Das NTRU-Kryptosystem

von
Anja Moldenhauer

Bachelorarbeit im Studiengang Mathematik (B. Sc.)
Universität Hamburg

29. September 2009

Betreuer: Prof. Dr. Ulf Kühn
Mitgutachter: PD Dr. Ralf Holtkamp

Versicherung

Die vorliegende Arbeit habe ich selbständig verfasst und keine anderen als die angegebenen Hilfsmittel - insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen - benutzt. Die Arbeit habe ich vorher nicht in einem anderen Prüfungsverfahren eingereicht. Die eingereichte schriftliche Fassung entspricht genau der auf dem elektronischen Speichermedium.

Anja Moldenhauer Datum

Inhaltsverzeichnis

1 Einführung

Meine Bachelorarbeit hat als Hauptgrundlage das Buch „An Introduction to Mathematical Cryptography“ [HPS1] und dessen Verbesserungen [HPS2]. Teilweise sind Passagen aus diesem sinngemäß übersetzt und übernommen worden. Für mehr Informationen zu dem Thema „NTRU“ kann das Kapitel 6 “Lattices and Cryptography“ (aus [HPS1]) empfohlen werden. Die Internetseite http://www.ntru.com bietet einführende und weiterführende Informationen.
Beginnen werde ich meine Arbeit mit einleitenden Erklärungen zu Kryptosystemen. Der Leser erhält damit einen ersten Überblick. Das 2. Kapitel ist den mathematischen Grundlagen gewidmet, die im Allgemeinen nicht in der Analysis und Linearen Algebra Vorlesung behandelt werden, die aber für die Erklärung des NTRU-Kryptosystems wichtig sind. Weiter wird der Stoff der elementaren Zahlentheorie, z. B. die Vorlesung [Kü], vorausgesetzt. Fortfahren werde ich mit der Beschreibung des NTRU-Verfahrens, dieses wird anhand des Ringes der Konvolutionspolynome erklärt. Im 4. Kapitel befasse ich mich mit dem zugrunde liegenden Problem und mit der Überführung zu den Gittern. Zum Schluss wird das NTRU-Kryptosystem in Bezug auf Gitter erläutert und es folgt ein Ausblick auf einen möglichen Angriff mit dem LLL-Algorithmus.

1.1 Einleitung

Schon seit Jahrtausenden versuchen Menschen geheime Nachrichten sicher zu übertragen. Das erste militärische Kryptographie-Verfahren, die Skytale, wurde schon im 5. Jahrhundert vor Christi Geburt von den Spartanern benutzt. Seitdem hat sich die Kryptographie immer weiter entwickelt (für mehr Informationen siehe [Si]). Aus dem Internet sind kryptographische Protokolle nicht mehr wegzudenken. Betrachten wir z. B. den Online-Einkauf, wobei der Käufer seine persönlichen Daten einem Unternehmen mitteilen muss, wenn er einen Vertrag mit ihm abschließt. Diese persönlichen Daten werden mit dem öffentlichen Schlüssel des Unternehmens verschlüsselt und diesem zugesandt. Das Unternehmen kann mit seinem privaten Schlüssel die Daten des Käufers wieder entschlüsseln ([Si]).
Im Jahr 1976 trafen sich Whitfield Diffie und Martin Hellman an der Westküste der USA in Kalifornien und führten dieses Konzept des asymmetrischen Kryptosystems ein. Ronald L. Rivest, Adi Shamir und Leonard Adleman erfanden 1977 das nach ihnen benannte RSA-Verfahren, das heutzutage als das asymmetrische Standardverfahren schlechthin gilt.
Es ist sinnvoll, viele unterschiedliche Kryptosysteme zu kennen, die auf verschiedenen schweren Problemen in der Mathematik beruhen. Kommt es in nächster Zeit zu einem Durchbruch beim Lösen eines schweren Problems in der Mathematik, so sollte dadurch nicht die Sicherheit aller Systeme gefährdet sein. Des weiteren sind gitterbasierende Kryptosysteme, wie das hier behandelte NTRU-Verfahren, häufig viel schneller (siehe Abschnitt 3.7) als faktorisierungsbasierende und diskrete Logarithmus basierende Kryptosysteme, wie das RSA-Verfahren und das ElGamal-Verfahren (für Informationen zu diesen Kryptosystemen, siehe z. B. [HPS1]).

1.2 Abstrakte Formulierung eines Kryptosystems

In der Kryptographie geht es im Allgemeinen um den sicheren und geheimen Informationsaustausch zwischen zwei Parteien. Zur Veranschaulichung nennen wir diese Alice und Bob. Es folgt ein Schema eines verschlüsselten Nachrichtenaustausches zwischen diesen Parteien:

Bob:

Er wählt seinen Klartext und wendet den Verschlüsselungsschlüssel auf diesen an. Hierzu muss der Klartext in einem vorgegebenen Datenformat dargestellt sein, z.B. ASCII. Bob erhält einen Geheimtext. Er sendet diesen an Alice.

Alice:

Sie wendet auf den Geheimtext den Entschlüsselungsschlüssel an. Alice erhält das bestimmte Datenformat zurück und kann dieses in den Klartext von Bob umwandeln.

1.3 Symmetrische und asymmetrische Kryptosysteme

Es gibt allgemein zwei Arten von Kryptosystemen, das symmetrische und das asymmetrische Kryptosystem. Beide werden im Folgenden kurz beschrieben.
Bei dem **symmetrischen Kryptosystem** einigen sich Bob und Alice gemeinsam auf einen Schlüssel, der sowohl für die Verschlüsselung als auch für die Entschlüsselung benutzt wird. Das Problem hierbei liegt darin, dass Bob und Alice sich absprechen müssen, welchen Schlüssel sie wählen. Deshalb treffen sie sich oder tauschen über einen sicheren Kanal die benötigten Informationen aus. Eine dritte Person, genannt Eve, darf nicht an die Schlüsselinformation gelangen. Ansonsten könnte Eve den Geheimtext entschlüsseln und somit die geheime Nachricht lesen. Das Kryptosystem wird symmetrisch genannt, da für die Ver- und Entschlüsselung dieselben Schlüssel benutzt werden. Ein Beispiel für ein solches Verfahren ist die Caesar-Verschlüsselung (siehe z. B. [Si]).

Bei dem **asymmetrischen Kryptosystem** werden für die Ver- und Entschlüsselung verschiedene Schlüssel benutzt. Möchte Bob eine Nachricht an Alice schicken, so erstellt diese zwei verschiedene Schlüssel, einen öffentlichen, zu dem jeder Zugang hat, und einen dazu passenden privaten Schlüssel, den sie geheim hält. Der öffentliche Schlüssel wird z. B. bei einer vertrauenswürdigen Verwaltungsstelle hinterlegt, so dass Bob sichergehen kann, dass er auch wirklich Alices öffentlichen Schlüssel erhält. Mit diesem öffentlichen Schlüssel kann er seine Nachricht verschlüsseln und dann an Alice verschicken. Diese kann mit ihrem privaten Schlüssel die Nachricht entschlüsseln. Solch ein Verfahren wird auch als „public key cryptosystem“ bezeichnet, weil es einen öffentlichen Schlüssel gibt, zu dem jeder Zugang hat. Asymmetrisch wird es genannt, da es zwei verschiedene Schlüssel gibt. Einen für die Verschlüsselung und einen anderen für die Entschlüsselung. Der Vorteil hierbei liegt darin, dass Bob und Alice sich nicht persönlich treffen müssen, bzw. sich nicht vor der Nachrichtenübermittlung auf einen gemeinsamen Schlüssel einigen brauchen. Eve kann durch belauschen nicht an den Schlüssel für die Entschlüsselung gelangen. Ein Nachteil entsteht, wenn dieselbe Nachricht an verschiedene Parteien geschickt werden soll. Jede Partei hat einen anderen öffentlichen Schlüssel und so muss die Nachricht jeweils neu mit

deren öffentlichen Schlüssel verschlüsselt werden. Beispiele für ein asymmetrische Kryptosystem sind das RSA-Verfahren oder das, in dieser Arbeit behandelte, NTRU-Verfahren.

1.4 Gruppen- und ringbasierende Kryptosysteme

Ein Kryptosystem, das heutzutage benutzt wird, ist das wohl bekannte gruppenbasierende RSA-Kryptosystem (z. B. aus der Vorlesung [Kü]). Dieses operiert im Restklassenring modulo m, verwendet aber nur die Multiplikation bzgl. der Gruppe der invertierbaren Elemente, deshalb wird es als gruppenbasierendes Kryptosystem bezeichnet. Das RSA-Verfahren ist in einem Ring eingebettet, somit könnte die zusätzliche Additionsoperation herangezogen werden, um dieses System anzugreifen. Wenn beide Operationen für einen Angriff benutzt werden können, macht es Sinn sich ein Kryptosystem zu überlegen, das auch beide Operationen für die Ver- bzw. Entschlüsselung verwendet. Das NTRU-Kryptosystem basiert auf dem Ring der Konvolutionspolynome (siehe Abschnitt 2.1). Bei diesem Verfahren, welches im 3. Kapitel (Beschreibung des NTRU-Verfahrens) ausführlich erläutert wird, benötigt sowohl die Verschlüsselung als auch die Entschlüsselung die Addition und die Multiplikation aus dem Ring. Das NTRU-Kryptosystem wird deshalb ringbasierend genannt.

1.5 NTRU

In dieser Bachelorarbeit geht es um das asymmetrische ringbasierende public key Kryptosystem NTRU. Es wurde 1996 von Jeffrey Hoffstein, Jill Pipher und Joseph H. Silverman auf der „Rump-Session“ der Crypto ’96 in Santa Barbara, Kalifornien, vorgestellt.
Für die Bezeichnung „NTRU“ finden sich folgende Erklärungsversuche (siehe [Ru]):
NTRU - **N**-th degree **TRU**ncated polynomial ring,
NTRU - **N**umber **T**heory **R**esearch **U**nit,
NTRU - **N**umber **T**heorists a**R**e **U**s.

2 Mathematische Grundlagen

In diesem Kapitel werden die mathematischen Grundlagen für das NTRU-Kryptosystem dargestellt. Zuerst wird der Ring der Konvolutionspolynome erläutert, der für die Beschreibung des NTRU-Kryptosystems benötigt wird. Danach kommt eine kurze Einführung in das Gebiet der Gitter. Es werden schwere Probleme im Gitter vorgestellt und eine Heuristik, die eine zu erwartende Länge für einen kürzesten Vektor in einem Gitter angibt.

2.1 Der Ring der Konvolutionspolynome

Kommen wir zur Erläuterung der speziellen polynomialen Quotientenringe. Im Folgenden wird mit $\mathbb{N}$ die Menge der natürlichen Zahlen ohne die Null bezeichnet.

Definition 2.1. Sei $N \in \mathbb{N}$. Der *Ring der Konvolutionspolynome $\mathcal{R}^N$ (vom Rang N)* ist der Quotientenring

$$\mathcal{R}^N = \mathbb{Z}[x] / \left(x^N - 1\right).$$

Der *Ring der Konvolutionspolynome* $\mathcal{R}_q^N$ *(modulo q)* ist der Quotientenring

$$\mathcal{R}_q^N = (\mathbb{Z}/q\mathbb{Z})[x] / \left(x^N - 1\right).$$

Die Koeffizienten werden aus dem Ring $\mathbb{Z}/q\mathbb{Z}$ gewählt. Hierbei ist $q \in \mathbb{N}$ nicht notwendigerweise eine Primzahl.

Bemerkungen 2.2.

(i) Jedes Element $a(x)$ aus $\mathcal{R}^N$ oder $\mathcal{R}_q^N$ ist eindeutig darstellbar als

$$a(x) = \sum_{i=0}^{N-1} a_i x^i,$$

mit Koeffizienten a_i aus $\mathbb{Z}$ bzw. aus $\mathbb{Z}/q\mathbb{Z}$.

(ii) Die Modulo-Rechnung der Polynome aus $\mathbb{Z}[x]$ bzw. $(\mathbb{Z}/q\mathbb{Z})[x]$ mit $x^N - 1$ ist das Gleiche, wie das Ersetzen von x^N durch 1. Ist im Polynomring $\mathcal{R}^N$ oder $\mathcal{R}_q^N$ z. B. eine Variable x^k mit $k \geq N$ vorhanden und wird dieser Polynomring modulo $x^N - 1$ gerechnet, dann wird k dargestellt als $k = iN + j$ mit $0 \leq j < N$ und somit gilt dann

$$x^k = x^{iN+j} = x^{iN}x^j = \left(x^N\right)^i x^j = 1^i x^j = x^j.$$

Mit anderen Worten: Wir rechnen den Exponenten von x modulo N.

(iii) Ein Polynom

$$a(x) = a_{N-1}x^{N-1} + a_{N-2}x^{N-2} + \ldots + a_1 x^1 + a_0 \in \mathcal{R}^N$$

identifizieren wir mit dem Vektor seiner Koeffizienten

$$(a_0, a_1, \ldots, a_{N-2}, a_{N-1}) \in \mathbb{Z}^N.$$

Analoges gilt für die Polynome aus $\mathcal{R}_q^N$.

(iv) Die Addition von Polynomen ist wie die Addition von Vektoren

$$a(x) + b(x) \longleftrightarrow (a_0 + b_0, a_1 + b_1, \ldots, a_{N-2} + b_{N-2}, a_{N-1} + b_{N-1}).$$

(v) Für die Multiplikation in $\mathcal{R}^N$ oder $\mathcal{R}_q^N$ wird das nachfolgende Lemma 2.3 benötigt. Es wird $\star$ geschrieben um zu verdeutlichen, dass die Multiplikation in $\mathcal{R}^N$ oder $\mathcal{R}_q^N$ gemeint ist. Diese Multiplikation wird auch das Konvolutionsprodukt genannt.

Lemma 2.3. *Das Produkt von zwei Polynomen $a(x)$, $b(x) \in \mathcal{R}^N$ wird durch die Formel*

$$a(x) \star b(x) = c(x) \qquad mit \qquad c_k = \sum_{i+j \equiv k \ (mod\ N)} a_i b_{k-i}$$

bestimmt. Wobei $0 \leq i, j \leq N-1$ gilt und die Bedingung $i + j \equiv k \ (mod\ N)$ erfüllt ist. Dies gilt auch für Polynome $a(x)$ und $b(x)$ aus $\mathcal{R}_q^N$, wobei die Koeffizienten c_k des Konvolutionsproduktes modulo q gerechnet werden müssen.

Beweis. Zuerst führen wir das Produkt der Polynome $a(x)$ und $b(x)$ aus. Wir betrachten also

$$\begin{aligned} a(x) \cdot b(x) &= \left(\sum_{i=0}^{N-1} a_i x^i \right) \cdot \left(\sum_{j=0}^{N-1} b_j x^j \right) \\ &= \sum_{0 \le i,j \le N-1} a_i b_j x^{i+j} \\ &= \sum_{k=0}^{2N-2} \left(\sum_{i+j=k} a_i b_j \right) x^k \\ &= \sum_{k=0}^{N-1} \left(\sum_{i+j=k} a_i b_j \right) x^k + \sum_{k=N}^{2N-2} \left(\sum_{i+j=k} a_i b_j \right) x^k. \end{aligned}$$

Um die Terme so zusammenzufassen, damit das Konvolutionsprodukt wieder im Ring der Konvolutionspolynome liegt, rechnen wir den Exponenten der Variablen x modulo N. Somit ergibt sich

$$\begin{aligned} a(x) \star b(x) &= \left(\sum_{i=0}^{N-1} a_i x^i \right) \star \left(\sum_{j=0}^{N-1} b_j x^j \right) \\ &= \sum_{k=0}^{N-1} \left(\sum_{i+j=k} a_i b_j \right) x^k + \sum_{k=N}^{2N-2} \left(\sum_{i+j=k} a_i b_j \right) x^{k-N} \\ &= \sum_{k=0}^{N-1} \left(\sum_{i+j=k} a_i b_j \right) x^k + \sum_{k=0}^{N-2} \left(\sum_{i+j=k+N} a_i b_j \right) x^k \\ &= \sum_{k=0}^{N-1} \left(\sum_{i+j \equiv k \,(\mathrm{mod\ N})} a_i b_j \right) x^k \\ &= \sum_{k=0}^{N-1} \left(\underbrace{\sum_{i+j \equiv k \,(\mathrm{mod\ N})} a_i b_{k-i}}_{=c_k} \right) x^k. \end{aligned}$$

□

Bemerkung 2.4. *Wenn das Konvolutionsprodukt $c(x) = a(x) \star b(x)$, von zwei Polynomen aus $\mathcal{R}^N$ mit N Koeffizienten, genauer betrachtet wird, dann erkennen wir, dass jeder Koeffizient c_k von dem Polynom $c(x) \in \mathcal{R}^N$ das Skalarprodukt von zwei Vektoren ist*

$$c_k = (a_m, a_{m-1}, \ldots, a_1, a_0, a_{N-1}, \ldots, a_{m+1}) \cdot (b_i, b_{i+1}, \ldots, b_{N-1}, b_0, b_1, \ldots, b_{i-1})^t.$$

Hierbei muss gelten $m + i \equiv k \ (mod\ N)$ für $k = 0, \ldots, N-1$ und $m, i \in \{0, \ldots, N-1\}$. Analoges gilt für Polynome aus $\mathcal{R}_q^N$, wobei darauf zu achten ist, dass die Koeffizienten aus dem Ring $\mathbb{Z}/q\mathbb{Z}$ sind.

Für das NTRU-Verfahren sind zwei Abbildungen wichtig.
Einmal gibt es eine natürliche Abbildung $\mathcal{R}^N \to \mathcal{R}_q^N$ mit $a'(x) \mapsto a(x)$ wobei $a'(x)(\text{mod } q) \equiv a(x)$ gilt. Bei der also die Koeffizienten eines Polynoms $a'(x) \in \mathcal{R}^N$ modulo q gerechnet werden. Für die zweite Abbildung von $\mathcal{R}_q^N \to \mathcal{R}^N$, definieren wir uns einen *zentralen Lift*.

Definition 2.5. Der *zentrale Lift* wird von $\mathcal{R}_q^N$ nach $\mathcal{R}^N$ definiert, indem $a(x) \in \mathcal{R}_q^N$ ein eindeutiges Polynom $a'(x) \in \mathcal{R}^N$ zugeordnet wird, mit der Eigenschaft:

$$a'(x) \ (\text{mod } q) \equiv a(x)$$

und mit Koeffizienten

$$-\frac{q}{2} < a_i' \leq \frac{q}{2}.$$

Um die NTRU-Schlüssel zu entwickeln, kann uns folgendes Lemma behilflich sein.

Lemma 2.6. *Sei q eine Primzahl. Das Polynom $a(x) \in \mathcal{R}_q^N$ hat genau dann ein multiplikatives Inverses, wenn*

$$ggT(a(x), x^N - 1) = 1 \qquad in \qquad (\mathbb{Z}/q\mathbb{Z})\,[x]$$

gilt. Wenn $a(x)$ und $x^N - 1$ teilerfremd sind, dann kann das Inverse $a(x)^{-1} \in \mathcal{R}_q^N$ mit dem erweiterten euklidischen Algorithmus berechnet werden, so dass die Polynome $u(x)$ und $v(x)$ in $(\mathbb{Z}/q\mathbb{Z})\,[x]$ sind, mit:

$$a(x)u(x) + (x^N - 1)v(x) = 1.$$

Dann ist $a(x)^{-1} = u(x)$ in $\mathcal{R}_q^N$.

Beweis. Als Erstes beweisen wir die notwendige Bedingung: Nach dem erweiterten euklidischen Algorithmus für Polynome ist bekannt, dass es Polynome $v(x)$, $u(x)$ im Polynomring $(\mathbb{Z}/q\mathbb{Z})[x]$ gibt, mit

$$a(x)u(x) + (x^N - 1)v(x) = \text{ggT}(a(x), x^N - 1).$$

Gilt nun $\text{ggT}(a(x), x^N - 1) = 1$, dann ergibt die Rechnung modulo $x^N - 1$ gerade $a(x) \star u(x) = 1$ in $\mathcal{R}_q^N$. Somit ist $u(x) = a(x)^{-1}$ in $\mathcal{R}_q^N$.
Nun zeigen wir die hinreichende Bedingung: Ist $a(x)$ eine Einheit in $\mathcal{R}_q^N$, dann gibt es ein Polynom $u(x)$, so dass $a(x) \star u(x) = 1$ in $\mathcal{R}_q^N$ ist. Nach der Definition von $\mathcal{R}_q^N$ bedeutet dies

$$a(x)u(x) \equiv 1 \ \ (\text{mod}(x^N - 1)).$$

Aufgrund der Definition der Kongruenz folgt, dass es ein Polynom $v(x)$ gibt, mit:

$$a(x)u(x) - 1 = (x^N - 1)v(x) \ \text{ in } (\mathbb{Z}/q\mathbb{Z})[x].$$

Hieraus schließen wir, dass jeder gemeinsame Teiler von $a(x)$ und $x^N - 1$ auch die 1 teilen muss. Daher folgt für den größten gemeinsamen Teiler: $\text{ggT}(a(x), x^N - 1) = 1$ in $(\mathbb{Z}/q\mathbb{Z})[x]$.
□

Die nächste Definition führt eine bestimmte Art von Konvolutionspolynomen ein.

Definition 2.7. Es seien d_1, d_2 und N aus $\mathbb{N}$, mit $d_1 + d_2 \leq N$. Es wird folgende Menge definiert:

$$\mathcal{T}(d_1, d_2, N) := \Bigg\{ p(x) = \textstyle\sum_{i=0}^{N-1} a_i x^i \in \mathcal{R}^N \Big| \; a_i \in \{-1, 0, 1\} \text{ für } i = 0, \ldots, N-1, \\ \text{mit } \; \#\{a_i | a_i = 1\} = d_1, \; \#\{a_i | a_i = -1\} = d_2 \Bigg\}.$$

Die Polynome $p(x) \in \mathcal{T}(d_1, d_2, N)$ werden *ternäre Polynome* genannt.

2.2 Gitter

Kommen wir zum Begriff des Gitters (vergleiche [HPS1], [Ne]).

Definition 2.8. Seien $v_1, v_2, \ldots, v_n \in \mathbb{R}^m$ linear unabhängige Vektoren. Das *Gitter* $\mathcal{L}$, das von $v_1, v_2, \ldots, v_n$ erzeugt wird, ist die Menge der Linearkombinationen von $v_1, v_2, \ldots, v_n$ mit Koeffizienten aus $\mathbb{Z}$

$$\mathcal{L} = \{\alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n | \alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{Z}\}.$$

Das Gitter heißt vollständig oder eine $\mathbb{Z}$-Struktur, wenn $m = n$ ist.

Ein Gitter ist ähnlich wie der Vektorraum $\mathbb{R}^n$, wobei die Koeffizienten beliebiger Linearkombinationen der Basisvektoren eingeschränkt sind auf den Zahlenbereich der ganzen Zahlen.

Als **Basis** vom Gitter $\mathcal{L}$ wird jede linear unabhängige Menge von Vektoren bezeichnet, die $\mathcal{L}$ erzeugt. Die **Dimension** gibt die Anzahl der Basisvektoren an. Diese Bezeichnungen sind analog, wie die Definitionen für Dimension und Basis im Vektorraum. Genau wie im Vektorraum haben zwei verschiedene Basen zu demselben Gitter die gleiche Dimension. Kommen wir in der folgenden Proposition zu einer weiteren Eigenschaft von zwei Basen.

Proposition 2.9. *Je zwei verschiedene Basen eines Gitters $\mathcal{L}$ können mit Hilfe einer Matrix U, mit Einträgen $u_{ij} \in \mathbb{Z}$ und $det(U) = \pm 1$, ineinander überführt werden.*

Beweis. Es sei $\mathcal{A} = \{v_1, v_2, \ldots, v_n\}$ eine Basis des Gitter $\mathcal{L}$, hierzu gehört folgende Matrix

$$A := \begin{pmatrix} v_1^t \\ v_2^t \\ \vdots \\ v_n^t \end{pmatrix} = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nn} \end{pmatrix}.$$

Eine weitere Basis des Gitters $\mathcal{L}$ sei die Menge $\mathcal{B} = \{w_1, w_2, \ldots, w_n\}$. Mit der dazu-

gehörigen Matrix

$$B := \begin{pmatrix} w_1^t \\ w_2^t \\ \vdots \\ w_n^t \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{pmatrix}.$$

Sollen nun die Vektoren der Basis $\mathcal{B}$ durch die Vektoren der Basis $\mathcal{A}$ dargestellt werden, so ergibt sich die Gleichung:

$$B = \begin{pmatrix} w_1^t \\ w_2^t \\ \vdots \\ w_n^t \end{pmatrix} = \overbrace{\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nn} \end{pmatrix}}^{U:=} \cdot \begin{pmatrix} v_1^t \\ v_2^t \\ \vdots \\ v_n^t \end{pmatrix}$$
$$= U \cdot A.$$

Die Vektoren aus der Basis $\mathcal{B}$ sind linear unabhängig und somit hat die Matrix U eine von Null verschiedene Determinante. Die inverse Matrix U^{-1} existiert.
Die Basis $\mathcal{A}$ kann durch die Vektoren der Basis $\mathcal{B}$ dargestellt werden, indem wir

$$U \cdot A = B \qquad \Leftrightarrow \qquad A = U^{-1} \cdot B$$

berechnen.
Nach Voraussetzung befinden wir uns in einem Gitter, somit müssen die Elemente der Matrizen U und U^{-1} aus den ganzen Zahlen stammen.
Um die Proposition zu beweisen, müssen wir noch folgende Äquivalenz zeigen: Für die Determinante einer Matrix U mit Einträgen aus den ganzen Zahlen gilt, $det(U) = \pm 1$ genau dann, wenn die Matrix U^{-1} nur Einträge aus $\mathbb{Z}$ besitzt.
Zuerst zeigen wir die notwendige Bedingung: Die Matrizen U und U^{-1} haben jeweils nur ganzzahlige Einträge, also ergibt sich zunächst, für dessen Determinanten

$$\det(U) \in \mathbb{Z} \text{ und } \det(U^{-1}) \in \mathbb{Z}. \tag{1}$$

Aus der Gleichung

$$1 = det(I) = det(U^{-1}U) = det(U^{-1})det(U)$$

können wir mit (1) schließen, dass $det(U) = \pm 1$ und auch $det(U^{-1}) = \pm 1$ gilt.
Zum Schluss beweisen wir die hinreichende Bedingung: Hat die Matrix U ganzzahlige Einträge und $det(U) = \pm 1$, dann folgt aus der Berechnung der Inversenmatrix U^{-1}, dass die Matrixelemente von U^{-1} nur aus $\mathbb{Z}$ stammen. Die Inverse der Matrix U kann mit Hilfe einer, aus der Linearen Algebra bekannten Formel, berechnet werden:

$$U^{-1} = \frac{1}{det(U)} \cdot C^t \qquad \text{mit } C = (c_{ij}) \in M(n \times n; \mathbb{Z}),\ c_{ij} := (-1)^{i+j} det(U'_{ij}),$$

wobei U'_{ij} die Streichungsmatrix ist, die entsteht wenn die i-te Zeile und j-te Spalte der Matrix U gestrichen wird.
Die Matrix U besitzt ganzzahlige Einträge, somit hat auch U'_{ij} Matrixeinträge aus den ganzen Zahlen und es gilt $det(U'_{ij}) \in \mathbb{Z}$. Weiter ist $det(U) = \pm 1$. Aus diesen Gründen ergibt sich insgesamt, dass die Matrix U^{-1} nur Einträge aus $\mathbb{Z}$ besitzt. □

Wird die Basis $\mathcal{A} = \{v_1, v_2, \ldots, v_n\}$ des Gitters $\mathcal{L}$ als Matrix A geschrieben und von links mit einer Matrix U multipliziert, welche nur Einträge aus den ganzen Zahlen und $det(U) = \pm 1$ hat, so ergibt sich eine neue Basis $\mathcal{B}$ des Gitters $\mathcal{L}$. Die Matrix U kommt aus der Menge der Matrizen, welche die allgemeine lineare Gruppe ($GL(n, \mathbb{Z})$) über $\mathbb{Z}$ genannt wird.

Kommen wir zu einer Definition einer wichtigen Menge in einem Gitter.

Definition 2.10. Sei $\mathcal{L}$ ein Gitter der Dimension n und die Vektoren $v_1, v_2, \ldots, v_n$ bilden eine Basis des Gitters. Die zu dieser Basis zugehörige Menge

$$\mathcal{F}(v_1, v_2, \ldots, v_n) = \{t_1 v_1 + t_2 v_2 + \ldots + t_n v_n | 0 \leq t_i < 1\}$$

wird *Grundmasche* genannt.

Proposition 2.11. *Es sei $\mathcal{L} \subset \mathbb{R}^n$ ein Gitter der Dimension n und $\mathcal{F}$ eine Grundmasche von $\mathcal{L}$. Dann kann jeder Vektor $w \in \mathbb{R}^n$ in der Form*

$$w = u + f,$$

für ein eindeutiges $f \in \mathcal{F}$ und ein eindeutiges $u \in \mathcal{L}$, dargestellt werden.

Beweis. Seien die Vektoren $v_1, v_2, \ldots, v_n$ eine Basis des Gitters $\mathcal{L}$, die die Grundmasche $\mathcal{F}$ angibt.
Zunächst wird die *Existenz* dieser Darstellung bewiesen.
Da die Menge $\{v_1, v_2, \ldots, v_n\}$ linear unabhängig ist und somit auch eine Basis des Vektorraums $\mathbb{R}^n$ bildet, kann jeder Vektor $w \in \mathbb{R}^n$ dargestellt werden als:

$$w = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n \qquad \text{mit} \qquad \alpha_1, \ldots, \alpha_n \in \mathbb{R}.$$

Wir schreiben jeden Koeffizienten α_i als

$$\alpha_i = a_i + t_i \qquad \text{mit} \qquad a_i \in \mathbb{Z} \text{ und } 0 \leq t_i < 1,$$

dann folgt

$$\begin{aligned} w &= (a_1 + t_1)v_1 + (a_2 + t_2)v_2 + \ldots + (a_n + t_n)v_n \\ &= (\underbrace{a_1 v_1 + a_2 v_2 + \ldots + a_n v_n}_{\text{das ist ein Vektor } u \in \mathcal{L}}) + (\underbrace{t_1 v_1 + t_2 v_2 + \ldots + t_n v_n}_{\text{das ist ein Vektor } f \in \mathcal{F}}). \end{aligned}$$

Somit ist gezeigt, dass jeder Vektor $w \in \mathbb{R}^n$ in der gewünschten Form darstellbar ist.
Nun wird die *Eindeutigkeit* dieser Darstellung bewiesen.
Angenommen es gibt zwei verschiedene Darstellungen eines Vektors $w \in \mathbb{R}^n$ als

$$w = u + f = u' + f' \qquad \text{mit } u, u' \in \mathcal{L} \text{ und } f, f' \in \mathcal{F}.$$

Also

$$\begin{aligned} w &= (a_1 + t_1)v_1 + (a_2 + t_2)v_2 + \ldots + (a_n + t_n)v_n \\ &= (a_1' + t_1')v_1 + (a_2' + t_2')v_2 + \ldots + (a_n' + t_n')v_n. \end{aligned}$$

Die Vektoren $v_1, v_2, \ldots, v_n$ bilden eine Basis und sind somit linear unabhängig, daher ergibt sich

$$a_i + t_i = a_i' + t_i' \qquad \text{für } \; i = 1, 2, \ldots, n.$$

Demzufolge ist

$$t_i - t_i' = a_i' - a_i \in \mathbb{Z}.$$

Für die Skalare t_i und t_i' gilt $0 \leq t_i, t_i' < 1$. Somit ist die einzige Möglichkeit, dass $t_i - t_i' \in \mathbb{Z}$ gilt, wenn die Gleichheit $t_i = t_i'$ erfüllt ist. Deshalb entspricht der Vektor f dem Vektor f' und aus den Gleichungen $u = w - f'$ und $u' = w - f$ folgern wir mit der Gleichheit $f = f'$ schließlich

$$u = w - f' = w - f = u'.$$

Danach sind die Vektoren $f \in \mathcal{F}$ und $u \in \mathcal{L}$ für die Darstellung des Vektors w eindeutig. □

Eine wichtige Größe in dem Gittern $\mathcal{L}$ ist die Determinante von $\mathcal{L}$.

Definition 2.12. Sei $\mathcal{L}$ ein Gitter der Dimension n und $\mathcal{F}$ eine Grundmasche von $\mathcal{L}$. Dann wird das n-dimensionale Volumen von $\mathcal{F}$ die *Determinante von* $\mathcal{L}$ genannt. Sie wird mit $\det(\mathcal{L})$ bezeichnet.

Die folgende Proposition erklärt, wie die Determinante eines Gitters berechnet wird.

Proposition 2.13. *Es sei $\mathcal{L} \subset \mathbb{R}$ ein Gitter der Dimension n, die Basis des Gitters $\mathcal{L}$ seien die Vektoren $v_1, v_2, \ldots, v_n$ und $\mathcal{F} = \mathcal{F}(v_1, v_2, \ldots, v_n)$ sei die dazugehörige Grundmasche. Die Koordinaten des i-ten Basisvektors werden geschrieben als*

$$v_i = (v_{i1}, v_{i2}, \ldots, v_{in}),$$

die Koordinaten des Vektors v_i werden für die Zeilen der Matrix F benutzt

$$F = F(v_1, v_2, \ldots, v_n) = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nn} \end{pmatrix}. \tag{2}$$

Dann ist das n-dimensionale Volumen von $\mathcal{F}$ gegeben durch die Formel

$$det(\mathcal{L}) = Vol(\mathcal{F}(v_1, v_2, \ldots, v_n)) = |det\,(F(v_1, v_2, \ldots, v_n))|\,.$$

Beweis. Für den Beweis benötigen wir mehrere Variablen. Das Volumen vom Gitter $\mathcal{L}$ fassen wir als ein Integral der konstanten Funktion 1 über $\mathcal{F}$ auf

$$Vol(\mathcal{F}) = \int_{\mathcal{F}} dx_1 dx_2 \dots dx_n.$$

Die Grundmasche ist die Menge aus der Definition 2.10; daher wird ein Variablentausch durchgeführt, von $x = (x_1, x_2, \dots, x_n)$ nach $t = (t_1, t_2, \dots, t_n)$ gemäß der Formel

$$(x_1, x_2, \dots, x_n) = t_1 v_1 + t_2 v_1 + \dots + t_n v_1.$$

Mit der Matrixdarstellung (2) von $F = F(v_1, v_2, \dots, v_n)$ kann der Variablenwechsel auch als Matrixgleichung $x = tF$ geschrieben werden. Die Jacobi-Matrix dieses Wechsels ist F und die Grundmasche $\mathcal{F}$ ist das Bild unter F auf dem Einheitswürfel $C_n = [0,1]^n$, dann ergibt sich:

$$\begin{aligned} det(\mathcal{L}) = Vol(\mathcal{F}(v_1, v_2, \dots, v_n)) = Vol(\mathcal{F}) &= \int_{\mathcal{F}} dx_1 dx_2 \dots dx_n \\ &= \int_{FC_n} dx_1 dx_2 \dots dx_n \\ &= \int_{C_n} |det(F)|\, dt_1 dt_2 \dots dt_n \\ &= |det(F)|\, Vol(C_n) \\ &= |det(F)| \\ &= |det(F\,(v_1, v_2, \dots, v_n))|\,. \end{aligned}$$

□

Eine wichtige Eigenschaft der Determinante wird durch das folgende Korollar beschrieben.

Korollar 2.14. *Es sei $\mathcal{L} \subset \mathbb{R}$ ein Gitter der Dimension n, dann hat jede Grundmasche vom Gitter $\mathcal{L}$ dasselbe Volumen. Somit ist $det(\mathcal{L})$ eine feste Größe des Gitters und unabhängig von der Grundmasche, die benutzt wurde um die Determinante zu berechnen.*

Beweis. Seien die Vektoren $v_1, v_2, \dots, v_n$ und $w_1, w_2, \dots, w_n$ zwei verschiedene Basen des Gitters $\mathcal{L}$ und $F(v_1, v_2, \dots, v_n)$ bzw. $F(w_1, w_2, \dots, w_n)$ die zugehörigen Matrizen, beschrieben wie bei Proposition 2.13 (2). Für $n \times n$ Matrizen mit Einträgen aus den ganzen Zahlen wissen wir aus Proposition 2.9, dass es eine Matrix U aus der allgemeinen linearen Gruppe über $\mathbb{Z}$ mit $det(U) = \pm 1$ gibt, wobei

$$F(v_1, v_2, \dots, v_n) = UF(w_1, w_2, \dots, w_n)$$

gilt. Wir rechnen:

$$\begin{aligned} Vol(\mathcal{F}(v_1, v_2, \dots, v_n)) &= |det(F(v_1, v_2, \dots, v_n))| && \text{(Proposition 2.13)} \\ &= |det(UF(w_1, w_2, \dots, w_n))| \\ &= |det(U)|\, |det(F(w_1, w_2, \dots, w_n))| \\ &= |\pm 1|\, |det(F(w_1, w_2, \dots, w_n))| \\ &= |det(F(w_1, w_2, \dots, w_n))| \\ &= Vol(\mathcal{F}(w_1, w_2, \dots, w_n)). && \text{(Proposition 2.13)} \end{aligned}$$

□

2.3 SVP und CVP

In Gittern können wir zwei fundamentale Probleme betrachten. Das Erste besteht darin, einen Gittervektor zu finden, der am nächsten an einem vorgegebenen Vektor liegt, der nicht im Gitter darstellbar ist. Das zweite Problem besteht darin, einen von Null verschiedenen Vektor zu finden, dessen Länge bezüglich einer vorgegebenen Norm unter allen Gittervektoren minimal ist. Wir verwenden im Folgenden die euklidische Norm, die mit $\|\cdot\|$ bezeichnet wird.

Definition 2.15 (SVP)**.** Das *Shortest Vector Problem (SVP)* ist die Suche nach einem nicht trivialen Gittervektor $v \in \mathcal{L}$ wobei $\|v\|$ minimal sein soll.

Definition 2.16 (CVP)**.** Unter dem *Closest Vector Problem (CVP)* wird die Suche bei einem gegebenen Vektor $w \in \mathbb{R}^m \backslash \mathcal{L}$ nach einem Vektor $v \in \mathcal{L}$, für den die Norm $\|w - v\|$ minimal sein soll, verstanden.

Es kann mehrere kürzeste und nächste Vektoren in einem Gitter $\mathcal{L}$ geben. Betrachten wir z. B. $\mathbb{Z}^2$, so sind $(\pm 1, 0)^t$ und $(0, \pm 1)^t$ jeweils kürzeste Vektoren. Zu einem Vektor $(0.5, 0)^t \in \mathbb{R}^2$ gibt es die beiden nächsten Vektoren $(0, 0)^t$ und $(1, 0)^t \in \mathbb{Z}^2$.
CVP und SVP sind grundlegende Probleme im Gitter, die schwerer zu lösen sind, je größer die Dimension des Gitters wird.
Das Shortest Vector Problem gehört zu den NP-vollständigen Problemen ([BNS]). Somit ist es ein Problem zu dem es keinen bekannten Algorithmus gibt, der dieses in polynomialer Zeit immer bestmöglich löst. Weiterhin sind somit alle NP-Probleme auf dieses zurückführbar. Das Closest Vector Problem gehört zu den NP-schweren Problemen. Dies sind Probleme, die nicht in NP liegen, wobei aber jedes NP-Problem auf ein solches in polynomialer Zeit zurückführbar ist.
Möchte man den Klartext rekonstruieren, so ist dieses Problem äquivalent zu einem CVP. Das Clostest Vector Problem erscheint in der Praxis als geringfügig schwieriger als das Shortest Vector Problem, da es oft auf ein SVP in unwesentlich höherer Dimension reduzierbar ist (S. 371 [HPS1]). Im Kapitel 5 (NTRU als Gitter) ist beschrieben, wie sich die Schlüsselrekonstruktion auf ein Shortest Vector Problem zurückführen lässt. Um das NTRU-Kryptosystem anzugreifen ist es schließlich besser das SVP zu lösen anstatt das CVP. Wir werden daher das Closest Vector Problem für einen Angriff auf das NTRU-Verfahren nicht weiter betrachten.

Es gibt einige Varianten des Shortest Vector Problems.

Definition 2.17 (apprSVP)**.** In einem Gitter $\mathcal{L}$ der Dimension n soll ein nicht trivialer Gittervektor gefunden werden, der nicht mehr als das Vielfache einer Funktion, $\psi(n)$, länger ist als ein kürzester vom Nullvektor verschiedener Vektor. Dieses Problem wird mit *approximate Shortest Vector Problem (apprSVP)* bezeichnet. Mit anderen Worten, ist $v_{\text{kürzest}}$ ein kürzester nicht trivialer Gittervektor in $\mathcal{L}$, finde einen von Null verschiedenen Vektor $v \in \mathcal{L}$ mit

$$\|v\| \leq \psi(n) \|v_{\text{kürzest}}\| .$$

Jede Wahl der Funktion $\psi(n)$ ergibt ein anderes apprSVP.

2.4 Die Heuristik von Gauß

Kommen wir nun zu einer Heuristik, mit der wir die Länge eines kürzesten Vektors in einem Gitter $\mathcal{L}$ abschätzen können.
Zunächst wird ein Satz eingeführt, der uns das Volumen eines Balls im $\mathbb{R}^n$ angibt. Dieser benötigt die aus der Analysis bekannte Γ-Funktion. Des weiteren werden wir die Stirlingsche Formel, aus der Analysis, für den Beweis des Satzes benötigen.

Proposition 2.18 (Stirlingsche Formel)**.** *Für große Werte von s gilt*

$$\Gamma(1+s)^{\frac{1}{s}} \approx \frac{s}{e}.$$

$\left(\textit{Genauer: } \ln\Gamma(1+s) = \ln\left(\frac{s}{e}\right)^s + \frac{1}{2}\ln(2\pi s) + \mathcal{O}(1), \textit{ wenn } s \longrightarrow \infty\right)$

Satz 2.19. *Es sei $\mathbb{B}_R(a)$ ein Ball vom Radius R mit Mittelpunkt a im $\mathbb{R}^n$. Das Volumen des Balls $\mathbb{B}_R(a)$ beträgt*

$$Vol(\mathbb{B}_R(a)) = \frac{\pi^{\frac{n}{2}} R^n}{\Gamma(1+\frac{n}{2})}. \tag{3}$$

Für große Werte von n ist das Volumen des Balls $\mathbb{B}_R(a) \subset R^n$ approximativ gegeben durch

$$Vol(\mathbb{B}_R(a))^{\frac{1}{n}} \approx \sqrt{\frac{2\pi e}{n}} R. \tag{4}$$

Beweis. Die Formel (3) ist aus der Analysis bekannt. Der Beweis der Abschätzung (4) kann mit der Stirlingschen Formel und (3) geführt werden:

$$\begin{aligned} Vol(\mathbb{B}_R(a)) = \frac{\pi^{\frac{n}{2}} R^n}{\Gamma(1+\frac{n}{2})} \Leftrightarrow Vol(\mathbb{B}_R(a))^{\frac{1}{n}} &= \frac{\pi^{\frac{1}{2}} R}{\Gamma(1+\frac{n}{2})^{\frac{1}{n}}} = \frac{\pi^{\frac{1}{2}} R}{\left(\Gamma(1+\frac{n}{2})^{\frac{1}{\frac{n}{2}}}\right)^{\frac{1}{2}}} \\ &\approx \frac{\pi^{\frac{1}{2}} R}{\left(\frac{n}{2}\frac{1}{e}\right)^{\frac{1}{2}}} = \sqrt{\frac{2\pi e}{n}} R. \end{aligned}$$

Somit ist also $Vol(\mathbb{B}_R(a))^{\frac{1}{n}} \approx \sqrt{\frac{2\pi e}{n}} R$ bewiesen. □

Wir können nun die Länge eines kürzesten Vektors in einem Gitter $\mathcal{L}$, mit einer großen Dimension n, mit Hilfe der folgenden Überlegung abschätzen.

Es sei $\mathbb{B}_R(0)$ ein großer Ball um den Ursprung. Die Anzahl von Gitterpunkten in diesem Ball kann dann annäherungsweise angegeben werden, indem das Volumen des Balls $\mathbb{B}_R(0)$ durch das Volumen einer Grundmasche $\mathcal{F}$ dividiert wird.

Dies können wir annehmen, da $\#(\mathbb{B}_R(0) \cap \mathcal{L})$ approximativ die Anzahl von Kopien der Grundmasche $\mathcal{F}$ ist, die in den Ball $\mathbb{B}_R(0)$ passen. Bei großen Dimensionen wird die Abschätzung des Fehlers schwierig. Nimmt die Dimension n zu, dann kann der Fehler, der von den Gitterpunkten in der Nähe der Grenze des Balls erzeugt wird, sehr groß werden, bis R riesig wird. Somit ist die Abschätzung

$$\#\{v \in \mathcal{L} \mid \|v\| \leq R\} \approx \frac{Vol(\mathbb{B}_R(0))}{Vol(\mathcal{F})} \tag{5}$$

etwas problematisch, wenn die Dimension n groß ist und der Radius R nicht allzu groß. Wir fragen nun nach dem Wert des Radius R, der die rechte Seite der Abschätzung (5) zu Eins werden lässt. Denn für diesen Wert erwarten wir als erstes einen vom Nullvektor verschiedenen Gitterpunkt in dem Ball.
Nehmen wir an, dass die Dimension n groß ist, dann können wir die Abschätzung (4) aus dem Satz 2.19 benutzen

$$Vol(\mathbb{B}_R(0))^{\frac{1}{n}} \approx \sqrt{\frac{2\pi e}{n}} R \Leftrightarrow Vol(\mathbb{B}_R(0)) \approx \left(\frac{2\pi e}{n}\right)^{\frac{n}{2}} R^n.$$

Um nun einen nicht trivialen Gittervektor zu erhalten, setzen wir nach obiger Annahme $1 = \frac{Vol(\mathbb{B}_R(0))}{Vol(\mathcal{F})}$, somit folgt $Vol(\mathbb{B}_R(0)) = Vol(\mathcal{F})$. Nach der Definition 2.12 ist
$Vol(\mathcal{F}) = det(\mathcal{L})$. Also erhalten wir

$$\left(\frac{2\pi e}{n}\right)^{\frac{n}{2}} R^n \approx Vol(\mathbb{B}_R(0)) = Vol(\mathcal{F}) = det(\mathcal{L}) \Leftrightarrow R \approx \sqrt{\frac{n}{2\pi e}} (det(\mathcal{L}))^{\frac{1}{n}}.$$

Dies führt uns zu der Heuristik von Gauß.

Definition 2.20. [Heuristik von Gauß] Es sei $\mathcal{L}$ ein Gitter der Dimension n. Die nach Gauß erwartete kürzeste Länge eines Vektor beträgt dann

$$\sigma(\mathcal{L}) = \sqrt{\frac{n}{2\pi e}} \left(det(\mathcal{L})\right)^{\frac{1}{n}}.$$

Die Gaußsche Heuristik besagt also, dass für den kürzesten Vektor $v_{\text{kürzest}}$ in einem gewählten Gitter $\mathcal{L}$ die Abschätzung

$$\|v_{\text{kürzest}}\| \approx \sigma(\mathcal{L})$$

gilt.

3 Beschreibung des NTRU-Verfahrens

Es folgt die Beschreibung des asymmetrischen ringbasierenden public key Kryptosystem NTRU, mit Hilfe des Ringes der Konvolutionspolynome.

3.1 Öffentliche Parameter

Vier Parameter werden gewählt und veröffentlicht

$$N, p, q, d \in \mathbb{N}.$$

Diese müssen folgende Bedingungen erfüllen:

(i) $2d + 1 \leq N$,

(ii) $\text{ggT}(p, q) = 1$,

(iii) N ist eine Primzahl und $\text{ggT}(N, q) = 1$,

(iv) $q > (6d + 1)p$.

Die Parameter p und q müssen keine Primzahlen sein und p sei immer wesentlich kleiner als q (siehe [NTRU], genauer http://www.ntru.com/cryptolab/pdf/ANTS97.pdf).

3.2 Schlüsselentwicklung

Wir betrachten den Ring der Konvolutionspolynome $\mathcal{R}^N$ und dessen Reduktionen $\mathcal{R}_q^N$ und $\mathcal{R}_p^N$. Alice rechnet geheim die Schlüssel aus. Sie muss für die Schlüsselentwicklung sowohl den öffentlichen als auch ihren privaten Schlüssel erzeugen. Hierfür benötigt sie zwei zufällige ternäre Polynome

$$f(x) \in \mathcal{T}(d+1, d, N) \qquad \text{und} \qquad g(x) \in \mathcal{T}(d, d, N).$$

Weiter sollen vom Polynom $f(x)$ die Inversen

$$F_q(x) = f(x)^{-1} \quad \text{in} \quad \mathcal{R}_q^N \qquad \text{und} \qquad F_p(x) = f(x)^{-1} \quad \text{in} \quad \mathcal{R}_p^N$$

existieren[1].
Sie kann nun den **öffentlichen Schlüssel** $h(x)$ berechnen:

$$h(x) = F_q(x) \star g(x) \pmod q \qquad \text{in} \qquad \mathcal{R}_q^N.$$

Ihr **privater Schlüssel** ist das Polynom $f(x)$. In der Praxis speichert Alice das Paar $(f(x), F_p(x))$ als ihren privaten Schlüssel, weil sie $F_p(x)$ zur Entschlüsselung benutzen wird.

3.3 Verschlüsselung

Bob möchte die Nachricht $m(x) \in \mathcal{R}_p^N$ an Alice schicken. Hierzu führt er einen zentralen Lift von $m(x) \in \mathcal{R}_p^N$ nach $m'(x) \in \mathcal{R}^N$ durch. Weiter nimmt er ein zufällig gewähltes ternäres Polynom $r(x) \in \mathcal{T}(d, d, N)$ hinzu, das auch kurzlebiger Schlüssel genannt wird. Zur Erstellung des verschlüsselten Textes benutzt er des weiteren den öffentlichen Schlüssel $h(x)$ und berechnet:

$$e(x) \equiv p \cdot h(x) \star r(x) + m'(x) \pmod{\text{q}}.$$

Die Nachricht $e(x)$ liegt im Polynomring $\mathcal{R}_q^N$.

3.4 Entschlüsselung

Für die Entschlüsselung von Bobs gesendeter Nachricht $e(x)$ berechnet Alice mit ihrem privaten Schlüssel $f(x)$ als erstes:

$$a(x) \equiv f(x) \star e(x) \pmod q.$$

Dann führt sie einen *zentralen Lift* von $a(x)$ nach $a'(x) \in \mathcal{R}^N$ durch. Zum Schluss berechnet Alice:

$$b(x) \equiv F_p(x) \star a'(x) \pmod p.$$

Das Polynom $b(x) \in \mathcal{R}_p^N$ entspricht der Nachricht $m(x)$ von Bob; dies werden wir im nächsten Abschnitt 3.5 nachweisen.

[1]Sind die Parameter p und q Primzahlen, dann kann mit Lemma 2.6 überprüft werden, ob die Inversen $F_q(x)$ und $F_p(x)$ existieren und mit Hilfe des euklidischen Algorithmus können diese berechnet werden. Wenn ein Inverses nicht existiert, muss Alice ein anderes ternäres Polynom $f(x) \in \mathcal{T}(d+1, d, N)$ auswählen.

3.5 Korrektheit

Um zu zeigen, wie das beschriebene Verfahren mit der Einschränkung $q > (6d+1)p$ funktioniert, also Alices Polynom $b(x)$ Bobs Nachricht $m(x)$ rekonstruiert, betrachten wir zunächst Alices Berechnungen genauer:

$$\begin{aligned} a(x) &\equiv f(x) \star e(x) \pmod q \\ &\equiv f(x) \star (p \cdot h(x) \star r(x) + m'(x)) \pmod q \\ &\equiv f(x) \star (p \cdot F_q(x) \star g(x) \star r(x) + m'(x)) \pmod q \\ &\equiv p \cdot \underbrace{f(x) \star F_q(x)}_{\equiv 1 \pmod q} \star g(x) \star r(x) + f(x) \star m'(x) \pmod q \\ &\equiv p \cdot g(x) \star r(x) + f(x) \star m'(x) \pmod q. \end{aligned}$$

Das Polynom

$$p \cdot g(x) \star r(x) + f(x) \star m'(x) \in \mathcal{R}^N$$

wird jetzt ohne modulo q Rechnung betrachtet. Wir überlegen uns hierbei, welche Werte der größte Exponent dieses Polynoms annehmen kann. Das Polynom besteht aus zwei Summanden. Beim ersten Summand $p \cdot g(x) \star r(x)$ liegen $g(x)$ und $r(x)$ in $\mathcal{T}(d, d, N)$. Wenn bei dem Konvolutionsprodukt $g(x) \star r(x)$ alle d Einsen so aufeinander treffen, dass sie jeweils die Koeffizienten von einem bestimmten x^i werden und auch alle d Minuseinsen so aufeinander treffen, dass sie miteinander multipliziert auch die Koeffizienten von diesem x^i werden, dann ist der größte Koeffizient den wir erhalten können $2d$. Insgesamt kann der erste Summand höchstens einen Koeffizienten der Größe $p \cdot 2d$ enthalten.
Beim zweiten Summanden gelangen wir mit derselben Überlegung dazu, dass ein Koeffizient höchstens den Wert $(2d+1) \cdot \frac{1}{2}p$ annehmen kann. Denn $f(x) \in \mathcal{T}(d+1, d, N)$ und es gilt $-\frac{1}{2}p < m'_i \leq \frac{1}{2}p$ für alle Koeffizienten von $m'(x)$.
Der größte Koeffizient der für das Polynoms $p \cdot g(x) \star r(x) + f(x) \star m'(x)$ möglich ist, tritt genau dann auf, wenn der größte Koeffizient des ersten Summanden mit dem größten Koeffizient des zweiten Summanden addiert wird. Somit kann das Polynom insgesamt höchstens einen Koeffizienten der Größe

$$p \cdot (2d) + (2d+1) \cdot \frac{1}{2}p = \left(2d + d + \frac{1}{2}\right) p = \left(3d + \frac{1}{2}\right) p$$

besitzen. Da nach Voraussetzung $q > (6d+1)\, p$ gilt, folgt:

$$q > (6d+1)\, p \;\Leftrightarrow\; \frac{1}{2}q > \frac{1}{2}\,(6d+1)\, p \;\Leftrightarrow\; \frac{1}{2}q > \left(3d + \frac{1}{2}\right) p.$$

Die Koeffizienten des Polynoms $p \cdot g(x) \star r(x) + f(x) \star m'(x)$ sind also alle echt kleiner als $\frac{1}{2}q$. Wenn Alice schließlich bei ihrer Entschlüsselung $a(x)$ modulo q berechnet und nach $\mathcal{R}^N$ zentral liftet, rekonstruiert sie den genauen Wert $p \cdot g(x) \star r(x) + f(x) \star m'(x)$, d.h sie berechnet

$$a'(x) = p \cdot g(x) \star r(x) + f(x) \star m'(x)$$

exakt in $\mathcal{R}^N$. Zum Schluss berechnet Alice

$$\begin{aligned} b(x) &\equiv F_p \star a'(x) (\mathrm{mod}\, p) \\ &\equiv F_p \star (\underbrace{p \cdot g(x) \star r(x)}_{\equiv 0 \,(\mathrm{mod}\, p)} + f(x) \star m'(x)) (\mathrm{mod}\, p) \\ &\equiv \underbrace{F_p \star f(x)}_{\equiv 1 \,(\mathrm{mod}\, p)} \star m'(x) \,(\mathrm{mod}\, p) \\ &\equiv m'(x) \,(\mathrm{mod}\, p). \end{aligned}$$

Bobs Nachricht $m(x)$ liegt in $\mathcal{R}_p^N$ und $m'(x)$ ist der zentrale Lift von $m(x)$. Hieraus folgt, dass $m(x) \equiv m'(x) \pmod p$ ist. Somit wissen wir, dass Alice mit $b(x)$ Bobs Nachricht $m(x)$ rekonstruiert hat.

Bemerkungen 3.1. *Die Abschätzung $q > (6d+1)p$ garantiert, dass die Entschlüsselung funktioniert. Es ist allerdings sehr unwahrscheinlich, dass die Einsen und Minuseinsen bei den Konvolutionsprodukten $g(x) \star r(x)$ und $f(x) \star m(x)$, wie oben beschrieben, aufeinander treffen und sich so alles zu dem höchsten Koeffizienten addiert. Die Entschlüsselung wird also auch für kleinere Werte von q gelingen, allerdings kann dann die Wahrscheinlichkeit, dass die Entschlüsselung nicht funktioniert nur schwer abgeschätzt werden. Es ist daher sehr wichtig, dass die Wahrscheinlichkeit, für fehlerhaftes Entschlüsseln sehr klein ist, z. B. kleiner als 2^{-80}.*

Bemerkungen 3.2. *Bob kann einen kurzlebigen Schlüssel $r(x) \in \mathcal{T}(d, d, N)$ wählen, um die Sicherheit des NTRU-Kryptosystem zu vergrößern. In Alices letztem Entschlüsselungsschritt fällt das ternäre Polynom $r(x)$, als ein Vielfaches von p, bei der Berechnung modulo p, weg. Es gibt viele verschiedene Verschlüsselungen $ph(x) \star r(x) + m(x)$, je nachdem, welcher kurzlebige Schlüssel $r(x)$ gewählt wurde. Wird der kurzlebige Schlüssel allerdings für zwei verschiedene Verschlüsselungstexte benutzt, oder ein Verschlüsselungstext mit zwei verschiedenen kurzlebigen Schlüsseln verschickt, so gibt es Möglichkeiten für Eve Informationen über den Klartext zu erlangen.*

3.6 Anmerkungen zu den öffentlichen Parametern

Zur Erinnerung wiederholen wir die Bedingungen der öffentlichen Parameter N, p, q und d:

(i) $2d + 1 \leq N$,

(ii) $\mathrm{ggT}(p, q) = 1$,

(iii) N ist eine Primzahl und $\mathrm{ggT}(N, q) = 1$,

(iv) $q > (6d + 1)p$.

Zu diesen Bedingungen sei jeweils folgendes angemerkt:

(i) Der private Schlüssel $f(x)$ kommt aus der Menge der ternären Polynome $\mathcal{T}(d + 1, d, N)$, die aus dem Ring der Konvolutionspolynome $\mathcal{R}^N$ mit N Koeffizienten kommen. Daher muss $2d + 1 \leq N$ gelten.

(ii) Sind p und q nicht teilerfremd, so kann Eve an die geheime Nachricht gelangen. Es könnten z. B. die Fälle

- $p|q$ oder
- $p = q$

auftreten. Beginnen wir mit dem ersten Fall:

- Gilt $p|q$ dann existiert ein $s \in \mathbb{N}$, so dass $s \cdot p = q$ gilt. Im Abschnitt 3.3 (Verschlüsselung) werden wir sehen, dass der Klartext wie folgt verschlüsselt wird

$$e(x) \equiv p \cdot h(x) \star r(x) + m'(x) \pmod{\text{q}}.$$

Somit ergibt sich:

$$\begin{aligned} e(x) &\equiv p \cdot h(x) \star r(x) + m'(x) \pmod{\text{q}} \\ &\Rightarrow s \cdot e(x) \equiv \underbrace{\underbrace{s \cdot p}_{=q} \cdot h(x) \star r(x)}_{\equiv 0 \bmod q} + s \cdot m'(x) \pmod{\text{q}} \\ &\Rightarrow s \cdot e(x) \equiv s \cdot m'(x) \pmod{\text{q}} \\ &\Rightarrow e(x) \equiv m'(x) \pmod{\frac{q}{(\text{ggT}(q,s))}} \\ &\Rightarrow e(x) \equiv m'(x) \pmod{p} \quad \text{da ggT } (q,s) = \text{ggT}(p \cdot s, s) = s \text{ und } \frac{q}{s} = p. \end{aligned}$$

Weiter ist $m(x) \equiv m'(x) \pmod{p}$. Wir haben also schließlich gezeigt, dass wenn der verschlüsselte Text $e(x)$ modulo q gerechnet wird, dann entspricht er dem Klartext $m(x)$.

- Für den Fall $q = p$ folgt

$$e(x) \equiv \underbrace{p \cdot h(x) \star r(x)}_{\equiv 0 \bmod p} + m'(x) \pmod{\text{p}}.$$

Mit $m(x) \equiv m'(x) \pmod{p}$ ergibt sich, dass der verschlüsselte Text $e(x)$ dem Klartext $m(x)$ entspricht, wenn Eve $e(x) \pmod{q}$ berechnet.

(iii) Diese Bedingungen sind nicht notwendig, damit das NTRU-Kryptosystem funktioniert. Sind die Bedingungen allerdings nicht erfüllt, so wird das Verfahren unsicher. Es ist ein Angriff über Gitter möglich ([HPS1], S. 392 bzw. S. 429).

(iv) Diese Ungleichung haben wir für den Nachweis der Korrektheit (Abschnitt 3.5) benötigt.

3.7 Zeitkomplexität

Wir betrachten die Geschwindigkeit des NTRU-Kryptosystems. Die meiste Zeit benötigt die Verschlüsselung und Entschlüsselung. Das Konvolutionsprodukt von zwei Polynomen $a(x) \star b(x)$ aus der Menge $\mathcal{R}^N$ braucht im Allgemeinen N^2 Multiplikationen, denn jeder

Koeffizient dieses Produktes ist genau betrachtet das Skalarprodukt von zwei Vektoren (siehe Bemerkung 2.4). Die Konvolutionsprodukte, die beim NTRU-Verfahren auftreten, sind

$$r(x) \star h(x), \qquad f(x) \star e(x), \qquad F_p(x) \star a'(x),$$

mit $r(x)$, $f(x)$ und $F_p(x)$ aus der Menge der ternären Polynome. Bei diesen Konvolutionsprodukten ist also jeweils ein Faktor aus der Menge der ternären Polynome. Somit führen wir bei den Konvolutionsprodukten keine Multiplikation aus, sondern ändern die Vorzeichen bei den Koeffizienten des anderen Faktors, lassen diese gleich, oder vernachlässigen die Koeffizienten. Somit führen wir effektiv nur Subtraktionen oder Additionen durch. Jedes Produkt benötigt insgesamt ungefähr $\frac{2}{3}N^2$ Additionen und Subtraktionen. Ist das d kleiner als $\frac{N}{3}$, dann brauchen die ersten beiden Konvolutionsprodukte nur $\frac{2}{3}dN$ Additionen und Multiplikationen. Die Ver- und Entschlüsselung vom NTRU-Verfahren benötigt also $\mathcal{O}(N^2)$ Schritte, bei dem jeder Schritt extrem schnell ist.
Um k bits für die Sicherheit zu archivieren, benötigt die Ver- und Entschlüsselung vom ElGamal-Verfahren und RSA-Verfahren $\mathcal{O}(k^3)$ Operationen, wobei Ver- und Entschlüsselung des gitterbasierenden Kryptosystemen nur $\mathcal{O}(k^2)$ Operationen benötigt.

3.8 Beispiel

Als Beispiel verschlüsseln wir den Klartext

„BACHELORARBEIT VON ANJA MOLDENHAUER“[2].

Zunächst legen wir die **öffentlichen Parameter**

$$N = 11, \qquad q = 41, \qquad p = 3, \qquad d = 2$$

fest. Diese erfüllen die Bedingungen:

$$2 \cdot 2 + 1 \leq 11,\ ggT(3, 41) = 1,\ 11 \in \mathbb{P},\ ggT(11, 41) = 1 \text{ und } 41 > (6 \cdot 2 + 1) \cdot 3 = 39.$$

Damit der Text verschlüsselt werden kann entwickeln wir die **Schlüssel**. Hierfür benötigen wir zwei zufällige ternäre Polynome:

(i) $f(x) \in \mathcal{T}(3, 2, 11)$: es sei $f(x) = x^7 - x^6 - x^4 + x^3 + x$,

(ii) $g(x) \in \mathcal{T}(2, 2, 11)$: es sei $g(x) = -x^9 + x^7 - x^2 + x$.

Mit dem erweiterten euklidischen Algorithmus erhalten wir die Inversen zu $f(x)$:

$$\begin{aligned} F_{41}(x) &= f^{-1}(x) \in \mathcal{R}_{41}^{11} \\ &= 2x^{10} + 29x^9 + 5x^8 + 4x^7 + 16x^6 + 10x^5 + 8x^4 + 32x^3 + 21x^2 + 15x + 23, \\ F_3(x) &= f^{-1}(x) \in \mathcal{R}_3^{11} \\ &= x^{10} + x^9 + 2x^8 + 2x^7 + x^6 + x^5 + x^4 + 2x^3 + x + 1, \end{aligned}$$

[2]Die Rechenschritte wurden mit Maple Classic Worksheet 9.5 durchgeführt. Im Anhang ist der Quellcode mit Erläuterungen zu finden.

mit $F_{41}(x) \star f(x) \equiv 1 \pmod{41}$ und $F_3(x) \star f(x) \equiv 1 \pmod{3}$.
Jetzt können wir den **öffentlichen Schlüssel** $h(x)$ berechnen

$$h(x) = F_{41}(x) \star g(x) = 40x^9 + x^8 + 40x^6 + x^5 + 40x + 1 \in \mathcal{R}_{41}^{11}.$$

Unser **privater Schlüssel** ist das Tupel $(f(x), F_3(x))$.

Um den Text „BACHELORARBEIT VON ANJA MOLDENHAUER“ zu verschlüsseln wandeln wir diesen in Polynome aus R^{11} um, mit ganzzahligen Koeffizienten a_i , wobei $-\frac{3}{2} < a_i \leq \frac{3}{2}$ gilt. Als erstes werden wir den Text mit Hilfe der Tabelle 1 codieren und in eine Zeichenfolge umwandeln. Diese wird dann in Vektoren aus dem $\mathbb{Z}^{11}$ zerlegt, indem jeweils elf aufeinander folgende Zeichen als ein Vektor aufgefasst werden. Um an die Polynome zu gelangen, werden die Einträge dieser Vektoren als dessen Koeffizienten geschrieben (siehe Bemerkung 2.2(*iii*)).

Blank	*A*	*B*	*C*	*D*	*E*	*F*
$-1,-1,-1$	$-1,-1,0$	$-1,-1,1$	$-1,0,-1$	$-1,0,0$	$-1,0,1$	$-1,1,-1$
G	*H*	*I*	*J*	*K*	*L*	*M*
$-1,1,0$	$-1,1,1$	$0,-1,-1$	$0,-1,0$	$0,-1,1$	$0,0,-1$	$0,0,0$
N	*O*	*P*	*Q*	*R*	*S*	*T*
$0,0,1$	$0,1,-1$	$0,1,0$	$0,1,1$	$1,-1,-1$	$1,-1,0$	$1,-1,1$
U	*V*	*W*	*X*	*Y*	*Z*	–
$1,0,-1$	$1,0,0$	$1,0,1$	$1,1,-1$	$1,1,0$	$1,1,1$	–

Tabelle 1: Öffentlicher Verschlüsselungscode 1 [3]

Um dieses Beispiel ausführen zu können benötigen wir erst einige Vorüberlegungen.
Unser Ausgangstext hat 35 Zeichen, deshalb wird eine Zeichenfolge der Länge $3 \cdot 35 = 105$ entstehen. Die Vektoren sind aus $\mathbb{Z}^{11}$, somit muss die Länge der Zeichenfolge durch 11 teilbar sein. Folglich füllen wir unseren codierten Text solange mit Minuseinsen auf, bis er eine Länge bekommt, die durch 11 teilbar ist. In diesem Fall bis zu 110 Zeichen. Wir wählen die Minusein als Auffüller, da in der Tabelle 1 das Blank mit lauter Minuseinsen erzeugt wird.
Mit Hilfe von Tabelle 1 verwandeln wir den Text „BACHELORARBEIT VON ANJA MOLDENHAUER“ in die Zeichenkette

-1,-1,1,-1,-1,0,-1,0,-1,-1,1,1,-1,0,1,0,0,-1,0,1,-1,1,-1,-1,-1,-1,0,1,-1,-1,-1,-1,1,-1,0,1,0,-1,-1,1,
-1,1,-1,-1,-1,1,0,0,0,1,-1,0,0,1,-1,-1,-1,-1,-1,0,0,0,1,0,-1,0,-1,-1,0,-1,-1,-1,0,0,0,0,1,-1,0,0,-1,
-1,0,0,-1,0,1,0,0,1,-1,1,1,-1,-1,0,1,0,-1,-1,0,1,1,-1,-1,-1,-1,-1,-1,-1.

[3]Die Tabelle 1 wurde aus der Vorlesung [Ru] übernommen.

Diese wird in Vektoren $m'_i \in \mathbb{Z}^{11}$ zerlegt und mit ihren Polynomen $m'_i(x) \in \mathcal{R}^{11}$ identifiziert:

$$\begin{aligned}
m'_1 &= (-1,-1,1,-1,-1,0,-1,0,-1,-1,1)\\
&\hat{=} m'_1(x) = x^{10} - x^9 - x^8 - x^6 - x^4 - x^3 + x^2 - x - 1,\\
m'_2 &= (1,-1,0,1,0,0,-1,0,1,-1,1)\\
&\hat{=} m'_2(x) = x^{10} - x^9 + x^8 - x^6 + x^3 - x + 1,\\
m'_3 &= (-1,-1,-1,-1,0,1,-1,-1,-1,-1,1)\\
&\hat{=} m'_3(x) = x^{10} - x^9 - x^8 - x^7 - x^6 + x^5 - x^3 - x^2 - x - 1,\\
m'_4 &= (-1,0,1,0,-1,-1,1,-1,1,-1,-1)\\
&\hat{=} m'_4(x) = -x^{10} - x^9 + x^8 - x^7 + x^6 - x^5 - x^4 + x^2 - 1,\\
m'_5 &= (-1,1,0,0,0,1,-1,0,0,1,-1)\\
&\hat{=} m'_5(x) = -x^{10} + x^9 - x^6 + x^5 + x - 1,\\
m'_6 &= (-1,-1,-1,-1,0,0,0,1,0,-1,0)\\
&\hat{=} m'_6(x) = -x^9 + x^7 - x^3 - x^2 - x - 1,\\
m'_7 &= (-1,-1,0,-1,-1,-1,0,0,0,0,1)\\
&\hat{=} m'_7(x) = x^{10} - x^5 - x^4 - x^3 - x - 1,\\
m'_8 &= (-1,0,0,-1,-1,0,0,-1,0,1,0)\\
&\hat{=} m'_8(x) = x^9 - x^7 - x^4 - x^3 - 1,\\
m'_9 &= (0,1,-1,1,1,-1,-1,0,1,0,-1)\\
&\hat{=} m'_9(x) = -x^{10} + x^8 - x^6 - x^5 + x^4 + x^3 - x^2 + x,\\
m'_{10} &= (-1,0,1,1,-1,-1,-1,-1,-1,-1,-1)\\
&\hat{=} m'_{10}(x) = -x^{10} - x^9 - x^8 - x^7 - x^6 - x^5 - x^4 + x^3 + x^2 - 1.
\end{aligned}$$

Wir wollen mehrere verschiedene Nachrichten $m_1, \cdots, m_{10}$ verschicken. Wegen der Bemerkung 3.2 wählen wir verschiedene ternäre Polynome als kurzlebige Schlüssel $r_i(x) \in \mathcal{T}(2,2,11)$. Hier benutzen wir:

$$\begin{aligned}
r_1(x) &= x^{10} + x^8 - x^6 - x^4,\\
r_2(x) &= x^9 - x^7 - x^4 + x^3,\\
r_3(x) &= -x^9 + x^5 + x^3 - 1,\\
r_4(x) &= -x^8 + x^3 - x^2 + x,\\
r_5(x) &= x^7 - x^5 + x^4 - x^3,\\
r_6(x) &= x^6 + x^5 - x^3 - x,\\
r_7(x) &= -x^5 + x^3 - x + 1,\\
r_8(x) &= -x^4 - x^3 + x^2 + 1,\\
r_9(x) &= x^4 + x^2 - x - 1,\\
r_{10}(x) &= x^3 - x^2 - x + 1.
\end{aligned}$$

Für die **Verschlüsselungen** $e_i(x) \in \mathcal{R}^{11}_{41}$ berechnen wir

$$e_i(x) \equiv 3 \cdot h(x) \star r_i(x) + m'_i(x) \pmod{41} \qquad \text{für } \; i = 1, \ldots, 10.$$

Für die verschlüsselten Polynome erhalten wir:

$$
\begin{aligned}
e_1(x) &= 7x^{10} + 34x^9 + 40x^8 + 6x^7 + 34x^6 + 3x^5 + 2x^4 + 34x^3 + 7x^2 + 40x + 34,\\
e_2(x) &= x^{10} + 37x^9 + 7x^8 + 35x^7 + 2x^6 + 6x^5 + 29x^4 + 7x^3 + 6x^2 + 31x + 4,\\
e_3(x) &= 7x^{10} + 37x^9 + 40x^8 + 2x^7 + 37x^6 + x^5 + 37x^3 + 2x^2 + 40x + 37,\\
e_4(x) &= 34x^{10} + 2x^9 + 4x^8 + 34x^7 + 7x^6 + 37x^5 + 37x^4 + 9x^3 + 33x^2 + 5,\\
e_5(x) &= 34x^{10} + 7x^9 + 35x^8 + 3x^7 + 2x^6 + 33x^5 + 9x^4 + 32x^2 + 10x + 40,\\
e_6(x) &= 6x^{10} + 40x^9 + 38x^8 + x^7 + 38x^6 + 3x^5 + 37x^3 + 5x^2 + 37x + 37,\\
e_7(x) &= x^{10} + 32x^9 + 6x^8 + 3x^7 + 38x^6 + 40x^5 + 37x^4 + 5x^3 + 31x + 8,\\
e_8(x) &= 6x^{10} + 39x^9 + 38x^8 + 2x^7 + 38x^6 + 6x^5 + 40x^4 + 34x^3 + 6x^2 + 38x + 37,\\
e_9(x) &= 2x^{10} + 3x^9 + 36x^8 + 6x^7 + 40x^6 + 34x^5 + 4x^4 + 39x^3 + 2x^2 + 4x + 35,\\
e_{10}(x) &= 40x^{10} + 31x^9 + 8x^8 + 40x^7 + 34x^6 + 2x^5 + 37x^4 + 7x^3 + x^2 + 32x + 8.
\end{aligned}
$$

Wir wählen einen zweiten öffentlichen Verschlüsselungscode aus.

0	1	2	3	4	5	6	7	8	9	10	11	12	13
✻	✼	✳	✡	❋	✵	✸	✴	❀	★	✽	✯	☆	✭
14	15	16	17	18	19	20	21	22	23	24	25	26	27
✯	✢	✾	✧	✤	❃	❀	✸	✵	❀	✶	✺	❄	❄
28	29	30	31	32	33	34	35	36	37	38	39	40	–
❁	✲	✼	✹	❁	✪	✾	☆	❀	✦	✲	✱	✷	–

Tabelle 2: Öffentlicher Verschlüsselungscode 2

Identifizieren wir die Koeffizienten der Polynome $e_i(x)$ wieder, wie in Bemerkung 2.2 (iii), mit ihren Vektoren, so erhalten wir die Vektoren $e_i \in \mathbb{Z}^{11}$:

$$
\begin{aligned}
e_1 &= (34, 40, 7, 34, 2, 3, 34, 6, 40, 34, 7),\\
e_2 &= (4, 31, 6, 7, 29, 6, 2, 35, 7, 37, 1),\\
e_3 &= (37, 40, 2, 37, 0, 1, 37, 2, 40, 37, 7),\\
e_4 &= (5, 0, 33, 9, 37, 37, 7, 34, 4, 2, 34),\\
e_5 &= (40, 10, 32, 0, 9, 33, 2, 3, 35, 7, 34),\\
e_6 &= (37, 37, 5, 37, 0, 3, 38, 1, 38, 40, 6),\\
e_7 &= (8, 31, 0, 5, 37, 40, 38, 3, 6, 32, 1),\\
e_8 &= (37, 38, 6, 34, 40, 6, 38, 2, 38, 39, 6),\\
e_9 &= (35, 4, 2, 39, 4, 34, 40, 6, 36, 3, 2),\\
e_{10} &= (8, 32, 1, 7, 37, 2, 34, 40, 8, 31, 40).
\end{aligned}
$$

Werden die Einträge der Vektoren e_i hintereinander aufgeschrieben und codiert, so ergibt sich mit Hilfe der Tabelle 2 der **verschlüsselte Text**:

Hierbei wird jedem Koeffizienten ein Symbol zugeordnet. Der verschlüsselte Text hat einen Bestand von 110 Zeichen.
Um diesen Text zu **entschlüsseln**, wird er wieder in die Polynome $e_i(x) \in \mathcal{R}_{41}^{11}$ umgewandelt und dann berechnen wir mit dem geheimen privaten Schlüssel $(f(x), F_3(x))$ den Klartext. Als erstes wird

$$a_i(x) \equiv f(x) \star e_i(x) \pmod{41} \in \mathcal{R}_{41}^{11} \qquad \text{für } \; i = 1, \ldots, 10$$

berechnet. Bei diesem Beispiel erhalten wir:

$$\begin{aligned}
a_1(x) &= 38x^{10} + 3x^9 + 40x^8 + 37x^7 + 3x^6 + 37x^5 + 5x^4 + 38x + 40,\\
a_2(x) &= 7x^{10} + 2x^9 + 37x^8 + 38x^7 + 4x^6 + 40x^5 + 4x^4 + 37x^3 + 3x^2 + 38x + 37,\\
a_3(x) &= 39x^{10} + 38x^7 + 6x^6 + 33x^5 + 2x^4 + 35x^3 + 4x^2 + 39x + 3,\\
a_4(x) &= x^{10} + 39x^9 + x^8 + 40x^7 + x^6 + 38x^5 + 6x^4 + 34x^3 + 5x^2 + 33x + 4,\\
a_5(x) &= 40x^{10} + 36x^9 + 5x^8 + 36x^6 + 4x^5 + 39x^4 + 6x^3 + 36x^2 + 2x + 1,\\
a_6(x) &= 40x^{10} + 36x^8 + x^7 + 4x^6 + 2x^5 + 35x^4 + 40x^3 + 40x^2 + 4x + 40,\\
a_7(x) &= 6x^{10} + 40x^9 + 37x^8 + 6x^7 + 38x^6 + 37x^5 + 2x^4 + 4x^3 + 35x^2 + 36x + 1,\\
a_8(x) &= 38x^{10} + x^9 + 2x^7 + 3x^6 + 34x^4 + x^3 + 6x + 35,\\
a_9(x) &= 4x^{10} + 5x^9 + 38x^8 + 36x^7 + 38x^6 + 3x^5 + 5x^3 + 37x^2 + 37x + 2,\\
a_{10}(x) &= 7x^{10} + 34x^9 + 36x^8 + 40x^7 + 40x^6 + 38x^5 + 8x^4 + x^3 + 35x^2 + 40x + 2.
\end{aligned}$$

Nun führen wir einen zentralen Lift von $a_i(x) \in \mathcal{R}_{41}^{11}$ nach $a_i'(x) \in \mathcal{R}^{11}$ durch. Es ergibt sich:

$$\begin{aligned}
a_1'(x) &= -3x^{10} + 3x^9 - x^8 - 4x^7 + 3x^6 - 4x^5 + 5x^4 - 3x - 1,\\
a_2'(x) &= 7x^{10} + 2x^9 - 4x^8 - 3x^7 + 4x^6 - x^5 + 4x^4 - 4x^3 + 3x^2 - 3x - 4,\\
a_3'(x) &= -2x^{10} - 3x^7 + 6x^6 - 8x^5 + 2x^4 - 6x^3 + 4x^2 - 2x + 3,\\
a_4'(x) &= x^{10} - 2x^9 + x^8 - x^7 + x^6 - 3x^5 + 6x^4 - 7x^3 + 5x^2 - 8x + 4,\\
a_5'(x) &= -x^{10} - 5x^9 + 5x^8 - 5x^6 + 4x^5 - 2x^4 + 6x^3 - 5x^2 + 2x + 1,\\
a_6'(x) &= -x^{10} - 5x^8 + x^7 + 4x^6 + 2x^5 - 6x^4 - x^3 - x^2 + 4x - 1,\\
a_7'(x) &= 6x^{10} - x^9 - 4x^8 + 6x^7 - 3x^6 - 4x^5 + 2x^4 + 4x^3 - 6x^2 - 5x + 1,\\
a_8'(x) &= -3x^{10} + x^9 + 2x^7 + 3x^6 - 7x^4 + x^3 + 6x - 6,\\
a_9'(x) &= 4x^{10} + 5x^9 - 3x^8 - 5x^7 - 3x^6 + 3x^5 + 5x^3 - 4x^2 - 4x + 2,\\
a_{10}'(x) &= 7x^{10} - 7x^9 - 5x^8 - x^7 - x^6 - 3x^5 + 8x^4 + x^3 - 6x^2 - x + 2.
\end{aligned}$$

Zum Schluss, um an den Klartext zu gelangen, berechnen wir

$$b_i(x) \equiv F_3(x) \star a_i'(x) \pmod{3} \qquad \text{für } \; i = 1, \ldots, 10.$$

Somit erhalten wir:

$$
\begin{aligned}
b_1 &= x^{10} + 2x^9 + 2x^8 + 2x^6 + 2x^4 + 2x^3 + x^2 + 2x + 2,\\
b_2 &= x^{10} + 2x^9 + x^8 + 2x^6 + x^3 + 2x + 1,\\
b_3 &= x^{10} + 2x^9 + 2x^8 + 2x^7 + 2x^6 + x^5 + 2x^3 + 2x^2 + 2x + 2,\\
b_4 &= 2x^{10} + 2x^9 + x^8 + 2x^7 + x^6 + 2x^5 + 2x^4 + x^2 + 2,\\
b_5 &= 2x^{10} + x^9 + 2x^6 + x^5 + x + 2,\\
b_6 &= 2x^9 + x^7 + 2x^3 + 2x^2 + 2x + 2,\\
b_7 &= x^{10} + 2x^5 + 2x^4 + 2x^3 + 2x + 2,\\
b_8 &= x^9 + 2x^7 + 2x^4 + 2x^3 + 2,\\
b_9 &= 2x^{10} + x^8 + 2x^6 + 2x^5 + x^4 + x^3 + 2x^2 + x,\\
b_{10} &= 2x^{10} + 2x^9 + 2x^8 + 2x^7 + 2x^6 + 2x^5 + 2x^4 + x^3 + x^2 + 2.
\end{aligned}
$$

Nun führen wir einen zentraler Lift von $b_i(x) \in \mathcal{R}_3^{11}$ nach $m_i'(x)$ $\mathcal{R}^{11}$ durch und erhalten wieder:

$$
\begin{aligned}
m_1'(x) &= x^{10} - x^9 - x^8 - x^6 - x^4 - x^3 + x^2 - x - 1\\
&\hat{=} m_1' = (-1,-1,1,-1,-1,0,-1,0,-1,-1,1),\\
m_2'(x) &= x^{10} - x^9 + x^8 - x^6 + x^3 - x + 1\\
&\hat{=} m_2' = (1,-1,0,1,0,0,-1,0,1,-1,1),\\
m_3'(x) &= x^{10} - x^9 - x^8 - x^7 - x^6 + x^5 - x^3 - x^2 - x - 1\\
&\hat{=} m_3' = (-1,-1,-1,-1,0,1,-1,-1,-1,-1,1),\\
m_4'(x) &= -x^{10} - x^9 + x^8 - x^7 + x^6 - x^5 - x^4 + x^2 - 1\\
&\hat{=} m_4' = (-1,0,1,0,-1,-1,1,-1,1,-1,-1),\\
m_5'(x) &= -x^{10} + x^9 - x^6 + x^5 + x - 1\\
&\hat{=} m_5' = (-1,1,0,0,0,1,-1,0,0,1,-1),\\
m_6'(x) &= -x^9 + x^7 - x^3 - x^2 - x - 1\\
&\hat{=} m_6' = (-1,-1,-1,-1,0,0,0,1,0,-1,0),\\
m_7'(x) &= x^{10} - x^5 - x^4 - x^3 - x - 1\\
&\hat{=} m_7' = (-1,-1,0,-1,-1,-1,0,0,0,0,1),\\
m_8'(x) &= x^9 - x^7 - x^4 - x^3 - 1\\
&\hat{=} m_8' = (-1,0,0,-1,-1,0,0,-1,0,1,0),\\
m_9'(x) &= -x^{10} + x^8 - x^6 - x^5 + x^4 + x^3 - x^2 + x\\
&\hat{=} m_9' = (0,1,-1,1,1,-1,-1,0,1,0,-1),\\
m_{10}'(x) &= -x^{10} - x^9 - x^8 - x^7 - x^6 - x^5 - x^4 + x^3 + x^2 - 1\\
&\hat{=} m_{10}' = (-1,0,1,1,-1,-1,-1,-1,-1,-1,-1).
\end{aligned}
$$

Schreiben wir die Vektoren hintereinander auf, so ergibt sich wieder die Zeichenfolge

-1,-1,1,-1,-1,0,-1,0,-1,-1,1,1,-1,0,1,0,0,-1,0,1,-1,1,-1,-1,-1,-1,0,1,-1,-1,-1,-1,1,-1,0,1,0,-1,-1,1,
-1,1,-1,-1,-1,1,0,0,0,1,-1,0,0,1,-1,-1,-1,-1,-1,0,0,0,1,0,-1,0,-1,-1,0,-1,-1,-1,0,0,0,0,1,-1,0,0,-1,
-1,0,0,-1,0,1,0,0,1,-1,1,1,-1,-1,0,1,0,-1,-1,0,1,1,-1,-1,-1,-1,-1,-1,-1.

Diese zerlegen wir in Tripel. Hierbei werden zwei Minuseinsen am Ende übrig bleiben. Dies waren Platzhalter und sie werden bei der Rückcodierung mit keinem Buchstaben versehen. Somit erhalten wir den Klartext „BACHELORARBEIT VON ANJA MOLDENHAUER“ zurück.

3.9 Sicherheitslevels

In der folgenden Tabelle werden Vorschläge für NTRU-Parameter für verschiedene Sicherheitslevels angegeben (siehe [NTRU] unter „The NTRU Public Key Cryptosystem“).

Sicherheit	N	q	p
Moderat	167	128	3
Standard	251	128	3
Hoch	347	128	3
Höchste	503	256	3

Tabelle 3: Sicherheitslevels

4 Zugrunde liegendes Problem

In diesem Kapitel betrachten wir das Problem, das entsteht, wenn jemand das NTRU-Kryptosystem im Ring der Konvolutionspolynome angreifen möchte. Versucht der Angreifer die ternären Polynome von Alice zu finden, die sie für die Schlüsselberechnung benutzt hat, so gelangt er zu dem NTRU-Schlüsselrekonstruktionsproblem. Bei der „brute-force“ Suche betrachten wir die Anzahl der ternären Polynome, die ein Angreifer durchprobieren muss, bis er ein, zur Entschlüsselung dienendes Polynom, gefunden hat. Danach kommen wir zu den Lösungen des Problems, also zu dem erwarteten Ausblick auf das nächste Kapitel, in dem das NTRU-Schlüsselrekonstruktionsproblem in Bezug zu einem NP-vollständigen Problem, das Shortest Vector Problem, in einem Gitter dargestellt wird.

4.1 NTRU-Schlüsselrekonstruktionsproblem

Ein Angreifer des NTRU-Kryptosystems hat die Informationen der öffentlichen Parameter (N, p, q, d) und den öffentlichen Schlüssel $h(x)$. Die Koeffizienten des öffentlichen Schlüssels scheinen willkürliche ganze Zahlen modulo q zu sein. Es gibt jedoch eine ver-

steckte Beziehung, denn aus $h(x) = F_q(x) \star g(x)$ folgt

$$f(x) \star h(x) = \underbrace{f(x) \star F_q(x)}_{\equiv\ 1\ (\mathrm{mod}\ q)} \star g(x)$$
$$\equiv g(x)\ (\mathrm{mod}\ q),$$

wobei $f(x) \in \mathcal{T}(d+1, d, N)$ und $g(x) \in \mathcal{T}(d, d, N)$ sind und somit sehr kleine Koeffizienten besitzen. Soll das NTRU-Kryptosystem angegriffen werden, indem der private Schlüssel gefunden wird, so erhalten wir das NTRU-Schlüsselrekonstruktionsproblem:

Sei der öffentliche Schlüssel $h(x)$ und die öffentlichen Parameter gegeben, dann finde ternäre Polynome $f(x)$ und $g(x)$, so dass die Kongruenz

$$f(x) \star h(x) \equiv g(x)(\mathrm{mod}\ q)$$

erfüllt ist.

Die Lösung des NTRU-Schlüsselrekonstruktionsproblems ist nicht eindeutig, denn wenn das Paar $(f(x), g(x))$ eine Lösung ist, dann ist auch $(x^k \star f(x), x^k \star g(x))$ eine Lösung des Problems für jedes $0 \leq k < N$. Das Polynom $x^k \star f(x)$ wird auch eine Rotation des ternären Polynoms $f(x)$ genannt, denn die Koeffizienten werden um k Positionen zyklisch rotiert. Die Rotation wirkt als privater Entschlüsselungsschlüssel, und zwar so, dass die Entschlüsselung mit $x^k \star f(x)$ den rotierten Klartext $x^k \star m(x)$ ergibt.
Allgemein dient jedes Paar $(f(x), g(x))$, mit ausreichend kleinen Koeffizienten und das die Bedingung $f(x) \star h(x) \equiv g(x)(\mathrm{mod}\ q)$ erfüllt, als ein NTRU-Entschlüsselungsschlüssel. Zum Beispiel sei das ternäre Polynom $f(x)$ der original Entschlüsselungschlüssel und $z(x)$ sei ein Polynom mit sehr kleinen Koeffizienten, dann wirkt $z(x) \star f(x)$ auch als Entschlüsselungsschlüssel.

4.2 „brute-force“ Suche

Eve möchte eine „brute-force“ Suche durchführen, wenn sie einen privaten Schlüssel $f(x)$ gefunden hat, für den $f(x) \star h(x)\ (\mathrm{mod}\ q)$ ein ternäres Polynom ist, dann kann sie aufhören zu suchen und weiß nach Abschnitt 4.1, dass sie einen Schlüssel, der zur Entschlüsselung dient, gefunden hat. Mit aller Wahrscheinlichkeit (siehe Abschnitt 4.3) erfüllt nur das Polynom $f(x)$ und seine Rotationen die Bedingung

$$f(x) \star h(x) \equiv g(x)(\mathrm{mod}\ q),$$

so dass $g(x)$ ein ternäres Polynom ist. Aber wenn Eve ein anderes ternäres Polynom mit dieser Eigenschaft findet, dann wird dies auch als Entschlüsselungsschlüssel dienen. Nun müssen wir die Menge aller ternären Polynome bestimmen, die Eve durchsuchen muss. Im Allgemeinen können wir ein Polynom aus der Menge $\mathcal{T}(d_1, d_2, N)$ bestimmen, indem wir erst d_1 Koeffizienten als 1 wählen und danach d_2 Koeffizienten aus den übrigen $N - d_1$ Koeffizienten gleich -1 wählen. Also

$$\#\mathcal{T}(d_1, d_2, N) = \binom{N}{d_1} \cdot \binom{N - d_1}{d_2} = \frac{N!}{d_1!(N - d_1)!} \cdot \frac{(N - d_1)!}{d_2!(N - d_1 - d_2)!}$$
$$= \frac{N!}{d_1! d_2! (N - d_1 - d_2)!}.$$

Für eine „brute-force“ Suche muss Eve jedes Polynom aus $\mathcal{T}(d+1,d,N)$ ausprobieren, bis sie einen Entschlüsselungsschlüssel findet. Allerdings ist jede Rotation des ternären Polynoms $f(x)$ ein Schlüssel, der zur Entschlüsselung dient, und somit gibt es insgesamt N Schlüssel, die sie an den Geheimtext gelangen lassen. Folglich wird sie keine Rotationen von einem möglichen Schlüssel ausprobieren. Somit kostet es Eve im schlimmsten Fall $W^{\#} := \frac{\#\mathcal{T}(d+1,d,N)}{N}$ Versuche, bis sie den Schlüssel $f(x)$ oder eine Rotation von diesem findet.

Beispiel 4.1. *Wir betrachten die Anzahl von Schlüsseln, die Eve höchstens ausprobieren muss, wenn sie die vorgeschlagenen Sicherheitslevels aus 3.9 (Tabelle 3) oder wenn sie das gegebene Beispiel 3.8 für das NTRU-Verfahren brechen möchte.*

Sicherheit	N	q	p	d_1 *mit* $q > (6d_1+1)p$	$W^{\#} = \frac{\#\mathcal{T}(d_1+1,d_1,N)}{N}$
Moderat	167	128	3	6	$W^{\#} \approx 8,03 \cdot 10^{19}$
Standard	251	128	3	13	$W^{\#} \approx 1,06 \cdot 10^{41}$
Hoch	347	128	3	19	$W^{\#} \approx 1,25 \cdot 10^{60}$
Höchste	503	256	3	27	$W^{\#} \approx 1,08 \cdot 10^{87}$
Beispiel 3.8	11	41	3	2	420

Tabelle 4: Anzahl der Schlüssel, die Eve bei einer „brute-force“ Suche ausprobieren muss [4]

Es gibt verschiedene Möglichkeiten für den Parameter d. Damit die Entschlüsselung garantiert funktioniert kann maximal der Wert d_1 aus der Tabelle 4 gewählt werden, da dieser die Ungleichung $q > (6d_1+1)p$ erfüllt. Nach Bemerkung 3.1 können aber auch andere Werte benutzt werden, wobei allerdings die Wahrscheinlichkeit, dass die Entschlüsselung nicht funktioniert nur schwer abzuschätzen ist.
Die Tabelle 4 zeigt, dass je höher der Sicherheitslevel ist, desto mehr Schlüssel muss Eve ausprobieren. Nur das Beispiel 3.8 kann sie vergleichsweise sehr schnell brechen, da Eve hierfür nur 420 verschiedene Schlüssel ausprobieren muss.

4.3 Lösungen des NTRU-Schlüsselrekonstruktionsproblems

Das ternäre Polynom $f(x)$ und deren Rotationen sind wahrscheinlich die einzigen zur Entschlüsselung dienenden Polynome aus der Menge $\mathcal{T}(d+1,d,N)$. Um dies einzusehen betrachten wir die Wahrscheinlichkeit, mit der ein willkürlich gewähltes Polynom $f(x) \in \mathcal{T}(d+1,d,N)$, die Eigenschaft hat, dass

$$g(x) \equiv f(x) \star h(x) \pmod{q}$$

ein ternäres Polynom ist. Wir behandeln die Koeffizienten des Polynoms $g(x)$ als unabhängige[5], willkürlich gleich verteilte ganze Zahlen modulo q. Die Wahrscheinlichkeit, dass ein bestimmter Koeffizient ternär ist, beträgt $\frac{3}{q}$. Mit der Annahme, dass die Koeffizienten unabhängig sind, beträgt die Wahrscheinlichkeit, dass alle Koeffizienten des

[4] Die Werte wurden mit Hilfe des Texas Instruments „TI-92 Plus“ berechnet.

[5] In Wirklichkeit sind die Zahlen nicht völlig unabhängig, aber sie sind unabhängig genug um dies als eine gute Approximation annehmen zu können.

Polynoms ternär sind, ungefähr $\left(\frac{3}{q}\right)^N$.
Also folgt insgesamt

$$\begin{aligned} J &:= \begin{pmatrix} \text{Erwartete Anzahl von Ent-} \\ \text{schlüsselungsschlüsseln in } \mathcal{T}(d+1,d,N) \end{pmatrix} \\ &\approx W'keit \begin{pmatrix} f(x) \in \mathcal{T}(d+1,d,N) \text{ ist ein} \\ \text{Entschlüsselungsschlüssel} \end{pmatrix} \times \#\mathcal{T}(d+1,d,N) \\ &= \left(\frac{3}{q}\right)^N \binom{N}{d+1} \binom{N-(d+1)}{d}. \end{aligned}$$

Beispiel 4.2. *Wir betrachten die erwartete Anzahl von Entschlüsselungsschlüsseln für verschiedene Sicherheitsstufen.*

Sicherheit	N	q	p	d_1 *mit* $q > (6d_1+1)p$	$J \approx \left(\frac{3}{q}\right)^N \binom{N}{d_1+1} \binom{N-(d_1+1)}{d_1}$
Moderat	167	128	3	6	$J \approx 7,99 \cdot 10^{-251}$
Standard	251	128	3	13	$J \approx 1,88 \cdot 10^{-366}$
Hoch	347	128	3	19	$J \approx 9,93 \cdot 10^{-504}$
Höchste	503	256	3	27	$J \approx 2,41 \cdot 10^{-882}$
Beispiel 3.8	11	41	3	2	$J \approx 1,12 \cdot 10^{-9}$

Tabelle 5: Erwartete Anzahl von Entschlüsselungsschlüsseln [6]

Anhand der Tabelle 5 können wir sehen, dass die erwartete Anzahl von Schlüsseln, die zur Entschlüsselung dienen, immer geringer wird, je höher der Sicherheitslevel ist. Des weiteren sei angemerkt, dass diese Anzahl extrem gering ist.

Wenn das Polynom $h(x)$ zu den jeweiligen Sicherheitsleveln ein öffentlicher Schlüssel ist, dann existieren dazu auch Entschlüsselungsschlüssel, denn wir benötigen den privaten Schlüssel $f(x)$ um den öffentlichen Schlüssel $h(x)$ zu konstruieren. Somit können wir nun schließen, dass es sehr unwahrscheinlich ist, dass noch andere Entschlüsselungsschlüssel, also Lösungen des NTRU-Schlüsselrekonstruktionsproblems, existieren, außer dem privaten Schlüssel $f(x)$ von Alice und dessen Rotationen.

4.4 NTRU-Schlüsselrekonstruktionsproblem in Bezug zu einem NP-vollständigen Problem

Wir haben gesehen, dass das NTRU-Schlüsselrekonstruktionsproblem praktisch nicht lösbar ist durch eine „brute-force“ Suche (siehe 4.2). Das Problem ist ebenso nicht lösbar durch

[6] Die Werte wurden mit Hilfe des Texas Instruments „TI-92 Plus“ berechnet.

eine „colission“ Suche (siehe z.B. [HPS1] S. 399). Im nächsten Kapitel (NTRU als Gitter) werden wir zeigen, dass das Lösen des NTRU-Schlüsselrekonstruktionsproblems in gewisser Weise äquivalent zum Lösen des SVP in einer bestimmten Art von Gittern ist. Dies setzt das NTRU-Schlüsselrekonstruktionsproblem in Bezug zu einem gut studierten Problem in der Mathematik. Die beste Methode um den NTRU-Privatenschlüssel aus dem öffentlichen Schlüssel zu rekonstruieren ist zur Zeit das Anwenden der Gitterreduktion. Allerdings ist es unbekannt, ob die Gitterreduktion die allerbeste Methode ist um das Problem zu lösen. Niemand weiß, ob es noch bessere Algorithmen gibt. Die Schwierigkeit des Problems kann abgeschätzt werden, indem der zurzeit schnellste Algorithmus auf dieses angewendet wird.

5 NTRU als Gitter

In diesem Kapitel wird erläutert, wie die NTRU-Schlüsselrekonstruktion als ein kürzestes Vektorproblem (SVP) in einer bestimmten Art von Gittern interpretiert werden kann.

Es sei

$$h(x) = h_0 + h_1 x + h_2 x^2 + \ldots + h_{N-1} x^{N-1}$$

ein öffentlicher NTRU-Schlüssel. Dann bestimmt das Polynom $h(x)$ das Gitter $\mathcal{L}_h^{NTRU}$, welches durch die Matrix

$$M_h^{NTRU} = \begin{pmatrix} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & 1 & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 \\ 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{pmatrix}$$

beschrieben wird. Diese wird kurz geschrieben als

$$M_h^{NTRU} = \begin{pmatrix} I & h \\ 0 & qI \end{pmatrix},$$

wobei I die Einheitsmatrix, h die zyklischen Permutationen der Koeffizienten des Polynoms $h(x)$ und 0 die Nullmatrix beschreibt. Der Eintrag qI gibt eine Matrix an, die auf der Diagonalen q's stehen hat und sonst nur Nullen. Jeder dieser Blöcke hat sowohl N Zeilen als auch N Spalten mit Koeffizienten aus dem Polynomring $\mathcal{R}^N$.

Zur Vereinfachung zunächst eine Definition.

Definition 5.1. Für zwei Polynome $a(x)$, $b(x)$ aus $\mathcal{R}^N$ mit jeweils den Koeffizienten a_i bzw. b_i mit $0 \leq i \leq N-1$ wird die Bezeichnung

$$(a, b) := (a_0, a_1, \ldots, a_{N-1}, b_0, b_1, \ldots, b_{N-1}) \in \mathbb{Z}^{2N}$$

definiert.

Mit der folgenden Proposition können wir zeigen, dass die ternären Polynome $f(x)$ und $g(x)$ Elemente in dem erzeugten NTRU-Gitter $\mathcal{L}_h^{NTRU}$ sind.

Proposition 5.2. *Seien $f(x)$ und $g(x)$ die privaten zufälligen Polynome aus der 3.2 Schlüsselentwicklung und $h(x)$ der öffentliche Schlüssel, der durch $h(x) = F_q(x) \star g(x)$ (mod q) erzeugt wird. Ist nun $f(x) \star h(x) \equiv g(x)$ (mod q) und sei $u(x) \in \mathcal{R}^N$ das Polynom, so dass mit einem $q \in \mathbb{N}$ gilt:*

$$f(x) \star h(x) = g(x) + q \cdot u(x).$$

Dann folgt:

$$(f, -u) M_h^{NTRU} = (f, g).$$

Der Vektor (f, g) liegt also im NTRU-Gitter $\mathcal{L}_h^{NTRU}$.

Beweis. Betrachte das Produkt

$$(f, -u) M_h^{NTRU} = (f, -u) \begin{pmatrix} I & h \\ 0 & qI \end{pmatrix}.$$

Für die ersten N Koordinaten berechnen wir $f \cdot I + 0 \cdot (-u)$. Hierbei kommt genau der Vektor f mit den Koeffizienteneinträgen f_i heraus. Weiter betrachten wir, was passiert wenn $f \cdot h - u \cdot q \cdot I$ berechnet wird

$$f \cdot h - u \cdot q \cdot I = \big((f_0 h_0, f_1 h_{N-1}, \ldots, f_{N-1} h_1, -q u_0), (f_0 h_1, f_1 h_0, \ldots, f_{N-1} h_2, -q u_1), \\ \ldots, (f_0 h_{N-1}, f_1 h_{N-2}, \ldots, f_{N-1} h_0, -q u_{N-1})\big).$$

Die k-te Komponente $(f_0 h_k, f_1 h_{k-1}, \ldots, f_{N-1} h_{k+1}, -q u_k)$ entspricht somit dem k-ten Eintrag des Vektors $f(x) \star h(x) - q \cdot u(x) = g(x)$. Hiermit ist also $(f, -u) M_h^{NTRU} = (f, g)$ bewiesen. Dieses bedeutet, dass wir den Vektor (f, g) erhalten, wenn wir eine bestimmte Linearkombination der Zeilen der Matrix M_h^{NTRU} wählen. Die privaten Schlüssel von Alice liegen somit im NTRU-Gitter $\mathcal{L}_h^{NTRU}$. □

Als nächstes zeigen wir, dass die privaten Schlüssel $f(x)$ und $g(x)$ äquivalent zu dem kürzestem Vektor in dem NTRU-Gitter $\mathcal{L}_h^{NTRU}$ sind.

Proposition 5.3. *Seien (N, p, q, d) die NTRU-Parameter, die zur Vereinfachung*

$$d \approx \frac{N}{3} \qquad \textit{und} \qquad q \approx 6dp \approx 2Np \ \big(\textit{da } q > (6d+1)p\big)$$

genügen. Sei $\mathcal{L}_h^{NTRU}$ ein NTRU-Gitter, das zu dem privaten Schlüssel (f,g) gehört.

Dann gilt

(i) $det\big(\mathcal{L}_h^{NTRU}\big) = q^N$,

(ii) $\|(f,g)\| \approx \sqrt{4d} \approx \sqrt{\frac{4N}{3}} \approx 1,155\sqrt{N}$,

(iii) Nach der Heuristik von Gauß gibt es eine große Wahrscheinlichkeit, dass der kürzeste von Null verschieden Vektor in dem NTRU-Gitter die Länge

$$\sigma\left(\mathcal{L}_h^{NTRU}\right) \approx \sqrt{\frac{Nq}{\pi e}} \approx \sqrt{\frac{2p}{\pi e}}\ N \approx 0,484\sqrt{p}\ N$$

besitzt.

Beweis.

(i) Nach Proposition 2.13 gilt $det\left(\mathcal{L}_h^{NTRU}\right) = det\left(M_h^{NTRU}\right)$, da die Matrix M_h^{NTRU} eine obere Dreiecksgestalt hat mit N Einsen auf der Diagonalen und weitere N mal den Eintrag q auf der Diagonalen, folgt $det\left(M_h^{NTRU}\right) = q^N$.

(ii) Jedes der Polynome $f(x) \in \mathcal{T}(d+1,d,N)$ und $g(x) \in \mathcal{T}(d,d,N)$ hat etwa d Koeffizienten, die den Wert 1 und d Koeffizienten die den Wert -1 annehmen. Bei der Berechnung der euklidischen Norm des Vektors (f,g) werden die Einsen und Minuseinsen des Vektors g quadriert und aufaddiert. Hieraus folgt der Wert $2d$. Analog bei dem Vektor f ergibt sich der Wert $2d+1$, so dass insgesamt $\|(f,g)\| \approx \sqrt{4d}$ folgt. Setzen wir nun noch $d \approx \frac{N}{3}$, so folgt der Rest der Behauptung.

(iii) Wir schätzen den kürzesten Vektor mit Hilfe der Heuristik von Gauß 2.20 ab. Dabei beachten wir, dass die Dimension $2N$ beträgt und nach (i) $\det\big(L_h^{NTRU}\big) = q^N$ gilt. Also folgt

$$\sigma\left(\mathcal{L}_h^{NTRU}\right) = \sqrt{\frac{2N}{2\pi e}} det\left(\mathcal{L}_h^{NTRU}\right)^{\frac{1}{2N}} \approx \sqrt{\frac{Nq}{\pi e}}.$$

□

Hieraus schließen wir, dass mit einer hohen Wahrscheinlichkeit bei einem großen N der kürzeste Vektor in dem Gitter $\mathcal{L}_h^{NTRU}$ der Vektor (f,g) und seine Rotationen ist. Weiter ist

$$\frac{\|(f,g)\|}{\sigma(\mathcal{L})} \approx \frac{1,155\sqrt{N}}{0,484\sqrt{p}N} \approx \frac{2,39}{\sqrt{pN}}.$$

Insgesamt liegt der Vektor (f,g) in dem Gitter $\mathcal{L}_h^{NTRU}$ und ist um einen Faktor $\mathcal{O}\left(1/\sqrt{N}\right)$ kürzer als die Heuristik von Gauß erwarten lässt. Daraus folgt, dass die NTRU-Schlüsselrekonstruktion ein Shortest Vector Problem in einer speziellen Art von Gittern ist.

6 Angriff mit dem LLL-Algorithmus

Im 5. Kapitel haben wir gesehen, dass Eve den privaten NTRU-Schlüssel bestimmen kann, wenn sie einen kürzesten Vektor im NTRU-Gitter $\mathcal{L}_h^{NTRU}$ findet. Die Sicherheit vom NTRU-Kryptosystem hängt zumindest von der Schwierigkeit ab ein SVP im Gitter $\mathcal{L}_h^{NTRU}$ zu lösen. Im Allgemeinen gilt, sofern Eve das apprSVP in $\mathcal{L}_h^{NTRU}$ mit einem Faktor von ungefähr N^ϵ für $\epsilon < \frac{1}{2}$ lösen kann, dann wird der Vektor, den sie findet, wahrscheinlich als Entschlüsselungsschlüssel dienen. Der LLL-Algorithmus läuft in polynomialer Zeit und löst apprSVP mit einem Faktor 2^N, aber wenn der Wert N des öffentlichen Schlüssles $h(x) \in \mathcal{R}_q^N$ groß ist, dann kann der LLL-Algorithmus keine sehr kleinen Vektoren in dem dazugehörigen Gitter $\mathcal{L}_h^{NTRU}$ finden.

Im Folgenden wird der LLL-Algorithmus vorgestellt. Dieser wurde 1982 von Lenstra, Lenstra und Lovász publiziert und nach ihnen benannt.

Wir gehen davon aus, dass die Basis $\mathcal{B} = \{v_1, \ldots, v_n\}$ eines Gitters $\mathcal{L}$ gegeben ist. Die Aufgabe ist nun diese Basis $\mathcal{B}$ in eine bessere Basis umzuformen. Mit einer besseren Basis ist hier gemeint, dass die Basisvektoren so kurz wie möglich, bzgl. der euklidischen Norm, sind. Beginnend mit dem kürzesten Basisvektor und dann aufsteigend angeordnet. Alternativ sollen die Vektoren in der besseren Basis möglichst orthogonal sein, d. h. das Skalarprodukt $v_i \cdot v_j$ soll so nahe an der Null liegen wie möglich.
Wenn uns dies gelingt haben wir einen kürzesten Vektor in einem Gitter gefunden.

Um eine bessere Basis zu konstruieren, beginnen wir aus der Bekannten eine orthogonale Basis zu berechnen. Dies geschieht mit Hilfe von Algorithmus 1, einer Variante des, aus der Linearen Algebra bekannten, Gram-Schmidt-Algorithmus.

Algorithmus 1 (Variante des Gram-Schmidt-Algorithmus)**.** *Die Vektoren $v_1, \ldots, v_n$ seien eine Basis des Vektorraums $V \subset \mathbb{R}^m$. Mit dem folgenden Algorithmus erhalten wir eine orthogonale Basis $v_1^*, \ldots, v_n^*$ für V:*
[1] *Setze $v_1^* = v_1$.*
[2] *Schleife für $i = 2, 3, \ldots, n$.*
[3] *Berechne $\mu_{i,j} = v_i \cdot v_j^* / \left\|v_j^*\right\|^2$ für $1 \leq j < i$.*
[4] *Setze $v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j^*$.*
[5] *Ende Schleife.*

Die beiden Basen haben die Eigenschaft das

$$span\{v_1, \ldots, v_n\} = span\{v_1^*, \ldots, v_n^*\} \quad \text{für } i = 1, 2, \ldots, n.$$

Die Vektoren der Basis $\mathcal{B}^* = \{v_1^*, \ldots, v_n^*\}$ bilden auch eine orthogonale Basis für den Vektorraum V, der von der Basis $\mathcal{B} = \{v_1, \ldots, v_n\}$ aufgespannt wird. Allerdings ist $\mathcal{B}^*$ keine Basis für das Gitter $\mathcal{L}$, das von $\mathcal{B}$ aufgespannt wird, denn der Algorithmus 1 benutzt Linearkombinationen bei denen nicht ganzzahlige Koeffizienten auftreten können.

Mit der nächsten Proposition ergibt sich, dass die beiden Basen dieselbe Determinante besitzen.

Proposition 6.1. *Sei $\mathcal{B} = \{v_1, \ldots, v_n\}$ eine Basis des Gitters $\mathcal{L}$ und sei $\mathcal{B}^* = \{v_1^*, \ldots, v_n^*\}$ die dazugehörige Gram-Schmidt orthogonale Basis, wie mit Algorithmus 1 berechnet wird. Dann gilt*

$$det(\mathcal{L}) = \prod_{i=1}^{n} \|v_i^*\| .$$

Beweis. Es sei $F = (v_1, \ldots, v_n)$ die Matrix (2) wie in Proposition 2.13 beschrieben. Also die Matrix, die als Zeilen die Einträge der Vektoren $v_1, \ldots, v_n$ besitzt. Nach der Proposition 2.13 wissen wir, dass $\det(\mathcal{L}) = |\det(F)|$ gilt. Es sei $F^* = (v_1^*, \ldots, v_n^*)$ die analoge Matrix mit den Vektoren $v_1^*, \ldots, v_n^*$ als Zeilen. Aus dem Algorithmus 1, Schritt [3] und [4], können wir schließen, dass die Matrizen F und F^* in Zusammenhang stehen, und zwar

$$MF^* = F,$$

wobei M die Matrix ist, die den Basiswechsel vollzieht.

$$M = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ \mu_{2,1} & 1 & 0 & \cdots & 0 & 0 \\ \mu_{3,1} & \mu_{3,2} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ \mu_{n-1,1} & \mu_{n-1,2} & \mu_{n-1,3} & \cdots & 1 & 0 \\ \mu_{n,1} & \mu_{n,2} & \mu_{n,3} & \cdots & \mu_{n,n-1} & 1 \end{pmatrix}.$$

Diese Matrix hat eine untere Dreiecksgestalt mit Einsen auf der Diagonalen. Für die Determinante von M bedeutet dies $\det(M)$=1. Schließlich ergibt sich

$$\det(\mathcal{L}) = |\det(F)| = |\det(MF^*)| = |\det(M)\ \det(F^*)| = |\det(F^*)| .$$

Die Zeilen der Matrix F^* bestehen aus den paarweise orthogonalen Vektoren $v_1^*, \ldots, v_n^*$. Für die Determinante der Matrix bedeutet dies:

$$|\det(F^*)| = \prod_{i=1}^{n} \|v_i^*\| .$$

□

Obwohl die durch den Algorithmus 1 berechnete Basis $\mathcal{B}^*$ keine Basis für das von der Basis $\mathcal{B}$ erzeugte Gitter $\mathcal{L}$ ist, benötigen wir die Basis $\mathcal{B}^*$, um ein wichtiges Konzept für den LLL-Algorithmus zu definieren.

Definition 6.2. Sei $\mathcal{B} = \{v_1, \ldots, v_n\}$ eine Basis des Gitters $\mathcal{L}$ und sei $\mathcal{B}^* = \{v_1^*, \ldots, v_n^*\}$ die dazugehörige Gram-Schmidt orthogonale Basis, wie mit Algorithmus 1 berechnet. Die Basis $\mathcal{B}$ wird *LLL reduziert* genannt, wenn sie die folgenden zwei Bedingungen erfüllt:

Größen Bedingung $$|\mu_{i,j}| = \frac{|v_i \cdot v_j|}{\|v_j^*\|} \leq \frac{1}{2} \quad \text{für alle } 1 \leq j < i \leq n.$$

Lovász Bedingung $$\|v_i^*\|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \|v_{i-1}^*\|^2 \quad 1 < i \leq n.$$

Lenstra, Lenstra und Lovász haben herausgefunden, dass es sich bei der LLL reduzierten Basis um eine gute Basis handelt und das es möglich ist diese in polynomialer Zeit zu berechnen. Wir betrachten die erwünschten Eigenschaften der LLL reduzierten Basis.

Satz 6.3. *Sei $\mathcal{L}$ ein Gitter der Dimension n. Jede LLL reduzierte Basis $\{v_1, v_2, \ldots, v_n\}$ für $\mathcal{L}$ hat die folgenden zwei Eigenschaften:*

$$\prod_{i=1}^{n} \|v_i\| \leq 2^{n(n-1)/4} det(\mathcal{L}),$$
$$\|v_j\| \leq 2^{(i-1)/2} \|v_i^*\| \quad \text{für alle } 1 \leq j \leq i \leq n.$$

Weiterhin genügt der erste Vektor in der LLL reduzierten Basis den Ungleichungen:

$$\|v_1\| \leq 2^{(n-1)/4} |det(\mathcal{L})|^{\frac{1}{n}} \quad \text{und} \quad \|v_1\| \leq 2^{(n-1)/2} \min_{0 \neq v \in \mathcal{L}} \|v\|.$$

Also löst eine LLL reduzierte Basis apprSVP mit einem Faktor von $2^{(n-1)/2}$.

Beweis. Zunächst machen wir ein paar Vorüberlegungen: Nach der Definition 6.2 wissen wir, dass

$$|\mu_{i,i-1}| \leq \frac{1}{2} \qquad \text{und} \qquad \|v_i^*\|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \left\|v_{i-1}^*\right\|^2$$

gilt. Hieraus folgt zusammen

$$\|v_i^*\|^2 \geq \frac{1}{2} \left\|v_{i-1}^*\right\|^2 \qquad \Leftrightarrow \qquad \left\|v_{i-1}^*\right\|^2 \leq 2 \|v_i^*\|^2 \qquad \Leftrightarrow \qquad \left\|v_{i-1}^*\right\|^2 \leq 2^{i-(i-1)} \|v_i^*\|^2.$$

Wenden wir diese Abschätzung öfter hintereinander an, so erhalten wir die allgemeine Abschätzung

$$\left\|v_j^*\right\|^2 \leq 2^{i-j} \|v_i^*\|^2 \text{ für } 1 \leq j < i \leq n.$$

Als erstes beweisen wir $\prod_{i=1}^{n} \|v_i\| \leq 2^{n(n-1)/4} \det(\mathcal{L})$:
Hierfür betrachten wir aus dem Algorithmus 1 den Schritt [4] und rechnen:

$$\begin{aligned}
\|v_i\|^2 &= \left\| v_i^* + \sum_{j=1}^{i-1} \mu_{i,j} v_j^* \right\|^2 \\
&= \|v_i^*\|^2 + \sum_{j=1}^{i-1} \mu_{i,j}^2 \left\|v_j^*\right\|^2 \qquad \text{da } v_1^*, \ldots, v_n^* \text{ paarweise orthogonal sind,} \\
&\leq \|v_i^*\|^2 + \sum_{j=1}^{i-1} \frac{1}{4} \left\|v_j^*\right\|^2 \qquad \text{da } |\mu_{i,j}| \leq \frac{1}{2}, \\
&\leq \|v_i^*\|^2 + \sum_{j=1}^{i-1} 2^{i-j-2} \|v_i^*\|^2 \qquad \text{da } \left\|v_j^*\right\|^2 \leq 2^{i-j} \|v_i^*\|^2, \\
&= \frac{1 + 2^{i-1}}{2} \|v_i^*\|^2 \\
&\leq 2^{i-1} \|v_i^*\|^2 \qquad \text{da } 1 \leq 2^{i-1} \text{ für alle } i \geq 1.
\end{aligned}$$

Multiplizieren wir das Ergebnis $\|v_i\|^2 \leq 2^{i-1} \|v_i^*\|^2$ mit sich selber für $1 \leq i \leq n$ so ergibt sich

$$\prod_{i=1}^{n} \|v_i\|^2 \leq \prod_{i=1}^{n} 2^{i-1} \|v_i^*\|^2 = 2^{n(n-1)/2} \prod_{i=1}^{n} \|v_i^*\|^2$$
$$= 2^{n(n-1)/2} (\det(\mathcal{L}))^2 \qquad \text{da nach Proposition 6.1 } \det(\mathcal{L}) = \prod_{i=1}^{n} \|v_i^*\| \text{ gilt.}$$

Somit ist $\prod_{i=1}^{n} \|v_i\| \leq 2^{n(n-1)/4} \det(\mathcal{L})$ gezeigt.
Zeigen wir nun $\|v_j\| \leq 2^{(i-1)/2} \|v_i^*\|$:
Für jedes $j \leq i$ benutzen wir das vorherige Ergebnis $\|v_i\|^2 \leq 2^{i-1} \|v_i^*\|^2$, jetzt mit $i = j$:

$$\|v_j\|^2 \leq 2^{j-1} \left\|v_j^*\right\|^2 \leq 2^{j-1} \cdot 2^{i-j} \left\|v_j^*\right\|^2 = 2^{i-1} \left\|v_j^*\right\|^2 .$$

Hieraus folgt nach ziehen der Wurzel $\|v_j\| \leq 2^{(i-1)/2} \|v_i^*\|$.
Als Drittes beweisen wir die Ungleichung $\|v_1\| \leq 2^{(n-1)/4} |\det(\mathcal{L})|^{\frac{1}{n}}$:
Wir benutzen die schon bewiesene Abschätzung $\|v_j\| \leq 2^{(i-1)/2} \|v_i^*\|$ und setzten $j = 1$. Multiplizieren wir dies über $1 \leq i \leq n$ und benutzen wieder $\det(\mathcal{L}) = \prod_{i=1}^{n} \|v_i^*\|$ (Proposition 6.1) so erhalten wir:

$$\|v_1\|^n \leq \prod_{i=1}^{n} 2^{(i-1)/2} \|v_i^*\| = 2^{n(n-1)/4} \prod_{i=1}^{n} \|v_i^*\| = 2^{n(n-1)/4} \det(\mathcal{L}).$$

Ziehen wir noch die n-te Wurzel, so erhalten wir die gewünschte Ungleichung
$\|v_1\| \leq 2^{(n-1)/4} |\det(\mathcal{L})|^{\frac{1}{n}}$.
Als letztes zeigen wir die Abschätzung $\|v_1\| \leq 2^{(n-1)/2} \ \min_{0 \neq v \in \mathcal{L}} \|v\|$:
Wir betrachten hierfür einen nicht trivialen Gittervektor $v \in \mathcal{L}$ und schreiben diesen als

$$v = \sum_{j=1}^{i} a_j v_j = \sum_{j=1}^{i} b_j v_j^*,$$

mit $a_i \neq 0$. Die Koeffizienten $a_1, \ldots, a_n$ sind ganze Zahlen, wobei die $b_1, \ldots, b_n$ aus den reellen Zahlen stammen. Insbesondere gilt $|a_i| \geq 1$. Nach Konstruktion sind die Vektoren $v_1^*, \ldots, v_n^*$ paarweise orthogonal und spannen denselben Raum auf wie die Vektoren $v_1, \ldots, v_n$. Daher gilt

$$v \cdot v_i^* = a_i v_i \cdot v_i^* = b_i v_i^* \cdot v_i^* \qquad \text{und} \qquad v_i \cdot v_i^* = v_i^* \cdot v_i^*.$$

Hieraus folgern wir $a_i = b_i$ und schließlich $|b_i| = |a_i| \geq 1$. Wir betrachten nun:

$$\|v\|^2 = \left\| \sum_{j=1}^{i} b_j v_j^* \right\|^2 = \sum_{j=1}^{i} b_j^2 \left\|v_j^*\right\|^2 \qquad \text{da die } v_1^*, \ldots, v_i^* \text{ paarweise orthogonal sind.}$$
$$\geq b_i^2 \|v_i^*\|^2$$
$$\geq \|v_i^*\|^2$$
$$\geq 2^{-(i-1)} \|v_1\|^2 .$$

Die letzte Abschätzung erhalten wir aus der schon bewiesenen Abschätzung $\|v_j\| \leq 2^{(i-1)/2} \|v_i^*\|$ mit $j = 1$. Denn

$$\|v_1\|^2 \leq 2^{i-1} \|v_i^*\|^2 \qquad \Leftrightarrow \qquad \|v_i^*\|^2 \geq 2^{-(i-1)} \|v_1\|^2 .$$

Schließlich folgt

$$2^{-(i-1)} \|v_1\|^2 \geq 2^{-(n-1)} \|v_1\|^2 .$$

Somit ergibt sich die zu zeigende Abschätzung:

$$\|v\|^2 \geq 2^{-(n-1)} \|v_1\|^2 \qquad \Leftrightarrow \qquad \|v\| \geq 2^{-(n-1)/2} \|v_1\| \qquad \Leftrightarrow \qquad \|v_1\| \leq 2^{(n-1)/2} \|v\| ,$$

hieraus folgt insbesondere, da $v \in \mathcal{L}$ ein beliebiger nicht trivialer Gittervektor ist,

$$\|v_1\| \leq 2^{(n-1)/2} \min_{0 \neq v \in \mathcal{L}} \|v\| .$$

□

Im Folgenden wird der LLL-Gitterreduktions-Algorithmus dargestellt.

Algorithmus 2.
[1] *Eingabe: eine Basis* $\{v_1, \ldots, v_n\}$ *des Gitters* $\mathcal{L}$.
[2] *Setze* $v_1^* = v_1$.
[3] *Setze* $k = 2$.
[4] *Schleife solange* $k \leq n$.
[5] *Schleife* $j = k-1, k-2, \ldots, 2, 1$.
[6] *Setze* $v_k = v_k - \lfloor \mu_{k,j} \rceil \, v_j$. [*Größenreduktion*]
[7] *Ende j Schleife.*
[8] *Wenn* $\|v_k^*\|^2 \geq \left(\frac{3}{4} - \mu_{k,k-1}^2\right) \left\|v_{k-1}^*\right\|^2$. [*Lovász Bedingug*]
[9] *Setze* $k = k + 1$.
[10] *Sonst*
[11] *tausche* v_{k-1} *und* v_k. [*Austausch Schritt*]
[12] *Setze* $k = max\,(k-1, 2)$.
[13] *Ende Wenn.*
[14] *Ende k Schleife.*
[15] *Ausgabe : LLL reduzierte Basis* $\{v_1, \ldots, v_n\}$.
Bei jedem Schritt ist $\{v_1^*, \ldots, v_k^*\}$ *die orthogonale Menge von Vektoren die wir erhalten, wenn der Gram-Schmidt-Algorithmus 1 auf die aktuellen Vektoren* $v_1, \ldots, v_k$ *angewendet wird und* $\mu_{i,j}$ *ist die dazugehörige Größe* ($\mu_{i,j} = v_i \cdot v_j^* / \left\|v_j^*\right\|^2$).

Satz 6.4 (LLL-Algorithmus)**.** *Sei* $\{v_1, \ldots, v_n\}$ *eine Basis des Gitters* $\mathcal{L}$. *Algorithmus 2 terminiert in einer endlichen Anzahl von Schritten und gibt eine LLL reduzierte Basis für das Gitter* $\mathcal{L}$ *aus.*
Exakter, sei $\mathcal{B} = \max \|v_i\|$, *dann führt der Algorithmus 2 die Hauptschleife über k* (*Schritt* [4] − [14]) *nicht mehr als* $\mathcal{O}(n^2 log\, n + n^2 log\, \mathcal{B})$ *durch. Insbesondere ist der LLL-Algorithmus ein polynomialer Algorithmus.*

Beweis. Der Beweis zu diesem Satz wird in [HPS1] auf den Seiten 413 und 414 skizziert. □

Wird der LLL-Algorithmus ausgeführt, so ergibt sich eine LLL reduzierte Basis. Der erste Basisvektor ist dann ein approximativ kürzester Vektor in dem Gitter $\mathcal{L}$.
Nach Proposition 5.3 kann dieser Basisvektor mit Polynomen identifiziert werden, die nach der Heuristik von Gauß wahrscheinlich die privaten Schlüssel (f, g) sind.
Der LLL-Algorithmus löst das apprSVP in einem Gitter mit einer großen Dimension n nicht mehr genau genug. Es gibt Varianten vom LLL-Algorithmus, z.B. der BKZ-LLL-Algorithmus (siehe z. B. [HPS1]), der für dieses Problem eine bessere Lösung angibt, allerdings mit einer Laufzeit die exponentiell mit der Dimension n wächst.

7 Fazit

Für mich sind die public key Kryptosysteme sehr faszinierend, denn aus den öffentlichen Schlüssel kann so leicht kein Schlüssel zur Entschlüsselung rekonstruiert werden. Das NTRU-Verfahren wird im Ring der Konvolutionspolynome durchgeführt, aber in diesem Gebiet der Mathematik kann es quasi nicht angegriffen werden. Die Rechenschritte für die Verschlüsselung sind einfach auszuführen. Der Angriff hingegen ist sehr schwer durchzuführen. Dies ist das Prinzip einer Einwegfunktion.
Wird das Problem in einen anderen Bereich der Mathematik gehoben, beim NTRU-Verfahren in das Gebiet der Gitter, so ergeben sich neue Möglichkeiten, mit dem ein besserer Angriff vollzogen werden kann (z.B. mit dem LLL-Algorithmus aus Kapitel 6).
Nachdem ich mich in Form meiner Bachelorarbeit über längere Zeit mit dem NTRU-Kryptosystem befasst habe und die Mathematik hinter dem Verfahren verstanden habe, kann ich mir vorstellen, dass es eine lohnende Alternative zu dem, zur Zeit meist benutzten, RSA-Verfahren werden könnte. Das RSA-Verfahren kann gebrochen werden, indem eine große Zahl in ihre zwei Primfaktoren zerlegt wird ([HPS1] Kapitel 3, 3.1). Allerdings ist es nicht bekannt, ob es hierfür einen guten Algorithmus gibt. Wird das NTRU-Verfahren gebrochen, indem der geheime Schlüssel im NTRU-Gitter $\mathcal{L}_h^{NTRU}$ gefunden wird, so ist dies äquivalent zum Lösen des NP-Vollständigen Shortest Vector Problem. Des weiteren haben wir im Abschnitt 3.7 (Zeitkomplexität) gesehen, dass das NTRU-Verfahren schneller ist als das RSA-Verfahren.
Bei den RFID Chips, was für Radio Frequency Identification steht, findet das NTRU-Verfahren schon seine Anwendung.
(Siehe [NTRU], genauer http://www.ntru.com/downloads/RFID_White_paper_FNL.pdf)

Literatur

[BNS] *Albrecht Beutelsbacher, Heike B. Neumann, Thomas Schwarzpaul*: Kryptografie in Theorie und Praxis, (Vieweg 2005)

[HPS1] *Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman*: An Introduction to Mathematical Cryptography, (Springer 2008)

[HPS2] *Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman*: *Internetseite*: www.math.brown.edu/∼jhs/MathCryptoHome.html, Errata List (update Mai 2009)

[Kü] *Ulf Kühn (Universität Hamburg)*: Vorlesungsskript „Elementare Zahlentheorie“, (Wintersemester 2008/2009)

[Ne] *Jürgen Neukirch*: Algebraische Zahlentheorie, (Springer 2006)

[NTRU] *Internetseite*: www.ntru.com

[Ru] *Wolfgang M. Ruppert (Universität Erlangen)*: Vorlesungsskript „Kryptographie und Gitter“, (Wintersemester 2007/2008)

[Si] *Simon Singh*: Geheime Botschaften (Die Kunst der Verschlüsselng von der Antike bis in die Zeit des Internets), (dtv 2006)

Anhang

Das Beispiel 3.8 wurde mit Hilfe von Classic Worksheet Maple 9.5 berechnet.

Öffentliche Parameter:

N=11, q=41, p=3, d=2.

Die zufällig gewählten ternären Polynome von Alice lauten:

```
> f:=x^7-x^6-x^4+x^3+x;
> g:=-x^9+x^7-x^2+x;
```

$$f := x^7 - x^6 - x^4 + x^3 + x$$
$$g := -x^9 + x^7 - x^2 + x$$

Mit Hilfe von Lemma 2.17 wird zunächst überprüft ob die Inversen zu dem Polynom f existieren. Danach werden die Inversen (F_q, F_p) mit Hilfe des erweiterten euklidischen Algorithmus berechnet:

```
> t:=x^11-1;
> gcdex(f,t,x,u,v);
> u,v;
> expand(f*u+t*v);
```

$$t := x^{11} - 1$$
$$1$$

$$-\frac{4}{23} + \frac{20}{23}x^4 - \frac{2}{23}x^3 - \frac{9}{23}x^2 + \frac{17}{23}x - \frac{1}{23}x^6 - \frac{16}{23}x^5 - \frac{8}{23}x^8 + \frac{10}{23}x^7 + \frac{5}{23}x^{10} + \frac{11}{23}x^9,$$
$$-1 - \frac{4}{23}x + \frac{17}{23}x^2 - \frac{13}{23}x^3 + \frac{19}{23}x^4 - \frac{6}{23}x^5 - \frac{5}{23}x^6$$
$$1$$

```
> Fq:=u mod 41;
> Fp:=u mod 3;
```

$$Fq := 23 + 8x^4 + 32x^3 + 21x^2 + 15x + 16x^6 + 10x^5 + 5x^8 + 4x^7 + 2x^{10} + 29x^9$$
$$Fp := 1 + x^4 + 2x^3 + x + x^6 + x^5 + 2x^8 + 2x^7 + x^{10} + x^9$$

Zur Sicherheit eine Kontrolle:

```
> expand(f*u+t*v);
> o:=expand(f*Fp) mod 3;
> l:=expand(f*Fq) mod 41;
```

$$1$$
$$o := x + x^2 + 2x^6 + 2x^4 + x^3 + x^{11} + 2x^{13} + 2x^{12} + x^{15} + 2x^{14} + x^{17}$$

$$l := 23x + 15x^2 + 39x^6 + 24x^4 + 3x^3 + x^{11} + 14x^5 + 26x^{13} + 18x^{12} + 17x^{15} + 38x^{14} + 2x^{17} + 27x^{16}$$

Die Polynome o und l werden von Hand im Ring der Konvolutionspolynome $\mathcal{R}^{11}$ dargestellt. Anschließend modulo 3 bzw. modulo 41 gerechnet, damit o im Ring $\mathcal{R}_3^{11}$ und l im Ring $\mathcal{R}_{41}^{11}$ liegen:

```
> o:=x^3+2*x+x^4+2*x^3+x^6+2*x^2+2*x^4+1+2*x^6+x+x^2 mod 3;
> l:=3*x^3+18*x+17*x^4+38*x^3+2*x^6+27*x^5+26*x^2+24*x^4+1+39*x^6+23*x+1
> 4*x^5+15*x^2 mod 41;
```

$$o := 1$$
$$l := 1$$

Der geheime Schlüssel ist nun das Tupel (f, F_p).

Berechne den öffentlichen Schlüssel h:

```
> h:=expand(Fq*g) mod 41;
```

$h := 40\,x^9 + 23\,x + 29\,x^7 + 33\,x^2 + 2\,x^6 + 11\,x^4 + 6\,x^3 + x^{11} + 3\,x^8 + 17\,x^5 + 8\,x^{13} + 17\,x^{12}$
$+ 30\,x^{15} + 35\,x^{14} + 38\,x^{17} + 25\,x^{16} + 39\,x^{19} + 12\,x^{18}$

Der öffentliche Schlüssel h wird von Hand im Ring der Konvolutionspolynome $\mathcal{R}^{11}$ dargestellt. Anschließend modulo 41 gerechnet, damit h im Ring $\mathcal{R}_{41}^{11}$ liegt:

```
> h:=
> 40*x^9+29*x^7+33*x^2+2*x^6+11*x^4+6*x^3+1+3*x^8+17*x^5+8*x^2+17*x+30*x
> ^4+35*x^3+38*x^6+25*x^5+39*x^8+12*x^7+23*x mod 41;
```

$$h := 40\,x^9 + 40\,x^6 + 1 + x^8 + x^5 + 40\,x$$

Dies sind die zu verschlüsselten Nachrichten $m_1, m_2, \ldots, m_{10}$:

```
> m1:=x^10-x^9-x^8-x^6-x^4-x^3+x^2-x-1;
> m2:=x^10-x^9+x^8-x^6+x^3-x+1;
> m3:=x^10-x^9-x^8-x^7-x^6+x^5-x^3-x^2-x-1;
> m4:=-x^10-x^9+x^8-x^7+x^6-x^5-x^4+x^2-1;
> m5:=-x^10+x^9-x^6+x^5+x-1;
> m6:=-x^9+x^7-x^3-x^2-x-1;
> m7:=x^10-x^5-x^4-x^3-x-1;
> m8:=x^9-x^7-x^4-x^3-1;
> m9:=-x^10+x^8-x^6-x^5+x^4+x^3-x^2+x;
> m10:=-x^10-x^9-x^8-x^7-x^6-x^5-x^4+x^3+x^2-1;
```

$$m1 := x^{10} - x^9 - x^8 - x^6 - x^4 - x^3 + x^2 - x - 1$$
$$m2 := x^{10} - x^9 + x^8 - x^6 + x^3 - x + 1$$
$$m3 := x^{10} - x^9 - x^8 - x^7 - x^6 + x^5 - x^3 - x^2 - x - 1$$
$$m4 := -x^{10} - x^9 + x^8 - x^7 + x^6 - x^5 - x^4 + x^2 - 1$$
$$m5 := -x^{10} + x^9 - x^6 + x^5 + x - 1$$
$$m6 := -x^9 + x^7 - x^3 - x^2 - x - 1$$
$$m7 := x^{10} - x^5 - x^4 - x^3 - x - 1$$
$$m8 := x^9 - x^7 - x^4 - x^3 - 1$$

$$m9 := -x^{10} + x^8 - x^6 - x^5 + x^4 + x^3 - x^2 + x$$
$$m10 := -x^{10} - x^9 - x^8 - x^7 - x^6 - x^5 - x^4 + x^3 + x^2 - 1$$

Die kurzlebigen Schlüssel $r_1, r_2, \ldots, r_{10}$ aus $\mathcal{T}(2, 2, 11)$:

```
> r1:=x^10+x^8-x^6-x^4;
> r2:=x^9-x^7-x^4+x^3;
> r3:=-x^9+x^5+x^3-1;
> r4:=-x^8+x^3-x^2+x;
> r5:=x^7-x^5+x^4-x^3;
> r6:=x^6+x^5-x^3-x;
> r7:=-x^5+x^3-x+1;
> r8:=-x^4-x^3+x^2+1;
> r9:=x^4+x^2-x-1;
> r10:=x^3-x^2-x+1;
```

$$r1 := x^{10} + x^8 - x^6 - x^4$$
$$r2 := x^9 - x^7 - x^4 + x^3$$
$$r3 := -x^9 + x^5 + x^3 - 1$$
$$r4 := -x^8 + x^3 - x^2 + x$$
$$r5 := x^7 - x^5 + x^4 - x^3$$
$$r6 := x^6 + x^5 - x^3 - x$$
$$r7 := -x^5 + x^3 - x + 1$$
$$r8 := -x^4 - x^3 + x^2 + 1$$
$$r9 := x^4 + x^2 - x - 1$$
$$r10 := x^3 - x^2 - x + 1$$

Für die Verschlüsselung des Textes wird folgendes berechnet:

```
> e1:=expand((3*r1)*h+m1) mod 41;
> e2:=expand((3*r2)*h+m2) mod 41;
> e3:=expand((3*r3)*h+m3) mod 41;
> e4:=expand((3*r4)*h+m4) mod 41;
> e5:=expand((3*r5)*h+m5) mod 41;
> e6:=expand((3*r6)*h+m6) mod 41;
> e7:=expand((3*r7)*h+m7) mod 41;
> e8:=expand((3*r8)*h+m8) mod 41;
> e9:=expand((3*r9)*h+m9) mod 41;
> e10:=expand((3*r10)*h+m10) mod 41;
```

$$e1 := 40 + 40\,x + 34\,x^9 + 3\,x^7 + x^2 + 37\,x^6 + 37\,x^4 + 40\,x^3 + 35\,x^{11} + 7\,x^{10} + 2\,x^8 + 3\,x^5 + 6\,x^{13} + 6\,x^{15} + 35\,x^{14} + 38\,x^{17} + 38\,x^{19} + 3\,x^{18}$$

$$e2 := 38\,x^{18} + 35\,x^{15} + 37\,x^9 + 3\,x^{17} + 3\,x^{14} + 3\,x^{16} + 6\,x^{13} + 38\,x^7 + 32\,x^{12} + 7\,x^8 + 35\,x^4 + 3\,x^5 + 4\,x^3 + 3\,x^{11} + x^{10} + 40\,x^6 + 40\,x + 1$$

$$e3 := 3\,x^{18} + 3\,x^{15} + 37\,x^9 + 38\,x^{17} + 35\,x^{14} + 7\,x^{10} + 3\,x^{13} + 38\,x^{12} + 2\,x^3 + 38\,x^4 + 37 + 2\,x + 40\,x^8 + 40\,x^7 + 40\,x^6 + x^5 + 40\,x^2$$

$$e4 := 3\,x^{17} + 3\,x^{14} + 4\,x^8 + 38\,x^{16} + 38\,x^{13} + 2\,x^9 + 38\,x^{12} + 6\,x^3 + 6\,x^{11} + 37\,x^4 + 36\,x^2 + 34\,x^{10} + 34\,x^7 + 3\,x + 4\,x^6 + 40\,x^5 + 40$$

$$e5 := 38\,x^{16} + 32\,x^{13} + 3\,x^7 + 3\,x^{15} + 9\,x^{12} + 35\,x^8 + 3\,x^{14} + 36\,x^5 + 34\,x^{10} + 2\,x^6 + 6\,x^4 + 7\,x^9 + 38\,x^3 + x + 40$$

$$e6 := 38\,x^{15} + 38\,x^6 + 38\,x^{11} + 3\,x^5 + 3\,x^{13} + 6\,x^{10} + 40\,x^9 + 37\,x^3 + 38\,x^8 + 3\,x^4 + 37\,x + 2\,x^2 + x^7 + 40$$

$$e7 := 3\,x^{14} + 6\,x^{11} + 38\,x^{13} + x^{10} + 38\,x^6 + 38\,x^{12} + 32\,x^9 + 2\,x^3 + 6\,x^8 + 37\,x^4 + 3\,x^7 + 34\,x + 3\,x^2 + 2 + 40\,x^5$$

$$e8 := 3\,x^{13} + 6\,x^{10} + 40\,x^4 + 39\,x^9 + 6\,x^5 + 34\,x^3 + 35\,x^{11} + 38\,x^8 + 3\,x^2 + 2\,x^7 + 38\,x^6 + 2 + 38\,x$$

$$e9 := 38\,x^{13} + 2\,x^{10} + 4\,x^4 + 3\,x^{12} + 3\,x^9 + 34\,x^5 + 38\,x^{11} + 36\,x^8 + 5\,x^2 + 6\,x^7 + 39\,x^3 + x + 40\,x^6 + 38$$

$$e10 := 38\,x^{12} + 31\,x^9 + 7\,x^3 + 6\,x^{11} + 8\,x^8 + 37\,x^4 + x^2 + 40\,x^{10} + 40\,x^7 + 35\,x + 34\,x^6 + 2 + 2\,x^5$$

Die Polynome e_i werden von Hand im Ring der Konvolutionspolynome $\mathcal{R}^{11}$ dargestellt, für i=1,2,...,10. Anschließend modulo 41 gerechnet, damit die Polynome im Ring $\mathcal{R}^{11}_{41}$ liegen:

```
> e1:=40+40*x+34*x^9+3*x^7+x^2+37*x^6+37*x^4+40*x^3+35+7*x^10+2*x^8+3*x^
> 5+6*x^2+6*x^4+35*x^3+38*x^6+38*x^8+3*x^7 mod 41;
> e2:=38*x^7+35*x^4+37*x^9+3*x^6+3*x^3+3*x^5+6*x^2+38*x^7+32*x+7*x^8+35*
> x^4+3*x^5+4*x^3+3+x^10+40*x^6+40*x+1 mod 41;
> e3:=3*x^7+3*x^4+37*x^9+38*x^6+35*x^3+7*x^10+3*x^2+38*x+2*x^3+38*x^4+37
> +2*x+40*x^8+40*x^7+40*x^6+x^5+40*x^2 mod 41;
> e4:=3*x^6+3*x^3+4*x^8+38*x^5+38*x^2+2*x^9+38*x+6*x^3+6+37*x^4+36*x^2+3
> 4*x^10+34*x^7+3*x+4*x^6+40*x^5+40 mod 41;
> e5:=38*x^5+32*x^2+3*x^7+3*x^4+9*x+35*x^8+3*x^3+36*x^5+34*x^10+2*x^6+6*
> x^4+7*x^9+38*x^3+x+40 mod 41;
> e6:=38*x^4+38*x^6+38+3*x^5+3*x^2+6*x^10+40*x^9+37*x^3+38*x^8+3*x^4+37*
> x+2*x^2+x^7+40 mod 41;
> e7:=3*x^3+6+38*x^2+x^10+38*x^6+38*x+32*x^9+2*x^3+6*x^8+37*x^4+3*x^7+34
> *x+3*x^2+2+40*x^5 mod 41;
> e8:=3*x^2+6*x^10+40*x^4+39*x^9+6*x^5+34*x^3+35+38*x^8+3*x^2+2*x^7+38*x
> ^6+2+38*x mod 41;
> e9:=38*x^2+2*x^10+4*x^4+3*x+3*x^9+34*x^5+38+36*x^8+5*x^2+6*x^7+39*x^3+
> x+40*x^6+38 mod 41;
> e10:=38*x+31*x^9+7*x^3+6+8*x^8+37*x^4+x^2+40*x^10+40*x^7+35*x+34*x^6+2
> +2*x^5 mod 41;
```

$$e1 := 34 + 40\,x + 34\,x^9 + 6\,x^7 + 7\,x^2 + 34\,x^6 + 2\,x^4 + 34\,x^3 + 7\,x^{10} + 40\,x^8 + 3\,x^5$$

$$e2 := 35\,x^7 + 29\,x^4 + 37\,x^9 + 2\,x^6 + 7\,x^3 + 6\,x^5 + 6\,x^2 + 31\,x + 7\,x^8 + 4 + x^{10}$$

$$e3 := 2\,x^7 + 37\,x^9 + 37\,x^6 + 37\,x^3 + 7\,x^{10} + 2\,x^2 + 40\,x + 37 + 40\,x^8 + x^5$$

$$e4 := 7\,x^6 + 9\,x^3 + 4\,x^8 + 37\,x^5 + 33\,x^2 + 2\,x^9 + 5 + 37\,x^4 + 34\,x^{10} + 34\,x^7$$

$$e5 := 33\,x^5 + 32\,x^2 + 3\,x^7 + 9\,x^4 + 10\,x + 35\,x^8 + 34\,x^{10} + 2\,x^6 + 7\,x^9 + 40$$
$$e6 := 38\,x^6 + 37 + 3\,x^5 + 5\,x^2 + 6\,x^{10} + 40\,x^9 + 37\,x^3 + 38\,x^8 + 37\,x + x^7$$
$$e7 := 5\,x^3 + 8 + x^{10} + 38\,x^6 + 31\,x + 32\,x^9 + 6\,x^8 + 37\,x^4 + 3\,x^7 + 40\,x^5$$
$$e8 := 6\,x^2 + 6\,x^{10} + 40\,x^4 + 39\,x^9 + 6\,x^5 + 34\,x^3 + 37 + 38\,x^8 + 2\,x^7 + 38\,x^6 + 38\,x$$
$$e9 := 2\,x^2 + 2\,x^{10} + 4\,x^4 + 4\,x + 3\,x^9 + 34\,x^5 + 35 + 36\,x^8 + 6\,x^7 + 39\,x^3 + 40\,x^6$$
$$e10 := 32\,x + 31\,x^9 + 7\,x^3 + 8 + 8\,x^8 + 37\,x^4 + x^2 + 40\,x^{10} + 40\,x^7 + 34\,x^6 + 2\,x^5$$

Für die Entschlüsselung rechnen wir:

```
> a1:=expand(f*e1) mod 41;
> a2:=expand(f*e2) mod 41;
> a3:=expand(f*e3) mod 41;
> a4:=expand(f*e4) mod 41;
> a5:=expand(f*e5) mod 41;
> a6:=expand(f*e6) mod 41;
> a7:=expand(f*e7) mod 41;
> a8:=expand(f*e8) mod 41;
> a9:=expand(f*e9) mod 41;
> a10:=expand(f*e10) mod 41;
```

$$a1 := 34\,x + 3\,x^9 + 37\,x^7 + 40\,x^2 + 37\,x^6 + 40\,x^4 + 40\,x^{11} + 38\,x^{10} + 40\,x^8 + 10\,x^5 + x^{13} + 4\,x^{12} + 6\,x^{15} + 7\,x^{17} + 27\,x^{16}$$

$$a2 := 4\,x + 2\,x^9 + 38\,x^7 + 31\,x^2 + 3\,x^6 + 34\,x^4 + 10\,x^3 + 37\,x^{11} + 7\,x^{10} + 37\,x^8 + 4\,x^5 + 13\,x^{13} + 34\,x^{12} + 11\,x^{15} + 27\,x^{14} + x^{17} + 36\,x^{16}$$

$$a3 := 37\,x + 38\,x^7 + 40\,x^2 + 40\,x^6 + 40\,x^4 + 39\,x^3 + 3\,x^{11} + 39\,x^{10} + 3\,x^5 + 5\,x^{13} + 2\,x^{12} + 3\,x^{15} + 37\,x^{14} + 7\,x^{17} + 30\,x^{16}$$

$$a4 := 5\,x + 39\,x^9 + 40\,x^7 + 8\,x^6 + 4\,x^4 + 38\,x^3 + 4\,x^{11} + x^{10} + x^8 + 29\,x^5 + 5\,x^{13} + 28\,x^{12} + 2\,x^{15} + 37\,x^{14} + 34\,x^{17} + 9\,x^{16}$$

$$a5 := 40\,x + 36\,x^9 + 10\,x^2 + 2\,x^6 + 11\,x^4 + 31\,x^3 + x^{11} + 40\,x^{10} + 5\,x^8 + 31\,x^5 + 26\,x^{13} + 3\,x^{12} + 28\,x^{15} + 16\,x^{14} + 34\,x^{17} + 14\,x^{16}$$

$$a6 := 37\,x + x^7 + 37\,x^2 + 39\,x^6 + 37\,x^4 + x^3 + 40\,x^{11} + 40\,x^{10} + 36\,x^8 + 9\,x^5 + 3\,x^{13} + 8\,x^{12} + 39\,x^{15} + 39\,x^{14} + 6\,x^{17} + 34\,x^{16}$$

$$a7 := 8\,x + 40\,x^9 + 6\,x^7 + 31\,x^2 + 37\,x^6 + 28\,x^4 + 8\,x^3 + x^{11} + 6\,x^{10} + 37\,x^8 + 6\,x^5 + 4\,x^{13} + 28\,x^{12} + 15\,x^{15} + 37\,x^{14} + x^{17} + 31\,x^{16}$$

$$a8 := 37\,x + x^9 + 2\,x^7 + 38\,x^2 + 38\,x^6 + 35\,x^4 + 2\,x^3 + 35\,x^{11} + 38\,x^{10} + 8\,x^5 + 3\,x^{13} + 10\,x^{12} + 40\,x^{15} + 40\,x^{14} + 6\,x^{17} + 33\,x^{16}$$

$$a9 := 35\,x + 5\,x^9 + 36\,x^7 + 4\,x^2 + 36\,x^6 + 8\,x^4 + 37\,x^3 + 2\,x^{11} + 4\,x^{10} + 38\,x^8 + 2\,x^5 + 33\,x^{13} + 2\,x^{12} + 33\,x^{15} + 9\,x^{14} + 2\,x^{17} + x^{16}$$

$$a10 := 8\,x + 34\,x^9 + 40\,x^7 + 32\,x^2 + 31\,x^4 + 9\,x^3 + 2\,x^{11} + 7\,x^{10} + 36\,x^8 + 6\,x^5 + 3\,x^{13} + 32\,x^{12} + 18\,x^{15} + 33\,x^{14} + 40\,x^{17} + 32\,x^{16}$$

Die Polynome a_i werden von Hand im Ring der Konvolutionspolynome $\mathcal{R}^{11}$ dargestellt, für i=1,2,...,10. Anschließend modulo 41 gerechnet, damit die Polynome im Ring $\mathcal{R}_{41}^{11}$ liegen:

```
> a1:=
> 34*x+3*x^9+37*x^7+40*x^2+37*x^6+40*x^4+40+38*x^10+40*x^8+10*x^5+x^2+4*
> x+6*x^4+7*x^6+27*x^5 mod 41;
> a2:=4*x+2*x^9+38*x^7+31*x^2+7*x^10+37*x^8+4*x^5+3*x^6+34*x^4+10*x^3+37
> +13*x^2+34*x+11*x^4+27*x^3+x^6+36*x^5 mod 41;
> a3:=37*x+38*x^7+40*x^2+39*x^10+3*x^5+40*x^6+40*x^4+39*x^3+3+5*x^2+2*x+
> 3*x^4+37*x^3+7*x^6+30*x^5 mod 41;
> a4:=5*x+39*x^9+40*x^7+8*x^6+4*x^4+38*x^3+4+x^10+x^8+29*x^5+5*x^2+28*x+
> 2*x^4+37*x^3+34*x^6+9*x^5 mod 41;
> a5:=40*x+36*x^9+10*x^2+2*x^6+11*x^4+31*x^3+1+40*x^10+5*x^8+31*x^5+26*x
> ^2+3*x+28*x^4+16*x^3+34*x^6+14*x^5 mod 41;
> a6:=37*x+x^7+37*x^2+39*x^6+37*x^4+x^3+40+40*x^10+36*x^8+9*x^5+3*x^2+8*
> x+39*x^4+39*x^3+6*x^6+34*x^5 mod 41;
> a7:=8*x+40*x^9+6*x^7+31*x^2+37*x^6+28*x^4+8*x^3+1+6*x^10+37*x^8+6*x^5+
> 4*x^2+28*x+15*x^4+37*x^3+x^6+31*x^5 mod 41;
> a8:=37*x+x^9+2*x^7+38*x^2+38*x^6+35*x^4+2*x^3+35+38*x^10+8*x^5+3*x^2+1
> 0*x+40*x^4+40*x^3+6*x^6+33*x^5 mod 41;
> a9:=35*x+5*x^9+36*x^7+4*x^2+36*x^6+8*x^4+37*x^3+2+4*x^10+38*x^8+2*x^5+
> 33*x^2+2*x+33*x^4+9*x^3+2*x^6+x^5 mod 41;
> a10:=8*x+34*x^9+40*x^7+32*x^2+31*x^4+9*x^3+2+7*x^10+36*x^8+6*x^5+3*x^2
> +32*x+18*x^4+33*x^3+40*x^6+32*x^5 mod 41;
```

$$a1 := 38\,x + 3\,x^9 + 37\,x^7 + 3\,x^6 + 5\,x^4 + 40 + 38\,x^{10} + 40\,x^8 + 37\,x^5$$
$$a2 := 38\,x + 2\,x^9 + 38\,x^7 + 3\,x^2 + 7\,x^{10} + 37\,x^8 + 40\,x^5 + 4\,x^6 + 4\,x^4 + 37\,x^3 + 37$$
$$a3 := 39\,x + 38\,x^7 + 4\,x^2 + 39\,x^{10} + 33\,x^5 + 6\,x^6 + 2\,x^4 + 35\,x^3 + 3$$
$$a4 := 33\,x + 39\,x^9 + 40\,x^7 + x^6 + 6\,x^4 + 34\,x^3 + 4 + x^{10} + x^8 + 38\,x^5 + 5\,x^2$$
$$a5 := 2\,x + 36\,x^9 + 36\,x^2 + 36\,x^6 + 39\,x^4 + 6\,x^3 + 1 + 40\,x^{10} + 5\,x^8 + 4\,x^5$$
$$a6 := 4\,x + x^7 + 40\,x^2 + 4\,x^6 + 35\,x^4 + 40\,x^3 + 40 + 40\,x^{10} + 36\,x^8 + 2\,x^5$$
$$a7 := 36\,x + 40\,x^9 + 6\,x^7 + 35\,x^2 + 38\,x^6 + 2\,x^4 + 4\,x^3 + 1 + 6\,x^{10} + 37\,x^8 + 37\,x^5$$
$$a8 := 6\,x + x^9 + 2\,x^7 + 3\,x^6 + 34\,x^4 + x^3 + 35 + 38\,x^{10}$$
$$a9 := 37\,x + 5\,x^9 + 36\,x^7 + 37\,x^2 + 38\,x^6 + 5\,x^3 + 2 + 4\,x^{10} + 38\,x^8 + 3\,x^5$$
$$a10 := 40\,x + 34\,x^9 + 40\,x^7 + 35\,x^2 + 8\,x^4 + x^3 + 2 + 7\,x^{10} + 36\,x^8 + 38\,x^5 + 40\,x^6$$

Wir führen für die Polynome a_i den zentraler Lift von $\mathcal{R}_{41}^{11}$ nach $\mathcal{R}^{11}$ per Hand durch:

```
> a1:=-3*x+3*x^9-4*x^7+3*x^6+5*x^4-1-3*x^10-1*x^8-4*x^5;
> a2:=-3*x+2*x^9-3*x^7+3*x^2+7*x^10-4*x^8-x^5+4*x^6+4*x^4-4*x^3-4;
> a3:=-2*x-3*x^7+4*x^2-2*x^10-8*x^5+6*x^6+2*x^4-6*x^3+3;
> a4:=-8*x-2*x^9-x^7+x^6+6*x^4-7*x^3+4+x^10+x^8-3*x^5+5*x^2;
> a5:=2*x-5*x^9-5*x^2-5*x^6-2*x^4+6*x^3+1-1*x^10+5*x^8+4*x^5;
> a6:=4*x+x^7-1*x^2+4*x^6-6*x^4-1*x^3-1-1*x^10-5*x^8+2*x^5;
> a7:=-5*x-1*x^9+6*x^7-6*x^2-3*x^6+2*x^4+4*x^3+1+6*x^10-4*x^8-4*x^5;
> a8:=6*x+x^9+2*x^7+3*x^6-7*x^4+x^3-6-3*x^10;
> a9:=-4*x+5*x^9-5*x^7-4*x^2-3*x^6+5*x^3+2+4*x^10-3*x^8+3*x^5;
> a10:=-x-7*x^9-x^7-6*x^2+8*x^4+x^3+2+7*x^10-5*x^8-3*x^5-x^6;
```

$$a1 := -3\,x + 3\,x^9 - 4\,x^7 + 3\,x^6 + 5\,x^4 - 1 - 3\,x^{10} - x^8 - 4\,x^5$$

$$a2 := -3\,x + 2\,x^9 - 3\,x^7 + 3\,x^2 + 7\,x^{10} - 4\,x^8 - x^5 + 4\,x^6 + 4\,x^4 - 4\,x^3 - 4$$
$$a3 := -2\,x - 3\,x^7 + 4\,x^2 - 2\,x^{10} - 8\,x^5 + 6\,x^6 + 2\,x^4 - 6\,x^3 + 3$$
$$a4 := -8\,x - 2\,x^9 - x^7 + x^6 + 6\,x^4 - 7\,x^3 + 4 + x^{10} + x^8 - 3\,x^5 + 5\,x^2$$
$$a5 := 2\,x - 5\,x^9 - 5\,x^2 - 5\,x^6 - 2\,x^4 + 6\,x^3 + 1 - x^{10} + 5\,x^8 + 4\,x^5$$
$$a6 := 4\,x + x^7 - x^2 + 4\,x^6 - 6\,x^4 - x^3 - 1 - x^{10} - 5\,x^8 + 2\,x^5$$
$$a7 := -5\,x - x^9 + 6\,x^7 - 6\,x^2 - 3\,x^6 + 2\,x^4 + 4\,x^3 + 1 + 6\,x^{10} - 4\,x^8 - 4\,x^5$$
$$a8 := 6\,x + x^9 + 2\,x^7 + 3\,x^6 - 7\,x^4 + x^3 - 6 - 3\,x^{10}$$
$$a9 := -4\,x + 5\,x^9 - 5\,x^7 - 4\,x^2 - 3\,x^6 + 5\,x^3 + 2 + 4\,x^{10} - 3\,x^8 + 3\,x^5$$
$$a10 := -x - 7\,x^9 - x^7 - 6\,x^2 + 8\,x^4 + x^3 + 2 + 7\,x^{10} - 5\,x^8 - 3\,x^5 - x^6$$

Berechne $b_i = F_p * a_i \pmod 3$, für i=1,2,...,10:

```
> b1:=expand(Fp*a1) mod 3;
> b2:=expand(Fp*a2) mod 3;
> b3:=expand(Fp*a3) mod 3;
> b4:=expand(Fp*a4) mod 3;
> b5:=expand(Fp*a5) mod 3;
> b6:=expand(Fp*a6) mod 3;
> b7:=expand(Fp*a7) mod 3;
> b8:=expand(Fp*a8) mod 3;
> b9:=expand(Fp*a9) mod 3;
> b10:=expand(Fp*a10) mod 3;
```

$$b1 := 2 + 2\,x + 2\,x^9 + x^7 + x^6 + x^4 + x^3 + x^{10} + 2\,x^8 + x^{13} + x^{15} + x^{14} + x^{17} + 2\,x^{18}$$
$$b2 := 2 + 2\,x + x^9 + 2\,x^4 + 2\,x^{11} + x^{10} + x^8 + 2\,x^5 + x^{15} + x^{14} + 2\,x^{17} + x^{16} + x^{20}$$

$$b3 := x + x^9 + 2\,x^2 + x^4 + x^3 + 2\,x^{11} + x^{10} + x^8 + x^{12} + 2\,x^{15} + x^{14} + 2\,x^{17} + x^{16} + x^{19} + 2\,x^{18} + x^{20}$$

$$b4 := 1 + 2\,x + x^9 + x^7 + 2\,x^4 + x^{11} + 2\,x^{10} + 2\,x^8 + x^{13} + x^{12} + x^{17} + 2\,x^{16} + 2\,x^{19} + x^{18} + x^{20}$$
$$b5 := 1 + 2\,x^9 + 2\,x^7 + x^{11} + 2\,x^{10} + x^5 + x^{12} + 2\,x^{17} + x^{18} + 2\,x^{20}$$
$$b6 := 2 + 2\,x^7 + 2\,x^3 + x^8 + 2\,x^{13} + 2\,x^{12} + 2\,x^{19} + 2\,x^{18} + 2\,x^{20}$$
$$b7 := 1 + 2\,x + 2\,x^7 + x^2 + x^{11} + x^{10} + x^8 + 2\,x^{13} + 2\,x^{15} + 2\,x^{14} + 2\,x^{16} + 2\,x^{19} + x^{18}$$
$$b8 := x^9 + x^7 + 2\,x^6 + x^3 + 2\,x^{11} + 2\,x^8 + 2\,x^5 + 2\,x^{15} + x^{14} + x^{17} + x^{16} + x^{19} + x^{18}$$

$$b9 := 2 + x + 2\,x^9 + 2\,x^7 + x^2 + x^6 + 2\,x^4 + 2\,x^3 + x^{11} + 2\,x^{10} + x^8 + 2\,x^5 + x^{13} + 2\,x^{15} + 2\,x^{14} + x^{17} + x^{18} + x^{20}$$

$$b10 := 2 + x + x^9 + 2\,x^2 + 2\,x^6 + 2\,x^3 + 2\,x^{10} + 2\,x^8 + 2\,x^{13} + 2\,x^{12} + 2\,x^{15} + 2\,x^{14} + 2\,x^{16} + 2\,x^{18} + x^{20}$$

Die Polynome b_i werden von Hand im Ring der Konvolutionspolynome $\mathcal{R}^{11}$ dargestellt, für i=1,2,...,10. Anschließend modulo 3 gerechnet, damit die Polynome im Ring $\mathcal{R}_3^{11}$ liegen:

```
> b1:=2+2*x+2*x^9+x^7+x^6+x^4+x^3+x^10+2*x^8+x^2+x^4+x^3+x^6+2*x^7 mod
> 3;
> b2:=2+2*x+x^9+x^10+x^8+2*x^5+2*x^4+2+x^9+x^4+x^3+2*x^6+x^5 mod 3;
```

```
> b3:=x+x^8+2*x^7+x^9+2*x^2+x^10+x^8+x^4+x^3+2+x^9+x+2*x^4+x^3+2*x^6+x^
> 5 mod 3;
> b4:=1+2*x+x^9+x^7+2*x^4+1+2*x^10+2*x^8+x^2+x+x^6+2*x^5+2*x^8+x^7+x^9
> mod 3;
> b5:=1+2*x^9+2*x^7+1+2*x^10+x^5+x+2*x^6+x^7+2*x^9 mod 3;
> b6:=2+2*x^7+2*x^3+x^8+2*x^2+2*x+2*x^8+2*x^7+2*x^9 mod 3;
> b7:=1+2*x+2*x^7+x^2+1+x^10+x^8+2*x^2+2*x^4+2*x^3+2*x^5+2*x^8+x^7 mod
> 3;
> b8:=x^9+x^7+2*x^6+x^3+2+2*x^8+2*x^5+2*x^4+x^3+x^6+x^5+x^8+x^7 mod 3;
> b9:=2+x+2*x^9+2*x^7+x^2+x^6+2*x^4+2*x^3+1+2*x^10+x^8+2*x^5+x^2+2*x^4+2
> *x^3+x^6+x^7+x^9 mod 3;
> b10:=2+x+x^9+2*x^2+2*x^6+2*x^3+2*x^10+2*x^8+2*x^2+2*x+2*x^4+2*x^3+2*x^
> 5+2*x^7+x^9 mod 3;
```

$$b1 := 2 + 2\,x + 2\,x^9 + 2\,x^6 + 2\,x^4 + 2\,x^3 + x^{10} + 2\,x^8 + x^2$$

$$b2 := 1 + 2\,x + 2\,x^9 + x^{10} + x^8 + x^3 + 2\,x^6$$

$$b3 := 2\,x + 2\,x^8 + 2\,x^7 + 2\,x^9 + 2\,x^2 + x^{10} + 2\,x^3 + 2 + 2\,x^6 + x^5$$

$$b4 := 2 + 2\,x^9 + 2\,x^7 + 2\,x^4 + 2\,x^{10} + x^8 + x^2 + x^6 + 2\,x^5$$

$$b5 := 2 + x^9 + 2\,x^{10} + x^5 + x + 2\,x^6$$

$$b6 := 2 + x^7 + 2\,x^3 + 2\,x^2 + 2\,x + 2\,x^9$$

$$b7 := 2 + 2\,x + x^{10} + 2\,x^4 + 2\,x^3 + 2\,x^5$$

$$b8 := x^9 + 2\,x^7 + 2\,x^3 + 2 + 2\,x^4$$

$$b9 := x + 2\,x^2 + 2\,x^6 + x^4 + x^3 + 2\,x^{10} + x^8 + 2\,x^5$$

$$b10 := 2 + 2\,x^9 + x^2 + 2\,x^6 + x^3 + 2\,x^{10} + 2\,x^8 + 2\,x^4 + 2\,x^5 + 2\,x^7$$

Führe für die Polynome b_i den zentraler Lift von $\mathcal{R}_3^{11}$ nach $\mathcal{R}^{11}$ per Hand durch:

```
> n1:=x^10-x^9-x^8-x^6-x^4-x^3+x^2-x-1;
> n2:=x^10-x^9+x^8-x^6+x^3-x+1;
> n3:=x^10-x^9-x^8-x^7-x^6+x^5-x^3-x^2-x-1;
> n4:=-x^10-x^9+x^8-x^7+x^6-x^5-x^4+x^2-1;
> n5:=-x^10+x^9-x^6+x^5+x-1;
> n6:=-x^9+x^7-x^3-x^2-x-1;
> n7:=x^10-x^5-x^4-x^3-x-1;
> n8:=x^9-x^7-x^4-x^3-1;
> n9:=-x^10+x^8-x^6-x^5+x^4+x^3-x^2+x;
> n10:=-x^10-x^9-x^8-x^7-x^6-x^5-x^4+x^3+x^2-1;
```

$$n1 := x^{10} - x^9 - x^8 - x^6 - x^4 - x^3 + x^2 - x - 1$$

$$n2 := x^{10} - x^9 + x^8 - x^6 + x^3 - x + 1$$

$$n3 := x^{10} - x^9 - x^8 - x^7 - x^6 + x^5 - x^3 - x^2 - x - 1$$

$$n4 := -x^{10} - x^9 + x^8 - x^7 + x^6 - x^5 - x^4 + x^2 - 1$$

$$n5 := -x^{10} + x^9 - x^6 + x^5 + x - 1$$

$$n6 := -x^9 + x^7 - x^3 - x^2 - x - 1$$

$$n7 := x^{10} - x^5 - x^4 - x^3 - x - 1$$

$$n8 := x^9 - x^7 - x^4 - x^3 - 1$$

$$n9 := -x^{10} + x^8 - x^6 - x^5 + x^4 + x^3 - x^2 + x$$

$$n10 := -x^{10} - x^9 - x^8 - x^7 - x^6 - x^5 - x^4 + x^3 + x^2 - 1$$