilde{x}\right) - \mathcal{D}\left(\tilde{x}\right)\right| \\
&= \frac{1}{2}\left(\sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|\leq\sigma\sqrt{m}}\left|\mathcal{D}_{\Lambda,\sigma,c}\left(\tilde{x}\right) - \mathcal{D}\left(\tilde{x}\right)\right| + \sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|>\sigma\sqrt{m}}\left|\mathcal{D}_{\Lambda,\sigma,c}\left(\tilde{x}\right) - \mathcal{D}\left(\tilde{x}\right)\right|\right)
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{1}{2}\left(\mu\left(1+\epsilon\right)+\sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|>\sigma\sqrt{m}}\mathcal{D}_{\Lambda,\sigma,c}\left(\tilde{x}\right)+\sum_{\tilde{x}\in\Lambda,\|\tilde{x}-c\|>\sigma\sqrt{m}}\mathcal{D}\left(\tilde{x}\right)\right)\\
&\leq \frac{1}{2}\left(\mu\left(1+\epsilon\right)+\frac{1+\epsilon}{1-\epsilon}2^{-m}+\mu\left(1+\epsilon\right)+\frac{1+\epsilon}{1-\epsilon}2^{-m}\right)\\
&= \mu\left(1+\epsilon\right)+\frac{1+\epsilon}{1-\epsilon}\epsilon.
\end{aligned}
$$

Da ϵ, ν und μ vernachlässigbar in m sind, gilt dies auch für $\Delta\left(\mathcal{D}_{\Lambda,\sigma,c}, \mathcal{D}\right)$. Somit ist die Verteilung, nach der der Algorithmus Gittervektoren ausgibt, statistisch nah zu $\mathcal{D}_{\Lambda,\sigma,c}$ und damit ist alles gezeigt. □

Es können aber auch Gittervektoren nach $\mathcal{D}_{\Lambda,\sigma,c}$ in Polynomialzeit ausgegeben werden, wenn die untere Schranke für $\sigma \in \mathbb{R}_{>0}$ deutlich kleiner ist. Dieses Resultat wurde als Theorem 4.1 in [GPV08] gezeigt. Der dort beschriebene Algorithmus wird von nun an mit *SampleGaussian* bezeichnet.

Theorem 6.1.2. *Es existiert ein probabilistischer Polynomialzeitalgorithmus* SampleGaussian*, der bei Eingabe eines Gitters $\Lambda \subseteq \mathbb{R}^m$ mit vollem Rang, einer invertierbaren Matrix $B \in \mathbb{R}^{m\times m}$ mit $\Lambda = \mathcal{L}\left(B\right)$, einem $\sigma \in \mathbb{R}_{>0}$, wobei $\sigma \geq \|\tilde{B}\| f\left(m\right)$ für ein $f \in \omega\left(\sqrt{\log m}\right)$ ist, sowie einem $c \in \mathbb{R}^m$ ein $x \in \Lambda$ nach einer Verteilung ausgibt, die statistisch nah zu $\mathcal{D}_{\Lambda,\sigma,c}$ (in m) ist.*

6.2. Weitere Algorithmen

Im zweiten Teil dieser Arbeit wird zudem ein effizienter Algorithmus benötigt, der für ein Gitter $\Lambda \subseteq \mathbb{R}^m$ und ein $t \in \mathbb{R}^m$ Vektoren $e \in \Lambda + t$ nach einer Verteilung ausgibt, die statistisch nah zu $\mathcal{D}_{\Lambda+t,\sigma,0}$ ist. Dabei werden spezielle verschobene Gitter betrachtet, die zunächst definiert werden.

Definition 6.2.1. *Sei $q \in \mathbb{N}$ eine Primzahl, $A \in \mathbb{Z}_q^{n\times m}$ und $u \in \mathbb{Z}_q^n$. Dann definiere*

$$\Lambda_q^u\left(A\right) := \left\{e \in \mathbb{Z}^m \mid Ae = u\right\}.$$

Folgendes Lemma verdeutlicht den Zusammenhang zwischen den Mengen aus obiger Definition und dem Gitter $\Lambda_q^\perp\left(A\right)$ aus Beispiel 3.1.5.

Lemma 6.2.2. *Sei $q \in \mathbb{N}$ eine Primzahl, $A \in \mathbb{Z}_q^{n\times m}$ und $u \in \mathbb{Z}_q^n$. Ist $\Lambda_q^u\left(A\right) \neq \emptyset$, so gilt für alle $t \in \Lambda_q^u\left(A\right)$, dass $\Lambda_q^u\left(A\right) = \Lambda_q^\perp\left(A\right) + t$ ist.*

Beweis. Sei $\Lambda_q^u(A) \neq \emptyset$ und $t \in \Lambda_q^u(A)$.

- $\Lambda_q^u(A) \subseteq \Lambda_q^\perp(A) + t$:
  Sei $e \in \Lambda_q^u(A)$. Dann gilt $A(e-t) = Ae - At = u - u = 0$. Also ist $e - t \in \Lambda_q^\perp(A)$ und daher folgt $e = (e-t) + t \in \Lambda_q^\perp(A) + t$.
- $\Lambda_q^u(A) \supseteq \Lambda_q^\perp(A) + t$:
  Sei $x \in \Lambda_q^\perp(A)$. Dann gilt $A(x+t) = Ax + At = 0 + u = u$. Also ist $x + t \in \Lambda_q^u(A)$.

□

Im nächsten Lemma wird gezeigt, dass nach einer Verteilung gesampelt werden kann, die statistisch nah zu $\mathcal{D}_{\Lambda_q^u(A),\sigma,0}$ ist, indem auf *SampleGaussian* zurückgegriffen wird.

Lemma 6.2.3. *Sei $q \in \mathbb{N}$ eine Primzahl. Dann existiert ein probabilistischer Polynomialzeitalgorithmus* SamplePre*, der bei Eingabe einer Matrix $A \in \mathbb{Z}_q^{n \times m}$, einer invertierbaren Matrix $T_A \in \mathbb{Z}^{m \times m}$ mit $\Lambda_q^\perp(A) = \mathcal{L}(T_A)$, einem $u \in \mathbb{Z}_q^n$ mit $\Lambda_q^u(A) \neq \emptyset$ sowie einem $\sigma \in \mathbb{R}_{>0}$, wobei $\sigma \geq \|\widetilde{T_A}\| f(m)$ für ein $f \in \omega\left(\sqrt{\log m}\right)$ ist, ein $e \in \Lambda_q^u(A)$ nach einer Verteilung ausgibt, die statistisch nah zu $\mathcal{D}_{\Lambda_q^u(A),\sigma,0}$ (in m) ist.*

Beweis. Als Erstes folgt eine Beschreibung des Algorithmus.

1. Bestimme ein $t \in \Lambda_q^u(A)$.
2. Sei $x \in \Lambda_q^\perp(A)$ eine Ausgabe von $\mathit{SampleGaussian}\left(\Lambda_q^\perp(A), T_A, \sigma, -t\right)$.
3. Gib $e := x + t \in \Lambda_q^u(A)$ aus.

Effizienz. Mit dem Gaußschen Eliminationsverfahren lässt sich effizient ein $t \in \Lambda_q^u(A)$ finden. Da *SampleGaussian* ein Polynomialzeitalgorithmus ist, ist somit alles gezeigt.

Korrektheit. Wegen Lemma 6.2.2 und wegen $x \in \Lambda_q^\perp(A)$ gilt $e = x + t \in \Lambda_q^u(A)$. Daher bleibt nur noch zu zeigen, dass die Verteilung von e statistisch nah zu $\mathcal{D}_{\Lambda_q^u(A),\sigma,0}$ ist. Bezeichne mit $\mathcal{D} : \Lambda_q^\perp(A) \to [0,1]$ die Wahrscheinlichkeitsfunktion der Verteilung von x. Nach Theorem 6.1.2 ist diese Verteilung statistisch nah zu $\mathcal{D}_{\Lambda_q^\perp(A),\sigma,-t}$. Ferner ist $\mathcal{D}' : \Lambda_q^u(A) \to [0,1], \tilde{e} \mapsto \mathcal{D}(\tilde{e} - t)$ die Wahrscheinlichkeitsfunktion der Verteilung von e, weil *SamplePre* den Vektor $\tilde{e} \in \Lambda_q^u(A)$ genau dann ausgibt, wenn *SampleGaussian* den Vektor $\tilde{e} - t \in \Lambda_q^\perp(A)$ ausgibt. Zudem gilt für alle $\tilde{x} \in \Lambda_q^\perp(A)$, dass $\rho_{\sigma,-t}(\tilde{x}) = e^{-\pi \frac{\|\tilde{x}+t\|^2}{\sigma^2}} = \rho_{\sigma,0}(\tilde{x} + t)$ und daher auch

$$\mathcal{D}_{\Lambda_q^\perp(A),\sigma,-t}(\tilde{x}) = \frac{\rho_{\sigma,-t}(\tilde{x})}{\sum\limits_{y \in \Lambda_q^\perp(A)} \rho_{\sigma,-t}(y)} = \frac{\rho_{\sigma,0}(\tilde{x}+t)}{\sum\limits_{y \in \Lambda_q^\perp(A)} \rho_{\sigma,0}(y+t)}$$

$$= \frac{\rho_{\sigma,0}(\tilde{x}+t)}{\sum\limits_{\tilde{y}\in\Lambda_q^u(A)} \rho_{\sigma,0}(\tilde{y})} = \mathcal{D}_{\Lambda_q^u(A),\sigma,0}(\tilde{x}+t)$$

ist. Insgesamt ergibt sich somit

$$\begin{aligned}
\Delta\left(\mathcal{D}_{\Lambda_q^u(A),\sigma,0}, \mathcal{D}'\right) &= \frac{1}{2} \sum_{\tilde{e}\in\Lambda_q^u(A)} \left|\mathcal{D}_{\Lambda_q^u(A),\sigma,0}(\tilde{e}) - \mathcal{D}'(\tilde{e})\right| \\
&= \frac{1}{2} \sum_{\tilde{e}\in\Lambda_q^u(A)} \left|\mathcal{D}_{\Lambda_q^\perp(A),\sigma,-t}(\tilde{e}-t) - \mathcal{D}(\tilde{e}-t)\right| \\
&= \frac{1}{2} \sum_{\tilde{x}\in\Lambda_q^\perp(A)} \left|\mathcal{D}_{\Lambda_q^\perp(A),\sigma,-t}(\tilde{x}) - \mathcal{D}(\tilde{x})\right| \\
&= \Delta\left(\mathcal{D}_{\Lambda_q^\perp(A),\sigma,-t}, \mathcal{D}\right).
\end{aligned}$$

Demnach ist $\mathcal{D}'$ auch statistisch nah zu $\mathcal{D}_{\Lambda_q^u(A),\sigma,0}$, womit alles gezeigt ist. □

Um *SamplePre* nutzen zu können, wird eine Basis für $\Lambda_q^\perp(A)$ benötigt. Wie eine Matrix A zusammen mit einer Gitterbasis generiert werden kann, wird in [AP09] gezeigt und dort in Theorem 3.1 formuliert. Hier wird ein Spezialfall dieses Theorems angegeben.

Theorem 6.2.4. *Seien n, m, $q \in \mathbb{N}$, so dass q eine Primzahl und $m \geq 5n\log q$ ist. Dann existiert ein probabilistischer Algorithmus* TrapGen*, der bei Eingabe von n, m und q in Polynomialzeit (in n, m und $\log q$) ein Paar $(A, T_A) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}^{m\times m}$ ausgibt, so dass*

- *die Verteilung von A statistisch nah zur Gleichverteilung auf $\mathbb{Z}_q^{n\times m}$ (in n) ist,*
- *$\Lambda_q^\perp(A) = \mathcal{L}(T_A)$ gilt und*
- *für jedes $f \in \omega\left(\sqrt{\log n}\right)$ bis auf eine in n vernachlässigbare Wahrscheinlichkeit $\|T_A\| \leq mf(n)$ ist.*

Als Nächstes wird gezeigt, dass zufällig gleichverteilt gewählte Matrizen $A \in \mathbb{Z}_q^{n\times m}$ mit $m \geq 2n\log q$ bis auf eine vernachlässigbare Wahrscheinlichkeit vollen Zeilenrang haben. Für alle $u \in \mathbb{Z}_q^n$ ist in diesem Fall $\Lambda_q^u(A) \neq \emptyset$ und *SamplePre* kann auf A, T_A, u und ein hinreichend großes σ angewendet werden. Somit liefert *TrapGen* Basen beschränkter Länge und sichert die Anwendbarkeit von *SamplePre* auf ein zufällig gewähltes $u \in \mathbb{Z}_q^n$.

Lemma 6.2.5. *Seien $n, m, q \in \mathbb{N}$, so dass q eine Primzahl und $m \geq 2n\log q$ ist. Ist $A \in \mathbb{Z}_q^{n\times m}$ zufällig gleichverteilt gewählt, so hat A bis auf eine in n vernachlässigbare Wahrscheinlichkeit vollen Zeilenrang.*

Beweis. Für $k \in \{1, \ldots, m\}$ gilt, dass k linear unabhängige Vektoren aus $\mathbb{Z}_q^m$ insgesamt q^k Vektoren erzeugen. Die Wahrscheinlichkeit, dass ein aus $\mathbb{Z}_q^m$ zufällig gleichverteilt gewählter Vektor linear unabhängig zu den bereits vorhandenen k Vektoren ist, ist damit $1 - \frac{q^k}{q^m}$. Außerdem ist die Wahrscheinlichkeit, dass der gewählte Vektor nicht der Nullvektor ist, $1 - \frac{q^0}{q^m}$. Da $A \in \mathbb{Z}_q^{n \times m}$ zufällig gleichverteilt gewählt werden kann, indem sukzessive die n Zeilenvektoren von A aus Z_q^m zufällig gleichverteilt und unabhängig voneinander gewählt werden, gilt

$$\Pr\left(\operatorname{rk}(A) = n\right) = \prod_{k=0}^{n-1} \left(1 - \frac{q^k}{q^m}\right).$$

Weil aber für alle $k \in \{0, \ldots, n-1\}$ gilt, dass $\frac{q^k}{q^m} < \frac{q^n}{q^m} \leq \frac{q^n}{q^{2n \log q}} = \frac{1}{q^{2n \log q - n}} \leq \frac{1}{q^n}$ ist, gilt auch

$$\Pr\left(\operatorname{rk}(A) = n\right) > \left(1 - \frac{1}{q^n}\right)^n = 1 + \sum_{j=1}^{n} \binom{n}{j} \left(-\frac{1}{q^n}\right)^j.$$

Dabei ist

$$\left| \sum_{j=1}^{n} \binom{n}{j} \left(-\frac{1}{q^n}\right)^j \right| \leq \sum_{j=1}^{n} \binom{n}{j} \left(\frac{1}{q^n}\right)^j = \sum_{j=1}^{n} \prod_{i=1}^{j} \frac{n-j+i}{iq^n},$$

weshalb $\sum_{j=1}^{n} \binom{n}{j} \left(-\frac{1}{q^n}\right)^j$ vernachlässigbar in n ist. □

Zum Schluss dieses Kapitels wird noch die Verteilung von Ae für eine Matrix $A \in \mathbb{Z}_q^{n \times m}$ und ein zufällig $\mathcal{D}_{\mathbb{Z}^m, \sigma, 0}$-verteiltes $e \in \mathbb{Z}^m$ betrachtet, weil dies im zweiten Teil dieser Arbeit benötigt wird. Zuvor wird die Determinante von $\Lambda_q^\perp(A)$ bestimmt. Dafür wird ein Spezialfall des Elementarteilersatzes benötigt, der zum Beispiel in Abschnitt 11.5 aus [KM03] erläutert wird.

Satz 6.2.6. *Für jede Matrix $B \in \mathbb{Z}^{m \times m}$ gibt es Matrizen $S \in \mathbb{Z}^{m \times m}$ und $T \in \mathbb{Z}^{m \times m}$ mit $|\det(S)| = |\det(T)| = 1$, so dass $SBT \in \mathbb{Z}^{m \times m}$ eine Diagonalmatrix ist.*

Lemma 6.2.7. *Sei q eine Primzahl und $A \in \mathbb{Z}_q^{n \times m}$ mit vollem Zeilenrang. Dann ist $\det\left(\Lambda_q^\perp(A)\right) = q^n$.*

Beweis. Weil $\Lambda_q^\perp(A) \subseteq \mathbb{Z}^m$ ein Gitter mit vollen Rang ist, existiert eine in $\mathbb{R}$ invertierbare

Matrix $B \in \mathbb{Z}^{m \times m}$ mit $\Lambda_q^\perp(A) = \mathcal{L}(B)$. Nach Satz 6.2.6 gibt es Matrizen $S \in \mathbb{Z}^{m \times m}$ und $T \in \mathbb{Z}^{m \times m}$ mit $|\det(S)| = |\det(T)| = 1$, so dass $D := SBT \in \mathbb{Z}^{m \times m}$ eine Diagonalmatrix ist. Es bezeichnen $d_1, \ldots, d_m$ die Diagonaleinträge von D. Dann ist

$$\det\left(\Lambda_q^\perp(A)\right) = \sqrt{\det(B^T B)} = |\det(B)| = |\det(D)| = \left|\prod_{i=1}^{m} d_i\right| = |\mathbb{Z}^m / \mathcal{L}(D)|.$$

Wegen Lemma 3.3.7 ist $\mathcal{L}(D) = \mathcal{L}(SBT) = \mathcal{L}(SB)$. Weil zudem

$$\begin{aligned} \mathbb{Z}^m / \Lambda_q^\perp(A) &\longrightarrow \mathbb{Z}^m / \mathcal{L}(SB), \\ x + \Lambda_q^\perp(A) &\longmapsto Sx + \mathcal{L}(SB) \end{aligned}$$

und

$$\begin{aligned} \mathbb{Z}^m / \Lambda_q^\perp(A) &\longrightarrow \mathbb{Z}_q^n, \\ x + \Lambda_q^\perp(A) &\longmapsto Ax \end{aligned}$$

Gruppenisomorphismen sind, ist

$$\det\left(\Lambda_q^\perp(A)\right) = |\mathbb{Z}^m / \mathcal{L}(D)| = |\mathbb{Z}^m / \mathcal{L}(SB)| = \left|\mathbb{Z}^m / \Lambda_q^\perp(A)\right| = \left|\mathbb{Z}_q^n\right| = q^n.$$

□

Lemma 6.2.8. *Sei $q \in \mathbb{N}$ eine Primzahl, $\sigma \in \mathbb{R}_{>0}$ und $A \in \mathbb{Z}_q^{n \times m}$ mit vollem Zeilenrang. Ferner seien $\delta := \delta(m), \epsilon := \epsilon(m) \in \mathbb{R}_{>0}$ als Funktionen in m vernachlässigbar, so dass $\sigma \geq \eta_\delta(\mathbb{Z}^m)$ und $\sigma \geq \eta_\epsilon\left(\Lambda_q^\perp(A)\right)$ ist. Ist $e \in \mathbb{Z}^m$ zufällig $\mathcal{D}_{\mathbb{Z}^m,\sigma,0}$-verteilt, so ist die Verteilung von Ae auf $\mathbb{Z}_q^n$ statistisch nah (in m) zur diskreten Gleichverteilung auf $\mathbb{Z}_q^n$.*

Beweis. Es bezeichne $\mathcal{D}$ die Verteilung von Ae auf $\mathbb{Z}_q^n$ sowie $\mathcal{U}$ die diskrete Gleichverteilung auf $\mathbb{Z}_q^n$. Sei $u \in \mathbb{Z}_q^n$ und $t \in \Lambda_q^u(A)$. Dann ist wegen Lemma 6.2.2

$$\begin{aligned} \mathcal{D}(u) &= \Pr\left(u = Ae \mid e \sim \mathcal{D}_{\mathbb{Z}^m,\sigma,0}\right) \\ &= \Pr\left(e \in \Lambda_q^u(A) \mid e \sim \mathcal{D}_{\mathbb{Z}^m,\sigma,0}\right) \\ &= \frac{\rho_{\sigma,0}\left(\Lambda_q^u(A)\right)}{\rho_{\sigma,0}(\mathbb{Z}^m)} \end{aligned}$$

$$= \frac{\rho_{\sigma,0}\left(t + \Lambda_q^\perp(A)\right)}{\rho_{\sigma,0}\left(\mathbb{Z}^m\right)}$$
$$= \frac{\rho_{\sigma,-t}\left(\Lambda_q^\perp(A)\right)}{\rho_{\sigma,0}\left(\mathbb{Z}^m\right)}.$$

Aus Lemma 5.2.7 und Lemma 6.2.7 folgt

$$\rho_{\sigma,-t}\left(\Lambda_q^\perp(A)\right) = \frac{1}{\det\left(\Lambda_q^\perp(A)\right)}\sigma^m\left(1 + \sum_{y\in\left(\Lambda_q^\perp(A)\right)^*\setminus\{0\}} \rho_{\frac{1}{\sigma},0}(y)\, e^{2\pi i\langle t,y\rangle}\right)$$
$$= \frac{\sigma^m}{q^n}\left(1 + \sum_{y\in\left(\Lambda_q^\perp(A)\right)^*\setminus\{0\}} \rho_{\frac{1}{\sigma},0}(y)\, e^{2\pi i\langle t,y\rangle}\right)$$

und wegen $(\mathbb{Z}^m)^* = \mathbb{Z}^m$ auch

$$\rho_{\sigma,0}\left(\mathbb{Z}^m\right) = \det\left(\mathbb{Z}^m\right)\sigma^m\left(1 + \sum_{y\in\mathbb{Z}^m\setminus\{0\}} \rho_{\frac{1}{\sigma},0}(y)\right)$$
$$= \sigma^m\left(1 + \rho_{\frac{1}{\sigma},0}\left(\mathbb{Z}^m\setminus\{0\}\right)\right).$$

Dabei ist

$$\left|\sum_{y\in\left(\Lambda_q^\perp(A)\right)^*\setminus\{0\}} \rho_{\frac{1}{\sigma},0}(y)\, e^{2\pi i\langle t,y\rangle}\right| \leq \sum_{y\in\left(\Lambda_q^\perp(A)\right)^*\setminus\{0\}} \rho_{\frac{1}{\sigma},0}(y) = \rho_{\frac{1}{\sigma},0}\left(\left(\Lambda_q^\perp(A)\right)^*\setminus\{0\}\right) \leq \epsilon$$

und

$$0 \leq \rho_{\frac{1}{\sigma},0}\left(\mathbb{Z}^m\setminus\{0\}\right) \leq \delta.$$

Demnach gilt

$$\mathcal{D}(u) = \frac{\rho_{\sigma,-t}\left(\Lambda_q^\perp(A)\right)}{\rho_{\sigma,0}\left(\mathbb{Z}^m\right)} \leq \frac{\frac{\sigma^m}{q^n}(1+\epsilon)}{\rho_{\sigma,0}\left(\mathbb{Z}^m\right)} \leq \frac{\frac{\sigma^m}{q^n}(1+\epsilon)}{\sigma^m} = \frac{1}{q^n}(1+\epsilon)$$

sowie

$$\mathcal{D}(u) = \frac{\rho_{\sigma,-t}\left(\Lambda_q^\perp(A)\right)}{\rho_{\sigma,0}\left(\mathbb{Z}^m\right)} \geq \frac{\frac{\sigma^m}{q^n}(1-\epsilon)}{\rho_{\sigma,0}\left(\mathbb{Z}^m\right)} \geq \frac{\frac{\sigma^m}{q^n}(1-\epsilon)}{\sigma^m(1+\delta)} = \frac{1}{q^n}\frac{1-\epsilon}{1+\delta}.$$

Nun kann $\left|\mathcal{D}(u) - \frac{1}{q^n}\right|$ abgeschätzt werden. Falls $\frac{1}{q^n} \geq \mathcal{D}(u)$ ist, so ist

$$\frac{1}{q^n} - \mathcal{D}(u) \leq \frac{1}{q^n}\left(1 - \frac{1-\epsilon}{1+\delta}\right) = \frac{1}{q^n}\frac{\delta+\epsilon}{1+\delta}.$$

Im Fall $\mathcal{D}(u) > \frac{1}{q^n}$ gilt

$$\mathcal{D}(u) - \frac{1}{q^n} \leq \frac{1}{q^n}(1+\epsilon-1) = \frac{1}{q^n}\epsilon.$$

Dementsprechend ist

$$\left|\mathcal{D}(u) - \frac{1}{q^n}\right| \leq \frac{1}{q^n}\mu,$$

wobei $\mu := \max\left\{\frac{\delta+\epsilon}{1+\delta}, \epsilon\right\}$ ist, und es gilt

$$\Delta(\mathcal{D},\mathcal{U}) = \frac{1}{2}\sum_{u\in\mathbb{Z}_q^n}\left|\mathcal{D}(u) - \frac{1}{q^n}\right| \leq \frac{1}{2}\sum_{u\in\mathbb{Z}_q^n}\frac{1}{q^n}\mu = \frac{\mu}{2}.$$

Weil μ vernachlässigbar in m ist, ist alles gezeigt. □

Teil II.

Attributbasierte Verschlüsselung mittels Gittermethoden - Verfahren und Sicherheitsbeweise

7. Attributbasierte und Fuzzy Identitätsbasierte Verschlüsselung

Im zweiten Teil dieser Arbeit werden Fuzzy Identitätsbasierte Verschlüsselungsverfahren vorgestellt. Daher werden zunächst das allgemeine Prinzip sowie der Nutzen dieser Verschlüsselung erläutert. Da Fuzzy Identitätsbasierte Verschlüsselung ein Spezialfall von Attributbasierter Verschlüsselung ist, wird zuerst auf Letztere eingegangen.

Der Begriff und das Prinzip der Attributbasierten Verschlüsselung wurde in [SW05] eingeführt, um folgendem Problem entgegenzuwirken. Bei einem Dateisystem mit teilweise gemeinsamen Dateien für verschiedene Nutzer muss sichergestellt sein, dass ein Nutzer nur auf solche Dateien zugreifen kann, für die er befugt ist. Die Überprüfung der Zugriffsrechte eines Nutzers kann zum Beispiel durch Software ohne Verwendung von Verschlüsselungsverfahren realisiert werden. Dann liegen aber alle gespeicherten Dateien im Klartext auf dem Dateisystem und wenn das System beispielsweise durch Ausnutzen eines Softwarefehlers erfolgreich angegriffen wurde, so können sämtliche Informationen dem Angreifer zugänglich sein. Alternativ können herkömmliche asymmetrische Verschlüsselungsverfahren verwendet werden. Für jeden Nutzer müssen dann separate Kopien der Dateien existieren, auf die er zugreifen darf, um diese Kopien mit dem öffentlichen Schlüssel des Nutzers zu verschlüsseln. Somit kann der Nutzer mit seinem geheimen Schlüssel die zu ihm gehörigen Dateikopien entschlüsseln. Müssen allerdings viele Dateien für viele verschiedene Nutzer zugänglich sein, so ist der Speicheraufwand dafür sehr groß. Eine andere Möglichkeit wäre, dass es für gemeinsame Dateien auch gemeinsame Schlüssel gibt, was aber zu sehr vielen Schlüsseln führen kann. Außerdem wäre es denkbar, dass jeder Eigentümer seine Dateien selbst verwaltet und bei Anfrage eines anderen Nutzers die angefragten Dateien entschlüsselt und weitergibt. Bei hohem Informationsaustausch kann dies sehr aufwändig sein.

Alle vorgestellten Alternativen weisen Probleme auf. Eine natürliche Herangehensweise liefert nun Attributbasierte Verschlüsselung, von der es prinzipiell zwei Arten gibt.

Bei Key-Policy Attributbasierter Verschlüsselung besitzt eine Datei bestimmte Attribute und wird unter diesen verschlüsselt. Am Institut für Informatik könnten solche Attribute zum Beispiel das Fachgebiet, dem die Datei zugeordnet ist, sowie das Erscheinungsjahr sein. Zudem hat jeder Nutzer Zugriffsrechte. Der geheime Schlüssel eines Nutzers wird dann so gebildet, dass der Nutzer eine Datei genau dann entschlüsseln kann, wenn ihre Attribute seine Zugriffsrechte erfüllen. So soll ein Nutzer beispielsweise alle Dateien entschlüsseln können, die vom Fachgebiet Rechnernetze oder aus dem Jahr 2012 stammen.

Die zweite Art Attributbasierter Verschlüsselung ist Ciphertext-Policy Attributbasierte Verschlüsselung. Hierbei hat jeder Nutzer bestimmte Attribute und zu jeder Datei gehören bestimmte Zugriffsrechte. In einem großen Unternehmen könnten Attribute eines Nutzers zum Beispiel seine Abteilung sowie sein Standort sein und eine Datei soll beispielsweise von allen Mitarbeitern des Controlling in Paderborn entschlüsselt werden können.

Bei beiden Arten der Attributbasierten Verschlüsselung muss pro Datei nur eine Kopie existieren und jeder Nutzer benötigt nur einen Schlüssel. Es können somit komplexere und feinere Zugriffsrechte ohne die oben angesprochenen Probleme realisiert werden. Die Arten und Vorteile von Attributbasierter Verschlüsselung werden auch in [GPSW06] erläutert.

Wie bereits erwähnt, ist Fuzzy Identitätsbasierte Verschlüsselung ein Spezialfall von Attributbasierter Verschlüsselung, welcher ebenfalls in [SW05] eingeführt wurde. Dabei besitzen sowohl Dateien als auch Nutzer Attribute. Eine Datei wird unter ihren Attributen verschlüsselt und ein Nutzer soll eine Datei genau dann entschlüsseln können, wenn eine Mindestanzahl seiner Attribute mit denen der Datei übereinstimmt. Fuzzy Identitätsbasierte Verschlüsselung kann daher sowohl als Key-Policy als auch als Ciphertext-Policy Attributbasierte Verschlüsselung aufgefasst werden, weil die Attribute eines Nutzers bzw. einer Datei festlegen, welche Möglichkeiten es für die Attribute einer Datei, welche der Nutzer entschlüsseln können soll, bzw. für die Attribute eines Nutzers, welcher die Datei entschlüsseln können soll, gibt. Die Menge der Attribute eines Nutzers bzw. einer Datei wird Identität genannt.

Ein typischer Anwendungsfall für Fuzzy Identitätsbasierte Verschlüsselung sind biometrische Charakteristiken wie ein Fingerabdruck oder die Iris. Von einem Nutzer wird ein Fingerabdruck oder eine Irisaufnahme gemacht und dann wird damit eine Datei verschlüsselt. Möchte der Nutzer auf die Datei zugreifen, so wird wieder ein Fingerabdruck

oder eine Irisaufnahme gemacht und als geheimer Schlüssel des Nutzers verwendet. Die neue Aufnahme muss dabei nicht vollständig mit der früheren Aufnahme übereinstimmen. Gibt es allerdings hinreichend viele Übereinstimmungen zwischen den Aufnahmen, so wird der Nutzer als der Richtige identifiziert und er kann die Datei entschlüsseln.

8. Konkrete Fuzzy Identitätsbasierte Verschlüsselungsverfahren

Im Folgenden werden zwei Fuzzy Identitätsbasierte Verschlüsselungsverfahren vorgestellt. Das erste Verfahren verwendet Gitter und seine Sicherheit basiert auf dem *Learning with Errors Problem*. Dahingegen benutzt das zweite Verfahren bilineare Abbildungen auf Gruppen und seine Sicherheit beruht auf einer modifizierten Version der Entscheidungsvariante des *Bilinear Diffie-Hellman Problem*. Letzteres Problem wird im nächsten Kapitels vorgestellt.

8.1. Aufbau Fuzzy Identitätsbasierter Verschlüsselungsverfahren

Beiden vorzustellenden Verfahren ist gemeinsam, dass sie aus vier Polynomialzeitalgorithmen bestehen. Diese werden zunächst in Anlehnung an Abschnitt 2.1 aus [ABV$^+$12] allgemein beschrieben.

- *Setup.*
  - Eingabe: Sicherheitsparameter $\lambda \in \mathbb{N}$, Anzahl aller möglichen Attribute $l \in \mathbb{N}$, Grenzwert $k \in \mathbb{N}$ mit $k \leq l$.
  - Ausgabe: öffentliche Parameter PP, Hauptschlüssel MK.

  Dieser Algorithmus wird von einer zentralen Instanz einmal zur Initialisierung des Verschlüsselungsverfahrens aufgerufen. k bezeichnet die erforderliche Mindestanzahl an Übereinstimmungen zwischen der Identität eines Teilnehmers und der Identität eines Schlüsseltextes, damit der Teilnehmer den Schlüsseltext entschlüsseln kann. Die öffentlichen Parameter sind allen Teilnehmern bekannt, der Hauptschlüssel hingegen nur der zentralen Instanz.

- *KeyGen.*
  - Eingabe: öffentliche Parameter PP, Hauptschlüssel MK, Identität id.
  - Ausgabe: geheimer Schlüssel $\mathrm{SK_{id}}$.

  Eine zentrale Instanz kann mit diesem Algorithmus für jeden Teilnehmer mit einer Identität einen geheimen Schlüssel berechnen.

- *Enc.*
  - Eingabe: öffentliche Parameter PP, Identität id$'$, Nachricht M.
  - Ausgabe: Schlüsseltext $\mathrm{CT_{id'}}$ zur Nachricht M.

  Jeder Teilnehmer kann hiermit eine Nachricht unter Verwendung einer Identität verschlüsseln.

- *Dec.*
  - Eingabe: öffentliche Parameter PP, Schlüsseltext $\mathrm{CT_{id'}}$, geheimer Schlüssel $\mathrm{SK_{id}}$.
  - Ausgabe: Nachricht M, falls id und id$'$ mindestens k Übereinstimmungen haben.

  Ein Teilnehmer mit der Identität id und einem geheimen Schlüssel zu dieser Identität kann mit diesem Algorithmus einen Schlüsseltext zur Identität id$'$ entschlüsseln, wenn id und id$'$ mindestens k Übereinstimmungen haben.

Die ersten drei Algorithmen sind probabilistisch. Dementsprechend kann die Korrektheit eines Fuzzy Identitätsbasierten Verschlüsselungsverfahrens wie folgt definiert werden.

Definition 8.1.1. *Betrachte ein Fuzzy Identitätsbasiertes Verschlüsselungsverfahren mit den Algorithmen* Setup, KeyGen, Enc *und* Dec. *Sei zudem* $\lambda \in \mathbb{N}$, $l \in \mathbb{N}$ *und* $k \in \mathbb{N}$ *mit* $k \leq l$. *Ferner seien id und id$'$ Identitäten mit mindestens k Übereinstimmungen und M sei eine Nachricht. Außerdem sei* $(PP, MK) \leftarrow \mathrm{Setup}(\lambda, l, k)$, $SK_{id} \leftarrow \mathrm{KeyGen}(PP, MK, id)$ *und* $CT_{id'} \leftarrow \mathrm{Enc}(PP, id', M)$. *Das Verfahren heißt* korrekt, *falls* $\mathrm{Dec}(PP, CT_{id'}, SK_{id}) = M$ *bis auf eine in* λ *vernachlässigbare Wahrscheinlichkeit über die zufällige Wahl von PP,* $CT_{id'}$ *sowie* SK_{id} *ist.*

8.2. Verfahren basierend auf Gitterproblemen

Als Nächstes folgt eine Beschreibung des Fuzzy Identitätsbasierten Verschlüsselungsverfahrens, welches Gitter verwendet. Dieses Verfahren wurde in leicht abgewandelter Form in Kapitel 4 aus [ABV$^+$12] definiert. Welche Änderungen hier am Verfahren vorgenommen wurden, wird am Ende dieses Abschnitts erläutert. Außerdem ist der Korrektheitsbeweis für das Verfahren in [ABV$^+$12] unvollständig und fehlerhaft und wird in diesem Abschnitt korrigiert.

Für die Beschreibung des Verfahrens werden folgende Parameter benötigt.

- $\lambda \in \mathbb{N}$ sei der Sicherheitsparameter.
- $m := m(\lambda) \in \mathbb{N}$ sei polynomiell in λ und die Dimension der im Verfahren verwendeten Gitter.
- $n := n(\lambda) \in \mathbb{N}$ sei polynomiell in λ und die Zeilenzahl der Matrizen, welche die verwendeten Gitter erzeugen.
- $q := q(\lambda) \in \mathbb{N}$ sei eine Primzahl, so dass $\log(q)$ polynomiell in λ ist. Die meisten Berechnungen im Verfahren werden in $\mathbb{Z}_q$ ausgeführt.
- $\sigma := \sigma(\lambda) \in \mathbb{R}_{>0}$ sei polynomiell in λ und wird als Parameter zum Sampeln von Vektoren aus verschobenen Gittern verwendet.
- $\alpha := \alpha(\lambda) \in (0, 1)$ sei der Parameter einer Wahrscheinlichkeitsverteilung auf $\mathbb{Z}_q$, die für zufälliges, additives Rauschen verwendet wird.
- $l \in \mathbb{N}$ mit $l \leq n$ bezeichne die Anzahl aller möglichen Attribute.
- $k \in \mathbb{N}$ mit $k \leq l$ bezeichne die Mindestanzahl an Übereinstimmungen zwischen der Identität eines Teilnehmers und der Identität eines Schlüsseltextes, damit der Teilnehmer den Schlüsseltext entschlüsseln kann.

Nun können die benötigten vier Algorithmen mit Polynomialzeit in λ beschrieben werden.

Verfahren 8.2.1.

- Setup (λ, l, k).
  1. *Für alle* $i \in \{1, \ldots, l\}$ *sei* $(A_i, T_i) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}^{m \times m}$ *eine Ausgabe von* $\mathrm{TrapGen}(n, m, q)$ *aus Theorem 6.2.4.*
  2. *Wähle* $u = (u_1, \ldots, u_n)^T \in \mathbb{Z}_q^n$ *zufällig gleichverteilt.*
  3. *Gib sowohl* $PP := (k, u, A_1, A_2, \ldots, A_l)$ *als auch* $MK := (T_1, T_2, \ldots, T_l)$ *aus.*
- $\mathrm{KeyGen}(PP, MK, id)$ *mit* $id = (id_1, \ldots, id_l)^T \in \{0,1\}^l$.
  1. *Für alle* $i \in \{1, \ldots, n\}$ *wähle* $p_i \in \mathbb{Z}_q[X]$ *zufällig gleichverteilt, so dass* $p_i(0) = u_i$ *ist und* p_i *den Grad* $k-1$ *hat.*
  2. *Für alle* $j \in \{1, \ldots, l\}$ *mit* $id_j = 1$ *sei* $\hat{u}_j := (p_1(j), \ldots, p_n(j))^T \in \mathbb{Z}_q^n$.
  3. *Für alle* $j \in \{1, \ldots, l\}$ *mit* $id_j = 1$ *sei weiter* $e_j \in \mathbb{Z}^m$ *eine Ausgabe von* $\mathrm{SamplePre}(A_j, T_j, \hat{u}_j, \sigma)$ *aus Lemma 6.2.3.*
  4. *Gib* $SK_{id} := (id, \{e_j \mid j \in \{1, \ldots, l\}, id_j = 1\})$ *aus.*
- $\mathrm{Enc}(PP, id', M)$ *mit* $id' = (id'_1, \ldots, id'_l)^T \in \{0,1\}^l$ *und* $M \in \{0,1\}$.
  1. *Setze* $D := (l!)^2$.
  2. *Wähle* $s \in \mathbb{Z}_q^n$ *zufällig gleichverteilt.*
  3. *Wähle* $x \in \mathbb{Z}_q$ *zufällig* $\overline{\Psi}_\alpha$*-verteilt mit der Wahrscheinlichkeitsverteilung aus Definition 4.3.3 und Definition 4.3.4.*
  4. *Für alle* $j \in \{1, \ldots, l\}$ *mit* $id'_j = 1$ *wähle* $x_j \in \mathbb{Z}_q^m$ *zufällig* $\overline{\Psi}_\alpha^m$*-verteilt.*
  5. *Setze* $c_0 := u^T s + Dx + M \left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q$.
  6. *Für alle* $j \in \{1, \ldots, l\}$ *mit* $id'_j = 1$ *setze* $c_j := A_j^T s + Dx_j \in \mathbb{Z}_q^m$.
  7. *Gib* $CT_{id'} := \left(id', c_0, \{c_j \mid j \in \{1, \ldots, l\}, id'_j = 1\}\right)$ *aus.*
- $\mathrm{Dec}(PP, CT_{id'}, SK_{id})$.
  1. *Setze* $J := \{j \mid j \in \{1, \ldots, l\}, id_j = id'_j = 1\}$.
  2. *Wenn* $|J| < k$ *ist, dann gib* $\perp$ *als Platzhalter für „keine Entschlüsselung“ aus.*
  3. *Wenn* $|J| \geq k$ *ist, dann berechne Folgendes:*
     a) *Wähle* $S \subseteq J$ *mit* $|S| = k$.

b) Für alle $j \in S$ *setze* $L_j := \prod_{i \in S \setminus \{j\}} \frac{-i}{j-i}$.

c) Setze $r := c_0 - \sum_{j \in S} L_j e_j^T c_j \in \mathbb{Z}_q$.

d) Wenn $\min\{r, q-r\} < \lfloor \frac{q}{4} \rfloor$ *ist, so gib 0 aus, ansonsten gib 1 aus.*

Die grobe Idee hinter diesem Verfahren beginnt damit, dass eine Nachricht M in ihrem Schlüsseltext verborgen wird, indem auf $M \lfloor \frac{q}{2} \rfloor$ der zufällig gleichverteilte Wert $u^T s \in \mathbb{Z}_q$ addiert wird. Um während der Entschlüsselung mit einem passendem geheimen Schlüssel diese Addition wieder rückgängig machen zu können, werden weitere Informationen, die ungefähr die Form $A_j^T s$ haben, dem Schlüsseltext hinzugefügt. Ein passender geheimer Schlüssel liefert dazu mindestens k Vektoren e_j mit $A_j e_j = \hat{u}_j$. Die Vektoren $\hat{u}_j$ erhalten aber für jedes im Algorithmus *KeyGen* gewählte Polynom vom Grad $k-1$ jeweils eine Stützstelle, so dass mittels Lagrange-Interpolation $\sum L_j \hat{u}_j = u$ folgt. Deshalb ist $M \lfloor \frac{q}{2} \rfloor + u^T s - \sum L_j e_j^T A_j^T s = M \lfloor \frac{q}{2} \rfloor$. Der Algorithmus *Dec* würde daher immer korrekt entschlüsseln, wenn die Elemente des Schlüsseltextes die Form $c_0 = u^T s + M \lfloor \frac{q}{2} \rfloor$ sowie $c_j = A_j^T s$ haben. Das Verfahren wäre aber nicht sicher, da zur Nutzung des Algorithmus *TrapGen* $m \geq n$ gewählt werden muss und somit aus den Gleichungen $c_j = A_j^T s$ der Vektor s mit dem Gaußschen Eliminationsverfahren rekonstruiert werden kann. Weil u Teil des öffentlichen Parameters ist, ließe sich sofort $u^T s$ und damit die Nachricht M ohne die Kenntnis eines geheimen Schlüssels berechnen. Jedem Element des Schlüsseltextes wird deswegen zufälliges, additives Rauschen hinzugefügt. Bereits in Abschnit 4.3 wurde festgestellt, dass dann die Rekonstruktion von s nicht mehr so einfach möglich ist.

Um überhaupt die Algorithmen *TrapGen* und *SamplePre* benutzen zu können, müssen bestimmte Einschränkungen an die Parameter gelten, welche nun untersucht werden. Dafür sei im Rest des Abschnitts $f : [1, \infty) \to \mathbb{R}, m \mapsto (\log m)^{\frac{1}{2}+\delta}$ für ein festes $\delta \in \left(0, \frac{1}{2}\right)$. Dann ist $f \in \omega\left(\sqrt{\log m}\right)$. Für die Verwendung des Algorithmus *TrapGen* muss zunächst $m \geq 5n \log q$ sein. Nach Theorem 6.2.4 gibt *TrapGen* dann bei einem Aufruf eine Matrix $A \in \mathbb{Z}_q^{n \times m}$ sowie eine Matrix $T_A \in \mathbb{Z}^{m \times m}$ aus, wobei die Spalten von T_A eine Basis von $\Lambda_q^{\perp}(A)$ bilden. Bis auf eine in n vernachlässigbare Wahrscheinlichkeit ist zudem wegen Lemma 3.2.2 $\|\widetilde{T_A}\| \leq \|T_A\| \leq mf(n) \leq mf(m)$. Wird dann $\sigma := m(f(m))^2$ gesetzt, so gilt $\sigma \geq \|\widetilde{T_A}\| f(m)$. Da außerdem die Verteilung von A statistisch nah zur Gleichverteilung auf $\mathbb{Z}_q^{n \times m}$ ist, kann *SamplePre* wegen Lemma 6.2.5 bis auf eine in n vernachlässigbare Wahrscheinlichkeit für alle $u \in \mathbb{Z}_q^n$ auf der Ein-

gabe (A, T_A, u, σ) ausgeführt werden. Von nun an sei deshalb $\sigma = m\,(f\,(m))^2$ sowie $m \geq 5n \log q$. Um Schritt 3c) im Algorithmus *Dec* berechnen zu können, muss außerdem $q \geq l$ gelten.

Im Rest dieses Abschnitts soll die Korrektheit des obigen Verfahrens gezeigt werden. Zur Vorbereitung wird zunächst eine Funktion definiert. Daraufhin werden einige ihrer Eigenschaften überprüft sowie vier Lemmata mit nützlichen Abschätzungen bewiesen, bevor dann in Theorem 8.2.8 der Beweis der Korrektheit folgt.

Definition 8.2.2. *Sei $p \in \mathbb{R}_{>0}$. Dann definiere*

$$\begin{aligned} |\cdot|_p : \mathbb{R} &\longrightarrow \left[0, \frac{p}{2}\right], \\ a &\longmapsto \min\{a \mod p, p - (a \mod p)\}. \end{aligned}$$

Für $r \in \mathbb{Z}_q$ ist mit dieser Definition $|r|_q = \min\{r, q - r\}$. Demnach muss für die Korrektheit des Verfahrens $|r|_q$ für obiges r aus dem Algorithmus *Dec* analysiert werden. Zunächst werden dafür einige Eigenschaften von $|\cdot|_p$ nachgerechnet.

Lemma 8.2.3. *Für $p \in \mathbb{R}_{>0}$, $z \in \mathbb{Z}$ und $a, b \in \mathbb{R}$ gelten folgende Eigenschaften:*

1. $|a|_p \leq |a|$,
2. $|pa|_p = p|a|_1$,
3. $|a + b|_p \leq |a|_p + |b|_p$,
4. $|-a|_p = |a|_p$,
5. $|za|_p \leq |z||a|_p$.

Beweis.

1. Falls $|a| \geq \frac{p}{2}$ ist, so gilt offensichtlich $|a|_p \leq \frac{p}{2} \leq |a|$. Betrachte daher nun den Fall, dass $|a| < \frac{p}{2}$ ist. Gilt $0 \leq a < \frac{p}{2}$, dann ist $|a|_p = a = |a|$. Für $-\frac{p}{2} < a < 0$ gilt $|a|_p = p - (a + p) = -a = |a|$.

2. Zunächst gilt

$$p\,(a \mod 1) = p\,(a - \lfloor a \rfloor) = pa - p\lfloor a \rfloor = pa - p\left\lfloor \frac{pa}{p} \right\rfloor = pa \mod p.$$

Daraus folgt dann

$$\begin{aligned} p|a|_1 &= p\min\{a \mod 1, 1-(a \mod 1)\} \\ &= \min\{p(a \mod 1), p-p(a \mod 1)\} \\ &= \min\{pa \mod p, p-(pa \mod p)\} = |pa|_p. \end{aligned}$$

3. Es ist klar, dass $|a+b|_p = |(a \mod p)+(b \mod p)|_p$ ist.
   - Fall 1: Seien $a \mod p, b \mod p \in [0, \frac{p}{2}]$.
     - Fall 1.1: Sei $(a \mod p)+(b \mod p) \in [0, \frac{p}{2}]$. Dann ist

$$|a+b|_p = (a \mod p)+(b \mod p) = |a|_p + |b|_p.$$

     - Fall 1.2: Sei $(a \mod p)+(b \mod p) \in (\frac{p}{2}, p)$. Dann ist

$$\begin{aligned} |a+b|_p &= p-((a \mod p)+(b \mod p)) \\ &< \frac{p}{2} < (a \mod p)+(b \mod p) = |a|_p + |b|_p. \end{aligned}$$

     - Fall 1.3: Sei $(a \mod p)+(b \mod p) = p$. Dann ist

$$|a+b|_p = 0 < (a \mod p)+(b \mod p) = |a|_p + |b|_p.$$

   - Fall 2: Seien $a \mod p \in [0, \frac{p}{2}], b \mod p \in (\frac{p}{2}, p)$.
     - Fall 2.1: Sei $(a \mod p)+(b \mod p) \in (\frac{p}{2}, p)$. Dann ist

$$\begin{aligned} |a+b|_p &= p-((a \mod p)+(b \mod p)) \\ &\leq (a \mod p)+p-(b \mod p) = |a|_p + |b|_p. \end{aligned}$$

     - Fall 2.2: Sei $(a \mod p)+(b \mod p) \in \left[p, \frac{3p}{2}\right)$. Dann ist

$$\begin{aligned} |a+b|_p &= (a \mod p)+(b \mod p)-p \\ &< a \mod p < (a \mod p)+p-(b \mod p) = |a|_p + |b|_p. \end{aligned}$$

   - Fall 3: Seien $a \mod p, b \mod p \in (\frac{p}{2}, p)$.

– Fall 3.1: Sei $(a \mod p) + (b \mod p) \in \left(p, \frac{3p}{2}\right]$. Dann ist

$$\begin{aligned}|a+b|_p &= (a \mod p) + (b \mod p) - p \\ &\leq 2p - ((a \mod p) + (b \mod p)) \\ &= p - (a \mod p) + p - (b \mod p) = |a|_p + |b|_p.\end{aligned}$$

– Fall 3.2: Sei $(a \mod p) + (b \mod p) \in \left(\frac{3p}{2}, 2p\right)$. Dann ist

$$\begin{aligned}|a+b|_p &= p - (((a \mod p) + (b \mod p)) - p) \\ &= p - (a \mod p) + p - (b \mod p) = |a|_p + |b|_p.\end{aligned}$$

4. Ist $a \in p\mathbb{Z}$, so ist $a \mod p = 0$ und daher folgt $|a|_p = 0 = |-a|_p$. Sei nun $a \notin p\mathbb{Z}$ sowie $a > 0$. Dann gilt für $n := -\left\lfloor \frac{-a}{p} \right\rfloor$, dass $n \in \mathbb{N}$ und $(-a) \mod p = -a + np$ ist. Außerdem ist $n - 1 = \left\lfloor \frac{a}{p} \right\rfloor$, woraus $a \mod p = a - (n-1)\,p$ folgt. Nach Definition von $|\cdot|_p$ gilt dann

$$\begin{aligned}|a|_p &= \min\{a - (n-1)\,p, p - (a - (n-1)\,p)\} = \min\{a - np + p, -a + np\} \\ &= \min\{-a + np, p - (-a + np)\} = |-a|_p.\end{aligned}$$

Für $a \notin p\mathbb{Z}$ und $a < 0$ ist somit $|-a|_p = |-(-a)|_p = |a|_p$.

5. Für $z = 0$ ist diese Aussage trivial. Sei nun $z > 0$. Dann ist

$$|za|_p = |\underbrace{a + \ldots + a}_{z\text{-mal}}|_p \leq \underbrace{|a|_p + \ldots + |a|_p}_{z\text{-mal}} = z|a|_p.$$

Für $z < 0$ gilt $|za|_p = |(-z)\,a|_p \leq -z|a|_p$.

□

Um $|r|_q$ abzuschätzen, werden nun vier weitere Abschätzungen zur Vorbereitung gezeigt.

Lemma 8.2.4. *Seien $s, t \in \mathbb{R}_{>0}$. Ferner sei $z \in \mathbb{R}$ zufällig $\mathcal{N}(0, s)$-verteilt. Dann ist*

$$\Pr(|z| > ts) < \frac{1}{t} e^{-\frac{t^2}{2}}.$$

Beweis. Bezeichne mit $\varphi_s : \mathbb{R} \to \mathbb{R}_{>0}, x \mapsto \frac{1}{s\sqrt{2\pi}} e^{-\frac{x^2}{2s^2}}$ die Wahrscheinlichkeitsdichte von $\mathcal{N}(0, s)$. Damit gilt, dass

$$\begin{aligned}\Pr(z > ts) &= \int_{ts}^{\infty} \varphi_s(x)\, dx \leq \frac{s}{t} \int_{ts}^{\infty} \frac{x}{s^2} \varphi_s(x)\, dx = -\frac{s}{t} \int_{ts}^{\infty} \varphi_s'(x)\, dx \\ &= \frac{s}{t} \varphi_s(ts) = \frac{s}{t} \frac{1}{s\sqrt{2\pi}} e^{-\frac{t^2 s^2}{2s^2}} = \frac{1}{t\sqrt{2\pi}} e^{-\frac{t^2}{2}}\end{aligned}$$

ist, und wegen der Symmetrie von φ_s folgt

$$\Pr(|z| > ts) \leq \frac{2}{t\sqrt{2\pi}} e^{-\frac{t^2}{2}} < \frac{1}{t} e^{-\frac{t^2}{2}}.$$

□

Lemma 8.2.5. *Sei $x \in \mathbb{Z}_q$ zufällig $\overline{\Psi}_\alpha$-verteilt. Dann gilt*

$$|x|_q \leq \frac{1}{2} + q\sqrt{m}\frac{\alpha}{\sqrt{2\pi}}$$

bis auf eine in m vernachlässigbare Wahrscheinlichkeit.

Beweis. Da x zufällig $\overline{\Psi}_\alpha$-verteilt ist, existiert ein $y \in [0, 1)$, welches zufällig Ψ_α-verteilt ist, so dass $x = \lfloor qy \rceil \mod q$ gilt. Damit existiert aber auch ein zufällig $\mathcal{N}\left(0, \frac{\alpha}{\sqrt{2\pi}}\right)$-verteiltes $w \in \mathbb{R}$ mit $y = w \mod 1$. Aufgrund von Lemma 8.2.4 gilt $\Pr\left(|w| > \sqrt{m}\frac{\alpha}{\sqrt{2\pi}}\right) < \frac{1}{\sqrt{m}} e^{-\frac{m}{2}}$, wobei letzterer Ausdruck vernachlässigbar in m ist. Bis auf eine in m vernachlässigbare Wahrscheinlichkeit gilt demnach

$$\begin{aligned}|x|_q = |\lfloor qy \rceil \mod q|_q &= |\lfloor qy \rceil|_q = |\lfloor qy \rceil - qy + qy|_q \\ &\leq |\lfloor qy \rceil - qy|_q + |qy|_q \leq |\lfloor qy \rceil - qy| + q|y|_1 \\ &\leq \frac{1}{2} + q|w \mod 1|_1 = \frac{1}{2} + q|w|_1 \\ &\leq \frac{1}{2} + q|w| \leq \frac{1}{2} + q\sqrt{m}\frac{\alpha}{\sqrt{2\pi}}.\end{aligned}$$

□

Folgendes Lemma wird als Lemma 1 in [ABV+12] ohne Beweis aufgeführt.

Lemma 8.2.6. *Sei $e \in \mathbb{Z}^m$ und $x \in \mathbb{Z}_q^m$ zufällig $\overline{\Psi}_\alpha^m$-verteilt. Dann gilt für alle*

$g \in \omega\left(\sqrt{\log m}\right)$ *mit* $g(m) > 0$

$$|e^T x|_q \leq \|e\| q\alpha g(m) + \|e\| \frac{\sqrt{m}}{2}$$

bis auf eine in m *vernachlässigbare Wahrscheinlichkeit.*

Beweis. Da x zufällig $\overline{\Psi}_\alpha^m$-verteilt ist, existiert ein $y \in [0,1)^m$, welches zufällig Ψ_α^m-verteilt ist, so dass $x = \lfloor qy \rceil \mod q$ gilt. Damit existiert aber auch ein zufällig $\mathcal{N}\left(0, \frac{\alpha}{\sqrt{2\pi}}\right)^m$-verteiltes $w \in \mathbb{R}^m$ mit $y = w \mod 1$.

Seien nun $s_1, s_2 \in \mathbb{R}_{>0}$. Zudem seien $z_1, z_2 \in \mathbb{R}$ unabhängig voneinander gewählt, wobei z_1 zufällig $\mathcal{N}(0, s_1)$-verteilt und z_2 zufällig $\mathcal{N}(0, s_2)$-verteilt ist. Dann ist $z_1 + z_2$ zufällig $\mathcal{N}\left(0, \sqrt{s_1^2 + s_2^2}\right)$-verteilt und für alle $c \in \mathbb{R} \setminus \{0\}$ ist cz_1 zufällig $\mathcal{N}(0, |c| s_1)$-verteilt [Ass00]. Weil ohne Beschränkung der Allgemeinheit $e \neq 0$ gilt, folgt damit, dass $e^T w = \sum_{i=1}^m e_i w_i$ zufällig normalverteilt mit Standardabweichung $\sqrt{\sum_{i=1}^m e_i^2 \frac{\alpha^2}{2\pi}} = \|e\| \frac{\alpha}{\sqrt{2\pi}}$ ist. $e^T w$ ist also zufällig $\mathcal{N}\left(0, \|e\| \frac{\alpha}{\sqrt{2\pi}}\right)$-verteilt. Mit Lemma 8.2.4 gilt daher für alle $g \in \omega\left(\sqrt{\log m}\right)$ mit $g(m) > 0$

$$\begin{aligned}
\Pr\left(|e^T w| > \|e\| \alpha g(m)\right) &= \Pr\left(|e^T w| > \sqrt{2\pi} g(m) \|e\| \frac{\alpha}{\sqrt{2\pi}}\right) \\
&< \frac{1}{\sqrt{2\pi} g(m)} e^{-\frac{\left(\sqrt{2\pi} g(m)\right)^2}{2}} \\
&= \frac{1}{\sqrt{2\pi} g(m)} e^{-\pi g^2(m)},
\end{aligned}$$

wobei letzterer Ausdruck vernachlässigbar in m ist, da $g^2 \in \omega(\log m)$ ist.

Außerdem ist für alle $i \in \{1, \ldots, m\}$ $|\lfloor qy_i \rceil - qy_i| \leq \frac{1}{2}$, woraus

$$\|\lfloor qy \rceil - qy\| = \sqrt{\sum_{i=1}^m \left(\lfloor qy_i \rceil - qy_i\right)^2} \leq \sqrt{m \frac{1}{4}} = \frac{\sqrt{m}}{2}$$

folgt. Damit ergibt sich mit der Cauchy-Schwarzschen Ungleichung

$$|e^T \lfloor qy \rceil - e^T (qy)| = |\langle e, \lfloor qy \rceil \rangle - \langle e, qy \rangle| = |\langle e, \lfloor qy \rceil - qy \rangle| \leq \|e\| \|\lfloor qy \rceil - qy\| \leq \|e\| \frac{\sqrt{m}}{2}.$$

Insgesamt gilt bis auf eine in m vernachlässigbare Wahrscheinlichkeit für alle

$g \in \omega\left(\sqrt{\log m}\right)$ mit $g(m) > 0$, dass

$$\begin{aligned}
|e^T x|_q &= |e^T \lfloor qy \rceil \mod q|_q \\
&= \left|\left(e^T \lfloor qy \rceil \mod q\right) - \left(e^T (qy) \mod q\right) + \left(e^T (qy) \mod q\right)\right|_q \\
&\leq \left|\left(e^T \lfloor qy \rceil \mod q\right) - \left(e^T (qy) \mod q\right)\right|_q + |e^T (qy) \mod q|_q \\
&= |e^T \lfloor qy \rceil - e^T (qy)|_q + |qe^T y \mod q|_q \\
&\leq |e^T \lfloor qy \rceil - e^T (qy)| + |qe^T y|_q \\
&\leq \|e\| \frac{\sqrt{m}}{2} + q|e^T y|_1 \\
&= \|e\| \frac{\sqrt{m}}{2} + q|e^T w \mod 1|_1 \\
&= \|e\| \frac{\sqrt{m}}{2} + q|e^T w|_1 \\
&\leq \|e\| \frac{\sqrt{m}}{2} + q|e^T w| \\
&\leq \|e\| \frac{\sqrt{m}}{2} + \|e\| q\alpha g(m)
\end{aligned}$$

ist. □

Nun wird eine letzte Normabschätzung durchgeführt. Dafür sei erinnert, dass der Algorithmus *TrapGen* Gitterbasen ausgibt, deren Norm nach Gram-Schmidtscher Orthogonalisierung bis auf eine vernachlässigbare Wahrscheinlichkeit durch $mf(m)$ beschränkt ist, und dass für diesen Abschnitt $m \geq 5n \log q$ sowie $\sigma = m(f(m))^2$ gilt, wobei $f : [1, \infty) \to \mathbb{R}, m \mapsto (\log m)^{\frac{1}{2}+\delta}$ für ein festes $\delta \in \left(0, \frac{1}{2}\right)$ ist.

Lemma 8.2.7. *Sei $A \in \mathbb{Z}_q^{n \times m}$ und $T_A \in \mathbb{Z}^{m \times m}$, wobei die Spalten von T_A eine Basis von $\Lambda_q^\perp(A)$ bilden und $\|\widetilde{T_A}\| \leq mf(m)$ gilt. Zudem sei $u \in \mathbb{Z}_q^n$ mit $\Lambda_q^u(A) \neq \emptyset$ und $\sigma = m(f(m))^2$. Ist $e \in \mathbb{Z}^m$ eine Ausgabe von* $\mathrm{SamplePre}(A, T_A, u, \sigma)$*, so gilt*

$$\|e\| \leq m^{\frac{3}{2}} (f(m))^2$$

bis auf eine in m vernachlässigbare Wahrscheinlichkeit.

Beweis. Nach Lemma 5.3.4 gibt es wegen $f \in \omega\left(\sqrt{\log m}\right)$ und $f(m) \geq f(5) > 0$ ein in m vernachlässigbares $\epsilon(m) \in \mathbb{R}_{>0}$, so dass

$$\eta_{\epsilon(m)}\left(\Lambda_q^\perp(A)\right) \leq f(m) \|\widetilde{T_A}\| \leq m(f(m))^2 = \sigma$$

ist. Bei Betrachten des Beweises von Lemma 6.2.3 lässt sich feststellen, dass es für einen Vektor $e \in \mathbb{Z}^m$, der eine Ausgabe von $\mathit{SamplePre}(A, T_A, u, \sigma)$ ist, ein $x \in \Lambda_q^{\perp}(A)$ und ein $t \in \Lambda_q^u(A)$ gibt, so dass $e = x + t$ und die Verteilung von x statistisch nah zu $\mathcal{D}_{\Lambda_q^{\perp}(A),\sigma,-t}$ ist. Weil $\epsilon(m) \in (0,1)$ angenommen werden kann, folgt aus Lemma 5.3.3, dass bis auf eine in m vernachlässigbare Wahrscheinlichkeit $\|e\| = \|x + t\| \leq \sigma\sqrt{m} = m^{\frac{3}{2}} (f(m))^2$ ist. □

Nach all diesen Abschätzungen wird nun bewiesen, dass obiges Verschlüsselungsverfahren korrekt ist.

Theorem 8.2.8. *Das Verfahren 8.2.1 ist bei geeigneter Wahl der Parameter n, m, q, l, k, σ und α korrekt.*

Beweis. Es ist hier zu zeigen, dass obiges Verfahren bis auf eine in λ vernachlässigbare Wahrscheinlichkeit bei geeigneter Parameterwahl korrekt entschlüsselt. Hierbei ist nur der Fall zu betrachten, dass der Algorithmus *Dec* einen Schlüsseltext und einen geheimen Schlüssel in der Eingabe hat, die zu Identitäten gehören, die genügend Übereinstimmungen haben. Dazu wird im Folgenden das im Algorithmus *Dec* definierte $r \in \mathbb{Z}_q$ analysiert. Alle Variablen seien dabei so definiert, wie sie in den Algorithmen *Setup*, *KeyGen*, *Enc* und *Dec* beschrieben wurden.

Weil für alle $j \in \{1, \ldots, l\}$ mit $\mathrm{id}_j = 1$ gilt, dass $e_j \in \Lambda_q^{\hat{u}_j}(A_j)$ ist, folgt für alle $j \in S$, dass $\hat{u}_j = A_j e_j$ ist. Außerdem haben die im Algorithmus *KeyGen* gewählten Polynome $p_1, \ldots, p_n$ den Grad $k - 1$, weshalb sich diese mit k gegebenen, paarweise verschiedenen Stützstellen mittels Lagrange-Interpolation berechnen lassen. Deswegen gilt $\sum_{j \in S} L_j \hat{u}_j = u$. Damit lässt sich nachrechnen, dass

$$\begin{aligned}
r &= c_0 - \sum_{j \in S} L_j e_j^T c_j \\
&= u^T s + Dx + M \left\lfloor \frac{q}{2} \right\rfloor - \sum_{j \in S} L_j e_j^T \left(A_j^T s + D x_j \right) \\
&= u^T s + Dx + M \left\lfloor \frac{q}{2} \right\rfloor - \sum_{j \in S} L_j e_j^T A_j^T s - \sum_{j \in S} L_j e_j^T D x_j \\
&= u^T s - \left(\sum_{j \in S} L_j A_j e_j \right)^T s + Dx - \sum_{j \in S} D L_j e_j^T x_j + M \left\lfloor \frac{q}{2} \right\rfloor
\end{aligned}$$

$$
\begin{aligned}
&= u^T s - \left(\sum_{j\in S} L_j \hat{u}_j\right)^T s + Dx - \sum_{j\in S} DL_j e_j^T x_j + M\left\lfloor \frac{q}{2}\right\rfloor \\
&= u^T s - u^T s + Dx - \sum_{j\in S} DL_j e_j^T x_j + M\left\lfloor \frac{q}{2}\right\rfloor \\
&= M\left\lfloor \frac{q}{2}\right\rfloor + Dx - \sum_{j\in S} DL_j e_j^T x_j
\end{aligned}
$$

ist. Dementsprechend ist $|r|_q = \left|M\left\lfloor \frac{q}{2}\right\rfloor + Dx - \sum_{j\in S} DL_j e_j^T x_j\right|_q$. Deshalb muss nur noch gezeigt werden, dass $\left|Dx - \sum_{j\in S} DL_j e_j^T x_j\right|_q < \left\lfloor \frac{q}{4}\right\rfloor$ bis auf eine in λ vernachlässigbare Wahrscheinlichkeit ist, weil dann der Algorithmus *Dec* bis auf eine vernachlässigbare Wahrscheinlichkeit die korrekte Nachricht M ausgibt.

Um dies einzusehen, betrachte zunächst den Fall $M = 0$. Dann ist bis auf eine vernachlässigbare Wahrscheinlichkeit

$$\min\{r, q-r\} = |r|_q = \left|Dx - \sum_{j\in S} DL_j e_j^T x_j\right|_q < \left\lfloor \frac{q}{4}\right\rfloor$$

und der Algorithmus *Dec* gibt bis auf eine vernachlässigbare Wahrscheinlichkeit korrekterweise 0 aus.

Betrachte nun den Fall $M = 1$. Damit obige Ungleichung überhaupt sinnvoll sein kann, muss $q \geq 5$ gelten. Außerdem ist bis auf eine vernachlässigbare Wahrscheinlichkeit

$$r = \left\lfloor \frac{q}{2}\right\rfloor + Dx - \sum_{j\in S} DL_j e_j^T x_j \in \left[\left\lfloor \frac{q}{2}\right\rfloor, \left\lfloor \frac{q}{2}\right\rfloor + \left\lfloor \frac{q}{4}\right\rfloor\right) \cup \left(\left\lfloor \frac{q}{2}\right\rfloor - \left\lfloor \frac{q}{4}\right\rfloor, \left\lfloor \frac{q}{2}\right\rfloor\right].$$

Nun werden zwei Fälle unterschieden.

1. $r \in \left[\left\lfloor \frac{q}{2}\right\rfloor, \left\lfloor \frac{q}{2}\right\rfloor + \left\lfloor \frac{q}{4}\right\rfloor\right)$. Dann ist

$$|r|_q \geq \left|\left\lfloor \frac{q}{2}\right\rfloor + \left\lfloor \frac{q}{4}\right\rfloor\right|_q = q - \left\lfloor \frac{q}{2}\right\rfloor - \left\lfloor \frac{q}{4}\right\rfloor > q - \frac{q}{2} - \frac{q}{4} = \frac{q}{4} > \left\lfloor \frac{q}{4}\right\rfloor.$$

2. $r \in \left(\left\lfloor \frac{q}{2}\right\rfloor - \left\lfloor \frac{q}{4}\right\rfloor, \left\lfloor \frac{q}{2}\right\rfloor\right]$. Dann ist

$$|r|_q \geq \left|\left\lfloor \frac{q}{2}\right\rfloor - \left\lfloor \frac{q}{4}\right\rfloor\right|_q = \left\lfloor \frac{q}{2}\right\rfloor - \left\lfloor \frac{q}{4}\right\rfloor \geq \left\lfloor \frac{q}{4}\right\rfloor,$$

weil $2\lfloor\frac{q}{4}\rfloor \leq 2\left(\frac{q}{4}-\frac{1}{4}\right) = \frac{q}{2}-\frac{1}{2} \leq \lfloor\frac{q}{2}\rfloor$ gilt.

Deswegen ist bis auf eine vernachlässigbare Wahrscheinlichkeit

$$\min\{r, q-r\} = |r|_q \geq \left\lfloor \frac{q}{4} \right\rfloor.$$

Daher gibt der Algorithmus *Dec* im Fall $M = 1$ wie gewünscht bis auf eine vernachlässigbare Wahrscheinlichkeit 1 aus.

Um die Korrektheit zu zeigen, sind also im Folgenden die Parameter n, m, q, l, k und α so zu wählen, dass $\left|Dx - \sum_{j\in S} DL_j e_j^T x_j\right|_q < \lfloor\frac{q}{4}\rfloor$ bis auf eine in λ vernachlässigbare Wahrscheinlichkeit ist. σ wurde bereits durch $\sigma = m\left(f\left(m\right)\right)^2$ festgesetzt. Für die weiteren Parameter wird zunächst überlegt, dass für alle $j \in S$ $DL_j \in \mathbb{Z}$ ist. Sei also $j \in S$ und setze $d_j := \prod_{i\in S\setminus\{j\}} (j-i)$. Weil für jedes $i \in S \setminus \{j\}$ gilt, dass $|j-i| \in \{1, \ldots, (l-1)\}$ ist und höchstens zwei verschiedene Elemente aus S denselben Abstand zu j haben, ergibt sich $d_j \mid (l!)^2$. Damit ist $DL_j = \frac{(l!)^2}{d_j} \prod_{i\in S\setminus\{j\}} (-i) \in \mathbb{Z}$ und außerdem ist $|DL_j| \leq (l!)^2 \prod_{i\in S\setminus\{j\}} i \leq (l!)^3$. Damit folgt nun

$$\begin{aligned}\left|Dx - \sum_{j\in S} DL_j e_j^T x_j\right|_q &\leq |Dx|_q + \sum_{j\in S} |DL_j e_j^T x_j|_q \leq D|x|_q + \sum_{j\in S} |DL_j||e_j^T x_j|_q \\ &\leq D|x|_q + \sum_{j\in S} (l!)^3 |e_j^T x_j|_q.\end{aligned}$$

Für die Korrektheit des Verfahrens reicht es demnach zu zeigen, dass bis auf eine in λ vernachlässigbare Wahrscheinlichkeit $D|x|_q + \sum_{j\in S} (l!)^3 |e_j^T x_j|_q < \lfloor\frac{q}{4}\rfloor$ ist.

Weil $\lfloor\frac{q}{4}\rfloor \geq \frac{q}{4} - \frac{3}{4}$ ist, reicht es für die Korrektheit des Verfahrens zu zeigen, dass $D|x|_q + \sum_{j\in S} (l!)^3 |e_j^T x_j|_q + \frac{3}{4} < \frac{q}{4}$ bis auf eine in λ vernachlässigbare Wahrscheinlichkeit ist. Im nächsten Kapitel wird erklärt, warum dieses Verfahren sicher ist. Für diese Überlegung wird benötigt, dass $n = \lambda$ und $\alpha > \frac{2\sqrt{n}}{q}$ ist. Um diese Anforderung im nächsten Kapitel zu erfüllen, sei von nun an $\alpha := \frac{3\sqrt{n}}{q}$. Außerdem sei $m := \lceil 5n \log q \rceil$, da $m \geq 5n \log q$ gelten soll. Zusammen mit Lemma 8.2.5, Lemma 8.2.6 und Lemma 8.2.7 ergibt sich bis auf eine in m – und damit auch bis auf eine in n – vernachlässigbare

Wahrscheinlichkeit

$$
\begin{aligned}
&D|x|_q + \sum_{j\in S} (l!)^3 \left|e_j^T x_j\right|_q + \frac{3}{4} \\
&\leq D\left(\frac{1}{2} + q\sqrt{m}\frac{\alpha}{\sqrt{2\pi}}\right) + l\,(l!)^3 \left\|e_j\right\| \left(\frac{\sqrt{m}}{2} + q\alpha f\,(m)\right) + \frac{3}{4} \\
&\leq D\left(\frac{1}{2} + q\sqrt{m}\frac{\alpha}{\sqrt{2\pi}}\right) + \frac{3}{4} + l\,(l!)^3\, m^{\frac{3}{2}}\,(f\,(m))^2 \left(\frac{\sqrt{m}}{2} + q\alpha f\,(m)\right) \\
&= (l!)^2 \left(\frac{1}{2} + \frac{3\sqrt{n}\sqrt{m}}{\sqrt{2\pi}}\right) + \frac{3}{4} + l\,(l!)^3\, m^{\frac{3}{2}}\,(f\,(m))^2 \left(\frac{\sqrt{m}}{2} + 3\sqrt{n} f\,(m)\right) \\
&\leq (l!)^2 \left(\frac{1}{2} + \frac{3\sqrt{n}\sqrt{5n\log q + 1}}{\sqrt{2\pi}}\right) + \frac{3}{4} \\
&\quad + l\,(l!)^3\,(5n\log q + 1)^{\frac{3}{2}}\,(f\,(5n\log q + 1))^2 \left(\frac{\sqrt{5n\log q + 1}}{2} + 3\sqrt{n} f\,(5n\log q + 1)\right) \\
&= (l!)^2 \left(\frac{1}{2} + \frac{3\sqrt{n}\sqrt{5n\log q + 1}}{\sqrt{2\pi}}\right) + \frac{3}{4} \\
&\quad + l\,(l!)^3\,(5n\log q + 1)^{\frac{3}{2}} \left((\log\,(5n\log q + 1))^{\frac{1}{2}+\delta}\right)^2 \\
&\quad \cdot \left(\frac{\sqrt{5n\log q + 1}}{2} + 3\sqrt{n}\,(\log\,(5n\log q + 1))^{\frac{1}{2}+\delta}\right) \\
&\leq (l!)^2 \left(\frac{1}{2} + \frac{3\sqrt{n}\sqrt{5n\log q + 1}}{\sqrt{2\pi}}\right) + \frac{3}{4} \\
&\quad + l\,(l!)^3\,(5n\log q + 1)^{\frac{3}{2}}\,(\log\,(5n\log q + 1))^2 \\
&\quad \cdot \left(\frac{\sqrt{5n\log q + 1}}{2} + 3\sqrt{n}\log\,(5n\log q + 1)\right) =: \beta_1\,(l, n, q)\,.
\end{aligned}
$$

Es reicht also nun zu zeigen, dass bei geeigneter Wahl der Parameter $\beta_1\,(l, n, q) < \frac{q}{4}$ gilt. Dies ist äquivalent zu

$$
\begin{aligned}
\frac{q}{l\,(l!)^3} &> \frac{4\beta_1\,(l, n, q)}{l\,(l!)^3} \\
&= \frac{1}{l\,(l!)}\left(2 + \frac{12\sqrt{n}\sqrt{5n\log q + 1}}{\sqrt{2\pi}}\right) + \frac{3}{l\,(l!)^3} \\
&\quad + 4\,(5n\log q + 1)^{\frac{3}{2}}\,(\log\,(5n\log q + 1))^2 \\
&\quad \cdot \left(\frac{\sqrt{5n\log q + 1}}{2} + 3\sqrt{n}\log\,(5n\log q + 1)\right).
\end{aligned}
$$

Dafür reicht es wiederum zu zeigen, dass

$$\begin{aligned}\frac{q}{l\left(l!\right)^3} &> \left(2+\frac{12\sqrt{n}\sqrt{5n\log q+1}}{\sqrt{2\pi}}\right)+3\\ &\quad+4\left(5n\log q+1\right)^{\frac{3}{2}}\left(\log\left(5n\log q+1\right)\right)^2\\ &\quad\cdot\left(\frac{\sqrt{5n\log q+1}}{2}+3\sqrt{n}\log\left(5n\log q+1\right)\right)\\ &= 5+\frac{12\sqrt{n}\sqrt{5n\log q+1}}{\sqrt{2\pi}}\\ &\quad+4\left(5n\log q+1\right)^{\frac{3}{2}}\left(\log\left(5n\log q+1\right)\right)^2\\ &\quad\cdot\left(\frac{\sqrt{5n\log q+1}}{2}+3\sqrt{n}\log\left(5n\log q+1\right)\right)=:\beta_2\left(n,q\right)\end{aligned}$$

erfüllt ist.

Um große Sicherheit zu gewährleisten, würde aktuell bei einer Implementierung des Verfahrens $n=\lambda>100$ gewählt werden. Dies wird in Abschnitt 9.2 begründet. Dann könnte $q\in\left[2^{11}n^6l\left(l!\right)^3,2^{12}n^6l\left(l!\right)^3\right]$ eine mögliche Wahl sein, um obige Ungleichung zu erfüllen. Dass mit dieser Wahl tatsächlich die Ungleichung erfüllt ist, wird im Folgenden unter der Annahme $n\geq 100$ begründet. Zum einen ist $\frac{q}{l(l!)^3}\geq 2^{11}n^6$ und zum anderen gilt $q\leq 2^{12}n^7\left(n!\right)^3\leq 2^{12}n^{3n+7}$, woraus $\log q\leq 12+(3n+7)\log n=3n\log n+7\log n+12$ folgt. Damit ergibt sich

$$\begin{aligned}5n\log q+1 &\leq 15n^2\log n+35n\log n+60n+1\leq 15n^2\log n+35n\log n+61n\\ &\leq 15n^2\log n+\frac{35}{100}n^2\log n+\frac{61}{100\log 100}n^2\log n\leq 16n^2\log n,\end{aligned}$$

weshalb auch $\log\left(5n\log q+1\right)\leq 4+2\log n+\log\log n$ gilt. Dies wird nun in den obigen Ausdruck eingesetzt. Somit folgt

$$\begin{aligned}\beta_2\left(n,q\right) &\leq 5+\frac{12\sqrt{n}\sqrt{16n^2\log n}}{\sqrt{2\pi}}\\ &\quad+4\left(16n^2\log n\right)^{\frac{3}{2}}\left(4+2\log n+\log\log n\right)^2\\ &\quad\cdot\left(\frac{\sqrt{16n^2\log n}}{2}+3\sqrt{n}\left(4+2\log n+\log\log n\right)\right)=:\beta_3\left(n\right)\end{aligned}$$

und es reicht daher zu zeigen, dass $2^{11}n^6>\beta_3\left(n\right)$ ist, um $\frac{q}{l(l!)^3}>\beta_2\left(n,q\right)$ und da-

mit die Korrektheit des Verfahrens zu erhalten. Es lässt sich sofort nachrechnen, dass $2^{11}100^6 > \beta_3(100)$ ist. Um $2^{11}n^6 > \beta_3(n)$ für alle $n \geq 100$ zu erhalten, betrachte $\frac{\beta_3(n)}{2^{11}n^6}$ und stelle nach komplettem Ausmultiplizieren fest, dass alle Summanden monoton fallende Funktionen in n auf dem Bereich $\mathbb{R}_{\geq 100}$ sind. Demnach ist $\frac{\beta_3(n)}{2^{11}n^6}$ ebenfalls monoton fallend, weshalb für alle $n \geq 100$ gilt, dass $\frac{\beta_3(n)}{2^{11}n^6} \leq \frac{\beta_3(100)}{2^{11}100^6} < 1$ ist.

Die Parameter können also wie folgt gewählt werden, damit das Fuzzy Identitätsbasierte Verschlüsselungsverfahren 8.2.1 korrekt ist.

- $n := \lambda \in \mathbb{N}$ mit $n \geq 100$
- $k, l \in \mathbb{N}$ mit $k \leq l \leq n$
- $q \in \left[2^{11}n^6 l\,(l!)^3, 2^{12}n^6 l\,(l!)^3\right]$ Primzahl
- $m := \lceil 5n \log q \rceil$
- $\alpha := \frac{3\sqrt{n}}{q}$
- $\sigma := m\left((\log m)^{\frac{1}{2}+\delta}\right)^2 = m\,(\log m)^{2\delta+1}$ für ein festes $\delta \in \left(0, \frac{1}{2}\right)$

Diese Parameterwahl erfüllt zudem offensichtlich die Bedingungen, die an die Parameter ganz zu Beginn dieses Abschnitts auf Seite 74 gestellt wurden. □

Auch für kleinere Werte von n können Wertebereiche für q gefunden werden, so dass das Verfahren korrekt ist. Werden die unteren Schranken für n immer kleiner, so müssen die Faktoren im Wertebereich von q größer werden. So kann für $n \geq 11$ $q \in \left[2^{15}n^6 l\,(l!)^3, 2^{16}n^6 l\,(l!)^3\right]$ gewählt werden. Für $n \geq 3$ ist $q \in \left[2^{19}n^6 l\,(l!)^3, 2^{20}n^6 l\,(l!)^3\right]$ eine mögliche Wahl. Für ein beliebiges $n \in \mathbb{N}$ kann $q \in \left[2^{23}n^6 l\,(l!)^3, 2^{24}n^6 l\,(l!)^3\right]$ sein. Diese Parameterwahlen können ähnlich wie die Wahl für $n \geq 100$ am Schluss des obigen Korrektheitsbeweises begründet werden.

Ferner sei hier erläutert, inwiefern sich Verfahren 8.2.1 von dem in [ABV+12] unterscheidet. Zum einen ist dort die Menge der Überschneidungen zwischen zwei Identitäten id und id′ definiert als $\{j \mid j \in \{1, \ldots, l\}, \mathrm{id}_j = \mathrm{id}'_j\}$. Hier werden nur komponentenweise Überschneidungen berücksichtigt, bei denen beide Komponenten 1 sind, was an Schritt 1 im Algorithmus *Dec* erkennbar ist. Beide Definitionen von Überschneidungen sind sinnvoll. Die Definition, die in dieser Arbeit verwendet wird, sorgt für eine bessere Vergleichbarkeit zum Verfahren aus dem nächsten Abschnitt und für kürzere öffentliche Parameter, Schlüssel und Schlüsseltexte. So werden in [ABV+12] im geheimen Schlüssel

alle Vektoren $e_1, \ldots, e_l$ sowie im Schlüsseltext alle $c_1, \ldots, c_l$ benötigt. Um solche Vektoren zu berechnen, auch wenn die entsprechende Komponente der Identität 0 ist, müssen doppelt so viele Matrizen im Algorithmus *Setup* erstellt werden. Die eine Hälfte der Matrizen wird dann verwendet, falls die Komponente einer Identität 1 ist, und die andere Hälfte, falls die Komponente 0 ist.

Zum anderen wird hier zur Entschlüsselung nur eine k-elementige Teilmenge der Überschneidungen verwendet, weil diese bereits ausreichend ist, und es gibt einen Unterschied im Schritt 3d), da an selbiger Stelle in [ABV+12] $\min\{r, q-r\} < \frac{q}{4}$ geprüft wird. Außerdem wird dort nur $\left|Dx - \sum_{j \in S} DL_j e_j^T x_j\right|_q < \frac{q}{4}$ bis auf eine vernachlässigbare Wahrscheinlichkeit in Abschnit 4.1 gefordert. Dies reicht für die Abschätzung auf Seite 84–85 im obigen Korrektheitsbeweis aber nicht aus. Um dies einzusehen, betrachte den Fall $M = 1$ und $q = 4a + 1$ für ein $a \in \mathbb{N}$. Dann ist bis auf eine vernachlässigbare Wahrscheinlichkeit $r > \lfloor \frac{q}{2} \rfloor - \frac{q}{4}$. Weil $r \in \mathbb{Z}_q$ ist, gilt somit $r \geq \lfloor \frac{q}{2} \rfloor - \lfloor \frac{q}{4} \rfloor$ bis auf eine vernachlässigbare Wahrscheinlichkeit. Insbesondere kann nach dieser Abschätzung auch der Fall $r = \lfloor \frac{q}{2} \rfloor - \lfloor \frac{q}{4} \rfloor = 2a - a = a < \frac{q}{4}$ auftreten und der Algorithmus *Dec* würde 0 statt 1 ausgeben. Um solche Fälle nicht extra behandeln zu müssen, wurden Schritt 3d) im Algorithmus *Dec* sowie die Forderung an $\left|Dx - \sum_{j \in S} DL_j e_j^T x_j\right|_q$ angepasst.

Die Sicherheit von Verfahren 8.2.1 wird in Abschnitt 9.3 gezeigt.

8.3. Verfahren basierend auf bilinearen Abbildungen

Im Folgenden wird ein zweites Fuzzy Identitätsbasiertes Verschlüsselungsverfahren betrachtet. Dieses Verfahren verwendet bilineare Abbildungen auf Gruppen und wurde in Abschnitt 4.1 aus [SW05] definiert. Dafür werden Gruppen verwendet, die folgende Eigenschaft erfüllen.

Definition 8.3.1. *G_1 und G_2 seien Gruppen mit derselben Primordnung.* G_1 hat eine zulässige bilineare Abbildung nach G_2, *falls es eine Abbildung $e : G_1 \times G_1 \to G_2$ gibt, so dass für alle $g \in \hat{G}_1$ Folgendes gilt:*

1. *e ist bilinear, d.h. für alle $a, b \in \mathbb{Z}$ gilt $e\left(g^a, g^b\right) = e(g, g)^{ab}$,*

2. *e ist nicht degeneriert, d.h. $e(g, g) \in \hat{G}_2$,*

3. *e ist effizient berechenbar, d.h. es gibt einen effizienten Algorithmus, der für alle* $h_1, h_2 \in G_1$ *das Gruppenelement* $e(h_1, h_2)$ *berechnet.*

Dabei sei bemerkt, dass formal Familien von Gruppenpaaren $\left((G_1, G_2)_q\right)_{q \in \mathbb{N} \text{ Primzahl}}$ betrachtet werden sollten, wobei q die Ordnung der jeweiligen Gruppen ist, so dass der effiziente Algorithmus als Polynomialzeitalgorithmus in $\lceil \log q \rceil$ zu verstehen ist. Gruppen, die obige Eigenschaft erfüllen, können konstruiert werden. Als bilineare Abbildung kann dafür zum Beispiel eine modifizierte Variante der Weil-Paarung verwendet werden [BF03].

Mit obiger Definition können die Parameter aufgelistet werden, welche für die Beschreibung des Verschlüsselungsverfahrens benötigt werden.

- $\lambda \in \mathbb{N}$ sei der Sicherheitsparameter.
- $q := q(\lambda) \in \mathbb{N}$ sei eine Primzahl, so dass $\log(q)$ polynomiell in λ ist. Die verwendeten Gruppen haben Ordnung q.
- G_1 und G_2 seien Gruppen der Ordnung q, so dass G_1 eine zulässige bilineare Abbildung nach G_2 hat. Die meisten Berechnungen im Verfahren werden in diesen Gruppen ausgeführt.
- $e : G_1 \times G_1 \to G_2$ sei eine zulässige bilineare Abbildung.
- Außerdem sei $g \in \hat{G_1}$. Weil G_1 Primordnung hat, ist somit g ein Erzeuger von G_1. $e(g, g)$ ist ein Erzeuger von G_2, da $e(g, g) \in \hat{G_2}$ ist und G_2 Primordnung hat.
- $l := l(\lambda) \in \mathbb{N}$ mit $l < q$ sei polynomiell in λ und bezeichne die Anzahl aller möglichen Attribute.
- $k \in \mathbb{N}$ mit $k \leq l$ bezeichne die Mindestanzahl an Übereinstimmungen zwischen der Identität eines Teilnehmers und der Identität eines Schlüsseltextes, damit der Teilnehmer den Schlüsseltext entschlüsseln kann.

Nun können die benötigten vier Algorithmen mit Polynomialzeit in λ beschrieben werden.

Verfahren 8.3.2.

- Setup (λ, l, k).
  1. *Für alle* $i \in \{1, \ldots, l\}$ *wähle* $t_i \in \mathbb{Z}_q \setminus \{0\}$ *zufällig gleichverteilt und berechne* $T_i := g^{t_i} \in G_1$.
  2. *Wähle* $y \in \mathbb{Z}_q$ *zufällig gleichverteilt und berechne* $Y := e(g,g)^y \in G_2$.
  3. *Gib sowohl* $PP := (k, Y, T_1, T_2, \ldots, T_l)$ *als auch* $MK := (y, t_1, t_2, \ldots, t_l)$ *aus.*
- KeyGen (PP, MK, id) *mit* $id = (id_1, \ldots, id_l)^T \in \{0,1\}^l$.
  1. *Wähle* $p \in \mathbb{Z}_q[X]$ *zufällig gleichverteilt, so dass* $p(0) = y$ *ist und* p *den Grad* $k-1$ *hat.*
  2. *Für alle* $i \in \{1, \ldots, l\}$ *mit* $id_i = 1$ *berechne* $d_i := \frac{p(i)}{t_i} \in \mathbb{Z}_q$ *und* $D_i := g^{d_i} \in G_1$.
  3. *Gib* $SK_{id} := (id, \{D_i \mid i \in \{1, \ldots, l\}, id_i = 1\})$ *aus.*
- Enc (PP, id', M) *mit* $id' = (id'_1, \ldots, id'_l)^T \in \{0,1\}^l$ *und* $M \in G_2$.
  1. *Wähle* $s \in \mathbb{Z}_q$ *zufällig gleichverteilt.*
  2. *Berechne* $E := MY^s \in G_2$.
  3. *Für alle* $i \in \{1, \ldots, l\}$ *mit* $id'_i = 1$ *berechne* $E_i := T_i^s \in G_1$.
  4. *Gib* $CT_{id'} := (id', E, \{E_i \mid i \in \{1, \ldots, l\}, id'_i = 1\})$ *aus.*
- Dec $(PP, CT_{id'}, SK_{id})$.
  1. *Setze* $J := \{i \mid i \in \{1, \ldots, l\}, id_i = id'_i = 1\}$.
  2. *Wenn* $|J| < k$ *ist, dann gib* $\perp$ *als Platzhalter für „keine Entschlüsselung“ aus.*
  3. *Wenn* $|J| \geq k$ *ist, dann berechne Folgendes:*
     a) *Wähle* $S \subseteq J$ *mit* $|S| = k$.
     b) *Für alle* $i \in S$ *setze* $L_i := \prod_{j \in S \setminus \{i\}} \frac{-j}{i-j} \in \mathbb{Z}_q$.
     c) *Berechne* $M' := E \left(\prod_{i \in S} (e(D_i, E_i))^{L_i} \right)^{-1} \in G_2$.
     d) *Gib* M' *aus.*

In diesem Verfahren wird eine Nachricht M durch Multiplikation mit dem zufällig gleichverteilten Gruppenelement $e(g,g)^{sy} \in G_2$ in ihrem Schlüsseltext verborgen. Mit

den zusätzlichen Schlüsseltextelementen der Form $E_i = g^{st_i}$ und einem passendem geheimen Schlüssel kann jeweils $e(D_i, E_i) = e(g,g)^{sp(i)}$ berechnet werden. Weil mit einem passendem Schlüssel k solche Gruppenelemente berechnet werden können, liefert Lagrange-Interpolation im Exponenten eine Rekonstruktion von $e(g,g)^{sy}$ und somit auch von M. Dies wird im nun folgenden Korrektheitsbeweis ersichtlich, der deutlich einfacher ist als beim Verfahren 8.2.1.

Theorem 8.3.3. *Das Verfahren 8.3.2 ist korrekt.*

Beweis. Das im Algorithmus *KeyGen* gewählte Polynom p hat den Grad $k-1$ und lässt sich deshalb mit k gegebenen, paarweise verschiedenen Stützstellen mittels Lagrange-Interpolation berechnen. Demnach ist $\sum_{i \in S} L_i p(i) = p(0)$ und es gilt

$$\begin{aligned}
M' &= E \left(\prod_{i \in S} \left(e(D_i, E_i) \right)^{L_i} \right)^{-1} \\
&= MY^s \left(\prod_{i \in S} \left(e\left(g^{d_i}, T_i^s \right) \right)^{L_i} \right)^{-1} \\
&= Me(g,g)^{sy} \left(\prod_{i \in S} \left(e\left(g^{\frac{p(i)}{t_i}}, g^{st_i} \right) \right)^{L_i} \right)^{-1} \\
&= Me(g,g)^{sy} \left(\prod_{i \in S} \left(e(g,g)^{sp(i)} \right)^{L_i} \right)^{-1} \\
&= Me(g,g)^{sy} \left(\prod_{i \in S} e(g,g)^{sL_i p(i)} \right)^{-1} \\
&= Me(g,g)^{sy} \left(e(g,g)^{s \sum_{i \in S} L_i p(i)} \right)^{-1} \\
&= Me(g,g)^{sy} \left(e(g,g)^{sp(0)} \right)^{-1} \\
&= Me(g,g)^{sy} \left(e(g,g)^{sy} \right)^{-1} \\
&= M.
\end{aligned}$$

Obiges Verfahren entschlüsselt also immer korrekt und damit ist alles gezeigt. □

Die Sicherheit von Verfahren 8.3.2 wird in Abschnitt 9.4 gezeigt.

9. Sicherheitsannahmen und -beweise

In diesem Kapitel wird die Sicherheit der beiden eingeführten Fuzzy Identitätsbasierten Verschlüsselungsverfahren gezeigt. Außerdem werden die Sicherheitsannahmen, auf denen diese Verfahren beruhen, miteinander verglichen. Zuvor wird das gemeinsame Sicherheitsmodell beider Verfahren eingeführt.

9.1. Sicherheitsmodell

Verschlüsselungsverfahren sollen Vertraulichkeit gewährleisten. Dies bedeutet, dass Nachrichten nur von befugten Empfängern gelesen werden dürfen. Dafür wäre es wünschenswert, dass ein Angreifer aus einem Schlüsseltext gar keine Informationen über den zugehörigen Klartext gewinnen kann. Verfahren, die dies gewährleisten, heißen perfekt sicher. Allerdings sind solche Verfahren sehr ineffizient, da sie mindestens so viele geheime Schlüssel wie mögliche Klartexte benötigen. Formal wird das Konzept der perfekten Sicherheit zum Beispiel in Kapitel 2 von [KL07] eingeführt.

In der modernen Kryptografie wird deshalb für die Sicherheit eines Verfahrens etwas weniger gefordert. Dafür wird ein vemutliches schweres Problem als Sicherheitsannahme verwendet. Wenn die Sicherheitsannahme zutrifft, d.h. das vemutlich schwere Problem ist tatsächlich schwer, dann soll ein probabilistischer Polynomialzeitangreifer bis auf eine vernachlässigbare Wahrscheinlichkeit nichts aus einem Schlüsseltext lernen können. Bei Attributbasierter Verschlüsselung soll zudem gewährleistet werden, dass mehrere Unbefugte selbst bei Zusammenarbeit nichts aus einem Schlüsseltext lernen, den keiner von ihnen alleine entschlüsseln kann.

Die Sicherheit der beiden vorgestellten Fuzzy Identitätsbasierten Verschlüsselungsverfahren wird allerdings mit einem schwächeren Sicherheitsmodell gezeigt. Dieses wird nun formal definiert, bevor darauf eingegangen wird, welche Abstriche dieses Modell mit sich bringt. Das Modell wird in dieser Form auch in Abschnitt 2.1 von [SW05] beschrieben. Der Parameter $k \in \mathbb{N}$ gibt wie auch im letzten Kapitel die Anzahl der Überschneidun-

gen zwischen zwei Identitäten an, die notwendig sind, um eine mit der einen Identität verschlüsselten Nachricht mit einem Schlüssel zur anderen Identität zu entschlüsseln.

Definition 9.1.1. *Das* Fuzzy Selective-ID Spiel *zwischen einem Angreifer und einem Herausforderer hat folgenden Ablauf.*

1. Zielfestlegung. *Der Angreifer gibt die Identität id^* bekannt, bezüglich welcher er herausgefordert werden möchte.*

2. Setup. *Der Herausforderer führt den Algorithmus* Setup *des Fuzzy Identitätsbasierten Verschlüsselungsverfahrens mit einem Sicherheitsparameter $\lambda \in \mathbb{N}$ aus und gibt dem Angreifer die öffentlichen Parameter.*

3. Phase 1. *Der Angreifer darf geheime Schlüssel für Identitäten mit weniger als k Übereinstimmungen mit id^* beim Herausforderer erfragen.*

4. Herausforderung. *Der Angreifer gibt zwei Nachrichten M_0, M_1 der gleichen Länge aus. Der Herausforderer wirft eine Münze $b \in \{0,1\}$ (das heißt er wählt $b \in \{0,1\}$ zufällig gleichverteilt) und gibt dem Angreifer den Schlüsseltext zur Nachricht M_b unter der Identität id^*.*

5. Phase 2. *Phase 1 wird wiederholt.*

6. Schätzung. *Der Angreifer gibt eine Schätzung $b' \in \{0,1\}$ von b aus.*

Da davon ausgegangen werden kann, dass ein Angreifer immer mit mindestens Wahrscheinlichkeit $\frac{1}{2}$ die richtige Schätzung ausgibt, kann der Vorteil eines Angreifers und damit die Sicherheit eines Verfahrens wie folgt definiert werden.

Definition 9.1.2.

- *Der* Vorteil *eines Angreifers im Fuzzy Selective-ID Spiel ist definiert als* $\Pr(b' = b) - \frac{1}{2}$*, wobei die Wahrscheinlichkeit über die zufälligen Wahlen von Angreifer und Herausforderer ist.*

- *Ein Fuzzy Identitätsbasiertes Verschlüsselungsverfahren heißt* sicher im Fuzzy Selective-ID Sicherheitsmodell*, falls jeder Angreifer mit Polynomialzeit in λ höchstens in λ vernachlässigbaren Vorteil im Fuzzy Selective-ID Spiel hat.*

Im Fuzzy Selective-ID Spiel darf der Angreifer sogar zwei Nachrichten festlegen und er soll trotzdem bis auf eine vernachlässigbare Wahrscheinlichkeit nicht entscheiden können, welche von beiden Nachrichten vom Herausforderer verschlüsselt wurde. Somit wird formalisiert, dass ein Angreifer keine Informationen aus einem Schlüsseltext erhält. Die Anfragen in den Phasen 1 und 2 decken den Fall ab, dass mehrere Unbefugte zusammenarbeiten.

Trotzdem liefert dieses Modell noch nicht die eigentlich gewünschte Sicherheit, denn der Angreifer muss zu Beginn seines Angriffs die Identität festlegen, unter welcher später eine der beiden Nachrichten verschlüsselt werden wird. Erst danach werden die öffentlichen Parameter und der Hauptschlüssel von einem Herausforderer generiert. Dann startet der eigentliche Angriff, bei dem der Angreifer geheime Schlüssel beim Herausforderer erfragen darf und entscheiden soll, welche von seinen zwei festgelegten Nachrichten unter der von ihm bereits festgelegten Identität durch den Herausforderer verschlüsselt wurde. In der Praxis wird aber natürlich selten der Fall eintreten, dass ein System erst aufgesetzt wird, nachdem ein Angreifer bekannt gegeben hat, was er angreifen möchte. Daher ist oben definiertes Modells deutlich schwächer als das analoge Modell ohne vorherige Festlegung der anzugreifenden Identität, welches das eigentlich gewünschte Modell ist. Nichtsdestotrotz ist es sinnvoll, Verfahren zu betrachten, deren Sicherheit nur im schwächeren Modell gezeigt werden kann, weil die Techniken aus dem schwächeren Modell für Sicherheitsbeweise im stärkeren Modell verwendet werden können [LW12].

9.2. Sicherheitsannahmen

Die Sicherheit der beiden vorgestellten Verfahren beruht auf je einem Problem, welches als schwer angenommen wird. Dies bedeutet, dass es einen Polynomialzeitalgorithmus für das vermutlich schwere Problem gibt, falls es einen Polynomialzeitangreifer mit nicht vernachlässigbarem Vorteil im obigen Spiel gibt.

Beim Verfahren 8.2.1 ist *Unterscheidungs-LWE*$_{q,\overline{\Psi}_\alpha}$ aus Definition 4.3.7 das vermutlich schwere Problem. In Abschnitt 4.3 wurde bereits diskutiert, warum dieses Problem als schwer angenommen wird. Einer der Hauptgründe ist die dort beschriebene Reduktion von O. Regev, die in Theorem 4.3.8 zusammengefasst wurde. In dieses Theorem werden nun die am Ende des Beweises zu Theorem 8.2.8 erhaltenen Parameter von Seite 88 eingesetzt. Es seien jetzt also $n \in \mathbb{N}$ mit $n \geq 100$, $l \in \mathbb{N}$ mit $l \leq n$, $q \in \left[2^{11} n^6 l\,(l!)^3, 2^{12} n^6 l\,(l!)^3\right]$ eine Primzahl und $\alpha := \frac{3\sqrt{n}}{q}$. Dann ist $l = n^\epsilon$ für ein $\epsilon \in [0,1]$ und für jede in n polyno-

mielle Funktion $p(n)$ gilt

$$\begin{aligned}|p(n) \cdot q| &\leq |p(n)|\, 2^{12} n^6 l\, (l!)^3 \leq |p(n)|\, 2^{12} n^6 l \left(l^l\right)^3 \\ &= |p(n)|\, 2^{12} n^6 l 2^{3l \log l} = |p(n)|\, 2^{12} n^6 n^{\epsilon} 2^{3n^{\epsilon} \epsilon \log n}.\end{aligned}$$

Ist $\epsilon = 0$, so ist $p(n) \cdot q$ polynomiell in n. Für $\epsilon > 0$ ist $p(n) \cdot q \in 2^{\tilde{O}(n^{\epsilon})}$. Weil $\frac{n}{\alpha} = \frac{nq}{3\sqrt{n}} = \frac{\sqrt{n}q}{3}$ ist, gilt Analoges für $\frac{n}{\alpha}$.

Im Folgenden wird davon ausgegangen, dass die Eingabegrößen für $GapSVP_{\gamma}$ und $SIVP_{\gamma}$ polynomiell im Gitterang n sind. Theorem 4.3.8 sagt nun aus, dass es auf n-dimensionalen Gittern mit vollem Rang sowohl einen Quantenalgorithmus mit Polynomialzeit für $GapSVP_{\gamma}$ als auch einen Quantenalgorithmus mit Polynomialzeit für $SIVP_{\gamma}$ gibt, wobei $\gamma \in \tilde{O}(n^c)$ für ein $c \in \mathbb{R}_{>0}$ ist, falls es einen Polynomialzeitalgorithmus für $Unterscheidungs\text{-}LWE_{q,\overline{\Psi}_{\alpha}}$ gibt und $\epsilon = 0$ ist. Wie in Abschnitt 4.1 angegeben, erreichen Polynomialzeitalgorithmen aber bisher nur subexponentielle Approximationsfaktoren für $GapSVP_{\gamma}$ und $SIVP_{\gamma}$. Deshalb lässt sich vermuten, dass $Unterscheidungs\text{-}LWE_{q,\overline{\Psi}_{\alpha}}$ mit obigen Parametern für $\epsilon = 0$ schwer ist.

Im Fall $\epsilon > 0$ ergibt sich aus Theorem 4.3.8, dass es auf n-dimensionalen Gittern mit vollem Rang einen Quantenalgorithmus für $GapSVP_{\gamma}$ sowie einen Quantenalgorithmus für $SIVP_{\gamma}$ gibt, wobei der Approximationsfaktor γ und die Laufzeit jeweils in $2^{\tilde{O}(n^{\epsilon})}$ sind, falls es einen Polynomialzeitalgorithmus für $Unterscheidungs\text{-}LWE_{q,\overline{\Psi}_{\alpha}}$ gibt. Über diese Größenordnung des Approximationsfaktors ist für $\epsilon \in (0,1)$ allerdings nichts bekannt. Wie in Abschnitt 4.1 erwähnt, erreichen Polynomialzeitalgorithmen für $GapSVP_{\gamma}$ bzw. $SIVP_{\gamma}$ nur Approximationsfaktoren $\gamma \in 2^{O\left(\frac{n \log\log n}{\log n}\right)}$. Aber für Approximationsfaktoren $\gamma \in 2^{\tilde{O}(n^{\epsilon})}$ mit $\epsilon \in (0,1)$ kann keine Aussage getroffen werden. Daher kann dann auch keine Aussage über die Schwere von $Unterscheidungs\text{-}LWE_{q,\overline{\Psi}_{\alpha}}$ mittels Theorem 4.3.8 getroffen werden.

Für $\epsilon = 1$ kann aufgrund der bereits bekannten Algorithmen für $GapSVP_{\gamma}$ bzw. $SIVP_{\gamma}$ kein Schluss auf die Schwere von $Unterscheidungs\text{-}LWE_{q,\overline{\Psi}_{\alpha}}$ mit Theorem 4.3.8 gezogen werden. Um auf $Unterscheidungs\text{-}LWE_{q,\overline{\Psi}_{\alpha}}$ als schweres Problem aufgrund von Theorem 4.3.8 hoffen zu können, sollte also ϵ nahe bei 0 gewählt werden.

Außerdem sollte wegen des Zusammenhangs zwischen $Unterscheidungs\text{-}LWE_{q,\overline{\Psi}_{\alpha}}$ und $GapSVP_{\gamma}$ $n > 100$ gewählt werden, damit die Sicherheitsannahme nicht zu schwach ist, denn Untersuchungen in [GN08] haben gezeigt, dass SVP bei Gittern mit Rang 60 innerhalb einer Stunde gelöst werden kann und dass sich vermuten lässt, dass SVP_n

bis zum Rang 250 im Worst Case sowie bis zum Rang 500 im Average Case leicht zu berechnen ist.

Die Sicherheit des Verfahrens 8.3.2 beruht auf einem völlig anderen Problem, welches eine Abwandlung der Entscheidungsvariante des *Bilinear Diffie-Hellman Problem* ist. Dafür seien G_1, G_2 Gruppen der Ordnung q, wobei $q \in \mathbb{N}$ eine Primzahl ist, und $e : G_1 \times G_1 \to G_2$ sei eine zulässige bilineare Abbildung. Zunächst wird das *Bilinear Diffie-Hellman Problem* zusammen mit einer abgewandelten Version definiert.

Definition 9.2.1.

- *Beim* Bilinear Diffie-Hellman Problem (BDH) *ist ein Tupel* $\left(g, g^a, g^b, g^c\right) \in G_1^4$ *gegeben, wobei* $g \in \hat{G_1}$ *und* $a, b, c \in \mathbb{Z}_q$ *sind. Das Ziel ist es,* $e\left(g, g\right)^{abc} \in G_2$ *auszugeben.*

- *Beim* Modified Bilinear Diffie-Hellman Problem (MBDH) *ist ein Tupel* $\left(g, g^a, g^b, g^c\right) \in G_1^4$ *gegeben, wobei* $g \in \hat{G_1}$ *und* $a, b, c \in \mathbb{Z}_q$ *mit* $c \neq 0$ *sind. Das Ziel ist es,* $e\left(g, g\right)^d \in G_2$ *mit* $d := \frac{ab}{c} \in \mathbb{Z}_q$ *auszugeben.*

Die Entscheidungsvariante von *MBDH*, auf welcher die Sicherheit von Verfahren 8.3.2 beruht, lässt sich wie folgt formulieren.

Definition 9.2.2.

- *Beim* Entscheidungs-MBDH *ist ein Tupel* $\left(g, g^a, g^b, g^c, e\left(g, g\right)^d\right) \in G_1^4 \times G_2$ *gegeben, wobei* $g \in \hat{G_1}$ *und* $a, b, c, d \in \mathbb{Z}_q$ *mit* $c \neq 0$ *sind. Das Ziel ist, zu entscheiden, ob* $d = \frac{ab}{c}$ *ist.*

- *Der* Vorteil *eines Algorithmus* $\mathcal{A}$*, der nur 1 und 0 ausgeben kann, bei* Entscheidungs-MBDH *ist definiert als*

$$\frac{1}{2}\left|\Pr\left(\mathcal{A}\left(g, g^a, g^b, g^c, e\left(g, g\right)^{\frac{ab}{c}}\right) = 1\right) - \Pr\left(\mathcal{A}\left(g, g^a, g^b, g^c, e\left(g, g\right)^d\right) = 1\right)\right|,$$

  wobei die Wahrscheinlichkeiten über die zufälligen Wahlen von $\mathcal{A}$ *und die zufällig gleichverteilten Wahlen von* $g \in \hat{G_1}$ *sowie* $a, b, d \in \mathbb{Z}_q$ *und* $c \in \mathbb{Z}_q \setminus \{0\}$ *sind.*

- *Ein Algorithmus* $\mathcal{A}$ *löst* Entscheidungs-MBDH*, falls er nur 1 oder 0 ausgeben kann und sein Vorteil bei* Entscheidungs-MBDH *nicht vernachlässigbar ist.*

Für letztere Definition sollten der Vollständigkeit halber wieder Familien von Gruppenpaaren $\left((G_1, G_2)_q\right)_{q\in\mathbb{N} \text{ Primzahl}}$ betrachtet werden, so dass die Vernachlässigbarkeit in $\lceil \log q \rceil$ zu verstehen ist. Es gibt Gruppen G_1 und G_2 mit Primordnung q und zulässiger bilinearer Abbildung $e : G_1 \times G_1 \to G_2$, von denen vermutet wird, dass es keinen (probabilistischen) Polynomialzeitalgorithmus gibt, der *Entscheidungs-MBDH* auf diesen Gruppen löst. Wie diese Gruppen konstruiert werden können, lässt sich zum Beispiel in Abschnitt 5.1 von [BF03] finden.

Im Gegensatz zum *Learning with Errors Problem* gibt es aber keine Beweise, die zeigen, dass *Entscheidungs-MBDH* NP-schwer ist und es gibt keine vergleichbaren Zusammenhänge zu Problemen im Worst Case wie in Theorem 4.3.8. Außerdem lässt sich der diskrete Logarithmus in G_1 auf Quantencomputern in Polynomialzeit berechen [Sho97], d.h. für jeden Erzeuger $g \in \hat{G_1}$ und jedes $h \in G_1$ lässt sich effizient $x \in \mathbb{Z}_q$ finden, so dass $g^x = h$ ist. Daher können aber auch in Polynomialzeit $a, b, c \in \mathbb{Z}_q$ aus $\left(g, g^a, g^b, g^c\right) \in G_1^4$ auf Quantencomputern berechnet werden und somit sowohl *MBDH* als auch *Entscheidungs-MBDH* gelöst werden. Beim *Learning with Errors Problem* scheinen Quantencomputer jedoch keinen Fortschritt gegenüber klassischen Computern zu bringen [Reg10]. Daher ist die Verwendung des *Learning with Errors Problem* als Sicherheitsannahme besser begründbar.

9.3. Sicherheitsbeweis des Verfahrens basierend auf Gitterproblemen

In diesem und dem folgendem Abschnitt wird die Sicherheit der beiden vorgestellten Verfahren im Sicherheitsmodell aus Abschnitt 9.1 gezeigt. Es wird mit Verfahren 8.2.1 begonnen. Dabei seien die Parameter $n = \lambda \in \mathbb{N}$, $q \in \mathbb{N}$ Primzahl, $5n \log q \leq m \in \mathbb{N}$ $n \geq l \in \mathbb{N}$, $l \geq k \in \mathbb{N}$, $\alpha \in (0, 1)$ und $\sigma = m\,(f\,(m))^2$ mit $f : [1, \infty) \to \mathbb{R}$, $m \mapsto (\log m)^{\frac{1}{2}+\delta}$ für ein festes $\delta \in \left(0, \frac{1}{2}\right)$ wie in der Beschreibung des Verfahrens. In [ABV+12] wird die Sicherheit des Verfahrens in Theorem 1 formuliert. Diese Formulierung ist aber fehlerhaft, ebenso wie die darauffolgende Beweisskizze. Dies wird nun korrigiert.

Theorem 9.3.1. *Wenn es einen probabilistischen Angreifer $\mathcal{A}$ mit Polynomialzeit in n und mit Vorteil $\epsilon \in \left(0, \frac{1}{2}\right]$ im Fuzzy Selective-ID Spiel für das Verfahren 8.2.1 gibt, dann existiert ein probabilistischer Algorithmus $\mathcal{B}$ mit Polynomialzeit in n und mit Vorteil $\frac{\epsilon}{2}$*

bei Unterscheidungs-LWE$_{q,\overline{\Psi}_\alpha}$.

Beweis. $\mathcal{B}$ benutzt zum Unterscheiden der Verteilungen $\mathcal{A}$, indem er das Fuzzy Selective-ID Spiel in der Rolle des Herausforderers wie folgt simuliert. Während der Simulation müssen die Verteilungen aller Parameter, Schlüssel und gültigen Schlüsseltexte, die $\mathcal{A}$ von $\mathcal{B}$ bekommt, statistisch nah zu den entsprechenden Verteilungen im Verfahren 8.2.1 sein, damit $\mathcal{A}$ keinen Unterschied zwischen der Simulation und dem Fuzzy Selective-ID Spiel zum Verschlüsselungsverfahren bemerkt und somit die Annahme, dass $\mathcal{A}$ den Vorteil ϵ hat, auch angewendet werden kann.

1. *Instanziierung.* $\mathcal{B}$ stellt $(lm+1)$ Anfragen an das gegebene Orakel $\mathcal{O}$ und erhält damit (w_1, v_1), (w_1^1, v_1^1), $(w_1^2, v_1^2), \ldots, (w_1^m, v_1^m), \ldots, (w_l^1, v_l^1)$, $(w_l^2, v_l^2), \ldots,$ $(w_l^m, v_l^m) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

2. *Zielfestlegung.* $\mathcal{A}$ gibt die Identität $\mathrm{id}^* \in \{0,1\}^l$ bekannt, bezüglich welcher er herausgefordert werden möchte.

3. *Setup.* $\mathcal{B}$ konstruiert die öffentlichen Parameter folgendermaßen:

   a) Für alle $i \in \{1, \ldots, l\}$ mit $\mathrm{id}_i^* = 1$ bestehe die Matrix $A_i \in \mathbb{Z}_q^{n \times m}$ aus den Spalten $w_i^1, \ldots, w_i^m$.
   Damit ist A_i zufällig gleichverteilt.

   b) Für alle $i \in \{1, \ldots, l\}$ mit $\mathrm{id}_i^* = 0$ sei $(A_i, T_i) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}^{m \times m}$ eine Ausgabe von $\mathit{TrapGen}(n, m, q)$.
   Daher ist die Verteilung von $A_i \in \mathbb{Z}_q^{n \times m}$ statistisch nah zur Gleichverteilung.

   c) Setze $u := w_1 \in \mathbb{Z}_q^n$.

   Die öffentlichen Parameter $\mathrm{PP} := (k, u, A_1, A_2, \ldots, A_l)$ werden nun $\mathcal{A}$ übergeben. Im Verfahren 8.2.1 ist für alle $i \in \{1, \ldots, l\}$ die Verteilung der Matrizen $A_i \in \mathbb{Z}_q^{n \times m}$ statistisch nah zur Gleichverteilung. Da dort außerdem der Vektor $u \in \mathbb{Z}_q^n$ zufällig gleichverteilt ist, ist die Verteilung der öffentlichen Parameter dieser Simulation statistisch nah zur Verteilung der öffentlichen Parameter im Verfahren 8.2.1.

4. *Phase 1 und 2.* Immer wenn $\mathcal{A}$ einen geheimen Schlüssel für eine Identität $\mathrm{id} \in \{0,1\}^l$ mit $|\{i \mid i \in \{1, \ldots, l\}, \mathrm{id}_i = \mathrm{id}_i^* = 1\}| < k$ erfragt, konstruiert $\mathcal{B}$ diesen Schlüssel wie folgt:

   a) Sei $I := \{i \mid i \in \{1, \ldots, l\}, \mathrm{id}_i = \mathrm{id}_i^* = 1\}$ sowie $t := |I| < k$.

b) Für alle $i \in I$ sei $e_i \in \mathbb{Z}^m$ eine Ausgabe von $\mathit{SampleGaussian}\left(\mathbb{Z}^m, I_m, \sigma, 0\right)$ aus Theorem 6.1.2 und $\hat{u}_i := A_i e_i \in \mathbb{Z}_q$.

Dann ist die Verteilung von $e_i \in \mathbb{Z}^m$ statistisch nah zu $\mathcal{D}_{\mathbb{Z}^m,\sigma,0}$. Aus Lemma 6.2.5 folgt, dass A_i bis auf eine vernachlässigbare Wahrscheinlichkeit vollen Zeilenrang hat, und wegen Theorem 6.2.4 gibt es bis auf eine vernachlässigbare Wahrscheinlichkeit eine Basis für $\Lambda_q^{\perp}\left(A_i\right)$, deren Norm durch $mf\left(n\right)$ beschränkt ist. Nach Lemma 5.3.4 gibt es daher in m vernachlässigbare $\epsilon_1 := \epsilon_1\left(m\right)$, $\epsilon_2 := \epsilon_2\left(m\right) \in \mathbb{R}_{>0}$, so dass $\eta_{\epsilon_1}\left(\mathbb{Z}^m\right) \leq f\left(m\right) \leq \sigma$ und $\eta_{\epsilon_2}\left(\Lambda_q^{\perp}\left(A_i\right)\right) \leq f\left(m\right) \cdot mf\left(n\right) \leq \sigma$ bis auf eine vernachlässigbare Wahrscheinlichkeit ist. Deshalb ist nach Lemma 6.2.8 bis auf eine vernachlässigbare Wahrscheinlichkeit die Verteilung von $\hat{u}_i$ statistisch nah zur Gleichverteilung auf $\mathbb{Z}_q^n$.

Wegen $\hat{u}_i = A_i e_i$ folgt, dass ebenfalls $e_i \in \Lambda_q^{\hat{u}_i}\left(A_i\right)$ gilt. Außerdem ist die Verteilung von e_i eingeschränkt auf $\Lambda_q^{\hat{u}_i}\left(A_i\right)$ statistisch nah zu $\mathcal{D}_{\Lambda_q^{\hat{u}_i}(A_i),\sigma,0}$. Um dies einzusehen, bezeichne $\mathcal{D}$ die Verteilung von $e_i \in \mathbb{Z}^m$. Dann gilt für $x \in \mathbb{Z}^m$

$$\begin{aligned}
\Pr\left(e_i = x \mid e_i \in \Lambda_q^{\hat{u}_i}\left(A_i\right)\right) &= \frac{\Pr\left(e_i \in \Lambda_q^{\hat{u}_i}\left(A_i\right) \mid e_i = x\right) \Pr\left(e_i = x\right)}{\Pr\left(e_i \in \Lambda_q^{\hat{u}_i}\left(A_i\right)\right)} \\
&= \Pr\left(e_i \in \Lambda_q^{\hat{u}_i}\left(A_i\right) \mid e_i = x\right) \frac{\mathcal{D}\left(x\right)}{\mathcal{D}\left(\Lambda_q^{\hat{u}_i}\left(A_i\right)\right)} \\
&= \begin{cases} 0 & , \text{falls } x \notin \Lambda_q^{\hat{u}_i}\left(A_i\right) \\ \frac{\mathcal{D}(x)}{\mathcal{D}\left(\Lambda_q^{\hat{u}_i}(A_i)\right)} & , \text{falls } x \in \Lambda_q^{\hat{u}_i}\left(A_i\right) \end{cases}
\end{aligned}$$

nach dem Bayestheorem. Deshalb ist $\frac{\mathcal{D}}{\mathcal{D}\left(\Lambda_q^{\hat{u}_i}(A_i)\right)}$ die Verteilung von e_i eingeschränkt auf $\Lambda_q^{\hat{u}_i}\left(A_i\right)$. Weil aber $\frac{\mathcal{D}}{\mathcal{D}\left(\Lambda_q^{\hat{u}_i}(A_i)\right)}$ statistisch nah zu $\frac{\mathcal{D}_{\mathbb{Z}^m,\sigma,0}}{\mathcal{D}_{\mathbb{Z}^m,\sigma,0}\left(\Lambda_q^{\hat{u}_i}(A_i)\right)}$ ist und ferner

$$\begin{aligned}
\frac{\mathcal{D}_{\mathbb{Z}^m,\sigma,0}\left(x\right)}{\mathcal{D}_{\mathbb{Z}^m,\sigma,0}\left(\Lambda_q^{\hat{u}_i}\left(A_i\right)\right)} &= \frac{\rho_{\sigma,0}\left(x\right)}{\rho_{\sigma,0}\left(\mathbb{Z}^m\right)} \frac{\rho_{\sigma,0}\left(\mathbb{Z}^m\right)}{\rho_{\sigma,0}\left(\Lambda_q^{\hat{u}_i}\left(A_i\right)\right)} = \frac{\rho_{\sigma,0}\left(x\right)}{\rho_{\sigma,0}\left(\Lambda_q^{\hat{u}_i}\left(A_i\right)\right)} \\
&= \mathcal{D}_{\Lambda_q^{\hat{u}_i}(A_i),\sigma,0}\left(x\right)
\end{aligned}$$

für $x \in \Lambda_q^{\hat{u}_i}\left(A_i\right)$ gilt, ist die Verteilung von e_i eingeschränkt auf $\Lambda_q^{\hat{u}_i}\left(A_i\right)$

statistisch nah zu $\mathcal{D}_{\Lambda_q^{\hat{u}_i}(A_i),\sigma,0}$.

c) Für jedes $i \in \{1,\ldots,l\}$ mit $\mathrm{id}_i = 1$ soll im Verschlüsselungsverfahren gelten, dass $\hat{u}_i$ von der Form $\hat{u}_i = u + a_1 i + a_2 i^2 + \ldots + a_{k-1} i^{k-1}$ mit $a_1,\ldots,a_{k-1} \in \mathbb{Z}_q^n$ ist. Um die Vektoren $a_1,\ldots,a_{k-1}$ eindeutig zu bestimmen, werden zusätzlich zum bekannten $u \in \mathbb{Z}_q^n$ $k-1$ Vektoren $\hat{u}_i \in \mathbb{Z}_q^n$ benötigt. t solche Vektoren wurden bereits in Schritt b) bestimmt, wobei deren Verteilung bis auf eine vernachlässigbare Wahrscheinlichkeit statistisch nah zur Gleichverteilung auf $\mathbb{Z}_q^n$ ist. Da $t \leq k-1$ ist, wählt $\mathcal{B}$ nun weitere $k-1-t$ Vektoren $\hat{u}_i \in \mathbb{Z}_q^n$ für $i \in J$ zufällig gleichverteilt, wobei $J \subseteq \{1,\ldots,l\} \setminus I$ mit $|J| = k-1-t$ ist. Damit sind dann sowohl $a_1,\ldots,a_{k-1} \in \mathbb{Z}_q^n$ als auch die restlichen $\hat{u}_i$ für $i \in \{1,\ldots,l\} \setminus (I \cup J)$ eindeutig bestimmt.

d) Für alle $i \in \{1,\ldots,l\} \setminus I$ mit $\mathrm{id}_i = 1$ sei $e_i \in \mathbb{Z}^m$ eine Ausgabe von $\mathit{SamplePre}(A_i, T_i, \hat{u}_i, \sigma)$.
Daher gilt $\hat{u}_i = A_i e_i$ und die Verteilung von e_i ist statistisch nah zu $\mathcal{D}_{\Lambda_q^{\hat{u}_i}(A_i),\sigma,0}$.

$\mathcal{B}$ gibt $\mathcal{A}$ den geheimen Schlüssel $\mathrm{SK}_{\mathrm{id}} := (\mathrm{id}, \{e_i \mid i \in \{1,\ldots,l\}, \mathrm{id}_i = 1\})$.
Nach Konstruktion des Schlüssels gilt für alle $i \in \{1,\ldots,l\}$ mit $\mathrm{id}_i = 1$, dass $\hat{u}_i = A_i e_i$ ist und dass die Verteilung von e_i auf $\Lambda_q^{\hat{u}_i}(A_i)$ statistisch nah zu $\mathcal{D}_{\Lambda_q^{\hat{u}_i}(A_i),\sigma,0}$ ist. Da im Verfahren 8.2.1 für alle $i \in \{1,\ldots,l\}$ mit $\mathrm{id}_i = 1$ die Verteilung der $e_i \in \mathbb{Z}^m$ statistisch nah zu $\mathcal{D}_{\Lambda_q^{\hat{u}_i}(A_i),\sigma,0}$ sowie $\hat{u}_i = A_i e_i$ ist und obige Konstruktion die zufällig gleichverteilte Wahl der Polynome aus dem Algorithmus *KeyGen* simuliert, ist die Verteilung der geheimen Schlüssel in der Simulation statistisch nah zur Verteilung der geheimen Schlüssel im Verfahren 8.2.1.

5. *Herausforderung.* $\mathcal{B}$ wirft eine Münze $b \in \{0,1\}$ und berechnet Folgendes:

   a) Es sei $c_0 := Dv_1 + b\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$, wobei wieder $D := (l!)^2$ ist.

   b) Für alle $i \in \{1,\ldots,l\}$ mit $\mathrm{id}_i^* = 1$ sei außerdem $c_i := \left(Dv_i^1,\ldots,Dv_i^m\right)^T \in \mathbb{Z}_q^m$.

   $\mathcal{B}$ gibt dann $\mathrm{CT}_{\mathrm{id}^*} := (\mathrm{id}^*, c_0, \{c_i \mid i \in \{1,\ldots,l\}, \mathrm{id}_i^* = 1\})$ aus.
   Hier ist zu beachten, dass $\mathcal{A}$ zu Beginn dieses Schrittes nicht zwei Nachrichten gleicher Länge ausgeben muss, da 0 und 1 die beiden einzigen möglichen Nachrichten sind.

6. *Schätzung.* $\mathcal{A}$ gibt eine Schätzung $b' \in \{0,1\}$ von b aus. $\mathcal{B}$ akzeptiert, falls $b' = b$ ist, und ansonsten lehnt $\mathcal{B}$ ab.

Um den Vorteil von $\mathcal{B}$ bei *Unterscheidungs-LWE*$_{q,\overline{\Psi}_\alpha}$ zu analysieren, werden folgende zwei Fälle betrachtet.

- Falls $\mathcal{O} = \mathcal{O}_U$ ist, so sind $v_1, v_1^1, v_1^2, \ldots, v_1^m, \ldots, v_l^1, v_l^2, \ldots, v_l^m \in \mathbb{Z}_q$ zufällig gleichverteilt und unabhängig voneinander. Damit sind ebenfalls $c_0 \in \mathbb{Z}_q$ sowie alle $c_i \in \mathbb{Z}_q^m$ mit $\mathrm{id}_i^* = 1$ zufällig gleichverteilt und unabhängig voneinander. In dieser Situation erhält damit $\mathcal{A}$ keine Information über b und somit gilt

$$\Pr\left(\mathcal{B} \text{ akzeptiert } \mid \mathcal{O} = \mathcal{O}_U\right) = \Pr\left(b' = b \mid \mathcal{O} = \mathcal{O}_U\right) = \frac{1}{2}.$$

- Falls $\mathcal{O} = \mathcal{O}_{\mathcal{A}_{s,\overline{\Psi}_\alpha}}$ für ein zufällig gleichverteiltes $s \in \mathbb{Z}_q^n$ ist, so gilt $v_1 = w_1^T s + x_1$ mit $x_1 \sim \overline{\Psi}_\alpha$ und für alle $i \in \{1, \ldots, l\}$ und alle $j \in \{1, \ldots, m\}$ gilt $v_i^j = \left(w_i^j\right)^T s + x_i^j$, wobei die $x_i^j \sim \overline{\Psi}_\alpha$ unabhängig voneinander und unabhängig von x_1 sind. Daher ist

$$c_0 = D\left(w_1^T s + x_1\right) + b\left\lfloor \frac{q}{2} \right\rfloor = u^T (Ds) + Dx_1 + b\left\lfloor \frac{q}{2} \right\rfloor.$$

  Ferner folgt für alle $i \in \{1, \ldots, l\}$ mit $\mathrm{id}_i^* = 1$

$$\begin{aligned} c_i &= D\left(v_i^1, \ldots, v_i^m\right)^T = D\left(\left(w_i^1\right)^T s + x_i^1, \ldots, \left(w_i^m\right)^T s + x_i^m\right)^T = D\left(A_i^T s + x_i\right) \\ &= A_i^T (Ds) + Dx_i \end{aligned}$$

  mit $x_i = \left(x_i^1, \ldots, x_i^m\right)^T$. Deshalb gibt $\mathcal{B}$ in dieser Situation $\mathcal{A}$ einen Schlüsseltext zur Nachricht b unter der Identität id^*, dessen Verteilung der Verteilung der von $Enc\left(PP, \mathrm{id}^*, b\right)$ erstellten Schlüsseltexte entspricht. Weil die Verteilung der öffentlichen Parameter und der von $\mathcal{A}$ erfragten geheimen Schlüssel in dieser Simulation statistisch nah zur Verteilung der öffentlichen Parameter und der geheimen Schlüssel im Verfahren 8.2.1 ist, bemerkt $\mathcal{A}$ keinen Unterschied zwischen der Simulation und dem Fuzzy Selective-ID Spiel zum Verschlüsselungsverfahren. Daher hat $\mathcal{A}$ in dieser Situation den Vorteil ϵ und es gilt somit

$$\Pr\left(\mathcal{B} \text{ akzeptiert } \,\middle|\, \mathcal{O} = \mathcal{O}_{\mathcal{A}_{s,\overline{\Psi}_\alpha}}\right) = \Pr\left(b' = b \,\middle|\, \mathcal{O} = \mathcal{O}_{\mathcal{A}_{s,\overline{\Psi}_\alpha}}\right) = \frac{1}{2} + \epsilon.$$

Demnach folgt, dass

$$\begin{aligned}&\frac{1}{2}\left|\Pr\left(\mathcal{B}\text{ akzeptiert }\middle|\ \mathcal{O}=\mathcal{O}_{\mathcal{A}_{s,\overline{\Psi}_\alpha}}\right)-\Pr\left(\mathcal{B}\text{ akzeptiert }\middle|\ \mathcal{O}=\mathcal{O}_U\right)\right|\\ =&\frac{1}{2}\left|\left(\frac{1}{2}+\epsilon\right)-\frac{1}{2}\right|=\frac{\epsilon}{2}\end{aligned}$$

ist. $\mathcal{B}$ hat also den Vorteil $\frac{\epsilon}{2}$.

Da $\mathcal{A}$ ein Polynomialzeitalgorithmus ist, gilt dies auch für $\mathcal{B}$ und daher ist alles gezeigt.

□

Falls es nun einen Angreifer wie im Theorem 9.3.1 mit in n nicht vernachlässigbarem Vorteil gäbe, so würde es einen effizienten Algorithmus geben, der *Unterscheidungs-LWE*$_{q,\overline{\Psi}_\alpha}$ löst. Deswegen ist Verfahren 8.2.1 sicher im Fuzzy Selective-ID Sicherheitsmodell unter der Annahme, dass es keinen effizienten Algorithmus für *Unterscheidungs-LWE*$_{q,\overline{\Psi}_\alpha}$ gibt.

9.4. Sicherheitsbeweis des Verfahrens basierend auf bilinearen Abbildungen

Mit einer Simulation des Fuzzy Selective-ID Spiels wird nun ebenfalls die Sicherheit von Verfahren 8.3.2 gezeigt. Dabei seien die Parameter $q \in \mathbb{N}$ Primzahl, $l \in \mathbb{N}$ und $l \geq k \in \mathbb{N}$, die Gruppen G_1 und G_2 der Ordnung q sowie die zulässige bilineare Abbildung $e : G_1 \times G_1 \to G_2$ wie in der Beschreibung des Verfahrens. Zudem sei der Sicherheitsparameter $\lambda = \lceil \log q \rceil$. Dieser Beweis ist in leicht abgewandelter Form auch in Kapitel 5 von [SW05] aufgeführt.

Theorem 9.4.1. *Wenn es einen probabilistischen Angreifer $\mathcal{A}$ mit Polynomialzeit in λ und mit Vorteil $\epsilon \in \left(0, \frac{1}{2}\right]$ im Fuzzy Selective-ID Spiel für das Verfahren 8.3.2 gibt, dann existiert ein probabilistischer Algorithmus $\mathcal{B}$ mit Polynomialzeit in λ und mit Vorteil $\frac{\epsilon}{2}$ bei* Entscheidungs-MBDH.

Beweis. $\mathcal{B}$ benutzt für seine Entscheidung $\mathcal{A}$, indem er das Fuzzy Selective-ID Spiel in der Rolle des Herausforderers wie folgt simuliert. Während der Simulation müssen die Verteilungen aller Parameter, Schlüssel und gültigen Schlüsseltexte, die $\mathcal{A}$ von $\mathcal{B}$ bekommt, statistisch nah zu den entsprechenden Verteilungen im Verfahren 8.3.2 sein, damit $\mathcal{A}$ keinen Unterschied zwischen der Simulation und dem Fuzzy Selective-ID Spiel

zum Verschlüsselungsverfahren bemerkt und somit die Annahme, dass $\mathcal{A}$ den Vorteil ϵ hat, auch angewendet werden kann. In diesem Beweis wird sogar gezeigt, dass die entsprechenden Verteilungen identisch sind.

1. *Instanziierung.* $\mathcal{B}$ erhält ein Tupel $\left(g, g^a, g^b, g^c, e\left(g,g\right)^d\right) \in G_1^4 \times G_2$, wobei $g \in \hat{G_1}$ sowie $a, b \in \mathbb{Z}_q$ und $c \in \mathbb{Z}_q \setminus \{0\}$ zufällig gleichverteilt und unabhängig voneinander gewählt sind. $d \in \mathbb{Z}_q$ wurde entweder ebenfalls zufällig gleichverteilt und unabhängig von g, a, b und c gewählt oder durch $d = \frac{ab}{c}$ festgelegt.

2. *Zielfestlegung.* $\mathcal{A}$ gibt die Identität $\mathrm{id}^* \in \{0,1\}^l$ bekannt, bezüglich welcher er herausgefordert werden möchte.

3. *Setup.* $\mathcal{B}$ konstruiert die öffentlichen Parameter folgendermaßen:

   a) Es sei $Y := e\left(g, g^a\right) \in G_2$.

   b) Für alle $i \in \{1, \ldots, l\}$ mit $\mathrm{id}_i^* = 1$ wähle $\beta_i \in \mathbb{Z}_q \setminus \{0\}$ zufällig gleichverteilt und berechne $T_i := \left(g^c\right)^{\beta_i} \in G_1$.

   c) Für alle $i \in \{1, \ldots, l\}$ mit $\mathrm{id}_i^* = 0$ wähle $w_i \in \mathbb{Z}_q \setminus \{0\}$ zufällig gleichverteilt und berechne $T_i := g^{w_i} \in G_1$.

   Die öffentlichen Parameter $\mathrm{PP} := (k, Y, T_1, T_2, \ldots, T_l)$ werden nun $\mathcal{A}$ übergeben. Weil $a \in \mathbb{Z}_q$ und $c \in \mathbb{Z}_q \setminus \{0\}$ sowie alle $\beta_i \in \mathbb{Z}_q \setminus \{0\}$ und alle $w_i \in \mathbb{Z}_q \setminus \{0\}$ zufällig gleichverteilt und unabhängig voneinander gewählt wurden, ist die Verteilung der öffentlichen Parameter dieser Simulation identisch zur Verteilung der öffentlichen Parameter im Verfahren 8.3.2.

4. *Phase 1 und 2.* Immer wenn $\mathcal{A}$ einen geheimen Schlüssel für eine Identität $\mathrm{id} \in \{0,1\}^l$ mit $|\{i \mid i \in \{1, \ldots, l\}, \mathrm{id}_i = \mathrm{id}_i^* = 1\}| < k$ erfragt, konstruiert $\mathcal{B}$ diesen Schlüssel wie folgt:

   a) Sei $I := \{i \mid i \in \{1, \ldots, l\}, \mathrm{id}_i = 1\}$ sowie $t := |I|$.

   b) Sei $J := \{i \mid i \in \{1, \ldots, l\}, \mathrm{id}_i = \mathrm{id}_i^* = 1\}$.

   c) Für alle $i \in J$ wähle $d_i \in \mathbb{Z}_q$ zufällig gleichverteilt und berechne $D_i := g^{d_i} \in G_1$.

   d) Falls $t \geq k - 1$ ist, berechne Folgendes:

      i. Sei $J' \subseteq I$ eine beliebige Teilmenge mit $J \subseteq J'$ und $|J'| = k - 1$.

ii. Für alle $i \in J' \setminus J$ wähle $\lambda_i \in \mathbb{Z}_q$ zufällig gleichverteilt und berechne $d_i := \frac{\lambda_i}{w_i} \in \mathbb{Z}_q$ und $D_i := g^{d_i} \in G_1$.
Im Algorithmus *KeyGen* aus Verfahren 8.3.2 wird ein zufällig gleichverteilt gewähltes Polynom $p \in \mathbb{Z}_q[X]$ vom Grad $k-1$ mit $p(0) = a$ zur Schlüsselerzeugung verwendet. Um dies zu simulieren, sind nach diesem Schritt $k-1$ Stützstellen für p zufällig gleichverteilt gewählt worden. Damit stehen insgesamt k Stützstellen fest und p ist eindeutig bestimmt. Wegen Schritt 2 im Algorithmus *KeyGen* ist für alle $i \in J$ zudem $d_i = \frac{p(i)}{c\beta_i}$, also $p(i) = c\beta_i d_i$, und für alle $i \in J' \setminus J$ gilt $d_i = \frac{\lambda_i}{w_i} = \frac{p(i)}{w_i}$, also $p(i) = \lambda_i$. Ohne a und c zu kennen, kann $\mathcal{B}$ nun für alle $i \in I \setminus J'$ auch $D_i = g^{d_i}$ mit $d_i := \frac{p(i)}{w_i} \in \mathbb{Z}_q$ wie folgt berechnen.

iii. Sei $S := J' \cup \{0\}$.

iv. Für alle $j \in S$ setze $L_j(X) := \prod_{m \in S \setminus \{j\}} \frac{X-m}{j-m} \in \mathbb{Z}_q[X]$.

v. Für alle $i \in I \setminus J'$ berechne

$$D_i := \left(\prod_{j \in J} (g^c)^{\frac{\beta_j d_j L_j(i)}{w_i}} \right) \left(\prod_{j \in J' \setminus J} g^{\frac{\lambda_j L_j(i)}{w_i}} \right) (g^a)^{\frac{L_0(i)}{w_i}} \in G_1.$$

Dann ist für alle $i \in I \setminus J'$ tatsächlich $D_i = g^{d_i}$, denn nach Lagrange-Interpolation gilt $p(i) = \sum_{j \in J} c\beta_j d_j L_j(i) + \sum_{j \in J' \setminus J} \lambda_j L_j(i) + aL_0(i)$.

e) Falls $t < k-1$ ist, berechne Folgendes:

i. Für alle $i \in I \setminus J$ wähle $\lambda_i \in \mathbb{Z}_q$ zufällig gleichverteilt und berechne $d_i := \frac{\lambda_i}{w_i} \in \mathbb{Z}_q$ und $D_i := g^{d_i} \in G_1$.
In diesem Fall muss nichts weiter beachtet werden, da weniger als k gewählte Stützstellen für das Polynom $p \in \mathbb{Z}_q[X]$ mit Grad $k-1$ das Polynom noch nicht festlegen. Somit kann für alle $i \in I$ bereits D_i bestimmt werden, ohne auf Polynominterpolation zurückzugreifen.

$\mathcal{B}$ gibt $\mathcal{A}$ den geheimen Schlüssel $SK_{\mathrm{id}} := (\mathrm{id}, \{D_i \mid i \in I\})$.
Bei obiger Wahl der $D_i \in G_1$ für $i \in \{1, \ldots, l\}$ mit $\mathrm{id}_i = 1$ wird implizit ein zufällig gleichverteiltes Polynom $p \in \mathbb{Z}_q[X]$ mit Grad $k-1$ und $p(0) = a$ gewählt, so dass $D_i = g^{d_i}$ ist, wobei $d_i = \frac{p(i)}{c\beta_i}$ für $i \in J$ und $d_i = \frac{p(i)}{w_i}$ für $i \in I \setminus J$ ist. In der Situation aus Schritt e) wird dieses Polynom nicht eindeutig festgelegt, was aber auch nicht

erforderlich ist. Somit ist die Verteilung der geheimen Schlüssel in der Simulation identisch zur Verteilung der geheimen Schlüssel im Verfahren 8.3.2.

5. *Herausforderung.* $\mathcal{A}$ gibt zwei Nachrichten $M_0, M_1 \in G_2$ aus. $\mathcal{B}$ wirft eine Münze $\tilde{b} \in \{0,1\}$ und berechnet Folgendes:
   a) Es sei $E := M_{\tilde{b}} \left(e\left(g,g\right)^d \right) \in G_2$.
   b) Für alle $i \in \{1,\ldots,l\}$ mit $\mathrm{id}_i^* = 1$ sei außerdem $E_i := \left(g^b\right)^{\beta_i} \in G_1$.

   $\mathcal{B}$ gibt dann $\mathrm{CT}_{\mathrm{id}^*} := (\mathrm{id}^*, E, \{E_i \mid i \in \{1,\ldots,l\}, \mathrm{id}_i^* = 1\})$ aus.

6. *Schätzung.* $\mathcal{A}$ gibt eine Schätzung $b' \in \{0,1\}$ von $\tilde{b}$ aus. $\mathcal{B}$ gibt 1 aus, falls $b' = \tilde{b}$ ist, und ansonsten gibt $\mathcal{B}$ 0 aus.

Um den Vorteil von $\mathcal{B}$ bei *Entscheidungs-MBDH* zu analysieren, werden folgende zwei Fälle betrachtet.

- Falls $d \in \mathbb{Z}_q$ ein zufällig gleichverteiltes Element ist, so ist $E \in G_2$ ebenfalls zufällig gleichverteilt. In dieser Situation erhält damit $\mathcal{A}$ keine Information über $\tilde{b}$ und somit gilt

$$\begin{aligned} &\Pr\left(\mathcal{B}\left(g, g^a, g^b, g^c, e\left(g,g\right)^d\right) = 1 \;\middle|\; d \in \mathbb{Z}_q \text{ ist zufällig gleichverteilt } \right) \\ &= \Pr\left(b' = \tilde{b} \;\middle|\; d \in \mathbb{Z}_q \text{ ist zufällig gleichverteilt } \right) = \frac{1}{2}. \end{aligned}$$

- Falls d durch $d = \frac{ab}{c}$ festgelegt wurde, so gilt für $s := \frac{b}{c} \in \mathbb{Z}_q$, dass

$$E = M_{\tilde{b}} \left(e\left(g,g\right)^d \right) = M_{\tilde{b}} \left(e\left(g,g\right)^{\frac{ab}{c}} \right) = M_{\tilde{b}} \left(e\left(g,g^a\right)^{\frac{b}{c}} \right) = M_{\tilde{b}} Y^s$$

  ist. Für alle $i \in \{1,\ldots,l\}$ mit $\mathrm{id}^* = 1$ ist ferner

$$E_i = \left(g^b\right)^{\beta_i} = \left(g^{\frac{b}{c}}\right)^{c\beta_i} = \left(g^s\right)^{c\beta_i} = \left(g^{c\beta_i}\right)^s = T_i^s.$$

  Weil $b \in \mathbb{Z}_q$ und $c \in \mathbb{Z}_q \setminus \{0\}$ zufällig gleichverteilt und unabhängig voneinander sind, ist auch $s \in \mathbb{Z}_q$ zufällig gleichverteilt. Deshalb gibt $\mathcal{B}$ in dieser Situation $\mathcal{A}$ einen Schlüsseltext zur Nachricht $M_{\tilde{b}}$ unter der Identität id^*, dessen Verteilung identisch ist zu der Verteilung der von $Enc\left(\mathrm{PP}, \mathrm{id}^*, M_{\tilde{b}}\right)$ erstellten Schlüsseltexte. Weil die Verteilung der öffentlichen Parameter und der von $\mathcal{A}$ erfragten geheimen Schlüssel in dieser Simulation identisch zur Verteilung der öffentlichen Parameter

und der geheimen Schlüssel im Verfahren 8.3.2 ist, bemerkt $\mathcal{A}$ keinen Unterschied zwischen der Simulation und dem Fuzzy Selective-ID Spiel zum Verschlüsselungsverfahren. Daher hat $\mathcal{A}$ in dieser Situation den Vorteil ϵ und es gilt

$$\begin{aligned}
&\Pr\left(\mathcal{B}\left(g, g^a, g^b, g^c, e\left(g,g\right)^{\frac{ab}{c}}\right) = 1\right) \\
&= \Pr\left(\mathcal{B}\left(g, g^a, g^b, g^c, e\left(g,g\right)^{d}\right) = 1 \;\middle|\; d := \frac{ab}{c}\right) \\
&= \Pr\left(b' = \tilde{b} \;\middle|\; d := \frac{ab}{c}\right) \\
&= \frac{1}{2} + \epsilon.
\end{aligned}$$

Insgesamt folgt demnach für $\tilde{d} \in \mathbb{Z}_q$, welches zufällig gleichverteilt und unabhängig von g, a, b und c gewählt wurde, dass

$$\begin{aligned}
&\frac{1}{2}\left|\Pr\left(\mathcal{B}\left(g, g^a, g^b, g^c, e\left(g,g\right)^{\frac{ab}{c}}\right) = 1\right) - \Pr\left(\mathcal{B}\left(g, g^a, g^b, g^c, e\left(g,g\right)^{\tilde{d}}\right) = 1\right)\right| \\
&= \frac{1}{2}\left|\left(\frac{1}{2} + \epsilon\right) - \frac{1}{2}\right| = \frac{\epsilon}{2}
\end{aligned}$$

ist. $\mathcal{B}$ hat also den Vorteil $\frac{\epsilon}{2}$.

Da $\mathcal{A}$ ein Polynomialzeitalgorithmus ist, gilt dies auch für $\mathcal{B}$ und daher ist alles gezeigt.

□

Falls es nun einen Angreifer wie im Theorem 9.4.1 mit in λ nicht vernachlässigbarem Vorteil gäbe, so würde es einen effizienten Algorithmus geben, der *Entscheidungs-MBDH* löst. Deswegen ist Verfahren 8.3.2 sicher im Fuzzy Selective-ID Sicherheitsmodell unter der Annahme, dass es keinen effizienten Algorithmus für *Entscheidungs-MBDH* gibt.

10. Vergleich der Effizienz und Erweiterbarkeit

10.1. Effizienz

In diesem Abschnitt wird untersucht, wie groß die erstellten Schlüsseltexte, Schlüssel und öffentlichen Parameter der beiden vorgestellten Verfahren sind. Um konkrete Werte zu erhalten, wird dies insbesondere für Identitäten der Länge 32 durchgeführt. Damit können 2^{32} verschiedene Identitäten realisiert werden. Dies ist mehr als die Hälfte der aktuellen Weltbevölkerung. Trotzdem könnten Identitäten solcher Länge zum Beispiel für biometrische Anwendungen denkbar sein. Für eine Identität $\mathrm{id} \in \{0,1\}^l$ sei zudem im Folgenden $a_{\mathrm{id}} := |\{i \in \{1,\dots,l\} \mid \mathrm{id}_i = 1\}|$. Beispielhaft wird der Fall $a_{\mathrm{id}} = 8$ zum Erhalten konkreter Werte betrachtet.

Vor dem Vergleich der beiden Verfahren müssen noch einige Parameter festgesetzt werden. Wie bereits im letzten Kapitel festgestellt wurde, sollte für Verfahren 8.2.1 der Sicherheitsparameter $\lambda = n > 100$ gewählt werden, um große Sicherheit zu gewährleisten. Weil alle hier untersuchten Schlüsseltext-, Schlüssel- und Parameterlängen bei wachsendem n zunehmen, sei von nun an $n := 100$. Dann können die übrigen Parameter wie auf Seite 88 gewählt werden; also sei insbesondere $q \in \left[2^{11}n^6 l\,(l!)^3, 2^{12}n^6 l\,(l!)^3\right]$ eine Primzahl sowie $m := \lceil 5n \log q \rceil$. Damit ist

$$\lceil \log q \rceil \geq \left\lceil \log\left(2^{11}n^6 l\,(l!)^3\right)\right\rceil \geq \left\lceil \log\left(2^{11}100^6\right)\right\rceil = 51,$$
$$m \geq \left\lceil 5n \log\left(2^{11}n^6 l\,(l!)^3\right)\right\rceil \geq \left\lceil 500 \log\left(2^{11}100^6\right)\right\rceil = 25432$$

und im Fall $l = 32$ sogar

$$\lceil \log q \rceil \geq \left\lceil \log\left(2^{11}100^6 32\,(32!)^3\right)\right\rceil = 409,$$
$$m \geq \left\lceil 500 \log\left(2^{11}100^6 32\,(32!)^3\right)\right\rceil = 204427.$$

Weiterhin sei $f : [1,\infty) \to \mathbb{R}, m \mapsto (\log m)^{\frac{1}{2}+\delta}$ für ein festes $\delta \in \left(0, \frac{1}{2}\right)$ und $\sigma := m\,(f\,(m))^2$.

Im Verfahren 8.3.2 werden Gruppen G_1 und G_2 der Primordnung q verwendet. In der Praxis ist G_1 eine Untergruppe einer Gruppe von Punkten einer elliptischen Kurve über $\mathbb{F}_p$, wobei $p \geq q$ eine Primzahl ist. Elemente in G_1 können demnach mit $\alpha_1 := 2\lceil \log p \rceil$ Bits dargestellt werden. G_2 ist eine Untergruppe von $\mathbb{F}_{p^r} \setminus \{0\}$ mit $r \in \mathbb{N}$ und $r \geq 2$. Daher können Elemente in G_2 mit $\alpha_2 := r\lceil \log p \rceil$ Bits dargestellt werden. Dabei wird q so gewählt, dass $\lceil \log q \rceil = 271$ ist. Dann wird p so bestimmt, dass $\lceil \log p \rceil \in [271, 542]$ ist, und r wird danach minimal gewählt, so dass $\alpha_2 \geq 2048$ ist. Demnach ist $r \in [4, 8]$. Diese Parameterwahl wird zum Beispiel in einem Projekt der Arbeitsgruppe „Codes und Kryptographie“ von der Universität Paderborn verwendet, um einer Sicherheit von RSA mit Primzahlen der Bitlänge 1024 zu entsprechen.

Nun werden die Längen der Schlüsseltexte, Schlüssel und öffentlichen Parameter verglichen. Bei der Schlüsseltextlänge muss beachtet werden, dass Verfahren 8.2.1 nur jeweils ein Bit verschlüsselt, wohingegen Verfahren 8.3.2 eine Nachricht aus G_2 verschlüsselt. Deswegen wird die Schlüsseltextlänge pro verschlüsseltem Bit untersucht.

1. *Schlüsseltextlänge pro verschlüsseltem Bit.*

   a) *Verfahren 8.2.1.* Ein Schlüsseltext zur Identität $\mathrm{id} \in \{0,1\}^l$ hat die Form $\mathrm{CT}_{\mathrm{id}} = (\mathrm{id}, c_0, \{c_j \mid j \in \{1,\ldots,l\}, \mathrm{id}_j = 1\})$, wobei $c_0 \in \mathbb{Z}_q$ und für alle $j \in \{1,\ldots,l\}$ mit $\mathrm{id}_j = 1$ $c_j \in \mathbb{Z}_q^m$ ist. Damit ist die Darstellungsgröße eines Schlüsseltextes

   $$\beta_{\mathrm{CT}}^G := l + \lceil \log q \rceil + a_{\mathrm{id}} m \lceil \log q \rceil \geq l + 51 + 1297032 a_{\mathrm{id}}.$$

   Im Fall $l = 32$ und $a_{\mathrm{id}} = 8$ ist

   $$\beta_{\mathrm{CT}}^G \geq 32 + 409 + 8 \cdot 204427 \cdot 409 = 668885585.$$

   b) *Verfahren 8.3.2.* Ein Schlüsseltext zur Identität $\mathrm{id} \in \{0,1\}^l$ hat die Form $\mathrm{CT}_{\mathrm{id}} = (\mathrm{id}, E, \{E_i \mid i \in \{1,\ldots,l\}, \mathrm{id}_i = 1\})$, wobei $E \in G_2$ und für alle $i \in \{1,\ldots,l\}$ mit $\mathrm{id}_i = 1$ $E_i \in G_1$ ist. Daher ist die Darstellungsgröße eines Schlüsseltextes pro verschlüsseltem Bit

   $$\beta_{\mathrm{CT}}^B := \frac{l + \alpha_2 + a_{\mathrm{id}}\alpha_1}{\alpha_2} \leq \frac{l}{2048} + 1 + a_{\mathrm{id}}\frac{2}{r} \leq \frac{l}{2048} + 1 + \frac{a_{\mathrm{id}}}{2}.$$

Im Fall $l = 32$ und $a_{\mathrm{id}} = 8$ gilt

$$\beta_{\mathrm{CT}}^{B} \leq \frac{1}{64} + 1 + 4 < 6.$$

2. *Länge geheimer Schlüssel.*

   a) *Verfahren 8.2.1.* Ein geheimer Schlüssel zur Identität $\mathrm{id} \in \{0,1\}^l$ ist von der Form $\mathrm{SK}_{\mathrm{id}} = (\mathrm{id}, \{e_j \mid j \in \{1,\ldots,l\}, \mathrm{id}_j = 1\})$, wobei für alle $j \in \{1,\ldots,l\}$ mit $\mathrm{id}_j = 1$ $e_j \in \mathbb{Z}^m$ ist. Wegen Lemma 8.2.7 ist zudem $\|e_j\| \leq m^{\frac{3}{2}} (f(m))^2$ bis auf eine vernachlässigbare Wahrscheinlichkeit. Ist $e_j = (e_{j,1},\ldots,e_{j,m})^T$, so kann deshalb davon ausgegangen werden, dass für alle $i \in \{1,\ldots,m\}$ $|e_{j,i}| \leq \left\lfloor m^{\frac{3}{2}} (f(m))^2 \right\rfloor$ ist und somit $e_{j,i}$ mit $\left\lfloor \log\left(\left\lfloor m^{\frac{3}{2}} (f(m))^2 \right\rfloor\right)\right\rfloor + 2$ Bits dargestellt werden kann. Damit ist die Darstellungsgröße eines geheimen Schlüssels

$$\begin{aligned}\beta_{\mathrm{SK}}^{G} &:= l + a_{\mathrm{id}} m \left(\left\lfloor \log\left(\left\lfloor m^{\frac{3}{2}} (f(m))^2 \right\rfloor\right)\right\rfloor + 2\right) \\ &\geq l + a_{\mathrm{id}} m \left(\left\lfloor \log\left(\left\lfloor m^{\frac{3}{2}} \log m \right\rfloor\right)\right\rfloor + 2\right) \geq l + 686664 a_{\mathrm{id}}.\end{aligned}$$

   Im Fall $l = 32$ und $a_{\mathrm{id}} = 8$ ist

$$\beta_{\mathrm{SK}}^{G} \geq 32 + 8 \cdot 6541664 = 52333344.$$

   b) *Verfahren 8.3.2.* Ein geheimer Schlüssel zur Identität $\mathrm{id} \in \{0,1\}^l$ ist von der Form $\mathrm{SK}_{\mathrm{id}} = (\mathrm{id}, \{D_i \mid i \in \{1,\ldots,l\}, \mathrm{id}_i = 1\})$, wobei für alle $i \in \{1,\ldots,l\}$ mit $\mathrm{id}_i = 1$ $D_i \in G_1$ ist. Damit ist die Darstellungsgröße eines geheimen Schlüssels

$$\beta_{\mathrm{SK}}^{B} := l + a_{\mathrm{id}} \alpha_1 \leq l + 1084 a_{\mathrm{id}}.$$

   Im Fall $l = 32$ und $a_{\mathrm{id}} = 8$ gilt

$$\beta_{\mathrm{SK}}^{B} \leq 32 + 1084 \cdot 8 = 8704.$$

3. *Länge der Hauptschlüssel.*

   a) *Verfahren 8.2.1.* Ein Hauptschlüssel hat die Form $\mathrm{MK} = (T_1,\ldots,T_l)$ mit

$T_1, \ldots, T_l \in \mathbb{Z}^{m \times m}$. Für alle $i \in \{1, \ldots, l\}$ ist bis auf eine vernachlässigbare Wahrscheinlichkeit $\|T_i\| \leq mf(m)$. Deshalb kann davon ausgegangen werden, dass alle Beträge der Matrizeneinträge höchstens $\lfloor mf(m) \rfloor$ sind und daher jeder Matrizeneintrag mit $\lfloor \log(\lfloor mf(m) \rfloor) \rfloor + 2$ Bits dargestellt werden kann. Damit ist die Darstellungsgröße eines Hauptschlüssels

$$\begin{aligned}\beta^G_{\mathrm{MK}} &:= lm^2 \left(\lfloor \log(\lfloor mf(m) \rfloor) \rfloor + 2 \right) \\ &\geq lm^2 \left(\left\lfloor \log\left(\lfloor m\sqrt{\log m} \rfloor \right) \right\rfloor + 2 \right) \geq 11642159232l.\end{aligned}$$

Im Fall $l = 32$ ist

$$\beta^G_{\mathrm{MK}} \geq 32 \cdot 877598364909 = 28083147677088.$$

b) *Verfahren 8.3.2.* Ein Hauptschlüssel hat die Form $\mathrm{MK} = (y, t_1, \ldots, t_l)$ mit $y \in \mathbb{Z}_q$ und $t_1, \ldots, t_l \in \mathbb{Z}_q \setminus \{0\}$. Damit ist die Darstellungsgröße eines Hauptschlüssels

$$\beta^B_{\mathrm{MK}} := (l+1) \lceil \log q \rceil = 271(l+1).$$

Im Fall $l = 32$ gilt

$$\beta^B_{\mathrm{MK}} = 271 \cdot 33 = 8943.$$

4. *Länge öffentlicher Parameter.*

a) *Verfahren 8.2.1.* Öffentliche Parameter sind von der Form $\mathrm{PP} = (k, u, A_1, \ldots, A_l)$ mit $k \in \mathbb{N}$, $k \leq l$, $u \in \mathbb{Z}_q^n$ sowie $A_1, \ldots, A_l \in \mathbb{Z}_q^{n \times m}$. Damit ist die Darstellungsgröße öffentlicher Parameter

$$\begin{aligned}\beta^G_{\mathrm{PP}} &:= \lfloor \log k \rfloor + 1 + n\lceil \log q \rceil + lnm\lceil \log q \rceil \\ &\geq \lfloor \log k \rfloor + 1 + 100 \cdot 51 + l \cdot 100 \cdot 25432 \cdot 51 \\ &= \lfloor \log k \rfloor + 5101 + 129703200l.\end{aligned}$$

Im Fall $l = 32$ ist

$$\beta^G_{\mathrm{PP}} \geq \lfloor \log k \rfloor + 1 + 100 \cdot 409 + 32 \cdot 100 \cdot 204427 \cdot 409$$

$$= \lfloor \log k \rfloor + 267554098501.$$

b) *Verfahren 8.3.2.* Öffentliche Parameter sind von der Form $\mathrm{PP} = (k, Y, T_1, \ldots, T_l)$ mit $k \in \mathbb{N}$, $k \leq l$, $Y \in G_2$ und $T_1, \ldots, T_l \in G_1$. Damit ist die Darstellungsgröße öffentlicher Parameter

$$\begin{aligned}\beta^B_{\mathrm{PP}} &:= \lfloor \log k \rfloor + 1 + \alpha_2 + l\alpha_1 \\ &\leq \lfloor \log k \rfloor + 1 + 511 \cdot 5 + l \cdot 2 \cdot 542 = \lfloor \log k \rfloor + 2556 + 1084l.\end{aligned}$$

Im Fall $l = 32$ gilt

$$\beta^B_{\mathrm{PP}} \leq \lfloor \log k \rfloor + 2556 + 1084 \cdot 32 = \lfloor \log k \rfloor + 37244.$$

Alle untersuchten Darstellungsgrößen sind bei Verfahren 8.2.1 deutlich größer als bei Verfahren 8.3.2. Außerdem würde in der Praxis der Sicherheitsparameter von Verfahren 8.2.1 eher noch größer gewählt werden als hier, so dass alle obigen Darstellungsgrößen noch größer würden. Damit ist dieses Verfahren in der Praxis kaum einsetzbar.

10.2. Erweiterbarkeit

Mit Attributbasierter Verschlüsselung können komplexere Zugriffsrechte als mit Fuzzy Identitätsbasierter Verschlüsselung modelliert werden. Deshalb stellt sich die Frage, ob mit den Techniken der vorgestellten Fuzzy Identitätsbasierten Verschlüsselungsverfahren auch Attributbasierte Verschlüsselungsverfahren realisiert werden können. Bei Verfahren 8.3.2 ist solch eine Erweiterung gelungen, die in [GPSW06] beschrieben wird. Inzwischen gibt es zudem noch einige weitere Attributbasierte Verschlüsselungsverfahren, die bilineare Abbildungen nutzen, beispielsweise in [OSW07, Wat11]. Die Erweiterung von Verfahren 8.2.1 auf Attributbasierte Verschlüsselung ist allerdings nicht ohne weiteres möglich. Im Folgenden soll in Anlehnung an Anhang B aus [ABV+12] erklärt werden, welche Probleme bei naiver Erweiterung des Verschlüsselungsverfahrens auftreten.

Das Verfahren in Anhang A aus [GPSW06] ist ein Key-Policy Attributbasiertes Verschlüsselungsverfahren. Nachrichten werden dort unter einer Menge von Attributen, welche als Vektor $\mathrm{id} \in \{0,1\}^l$ modelliert ist, wie in Verfahren 8.3.2 verschlüsselt. Die Zugriffsrechte eines Teilnehmers werden als Boolesche Funktion $F : \{0,1\}^l \to \{0,1\}$ modelliert.

Ein Teilnehmer mit Boolescher Funktion F soll eine unter id verschlüsselte Nachricht genau dann entschlüsseln können, wenn $F(\mathrm{id}) = 1$ ist. Dabei können mit dem Verfahren nur monotone Boolesche Funktionen realisiert werden.

Definition 10.2.1. *Eine Boolesche Funktion $F : \{0,1\}^l \to \{0,1\}$ heißt* monoton, *falls für alle $x := (x_1, \ldots, x_l)$, $y := (y_1, \ldots, y_l) \in \{0,1\}^l$, wobei $F(x) = 1$ ist und für alle $i \in \{1, \ldots, l\}$ gilt, dass $x_i \leq y_i$ ist, ebenfalls $F(y) = 1$ ist.*

Im Algorithmus *KeyGen* aus Verfahren 8.3.2 wurde y aus dem Hauptschlüssel in einem Polynom versteckt und dann wurden Informationen über das Polynom auf die einzelnen Elemente des konstruierten geheimen Schlüssels aufgeteilt. Anstelle von Polynomen werden in dem Verfahren aus [GPSW06] monotone Spann-Programme verwendet.

Definition 10.2.2.

- *Sei $\mathbb{K}$ ein Körper und $l \in \mathbb{N}$. Ein* monotones Spann-Programm *über $\mathbb{K}$ ist ein Paar (M, ρ), wobei $M \in \mathbb{K}^{d \times t}$ und $\rho : \{1, \ldots, d\} \to \{1, \ldots, l\}$ ist.*

- *Sei $id := (id_1, \ldots, id_l) \in \{0,1\}^l$ und (M, ρ) ein monotones Spann-Programm wie oben. Dann ist $M_{id} \subseteq \mathbb{K}^t$, wobei $v \in M_{id}$ genau dann ist, wenn v die i-te Zeile von M für ein $i \in \{1, \ldots, d\}$ mit $id_{\rho(i)} = 1$ ist. Zudem sei $\epsilon := (1, 0, \ldots, 0)^T \in \mathbb{K}^t$.* (M, ρ) akzeptiert *die Eingabe id genau dann, wenn $\epsilon \in \mathrm{span}(M_{id})$ ist.*

Es ist klar, dass es für ein monotones Spann-Programm (M, ρ) eine monotone Boolesche Funktion $F : \{0,1\}^l \to \{0,1\}$ gibt, so dass für jedes $\mathrm{id} \in \{0,1\}$ genau dann $F(\mathrm{id}) = 1$ ist, wenn $\epsilon \in \mathrm{span}(M_{\mathrm{id}})$ ist. Andersherum kann auch für jede monotone Boolesche Funktion F ein monotones Spann-Programm über $\mathbb{Z}_q$ mit $q \in \mathbb{N}$ Primzahl konstruiert werden, welches genau die Eingaben id akzeptiert, für die $F(\mathrm{id}) = 1$ ist [LC10].

So wie Verfahren 8.3.2 zu dem Verfahren in Anhang A aus [GPSW06] erweitert wurde, könnte analog Verfahren 8.2.1 zu einem Key-Policy Attributbasierten Verschlüsselungsverfahren wie folgt erweitert werden. Dafür sei $\lambda \in \mathbb{N}$ ein Sicherheitsparameter und $n := n(\lambda) \in \mathbb{N}$ sowie $m := m(\lambda) \in \mathbb{N}$ weitere Parameter, die polynomiell in λ sind. Ferner sei $q := q(\lambda) \in \mathbb{N}$ eine Primzahl, so dass $\log(q)$ polynomiell in λ ist. $\sigma := \sigma(\lambda) \in \mathbb{R}_{>0}$ sei ebenfalls polynomiell in λ und es gelte $\alpha := \alpha(\lambda) \in (0,1)$ sowie $l \in \mathbb{N}$ mit $l \leq n$.

Verfahren 10.2.3.

- Setup (λ, l).
  1. *Für alle* $i \in \{1, \ldots, l\}$ *sei* $(A_i, T_i) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}^{m \times m}$ *eine Ausgabe von* TrapGen (n, m, q).
  2. *Wähle* $u = (u_1, \ldots, u_n)^T \in \mathbb{Z}_q^n$ *zufällig gleichverteilt.*
  3. *Gib sowohl* $PP := (u, A_1, A_2, \ldots, A_l)$ *als auch* $MK := (T_1, T_2, \ldots, T_l)$ *aus.*
- KeyGen $(PP, MK, (M, \rho))$ *mit* $M \in \mathbb{Z}_q^{d \times t}$ *und* $\rho : \{1, \ldots, d\} \to \{1, \ldots, l\}$.
  1. *Für alle* $i \in \{1, \ldots, n\}$ *wähle* $v_i \in \mathbb{Z}_q^t$ *zufällig, so dass* $\epsilon^T v_i = u_i$ *ist.*
  2. *Es bezeichnen* $M_1, \ldots, M_d$ *die Zeilen der Matrix* M. *Für alle* $j \in \{1, \ldots, d\}$ *sei* $\hat{u}_j := \left(M_j^T v_1, \ldots, M_j^T v_n\right)^T \in \mathbb{Z}_q^n$.
  3. *Für alle* $j \in \{1, \ldots, d\}$ *sei weiter* $e_j \in \mathbb{Z}^m$ *eine Ausgabe von* SamplePre $\left(A_{\rho(j)}, T_{\rho(j)}, \hat{u}_j, \sigma\right)$.
  4. *Gib* $SK_{(M,\rho)} := ((M, \rho), \{e_1, \ldots, e_d\})$ *aus.*
- Enc (PP, id, m) *mit* $id = (id_1, \ldots, id_l)^T \in \{0,1\}^l$ *und* $m \in \{0, 1\}$.
  1. *Wähle* $s \in \mathbb{Z}_q^n$ *zufällig gleichverteilt.*
  2. *Wähle* $x \in \mathbb{Z}_q$ *zufällig* $\overline{\Psi}_\alpha$*-verteilt.*
  3. *Für alle* $j \in \{1, \ldots, l\}$ *mit* $id_j = 1$ *wähle* $x_j \in \mathbb{Z}_q^m$ *zufällig* $\overline{\Psi}_\alpha^m$*-verteilt.*
  4. *Setze* $c_0 := u^T s + x + m \left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q$.
  5. *Für alle* $j \in \{1, \ldots, l\}$ *mit* $id_j = 1$ *setze* $c_j := A_j^T s + x_j \in \mathbb{Z}_q^m$.
  6. *Gib* $CT_{id} := (id, c_0, \{c_j \mid j \in \{1, \ldots, l\}, id_j = 1\})$ *aus.*
- Dec $\left(PP, CT_{id}, SK_{(M,\rho)}\right)$.
  1. *Wenn* $\epsilon \notin \text{span}\,(M_{id})$ *ist, dann gib* $\perp$ *als Platzhalter für „keine Entschlüsselung“ aus.*
  2. *Wenn* $\epsilon \in \text{span}\,(M_{id})$ *ist, dann berechne Folgendes:*
     a) *Setze* $J := \{j \mid j \in \{1, \ldots, d\}, id_{\rho(j)} = 1\}$.
     b) *Für alle* $j \in J$ *berechne* $a_j \in \mathbb{Z}_q$, *so dass* $\sum_{j \in J} a_j M_j = \epsilon$ *ist.*
     c) *Setze* $r := c_0 - \sum_{j \in J} a_j e_j^T c_{\rho(j)} \in \mathbb{Z}_q$.

d) Wenn $\min\{r, q-r\} < \lfloor \frac{q}{4} \rfloor$ *ist, so gib 0 aus, ansonsten gib 1 aus.*

Zunächst kann sich ähnlich wie im Korrektheitsbeweis des Verfahrens 8.2.1 in Theorem 8.2.8 überlegt werden, dass obiges Verfahren bei geeigneter Wahl der Parameter n, m, q, l, σ und α korrekt ist, wenn davon ausgegangen wird, dass monotone Spann-Programme (M, ρ) verwendet werden, für die die Koeffizienten $a_j \in \mathbb{Z}_q$ sowie die Anzahl d der Zeilen von M nicht zu groß werden. Für die Korrektheit ist zu zeigen, dass Verfahren 10.2.3 bis auf eine in λ vernachlässigbare Wahrscheinlichkeit korrekt entschlüsselt, falls ein Teilnehmer mit monotonem Spann-Programm (M, ρ) befugt ist, einen Schlüsseltext zur Identität id zu entschlüsseln, d.h. es gilt $\epsilon \in \text{span}\,(M_{\text{id}})$. Zuerst lässt sich feststellen, dass

$$\begin{aligned}
\sum_{j\in J} a_j \hat{u}_j &= \sum_{j\in J} a_j \left(M_j^T v_1, \ldots, M_j^T v_n\right)^T \\
&= \left(\sum_{j\in J} a_j M_j^T v_1, \ldots, \sum_{j\in J} a_j M_j^T v_n\right)^T \\
&= \left(\left(\sum_{j\in J} a_j M_j\right)^T v_1, \ldots, \left(\sum_{j\in J} a_j M_j\right)^T v_n\right)^T \\
&= \left(\epsilon^T v_1, \ldots, \epsilon^T v_n\right)^T = (u_1, \ldots, u_n)^T = u
\end{aligned}$$

ist. Für alle $j \in \{1, \ldots, d\}$ ist außerdem $e_j \in \Lambda_q^{\hat{u}_j}\left(A_{\rho(j)}\right)$ und deswegen gilt $A_{\rho(j)} e_j = \hat{u}_j$. Damit lässt sich nachrechnen, dass

$$\begin{aligned}
r &= c_0 - \sum_{j\in J} a_j e_j^T c_{\rho(j)} \\
&= u^T s + x + m\left\lfloor \frac{q}{2} \right\rfloor - \sum_{j\in J} a_j e_j^T \left(A_{\rho(j)}^T s + x_{\rho(j)}\right) \\
&= u^T s + x + m\left\lfloor \frac{q}{2} \right\rfloor - \sum_{j\in J} a_j e_j^T A_{\rho(j)}^T s - \sum_{j\in J} a_j e_j^T x_{\rho(j)} \\
&= u^T s - \left(\sum_{j\in J} a_j A_{\rho(j)} e_j\right)^T s + x - \sum_{j\in J} a_j e_j^T x_{\rho(j)} + m\left\lfloor \frac{q}{2} \right\rfloor \\
&= u^T s - \left(\sum_{j\in J} a_j \hat{u}_j\right)^T s + x - \sum_{j\in J} a_j e_j^T x_{\rho(j)} + m\left\lfloor \frac{q}{2} \right\rfloor
\end{aligned}$$

$$= u^T s - u^T s + x - \sum_{j \in J} a_j e_j^T x_{\rho(j)} + m \left\lfloor \frac{q}{2} \right\rfloor$$
$$= m \left\lfloor \frac{q}{2} \right\rfloor + x - \sum_{j \in J} a_j e_j^T x_{\rho(j)}$$

ist. Wie im Beweis von Theorem 8.2.8 muss daher nur noch gezeigt werden, dass $\left| x - \sum_{j \in J} a_j e_j^T x_{\rho(j)} \right|_q < \lfloor \frac{q}{4} \rfloor$ bis auf eine vernachlässigbare Wahrscheinlichkeit ist. Sind die Koeffizienten $a_j \in \mathbb{Z}_q$ sowie $d \in \mathbb{N}$ nicht zu groß, so kann dafür eine Abschätzung analog zum Beweis von Theorem 8.2.8 durchgeführt werden.

Problematisch ist allerdings die Sicherheit des obigen Verfahrens. Das Sicherheitsspiel hat den gleichen Ablauf wie das Fuzzy Selective-ID Spiel aus Definition 9.1.1, außer dass der Angreifer in den Phasen 1 und 2 geheime Schlüssel zu monotonen Spann-Programmen (M, ρ) mit $\epsilon \notin \text{span}\,(M_{\text{id}^*})$ beim Herausforderer erfragen darf. Dabei bezeichnet id^* wieder die Identität, bezüglich welcher der Angreifer herausgefordert werden möchte.

Betrachte nun die Situation, dass der Angreifer in der Phase der Zielfestlegung $\text{id}^* = (1, 1, 0, \ldots, 0)^T \in \{0,1\}^l$ wählt. Nach Erhalten der öffentlichen Parameter erfragt der Angreifer einen geheimen Schlüssel zum monotonen Spann-Programm (M', ρ'), wobei

$$M' := \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & q-1 \end{pmatrix} \in \mathbb{Z}_q^{3 \times 2}$$

und

$$\rho' : \{1,2,3\} \longrightarrow \{1, \ldots, l\},$$
$$\rho'(1) := 1, \rho'(2) := 2, \rho'(3) := 3$$

ist. Dann ist $M'_{\text{id}^*} = \left\{ (1,1)^T \right\}$ und somit $\epsilon = (1,0)^T \notin \text{span}\,(M'_{\text{id}^*})$. Der Angreifer erhält einen geheimen Schlüssel

$$\text{SK}_{(M', \rho')} = \left((M', \rho'), \{e_1, e_2, e_3\} \right).$$

Weil $M'_1 = M'_2$ ist, gilt $A_1 e_1 = \hat{u}_1 = \hat{u}_2 = A_2 e_2$. Es bezeichne $e_{1,2} \in \mathbb{Z}^{2m}$ den Vektor,

der in der ersten Hälfte aus e_1 und der zweiten Hälfte aus $-e_2$ besteht. Zudem sei $A_{1,2} \in \mathbb{Z}_q^{n\times(2m)}$ die Matrix, deren linke Hälfte A_1 und deren rechte Hälfte A_2 ist. Dann ist $A_{1,2}e_{1,2} = A_1e_1 - A_2e_2 = 0$. Demnach ist $e_{1,2} \in \Lambda_q^{\perp}(A_{1,2})$.

Außerdem ist $e_{1,2}$ ein relativ kurzer Vektor. Dafür sei wieder $m \geq 5n\log q$, $f : [1,\infty) \to \mathbb{R}, m \mapsto (\log m)^{\frac{1}{2}+\delta}$ für ein festes $\delta \in \left(0, \frac{1}{2}\right)$ sowie $\sigma := m\,(f(m))^2$. Wegen Theorem 6.2.4 ist für jedes $i \in \{1,\dots,l\}$ bis auf eine vernachlässigbare Wahrscheinlichkeit $\|\widetilde{T_i}\| \leq mf(m)$ und damit $\sigma \geq \|\widetilde{T_i}\| f(m)$. Aufgrund von Lemma 8.2.7 ist für $j \in \{1,2\}$ bis auf eine vernachlässigbare Wahrscheinlichkeit $\|e_j\| \leq m^{\frac{3}{2}}\,(f(m))^2$ und somit ist

$$\|e_{1,2}\| = \sqrt{\|e_1\|^2 + \|e_2\|^2} \leq \|e_1\| + \|e_2\| \leq 2m^{\frac{3}{2}}\,(f(m))^2\,.$$

Mit mehreren Anfragen der obigen Art könnte der Angreifer eine Basis $T_{1,2} \in \mathbb{Z}^{(2m)\times(2m)}$ für $\Lambda_q^{\perp}(A_{1,2})$ erhalten, wobei bis auf eine vernachlässigbare Wahrscheinlichkeit $\|\widetilde{T_{1,2}}\| \leq 2m^{\frac{3}{2}}\,(f(m))^2$ ist. Weil die Verteilung von A_1 und A_2 nach Theorem 6.2.4 statistisch nah zur Gleichverteilung auf $\mathbb{Z}_q^{n\times m}$ ist, ist die Verteilung von $A_{1,2}$ statistisch nah zur Gleichverteilung auf $\mathbb{Z}_q^{n\times(2m)}$ und wegen Lemma 6.2.5 ist bis auf eine vernachlässigbare Wahrscheinlichkeit $\Lambda_q^u(A_{1,2}) \neq \emptyset$. Deswegen kann der Angreifer *SamplePre* bis auf eine vernachlässigbare Wahrscheinlichkeit auf der Eingabe $\left(A_{1,2}, T_{1,2}, u, \|\widetilde{T_{1,2}}\| f(2m)\right)$ ausführen und er erhält einen Vektor $v \in \Lambda_q^u(A_{1,2})$. Wie im Beweis von Lemma 8.2.7 kann gezeigt werden, dass bis auf eine venachlässigbare Wahrscheinlichkeit

$$\|v\| \leq \left(\|\widetilde{T_{1,2}}\| f(2m)\right)\sqrt{2m} \leq \sqrt{8}m^2\,(f(m))^2\,f(2m)$$

ist.

Nun seien $v_1, v_2 \in \mathbb{Z}^m$ so, dass v_1 die obere Hälfte und v_2 die untere Hälfte von v ist. Dann ist $A_1v_1 + A_2v_2 = A_{1,2}v = u$. In der Phase der Herausforderung erhält der Angreifer einen Schlüsseltext $\mathrm{CT}_{\mathrm{id}^*} = (\mathrm{id}^*, c_0, \{c_1, c_2\})$ zu einer Nachricht $b \in \{0,1\}$. Damit kann er $r := c_0 - v_1^T c_1 - v_2^T c_2 \in \mathbb{Z}_q$ berechnen. Dabei ist

$$\begin{aligned} r &= c_0 - v_1^T c_1 - v_2^T c_2 \\ &= u^T s + x + b\left\lfloor \frac{q}{2} \right\rfloor - v_1^T\left(A_1^T s + x_1\right) - v_2^T\left(A_2^T s + x_2\right) \end{aligned}$$

$$
\begin{aligned}
&= u^T s - (A_1 v_1 + A_2 v_2)^T s + x - v_1^T x_1 - v_2^T x_2 + b \left\lfloor \frac{q}{2} \right\rfloor \\
&= u^T s - u^T s + x - v_1^T x_1 - v_2^T x_2 + b \left\lfloor \frac{q}{2} \right\rfloor \\
&= b \left\lfloor \frac{q}{2} \right\rfloor + x - v_1^T x_1 - v_2^T x_2.
\end{aligned}
$$

$\|v\|$ kann allerdings auch deutlich kleiner sein als obige Abschätzung, so dass $\left| x - v_1^T x_1 - v_2^T x_2 \right|_q < \lfloor \frac{q}{4} \rfloor$ ist. Dann könnte der Angreifer durch Prüfung, ob $|r|_q < \lfloor \frac{q}{4} \rfloor$ ist, den erhaltenen Schlüsseltext korrekt entschlüsseln.

Vor Kurzem wurde in [Boy12] ein Key-Policy Attributbasiertes Verschlüsselungsverfahren vorgestellt, welches mit Gittern arbeitet und dessen Sicherheit auf dem *Learning with Erros Problem* beruht. Es verwendet jedoch neue Techniken, so dass oben geschildertes Problem nicht auftritt. Außerdem benutzt es eine Konstruktion für monotone Spann-Programme aus monotonen Booleschen Funktionen, bei der die Koeffizienten $a_j \in \mathbb{Z}_q$ aus Schritt 2b) des Algorithmus *Dec* nicht zu groß werden, um Korrektheit zu gewährleisten.

Literaturverzeichnis

[ABV+12] Agrawal, Shweta ; Boyen, Xavier ; Vaikuntanathan, Vinod ; Voulgaris, Panagiotis ; Wee, Hoeteck: Functional Encryption for Threshold Functions (or Fuzzy IBE) from Lattices. In: *Public Key Cryptography–PKC 2012*. Berlin : Springer, 2012, S. 280–297

[AG11] Arora, Sanjeev ; Ge, Rong: New Algorithms for Learning in Presence of Errors. In: *Automata, Languages and Programming*. Berlin : Springer, 2011, S. 403–415

[Ajt98] Ajtai, Miklós: The Shortest Vector Problem in L_2 is NP-hard for Randomized Reductions. In: *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*. Association for Computing Machinery, 1998, S. 10–19

[Ajt04] Ajtai, Miklós: Generating Hard Instances of Lattice Problems. In: *Complexity of computations and proofs* Bd. 13. Neapel, Caserta : Dipartimento di Matematica, Seconda Università di Napoli, 2004 (quaderni di matematica), S. 1–32

[AP09] Alwen, Joël ; Peikert, Chris: Generating Shorter Bases for Hard Random Lattices. In: *26th International Symposium on Theoretical Aspects of Computer Science*. Wadern : Leibniz Center for Informatics, 2009, S. 75–86

[Ass00] Assenmacher, Walter: *Induktive Statistik*. Berlin : Springer, 2000

[BF03] Boneh, Dan ; Franklin, Matthew: Identity-Based Encryption from the Weil Pairing. In: *SIAM Journal on Computing* Bd. 32, Nr. 3. Philadelphia : Society for Industrial and Applied Mathematics, 2003, S. 586–615

[BMVT78] Berlekamp, Elwyn R. ; McEliece, Robert J. ; Van Tilborg, Henk C. A.: On the Inherent Intractability of Certain Coding Problems. In: *IEEE Transactions on Information Theory* Bd. 24, Nr. 3. IEEE Information Theory Society, 1978, S. 384–386

[Boy12] Boyen, Xavier: Attribute-Based Functional Encryption on Lattices. In: *Cryptology ePrint Archive* Nr. 716. International Association for Cryptologic Research, 2012. – http://eprint.iacr.org/2012/716.pdf

[Fri08] Fritzsche, Klaus: *Grundkurs Funktionentheorie: Eine Einführung in die komplexe Analysis und ihre Anwendungen*. Heidelberg : Spektrum Akademischer Verlag, 2008

[GN08] Gama, Nicolas ; Nguyen, Phong Q.: Predicting Lattice Reduction. In: *Advances in Cryptology–EUROCRYPT 2008*. Berlin : Springer, 2008, S. 31–51

[GPSW06] Goyal, Vipul ; Pandey, Omkant ; Sahai, Amit ; Waters, Brent: Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In: *Cryptology ePrint Archive* Nr. 309. International Association for Cryptologic Research, 2006. – http://eprint.iacr.org/2006/309.pdf

[GPV08] Gentry, Craig ; Peikert, Chris ; Vaikuntanathan, Vinod: How to Use a Short Basis: Trapdoors for Hard Lattices and New Cryptographic Constructions, 2008. – http://www.cs.toronto.edu/∼vinodv/GentryPeikertVSTOC2008.pdf

[HR07] Haviv, Ishay ; Regev, Oded: Tensor-based Hardness of the Shortest Vector Problem to within Almost Polynomial Factors. In: *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*. Association for Computing Machinery, 2007, S. 469–477

[Kö04] Königsberger, Konrad: *Analysis 2*. Berlin : Springer, 2004

[Kar72] Karp, Richard M.: Reducibility among Combinatorial Problems. In: *Complexity of Computer Computations*. New York : Plenum Press, 1972, S. 85–103

[Kho05] Khot, Subhash: Hardness of Approximating the Shortest Vector Problem in Lattices. In: *Journal of the ACM* Bd. 52, Nr. 5. New York : Association for Computing Machinery, 2005, S. 789–808

[KL07] Katz, Jonathan ; Lindell, Yehuda: *Introduction to Modern Cryptography*. Boca Raton, Florida : Chapman & Hall/CRC, 2007 (Chapman & Hall/Crc Cryptography and Network Security Series)

[KM03] Kowalsky, Hans J. ; Michler, Gerhard O.: *Lineare Algebra*. Berlin : De Gruyter, 2003 (de Gruyter Lehrbuch)

[Lap08] Lapidus, Michel L.: *In Search of the Riemann Zeros: Strings, Fractal Membranes and Noncommutative Spacetimes*. Washington, DC : American Mathematical Society, 2008

[LC10] Liu, Zhen ; Cao, Zhenfu: On Efficiently Transferring the Linear Secret-Sharing Scheme Matrix in Ciphertext-Policy Attribute-Based Encryption. In: *Cryptology ePrint Archive* Nr. 374. International Association for Cryptologic Research, 2010. – http://eprint.iacr.org/2010/374.pdf

[LW12] Lewko, Allison ; Waters, Brent: New Proof Methods for Attribute-Based Encryption: Achieving Full Security through Selective Techniques. In: *Advances in Cryptology–CRYPTO 2012*. Berlin : Springer, 2012, S. 180–198

[MG02] Micciancio, Daniele ; Goldwasser, Shafi: *Complexity of Lattice Problems: A Cryptographic Perspective.* Boston, Massachusetts : Kluwer Academic Publishers, 2002

[MR07] Micciancio, Daniele ; Regev, Oded: Worst-Case to Average-Case Reductions Based on Gaussian Measures. In: *SIAM Journal on Computing* Bd. 37, Nr. 1. Philadelphia : Society for Industrial and Applied Mathematics, 2007, S. 267–302

[MR09] Micciancio, Daniele ; Regev, Oded: Lattice-based Cryptography. In: *Post-Quantum Cryptography.* Berlin : Springer, 2009, S. 147–191

[OSW07] Ostrovsky, Rafail ; Sahai, Amit ; Waters, Brent: Attribute-Based Encryption with Non-Monotonic Access Structures. In: *Proceedings of the 14th ACM Conference on Computer and Communications Security.* Association for Computing Machinery, 2007, S. 195–203

[Pei09] Peikert, Chris: Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing.* Association for Computing Machinery, 2009, S. 333–342

[Reg04] Regev, Oded: Vorlesung „Lattices in Computer Science", 2004. – http://www.cims.nyu.edu/∼regev/teaching/lattices_fall_2004/

[Reg09] Regev, Oded: On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In: *Journal of the ACM* Bd. 56, Nr. 6. New York : Association for Computing Machinery, 2009, S. 34:1–34:40

[Reg10] Regev, Oded: The Learning with Errors Problem (Invited Survey). In: *Proceedings of the 2010 IEEE 25th Annual Conference on Computational Complexity.* IEEE Computer Society, 2010, S. 191–204

[Sho97] Shor, Peter W.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. In: *SIAM Journal on Computing* Bd. 26, Nr. 5. Philadelphia : Society for Industrial and Applied Mathematics, 1997, S. 1484–1509

[SW05] Sahai, Amit ; Waters, Brent: Fuzzy Identity-Based Encryption. In: *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques.* Berlin : Springer, 2005, S. 457–473

[Wat11] Waters, Brent: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In: *Public Key Cryptography–PKC 2011.* Berlin : Springer, 2011, S. 53–70