KAPITEL 4

NTRU

Entwickelt wurde die im Folgenden beschriebene NTRU-Verschlüsselung von den Mathematikern Jeffrey Hoffstein, Jill Pipher und Joseph Silverman in den Jahren 1994-1996. Zusammen mit Daniel Lieman gründeten sie 1996 die Firma NTRU Cryptosystems, Inc.. Es gibt ein paar Erklärungsversuche für die Bezeichnung NTRU:

- **N**-th degree **tru**ncated polynomial ring
- **N**umber **T**theory **R**esearch **U**nit
- **N**umber **T**heorits a**R**e **U**s

Es gibt verschiedene Varianten von NTRU, auch Internetseiten zu NTRU kommen und gehen.

- Zwei NTRU-Varianten sind unter den Kandidaten der 3. Runde für einen Post-Quanten-Kryptographie-Standard von NIST: NTRU und NTRU Prime.
- Informationen finden man auf der englischen Wikipedia-Seite NTRU.
- Die Wikipedia-Seite NTRUEncrypt beschreibt NTRU ähnlich wie es in diesem Kapitel geschieht. Die Seite enthält auch Vorschläge für aktuelle (2020) Schlüsselparameter.

1. $\mathbb{R}^N$ mit dem Konvolutionsprodukt

Wir beginnen mit dem Polynomring

$$\mathbb{R}[X] = \{a_0 + a_1X + a_2X^2 + \cdots + a_nX^n : n \in \mathbb{N}_0, a_i \in \mathbb{R}\}.$$

Gibt man sich $N \in \mathbb{N}$ vor und rechnet man modulo $X^N - 1$, so erhält man den Ring

$$R = \mathbb{R}[X]/(X^N - 1).$$

Wir schreiben x für das Bild von X in R und haben dann $x^N = 1$ und

$$R = \{a_0 + a_1x + a_2x^2 + \cdots + a_{N-1}x^{N-1} : a_i \in \mathbb{R}\}.$$

(Aus $x^N = 1$ folgt sofort $x^m = x^n$ für alle $m, n \in \mathbb{N}_0$ mit $m \equiv n \bmod N$.) R ist ein N-dimensionaler $\mathbb{R}$-Vektorraum mit Basis $1, x, \ldots, x^{N-1}$. Wir erhalten einen $\mathbb{R}$-Vektorraum-Isomorphismus

$$\Phi : R \to \mathbb{R}^N, \quad a_0 + a_1x + \cdots + a_{N-1}x^{N-1} \mapsto (a_0, a_1, \ldots, a_{N-1})$$

mit der Umkehrabbildung

$$\Psi : \mathbb{R}^N \to R, \quad (a_0, a_1, \ldots, a_{N-1}) \mapsto a_0 + a_1x + \cdots + a_{N-1}x^{N-1}.$$

Lemma. *Für die Multiplikation in R gilt*

$$\sum_{i=0}^{N-1} a_ix^i \cdot \sum_{j=0}^{N-1} b_jx^j = \sum_{k=0}^{N-1} \left(\sum_{i=0}^{k} a_ib_{k-i} + \sum_{i=k+1}^{N-1} a_ib_{N+k-i} \right) x^k.$$

Datei: kii_ntru.tex. Version vom 21.7.2022

Beweis:

$$\begin{aligned}
\sum_{i=0}^{N-1} a_i x^i \cdot \sum_{j=0}^{N-1} b_j x^j &= \sum_{\substack{0\le i\le N-1\\ 0\le j\le N-1}} a_i b_j x^{i+j} = \sum_{k=0}^{2N-2} \Big(\sum_{\substack{0\le i\le N-1\\ 0\le j\le N-1\\ i+j=k}} a_i b_j \Big) x^k = \\
&= \sum_{k=0}^{2N-2} \Big(\sum_{\substack{0\le i\le N-1\\ 0\le k-i\le N-1}} a_i b_{k-i} \Big) x^k = \sum_{k=0}^{2N-2} \Big(\sum_{\substack{\max(0,k-N+1)\le i\\ i\le \min(N-1,k)}} a_i b_{k-i} \Big) x^k = \\
&= \sum_{k=0}^{N-1} \Big(\sum_{\substack{\max(0,k-N+1)\le i\\ i\le \min(N-1,k)}} a_i b_{k-i} \Big) x^k + \sum_{k=N}^{2N-2} \Big(\sum_{\substack{\max(0,k-N+1)\le i\\ i\le \min(N-1,k)}} a_i b_{k-i} \Big) x^k = \\
&= \sum_{k=0}^{N-1} \Big(\sum_{0\le i\le k} a_i b_{k-i} \Big) x^k + \sum_{k=N}^{2N-2} \Big(\sum_{k-N+1\le i\le N-1} a_i b_{k-i} \Big) x^k \overset{k=l+N}{=} \\
&= \sum_{k=0}^{N-1} \Big(\sum_{0\le i\le k} a_i b_{k-i} \Big) x^k + \sum_{l=0}^{N-2} \Big(\sum_{l+1\le i\le N-1} a_i b_{N+l-i} \Big) x^{N+l} \overset{x^N=1}{=} \\
&= \sum_{k=0}^{N-1} \Big(\sum_{i=0}^{k} a_i b_{k-i} \Big) x^k + \sum_{k=0}^{N-2} \Big(\sum_{i=k+1}^{N-1} a_i b_{N+k-i} \Big) x^k = \\
&= \sum_{k=0}^{N-1} \Big(\sum_{i=0}^{k} a_i b_{k-i} + \sum_{i=k+1}^{N-1} a_i b_{N+k-i} \Big) x^k,
\end{aligned}$$

was zu zeigen war. ■

Vermöge der Isomorphismen Φ und Ψ definieren wir jetzt ein Multiplikation auf $\mathbb{R}^N$, das Konvolutionsprodukt:

$$a * b = \Phi(\Psi(a) \cdot \Psi(b)).$$

Damit wird $\mathbb{R}^N$ zu einem kommutativen Ring mit 1. Mit den Formeln des vorangegangenen Lemmas gilt also

$$(a_0, \dots, a_{N-1}) * (b_0, \dots, b_{N-1}) = (c_0, \dots, c_{N-1}) \text{ mit } c_k = \sum_{i=0}^{k} a_i b_{k-i} + \sum_{i=k+1}^{N-1} a_i b_{N+k-i}.$$

Beispiele:

(1) Wir wollen für $a = (1, 2, 3)$ und $b = (4, 5, 6)$ das $*$-Produkt in $\mathbb{R}^3$ berechnen. Wir berechnen $\Psi(a) \cdot \Psi(b)$ in R:

$$\Psi(a) \cdot \Psi(b) = (1 + 2x + 3x^2)(4 + 5x + 6x^2) = 4 + 13x + 28x^2 + 27x^3 + 18x^4.$$

Nun ist aber $x^3 = 1$ und $x^4 = x$, sodass wir erhalten

$$\Psi(a) \cdot \Psi(b) = 31 + 31x + 28x^2.$$

Daher gilt

$$(1, 2, 3) * (4, 5, 6) = (31, 31, 28).$$

(2) Für $a, b_0, \dots, b_{N-1} \in \mathbb{R}$ gilt

$$\begin{aligned}
(a, 0, \dots, 0) * (b_0, \dots, b_{N-1}) &= \Phi(a \cdot (b_0 + \cdots + b_{N-1}x^{N-1})) = \Phi(ab_0 + \cdots + ab_{N-1}x^{N-1}) = \\
&= (ab_0, \dots, ab_{N-1}) = a(b_0, \dots, b_{N-1}),
\end{aligned}$$

wobei $a(b_0, \dots, b_{N-1})$ die Multiplikation des Vektors $(b_0, \dots, b_{N-1})$ mit dem Skalar a ist.

(3) Wegen $(1, 0, \dots, 0) * (a_0, a_1, \dots, a_{N-1}) = (a_0, a_1, \dots, a_{N-1})$ spielt $(1, 0, \dots, 0)$ die Rolle der 1 im Ring $\mathbb{R}^N$. (Manchmal steht dafür auch kurz 1, wenn die Bedeutung klar ist.)

(4) Für $b_0, \ldots, b_{N-1} \in \mathbb{R}$ gilt.

$$\begin{aligned}(0,1,0,\ldots,0) * (b_0,\ldots,b_{N-1}) &= \Phi(x \cdot (b_0 + \cdots + b_{N-1}x^{N-1})) = \\ &= \Phi(b_0x + b_1x^2 + \cdots + b_{N-2}x^{N-1} + b_{N-1}x^N) = \\ &= \Phi(b_{N-1} + b_0x + b_1x^2 + \cdots + b_{N-2}x^{N-1}) = \\ &= (b_{N-1}, b_0, b_1, \ldots, b_{N-2}).\end{aligned}$$

Multiplikation mit $(0,1,0,\ldots,0)$ bewirkt also eine zyklische Rechtsverschiebung des Vektors $(b_0,\ldots,b_{N-1})$.

(5) Wir schreiben kurz σ für die zyklische Rechtsverschiebung um 1, also

$$\sigma : \mathbb{R}^N \to \mathbb{R}^N \text{ mit } \sigma((a_0, a_1, \ldots, a_{N-2}, a_{N-1})) = (a_{N-1}, a_0, a_1, \ldots, a_{N-2}).$$

Wendet man für $k \in \mathbb{N}$ die Abbildung σ nun k-fach an, d.h. bildet man $\sigma^k(a) = \underbrace{(\sigma \circ \cdots \circ \sigma)}_{k\text{mal}}(a)$, so hat man die zyklische Rechtsverschiebung um k. Dann ist klar, dass σ^N wieder die Identität ist. Was ist $\sigma^{N-1}(a)$? Schiebt man um 1 weiter, so erhält man $\sigma(\sigma^{N-1}(a)) = \sigma^N(a) = a$, also den Ausgangsvektor. Daher ist σ^{N-1} die zyklische Linksverschiebung um 1. Schreiben wir $s = (0,1,0,\ldots,0)$, so gilt nach dem zuvor Gezeigten $s * a = \sigma(a)$. Das können wir iterieren:

$$\begin{aligned}s * a &= \sigma(a), \\ s * s * a &= s * \sigma(a) = \sigma(\sigma(a)) = \sigma^2(a), \\ s * s * s * a &= s * \sigma^2(a) = \sigma^3(a), \\ &\vdots \quad\quad \vdots \\ \underbrace{s * \cdots * s}_{k \text{ mal}} * a &= \sigma^k(a).\end{aligned}$$

Es folgt

$$a * b = \sigma^N(a * b) = \underbrace{s * \cdots * s}_{N \text{ mal}} * a * b = s * a * \underbrace{s * \cdots * s}_{(N-1) \text{ mal}} * b = \sigma(a) * \sigma^{N-1}(b).$$

Schreibt man das aus, so erhält man

$$(a_0, a_1, \ldots, a_{N-1}) * (b_0, b_1, \ldots, b_{N-1}) = (a_{N-1}, a_0, a_1, \ldots, a_{N-2}) * (b_1, \ldots, b_{N-1}, b_0).$$

(6) Die Eigenschaft des Vektors $(0,1,0,\ldots,0)$ führt zu

$$(0,1,0,\ldots,0) * (1,1,\ldots,1) = (1,1,\ldots,1) = (1,0,\ldots,0) * (1,1,\ldots,1),$$

was dann

$$(1,-1,0,\ldots,0) * (1,1,1,\ldots,1) = 0$$

liefert. Für $N \geq 2$ gibt es also Nullteiler im Ring $\mathbb{R}^N$ mit der $*$-Multiplikation.

Bemerkung: In Python stellen wir die Elemente von $\mathbb{R}^N$ als Listen der Länge N dar. Die Multiplikation könnte dann so aussehen:

```
def ntru_mult(a,b):
  N=len(a)
  c=N*[0]
  for k in range(N):
    for i in range(k+1):
      c[k]+=a[i]*b[k-i]
    for i in range(k+1,N):
      c[k]+=a[i]*b[N+k-i]
  return c
```

Für $a = (a_0, a_1, \ldots, a_{N-1}) \in \mathbb{R}^N$ definieren wir eine $N \times N$-Matrix:

$$M(a) = \begin{pmatrix} a_0 & a_1 & a_2 & \ldots & a_{N-1} \\ a_{N-1} & a_0 & a_1 & \ldots & a_{N-2} \\ \vdots & \vdots & \vdots & & \vdots \\ a_1 & a_2 & a_3 & \ldots & a_0 \end{pmatrix}$$

(Die i-te Zeile geht also aus der 1. Zeile durch $(i-1)$-faches zyklisches Verschieben um 1 nach rechts hervor.) Dann gilt:

SATZ. (1) *Für $a, b \in \mathbb{R}^N$ gilt*

$$a * b = aM(b),$$

wo $aM(b)$ die Matrizenmultiplikation einer $1 \times N$-Matrix mit einer $N \times N$-Matrix bezeichnet.

(2) *Die Abbildung $\mathbb{R}^N \to \mathrm{M}_N(\mathbb{R})$, $a \mapsto M(a)$ ist $\mathbb{R}$-linear und ein Ringhomomorphismus. Insbesondere gilt für $a, b \in \mathbb{R}^N$*

$$M(a * b) = M(a)M(b) \quad \textit{und} \quad M((1, 0, \ldots, 0)) = \mathbf{1}_N.$$

Beweis:

(1) Es ist $(a_0, \ldots, a_{N-1}) * (b_0, \ldots, b_{N-1}) = (c_0, \ldots, c_{N-1})$

$$c_k = \sum_{i=0}^{k} a_i b_{k-i} + \sum_{i=k+1}^{N-1} a_i b_{N+k-i} = (a_0, a_1, \ldots, a_k, a_{k+1}, \ldots, a_{N-1}) \begin{pmatrix} b_k \\ b_{k-1} \\ \vdots \\ b_0 \\ b_{N-1} \\ \vdots \\ b_{k+1} \end{pmatrix},$$

woraus man die Behauptung ersieht.

(2) Für $c \in \mathbb{R}^N$ gilt mit (1)

$$cM(a)M(b) = (cM(a))M(b) = (c * a)M(b) = (c * a) * b = c * (a * b) = cM(a * b).$$

Da dies für jeden Vektor $c \in \mathbb{R}^N$ gilt, folgt $M(a)M(b) = M(a * b)$, wie behauptet. Der Rest ist klar. ■

Beispiel: Mit der ersten Formel des letzten Satzes kann man Multiplikationen auch mittels Matrizenmultiplikation ausrechen. Beispielsweise ist

$$(1, 2, 3) * (4, 5, 6) = (1, 2, 3) \begin{pmatrix} 4 & 5 & 6 \\ 6 & 4 & 5 \\ 5 & 6 & 4 \end{pmatrix} = (31, 31, 28),$$

was wir schon zuvor mit Hilfe von Polynomen berechnet hatten.

Ein Element $a \in \mathbb{R}^N$ heißt Einheit oder invertierbar bzgl. der $*$-Multiplikation, wenn es ein $b \in \mathbb{R}^N$ gibt mit $a * b = 1$. In diesem Fall heißt b das zu a Inverse oder invers zu a.

LEMMA. *Sei $a \in \mathbb{R}^N$.*

(1) *Es gilt die Äquivalenz:*

$$a \textit{ invertierbar} \quad \Longleftrightarrow \quad \det M(a) \neq 0.$$

Ist diesem Fall b die erste Zeile der Matrix $M(a)^{-1}$, so gilt $a * b = 1$, d.h. b ist das zu a inverse Element.

(2) *Ist a nicht invertierbar, also $\det M(a) = 0$, so gibt es ein $b \in \mathbb{R}^N \setminus \{0\}$ mit $a * b = 0$.*

Beweis:

(1) Ist a invertierbar, so gibt es ein $b \in \mathbb{R}^N$ mit $a * b = 1$ und damit $M(a)M(b) = M(1) = \mathbf{1}_N$. Es folgt $\det M(a) \neq 0$ und $M(b) = M(a)^{-1}$, woraus man auch sieht, dass b die erste Zeile der Matrix $M(a)^{-1}$ ist.

(2) Wir betrachten die lineare Abbildung

$$f : \mathbb{R}^N \to \mathbb{R}^N, \quad x \mapsto xM(a).$$

Ist $\det M(a) \neq 0$, so ist f bijektiv, also gibt es ein $b \in \mathbb{R}^N$ mit $bM(a) = (1, 0, \ldots, 0)$, was sofort $b * a = (1, 0, \ldots, 0)$ liefert.
Ist $\det M(a) = 0$, so hat die Abbildung einen nichttrivialen Kern, es gibt also ein $b \in \mathbb{R}^N \setminus \{0\}$ mit $bM(a) = 0$, was übersetzt $b * a = 0$ bedeutet. Daraus folgen alle Behauptungen. ■

Ist N nicht zu groß, so kann man mit dem letzten Lemma auch Inverse berechnen.

Beispiele:

(1) Für $a = (1, 2, 3)$ ist

$$M(a) = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{pmatrix}, \quad \det M(a) = 18, \quad M(a)^{-1} = \begin{pmatrix} -\frac{5}{18} & \frac{7}{18} & \frac{1}{18} \\ \frac{1}{18} & -\frac{5}{18} & \frac{7}{18} \\ \frac{7}{18} & \frac{1}{18} & -\frac{5}{18} \end{pmatrix},$$

also gilt

$$a * b = 1 \quad \text{mit} \quad b = (-\frac{5}{18}, \frac{7}{18}, \frac{1}{18}).$$

(2) Für $a = (1, 2, -3)$ gilt

$$M(a) = \begin{pmatrix} 1 & 2 & -3 \\ -3 & 1 & 2 \\ 2 & -3 & 1 \end{pmatrix}, \quad \det M(a) = 0 \quad \text{und} \quad a * (1, 1, 1) = (1, 2, -3) \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = 0.$$

Wir werden später noch einen anderen Weg zur Inversenberechnung kennenlernen, der Polynome verwendet.

2. $\mathbb{Z}^N$ mit dem Konvolutionsprodukt

Die expliziten Formeln der $*$-Multiplikation auf $\mathbb{R}^N$ garantieren, dass $\mathbb{Z}^N$ und auch $\mathbb{Q}^N$ abgeschlossen bzgl. der $*$-Multiplikation sind. Sie erhalten dadurch auch eine Ringstruktur:

SATZ. *$\mathbb{Z}^N$ und $\mathbb{Q}^N$ sind kommutative Ringe mit Eins bzgl. der $*$-Multiplikation.*

Kongruenzrechnung modulo m für $m \in \mathbb{N}$: Wir können auch leicht die Kongruenzrechnung von $\mathbb{Z}$ auf $\mathbb{Z}^N$ übertragen. Für $m \in \mathbb{N}$ und $a, b \in \mathbb{Z}^N$ definiert man

$$a \equiv b \bmod m \quad \Longleftrightarrow \quad a = b + mc \text{ für ein } c \in \mathbb{Z}^N.$$

Man sieht sofort, dass dies nichts anderes als die komponentenweise Kongruenz ist:

$$(a_0, \ldots, a_{N-1}) \equiv (b_0, \ldots, b_{N-1}) \bmod m \quad \Longleftrightarrow \quad a_i \equiv b_i \bmod m \text{ für alle } i.$$

LEMMA. *Für $m \in \mathbb{N}$ und $a_1, a_2, b_1, b_2 \in \mathbb{Z}^N$ gilt:*

$$a_1 \equiv a_2 \bmod m, \quad b_1 \equiv b_2 \bmod m \quad \Longrightarrow \quad a_1 + b_1 \equiv a_2 + b_2 \bmod m \text{ und } a_1 * b_1 \equiv a_2 * b_2 \bmod m.$$

Beweis: Wegen der Kongruenzen auf der linken Seite gibt es $a, b \in \mathbb{Z}^N$ mit $a_2 = a_1 + ma$, $b_2 = b_1 + mb$. Es folgt

$$a_2 * b_2 = (a_1 + ma) * (b_1 + mb) = a_1 * b_1 + m(a_1 * b + a * b_1 + ma * b),$$

was die zweite Aussage beweist; die erste ist klar. ■

Für $m \in \mathbb{N}$ heißt $a \in \mathbb{Z}^N$ invertierbar modulo m, wenn es ein $b \in \mathbb{Z}^N$ gibt mit

$$a * b \equiv 1 \bmod m,$$

wobei 1 für $(1,0,\dots,0)$ steht.

Erinnerung:

(1) Ist A eine quadratische $n\times n$-Matrix, bezeichnet A_{ij} die $(n-1)\times(n-1)$-Untermatrix von A, die man durch Streichen von Zeile i und Spalte j aus A erhält, so wird die zu A **adjungierte Matrix** A^{ad} definiert durch

$$A^{\mathrm{ad}} = \big((-1)^{i+j}\det A_{ji}\big)_{1\le i,j\le n}.$$

(In Zeile i und Spalte j von A^{ad} steht also $(-1)^{i+j}\det A_{ji}$.) Es gilt dann

$$AA^{\mathrm{ad}} = A^{\mathrm{ad}}A = (\det A)\cdot \mathbf{1}_n.$$

Für 2×2-Matrizen gilt

$$\begin{pmatrix} a & b\\ c & d\end{pmatrix}^{\mathrm{ad}} = \begin{pmatrix} d & -b\\ -c & a\end{pmatrix}.$$

(2) Ist A invertierbar, so erhält man die Formel

$$A^{-1} = \frac{1}{\det A}A^{\mathrm{ad}}.$$

(3) Hat A Einträge aus $\mathbb{Z}$, so sieht man aus dem Bildungsgesetz der adjungierten Matrix sofort, dass auch $\widetilde{A}$ nur Einträge aus $\mathbb{Z}$ hat:

$$A\in \mathrm{M}_n(\mathbb{Z}) \quad\Longrightarrow\quad A^{\mathrm{ad}}\in \mathrm{M}_n(\mathbb{Z}).$$

(4) SAGE berechnet die zur Matrix M adjungierte Matrix mit `M.adjugate()`. (Mit Python geht das genauso, wenn man zuvor `from sympy import Matrix` eingibt.)

LEMMA. *Sei $a\in\mathbb{Z}^N$ und $b\in\mathbb{Z}^N$ die erste Zeile der Matrix $M(a)^{\mathrm{ad}}$. Dann gilt:*

(1)
$$a*b = (\det M(a),0,\dots,0) = \det M(a)\cdot(1,0,\dots,0).$$

(2) *Für $m\in\mathbb{N}$ und $a\in\mathbb{Z}^N$ gilt:*

$$a \text{ ist invertierbar modulo } m \quad\Longleftrightarrow\quad \mathrm{ggT}(\det M(a), m) = 1.$$

Ist das der Fall, ist $d\in\mathbb{Z}$ mit $d\det M(a)\equiv 1 \bmod m$, so gilt

$$a*(db)\equiv 1 \bmod m.$$

Beweis:

(1) Ist b die erste Zeile der zu $M(a)$ adjungierten Matrix, so ist $bM(a)$ die erste Zeile der Matrix

$$M(a)^{\mathrm{ad}}M(a) = (\det M(a))\mathbf{1}_N,$$

also gilt $bM(a) = (\det M(a),0,\dots,0)$ und damit

$$a*b = b*a = bM(a) = (\det M(a))(1,0,\dots,0).$$

(2) Wir betrachten den Fall, dass $\mathrm{ggT}(\det A, m) = 1$ ist. Dann gibt es ein $d\in\mathbb{Z}$ mit $d(\det A)\equiv 1 \bmod m$. Es folgt dann

$$a*(db) = d(a*b) = d(\det A)(1,0,\dots,0)\equiv(1,0,\dots,0) \bmod m.$$

Also ist a invertierbar modulo m und wir haben auch ein Inverses konstruiert.

(3) Sei umgekehrt a invertierbar modulo m, d.h. es gibt $b',c\in\mathbb{Z}^N$ mit $a*b' = 1+mc$. Es folgt

$$M(a)M(b') = M(a*b') = M(1+mc) = \mathbf{1}_N + mM(c)$$

und damit

$$\det M(a)\cdot\det M(b) = \det(\mathbf{1}_N + mM(c))\equiv 1 \bmod m,$$

was $\mathrm{ggT}(\det M(a), m) = 1$ liefert. Damit sind die Behauptungen bewiesen. ■

Beispiel: Für $a = (1, 2, 3)$ gilt

$$M(a) = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{pmatrix} \quad \text{und} \quad M(a)^{\mathrm{ad}} = \begin{pmatrix} -5 & 7 & 1 \\ 1 & -5 & 7 \\ 7 & 1 & -5 \end{pmatrix}$$

und

$$(1, 2, 3) * (-5, 7, 1) = (18, 0, 0).$$

Multiplikation mit 2 liefert modulo 5

$$(1, 2, 3) * (0, 4, 2) \equiv 1 \bmod 5.$$

Bemerkung: Mit Python könnte man Inverse modulo m (unter Zuhilfenahme von sympy) folgendermaßen berechnen:

```
def ntru_inv_mod_0(a,m):
  M=[]
  for i in range(len(a)):
    M.append(a[-i:]+a[:-i])
  from sympy import Matrix
  b=list(Matrix(M).adjugate().row(0))
  c=ntru_mult(a,b)[0]
  if ggT(m,c)>1:
    return False
  c1=pow(c,-1,m)
  return [(c1*b_i)%m for b_i in b]
```

Da hierbei aber die adjungierte Matrix berechnet werden muss, ist dieses Verfahren für größere Werte von N sicher nicht praktikabel.

Wir wollen nun im Fall $m = p$ und $m = p^e$ noch einen anderen Weg für die Inversenberechnung modulo m aufzeigen:

SATZ. *Sei p eine Primzahl und $a = (a_0, a_1, \ldots, a_{N-1}) \in \mathbb{Z}^N$ und $\alpha(X) = \overline{a_0} + \overline{a_1}X + \cdots + \overline{a_{N-1}}X^{N-1} \in \mathbb{F}_p[X]$. Mit dem erweiterten euklidischen Algorithmus (in $\mathbb{F}_p[X]$) findet man $\beta(X), \gamma(X) \in \mathbb{F}_p[X]$ mit* $\operatorname{grad}(\beta(X)) \leq N - 1$ *und*

$$\alpha(X)\beta(X) + \gamma(X)(X^N - 1) = \operatorname{ggT}(\alpha(X), X^N - 1) \in \mathbb{F}_p[X].$$

(1) *Genau dann ist a invertierbar modulo p, wenn* $\operatorname{ggT}(\alpha(X), X^N - 1) = 1$ *(in $\mathbb{F}_p[X]$) gilt.*
(2) *Ist* $\operatorname{ggT}(\alpha(X), X^N - 1) = 1$ *und sind $b_0, b_1, \ldots, b_{N-1} \in \mathbb{Z}$ mit $\beta(X) = \overline{b_0} + \overline{b_1}X + \cdots + \overline{b_{N-1}}X^{N-1}$, so ist $b = (b_0, b_1, \ldots, b_{N-1})$ ein zu a Inverses modulo p.*

Beweis:

(1) Ist a invertierbar modulo p, so gibt es ein $b \in \mathbb{Z}^N$ mit $a * b \equiv 1 \bmod p$. Es gibt also ein $m \in \mathbb{Z}^N$ mit $a * b = 1 + pm$. Schreiben wir $A(X) = a_0 + a_1X + \cdots + a_{N-1}X^{N-1}$, $B(X) = b_0 + b_1X + \cdots + b_{N-1}X^{N-1}$, $M(X) = m_0 + m_1X + \cdots + m_{N-1}X^{N-1}$, so gibt es ein Polynom $\widetilde{C}(X) \in \mathbb{Z}[X]$ mit

$$A(X)B(X) = 1 + pM(X) + \widetilde{C}(X)(X^N - 1).$$

Reduktion modulo p macht aus $A(X)$ das Polynom $\alpha(X)$, aus $\widetilde{C}(X)$ das Polynom $\widetilde{\gamma}(X)$, sodass wir in $\mathbb{F}_p[X]$ erhalten

$$\alpha(X)\beta(X) = 1 + \widetilde{\gamma}(X)(X^N - 1).$$

Aus dieser Gleichung folgt sofort $\operatorname{ggT}(\alpha(X), X^N - 1) = 1$ in $\mathbb{F}_p[X]$.

(2) Ist umgekehrt $\mathrm{ggT}(\alpha(X), X^N-1) = 1$ in $\mathbb{F}_p[X]$, so findet man mit dem erweiterten euklidischen Algorithmus Polynome $\beta(X), \gamma(X) \in \mathbb{F}_p[X]$ mit

$$\alpha(X)\beta(X) + \gamma(X)(X^N - 1) = 1 \text{ in } F_p[X],$$

wobei man $\mathrm{grad}(\beta(X)) < \mathrm{grad}(X^N-1) = N$ annehmen kann. Schreibt man $A(X) = a_0 + a_1X + \cdots + a_{N-1}X^{N-1}$ und wählt man Repräsentanten $B(X), C(X) \in \mathbb{Z}[X]$ für $\beta(X), \gamma(X)$, so gibt es ein Polynom $M(X) \in \mathbb{Z}[X]$ mit

$$A(X)B(X) + C(X)(X^N - 1) = 1 + pM(X).$$

Betrachten wir das Bild in $R = \mathbb{Z}[X]/(X^N - 1)$, so erhalten wir

$$A(x)B(x) = 1 + pM(x),$$

also $\Phi(A(x))\Phi(B(x)) = 1 + p\Phi(M(x))$, d.h.

$$a * \Phi(B(x)) \equiv 1 \bmod p.$$

Dies war zu zeigen. ■

Beispiel: Wir wollen die Invertierbarkeit von $a = (1, 2, 3)$ modulo 5 untersuchen und gegebenenfalls ein Inverses bestimmen.

(1) Sei $\alpha(X) = 3X^2 + 2X + 1 \in \mathbb{F}_5[X]$. Wir wenden den euklidischen Algorithmus (in $\mathbb{F}_5[X]$) auf $X^3 - 1$ und $\alpha(X)$ an und erhalten (mit hier nicht ausgeschriebener Polynomdivision)

$$\begin{aligned} X^3 - 1 &= (2X + 2) \cdot \alpha(X) + (4X + 2), \\ \alpha(X) &= (2X + 2) \cdot (4X + 2) + 2. \end{aligned}$$

(2) Man erhält nun (durch Einsetzen)

$$2 = (4X^2 + 3X) \cdot \alpha(X) + (3X + 3) \cdot (X^3 - 1),$$

und nach Multiplikation mit 3

$$1 = (2X^2 + 4X) \cdot \alpha(X) + (4X + 4) \cdot (X^3 - 1).$$

Daher ist a invertierbar modulo 5 und $b = (0, 4, 2)$ ist invers zu a modulo 5.

LEMMA. *Sei p eine Primzahl und $a \in \mathbb{Z}^N$ invertierbar modulo p, $b_0 \in \mathbb{Z}^N$ ein zu a Inverses modulo p. Definiert man für $i \geq 1$*

$$b_i = b_{i-1} * (2 - a * b_{i-1}),$$

so gilt

$$a * b_i \equiv 1 \bmod p^{2^i} \text{ für alle } i \geq 0.$$

Beweis: Wir beweisen die Aussage durch Induktion nach i. Für $i = 0$ ist nach Voraussetzung $a * b_0 \equiv 1 \bmod p$, die Aussage also richtig. Es gelte nun bereits $a * b_i \equiv 1 \bmod p^{2^i}$, d.h. es existiert ein $c_i \in \mathbb{Z}^N$ mit

$$a * b_i = 1 + p^{2^i} \cdot c_i.$$

Es folgt

$$\begin{aligned} a * b_{i+1} &= a * b_i * (2 - a * b_i) = \\ &= (1 + p^{2^i} c_i) * (2 - (1 + p^{2^i} c_i)) = (1 + p^{2^i} c_i) * (1 - p^{2^i} c_i) = \\ &= 1 - p^{2^{i+1}} c_i * c_i, \end{aligned}$$

und damit $a * b_{i+1} \equiv 1 \bmod p^{2^{i+1}}$, was zu zeigen war. ■

SATZ. *Sei p eine Primzahl und $a \in \mathbb{Z}^N$. Für $e \in \mathbb{N}$ ist a genau dann invertierbar modulo p^e, wenn a invertierbar modulo p ist.*

Beweis: Ist a invertierbar modulo p^e, so ist a natürlich auch modulo p invertierbar. Die Umkehrung folgt sofort aus dem vorangegangenen Lemma. ■

Wir erinnern noch kurz an den erweiterten euklidischen Algorithmus für Polynome:

SATZ (Erweiterter euklidischer Algorithmus für Polynome). *Sei k ein Körper und seien $a_0, a_1 \in k[X]$ Polynome mit $a_1 \neq 0$. Rekursiv definiert man Polynome a_i und q_i wie folgt, wobei man mit $i = 1$ beginnt: Ist $a_i \neq 0$, so dividiert man a_{i-1} durch a_i, erhält einen Quotienten q_i und einen Rest a_{i+1}, also*

$$a_{i-1} = q_i a_i + a_{i+1} \quad \textit{mit } \operatorname{grad}(a_{i+1}) < \operatorname{grad}(a_i).$$

Das Verfahren endet nach endlich vielen Schritten:

$$\begin{aligned} a_0 &= q_1 a_1 + a_2, \\ a_1 &= q_2 a_2 + a_3, \\ a_2 &= q_3 a_3 + a_4, \\ &\vdots \quad\quad \vdots \\ a_i &= q_{i+1} a_{i+1} + a_{i+2}, \\ &\vdots \quad\quad \vdots \\ a_{n-2} &= q_{n-1} a_{n-1} + a_n, \\ a_{n-1} &= q_n a_n + 0. \end{aligned}$$

Definiert man nun rekursiv Polynome x_i, y_i durch

$$x_0 = 1, \quad y_0 = 0, \quad x_1 = 0, \quad y_1 = 1, \quad x_i = x_{i-2} - q_{i-1} x_{i-1}, \quad y_i = y_{i-2} - q_{i-1} y_{i-1} \textit{ für } i \geq 2,$$

so gilt $a_i = x_i a_0 + y_i a_1$. Ist ℓ der höchste Koeffizient von a_n, so gilt

$$\operatorname{ggT}(a_0, a_1) = \frac{1}{\ell} a_n \quad \textit{und} \quad \frac{1}{\ell} x_n a_0 + \frac{1}{\ell} y_n a_1 = \operatorname{ggT}(a_0, a_1).$$

Beweis: Wir zeigen $a_i = x_i a_0 + y_i a_1$ durch Induktion, wobei die Fälle $i = 0, 1$ per definitionem richtig sind. Sei nun $i \geq 2$ und die Behauptung bereits für kleinere Werte von i gezeigt. Es folgt

$$\begin{aligned} a_i &= a_{i-2} - q_{i-1} a_{i-1} = \\ &= (x_{i-2} a_0 + y_{i-2} a_1) - q_{i-1}(x_{i-1} a_0 + y_{i-1} a_1) = \\ &= (x_{i-2} - q_{i-1} x_{i-1}) a_0 + (y_{i-2} - q_{i-1} y_{i-1}) a_1 = \\ &= x_i a_0 + y_i a_1, \end{aligned}$$

was die Behauptung durch Induktion beweist. Dass a_n bis auf eine Konstante der ggT von a_0, a_1 ist, überlegt man sich mit den obigen Gleichungen sofort. ■

Bemerkung: Rechnen wir modulo m, so hatten wir meist das Repräsentantensystem $\{0, 1, 2, \ldots, m-1\}$ benutzt. Maple findet den Repräsentanten zu x mit `x\bmod m` oder `mod(x,m)`. Im Folgenden spielt das Repräsentantensystem

$$\{i \in \mathbb{Z} : -\frac{m}{2} < i \leq \frac{m}{2}\} = \begin{cases} \{-\frac{m}{2} + \frac{1}{2}, -\frac{m}{2} + \frac{3}{2}, \ldots \frac{m}{2} - \frac{3}{2}, \frac{m}{2} - \frac{1}{2}\} & \text{für ungerades } m, \\ \{-\frac{m}{2} + 1, -\frac{m}{2} + 2, \ldots, \frac{m}{2} - 1, \frac{m}{2}\} & \text{für gerades } m \end{cases}$$

eine wichtige Rolle. Maple berechnet den Repräsentanten zu x mit `mods(x,m)`.

3. Die NTRU-Verschlüsselung

Es gibt verschiedene NTRU-Varianten. Im Folgenden haben wir uns auf die Darstellung einer Variante beschränkt.

NTRU-Parameter:

(1) Man braucht drei natürliche Zahlen N, p und q. (Die Zahlen müssen keine Primzahlen sein.)

(2) Hier sind Parameter-Vorschläge aus dem „NTRU PKCS Tutorial“ von 2014:

	N	q	p
Small Illustration Parameters	11	32	3
Moderate Security	167	128	3
Standard Security	251	128	3
High Security	347	128	3
Highest Security	503	256	3

(3) Dann werden noch weitere Zahlen benutzt, wie d_f, d_g, d_r.

Die Teilmengen $L_N(d_1, d_2) = L(d_1, d_2)$: $L(d_1, d_2)$ besteht aus allen Vektoren aus $\mathbb{Z}^N$, bei denen d_1 Koeffizienten $+1$, d_2 Koeffizienten -1 und $N-d_1-d_2$ Koeffizienten 0 sind. Natürlich sollte sinnvollerweise $d_1 + d_2 \leq N$ vorausgesetzt werden. Es gilt dann

$$\#L(d_1, d_2) = \frac{N!}{d_1!d_2!(N - d_1 - d_2)!}.$$

Schlüsselerzeugung:

(1) Es wird zufällig ein $f \in L_N(d_f, d_f - 1)$ gewählt.
(2) f muss invertierbar modulo p und modulo q sein. Ist das nicht der Fall, geht man zurück zu (1). Sonst berechnet man sich $f_p \in \mathbb{Z}^N$ und $f_q \in \mathbb{Z}^N$ mit
$$f * f_p \equiv 1 \bmod p, \quad f * f_q \equiv 1 \bmod q.$$
(3) Es wird zufällig $g \in L_N(d_g, d_g)$ gewählt.
(4) Nun berechnet man $h \equiv pf_q * g \bmod q$. (In welchem Intervall die Koeffizienten von h modulo q gewählt werden, spielt keine Rolle.)
(5) h ist der öffentliche Schlüssel, f und f_p bilden den privaten Schlüssel. (Man merkt sich auch f_p, da die Berechnung doch nicht so schnell geht.)

Bemerkungen:

(1) Mitunter wird auch f in der Form

$f = 1 + pF$ mit $F = f_1 * f_2 + f_3$ und $f_1 \in L_N(d_1, d_1), f_2 \in L_N(d_2, d_2), f_3 \in L_N(d_3, d_3)$

gewählt, wie bei der „NTRU challenge“. Dann ist f nämlich invertierbar modulo p und man kann $f_p = 1$ wählen.
(2) Es gibt auch den Vorschlag $p = (2, 1, 0, \ldots, 0) = \Phi(2 + x)$ zu verwenden. Dann braucht man statt „ternären“ Vektoren bzw. Polynomen „binäre“ Vektoren bzw. Polynome, die nur 0 und 1 enthalten.

Beispiel: Wir legen die Parameter

$$N = 11, \quad p = 3, \quad q = 32, \quad d_f = 4, \quad d_g = 3, \quad d_r = 3$$

zugrunde und wollen einen NTRU-Schlüssel erzeugen.

(1) Wir wählen ein $f \in L_{11}(4, 3)$, nämlich
$$f = (0, 1, -1, 1, 1, -1, 0, 0, -1, 1, 0).$$
Wegen $\det M(f) = 529 = 23^2$ ist f modulo $p = 3$ und modulo $q = 32$ invertierbar. Wie zuvor beschrieben berechnet man dann
$$f_p = (2, 1, 1, 2, 0, 0, 1, 0, 1, 0, 2) \quad \text{und} \quad f_q = (28, 3, 14, 18, 21, 22, 14, 21, 13, 11, 28)$$
mit $f_p * f \equiv 1 \bmod p$, $f_q * f \equiv 1 \bmod q$.
(2) Nun wählen wir zufällt ein $g \in L_{11}(3, 3)$, nämlich
$$g = (-1, 0, -1, 1, 0, 0, 1, 0, -1, 0, 1).$$
(3) Mit f_q und g berechnen wir $h = pf_q * g \bmod q$ und erhalten
$$h = (7, 25, 9, 17, 4, 9, 31, 26, 8, 11, 13).$$

Der öffentliche NTRU-Schlüssel ist also h, der private besteht aus f und f_p.

Umwandlung von Text in eine Folge von Vektoren aus $\mathbb{Z}^N$: Man einigt sich auf ein Verfahren, wie man Text in eine Folge von Vektoren $m_i \in \mathbb{Z}^N$ umsetzt, wobei die Koeffizienten von m_i aus $\{-1, 0, 1\}$ sein sollen, d.h. es soll gelten $\|m_i\|_\infty \leq 1$.

Beispiel:

(1) Wir berücksichtigen hier nur 27 Zeichen, nämlich die Großbuchstaben und das Leerzeichen. Mit nachfolgender Tabelle wird jedes Zeichen in ein Zahlentripel mit Zahlen aus $\{-1, 0, 1\}$ verwandelt:

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

Hat der Ausgangstext also n Zeichen, so entsteht daraus eine Zahlenfolge mit $3n$ Zeichen.

(2) Jetzt füllen wir die Folge auf, z.B. mit -1, bis eine Zahlenfolge mit einer durch N teilbaren Anzahl entsteht. Dann unterteilen wir in Blöcke der Länge N und erhalten so eine Folge $m_i \in \mathbb{Z}^N$, wo in m_i nur Zahlen aus $\{-1, 0, 1\}$ stehen.

(3) Wir wählen den Text „KRYPTOGRAPHIE UND GITTER“und wandeln in nach obigem Schema in eine Zahlenfolge (mit 72 Zahlen) um:

```
0, -1, 1, 1, -1, -1, 1, 1, 0, 0, 1,
0, 1, -1, 1, 0, 1, -1, -1, 1, 0, 1,
-1, -1, -1, -1, 0, 0, 1, 0, -1, 1, 1,
0, -1, -1, -1, 0, 1, -1, -1, -1, 1, 0,
-1, 0, 0, 1, -1, 0, 0, -1, -1, -1, -1,
1, 0, 0, -1, -1, 1, -1, 1, 1, -1, 1,
-1, 0, 1, 1, -1, -1
```

Damit die Anzahl durch $N = 11$ teilbar ist, hängen wir fünfmal -1 an. Nun unterteilen wir in Blöcke der Länge $N = 11$ und erhalten folgende Vektorfolge:

$$\begin{aligned}
m_1 &= (0, -1, 1, 1, -1, -1, 1, 1, 0, 0, 1),\\
m_2 &= (0, 1, -1, 1, 0, 1, -1, -1, 1, 0, 1),\\
m_3 &= (-1, -1, -1, -1, 0, 0, 1, 0, -1, 1, 1),\\
m_4 &= (0, -1, -1, -1, 0, 1, -1, -1, -1, 1, 0),\\
m_5 &= (-1, 0, 0, 1, -1, 0, 0, -1, -1, -1, -1),\\
m_6 &= (1, 0, 0, -1, -1, 1, -1, 1, 1, -1, 1),\\
m_7 &= (-1, 0, 1, 1, -1, -1, -1, -1, -1, -1, -1).
\end{aligned}$$

Verschlüsselung: Man hat also einen öffentlichen NTRU-Schlüssel h und eine Folge von Vektoren $m_i \in \mathbb{Z}^N$. Für jedes i wählt man zufällig ein $r_i \in L(d_r, d_r)$ und berechnet

$$e_i \equiv r_i * h + m_i \bmod q.$$

Die Folge $e_1, e_2, e_3, \ldots$ ist dann die verschlüsselte Folge.

Beispiel: Mit dem oben erstellten öffentlichen Schlüssel

$$h = (7, 25, 9, 17, 4, 9, 31, 26, 8, 11, 13)$$

wollen wir nun den gerade in eine Vektorfolge $m_1, \ldots, m_7$ umgewandelten Text verschlüsseln. Dazu wählen wir zufällig Vektoren $r_i \in L_{11}(3,3)$ und berechnen dann $e_i = r_i * h + m_i \bmod q$.

m_i	r_i	e_i
$(0,-1,1,1,-1,-1,1,1,0,0,1)$	$(1,-1,0,0,0,-1,0,1,1,-1,0)$	$(7,19,13,14,8,22,14,28,16,26,27)$
$(0,1,-1,1,0,1,-1,-1,1,0,1)$	$(0,-1,0,-1,1,0,-1,1,0,0,1)$	$(25,10,26,20,3,10,14,3,29,7,15)$
$(-1,-1,-1,-1,0,0,1,0,-1,1,1)$	$(0,0,0,1,-1,-1,0,1,1,0,-1)$	$(10,12,16,3,30,29,14,22,8,8,6)$
$(0,-1,-1,-1,0,1,-1,-1,-1,1,0)$	$(-1,-1,0,1,0,0,-1,1,0,0,1)$	$(8,29,0,2,10,26,8,0,1,18,22)$
$(-1,0,0,1,-1,0,0,-1,-1,-1,-1)$	$(0,0,0,1,1,0,-1,1,0,-1,-1)$	$(26,3,8,26,20,7,30,15,4,11,5)$
$(1,0,0,-1,-1,1,-1,1,1,-1,1)$	$(0,-1,0,0,0,0,-1,-1,1,1,1)$	$(26,15,12,0,29,6,15,2,4,10,10)$
$(-1,0,1,1,-1,-1,-1,-1,-1,-1,-1)$	$(0,-1,0,0,0,1,-1,1,0,1,-1)$	$(28,5,8,26,14,27,3,26,8,13,28)$

$e_1, \ldots, e_7$ ist also die Verschlüsselung unseres Ausgangstexts mit dem angegebenen öffentlichen NTRU-Schlüssel h.

Entschlüsselung: Wir haben eine Vektorfolge $e_i \in \mathbb{Z}^N$, die mit dem zu (f, f_p) gehörigen öffentlichen Schlüssel verschlüsselt wurde.

(1) Wir berechnen a_i mit
$$a_i \equiv f * e_i \bmod q,$$
aber so, dass die Einträge von a_i im Intervall $(-\frac{q}{2}, \frac{q}{2}]$ liegen.
(2) Wir berechnen b_i mit
$$b_i \equiv f_p * a_i \bmod p,$$
aber so, dass die Einträge von b_i im Intervall $(-\frac{p}{2}, \frac{p}{2}]$ liegen.
(3) Ist kein Fehler passiert, dann sollte gelten $b_i = m_i$ und wir haben wieder die Ausgangsfolge.

Beispiel: Wir erhalten den Vektor
$$e = (6, 29, 26, 16, 14, 2, 27, 28, 4, 3, 30),$$
der mit dem zu
$$f = (0,1,-1,1,1,-1,0,0,-1,1,0), \quad f_p = (2,1,1,2,0,0,1,0,1,0,2)$$
gehörigen öffentlichen Schlüssel verschlüsselt wurde. Wir berechnen nacheinander:

$$\begin{aligned} f * e &= (42,-43,64,5,-6,71,2,2,25,-32,55), \\ a &= (10,-11,0,5,-6,7,2,2,-7,0,-9) \text{ mit } a \equiv f * e \bmod 32, \\ f_p * a &= (-13,-25,0,2,-13,1,29,-33,14,-25,-7), \\ b &= (-1,-1,0,-1,-1,1,-1,0,-1,-1,-1) \text{ mit } b \equiv f_p * a \bmod 3. \end{aligned}$$

Streicht man die zwei letzten Zahlen weg, damit die Anzahl durch 3 teilbar ist, dann erhält man den Text „ABC“.

Warum funktioniert die Entschlüsselung?

(1) Es ist $h \equiv p f_q * g \bmod q$, $f_q * f \equiv 1 \bmod q$ und $f_p * f \equiv 1 \bmod p$. Aus der ersten Gleichung erhält man dann
$$f * h \equiv pg \bmod q.$$
(2) Es ist $e_i \equiv r_i * h + m_i \bmod q$, was nun
$$a_i \equiv f * e_i \equiv r_i * f * h + f * m_i \equiv p r_i * g + f * m_i \bmod q$$
liefert. Dabei wurde a_i so gewählt, dass die Einträge im Intervall $(-\frac{q}{2}, \frac{q}{2}]$ liegen, d.h. dass
$$a_i \in (-\frac{q}{2}, \frac{q}{2}]^N$$
gilt.

(3) Nun kommt das zentrale Argument: p ist klein im Verhältnis zu q, die Vektoren f, g, r_i, m_i sind klein im Sinne von $f \in L_N(d_f, d_f - 1)$, $g \in L_N(d_g, d_g)$, $r_i \in L_N(d_r, d_r)$, $m_i \in \{-1, 0, 1\}^N$, d.h. sie haben nur Einträge aus $\{-1, 0, 1\}$. Also hat wahrscheinlich auch $pr_i * g + f * m_i$ kleine Einträge, d.h. wahrscheinlich Einträge aus dem Intervall $(-\frac{q}{2}, \frac{q}{2}]$. Wir bezeichnen diese Annahme als Entschlüsselungsbedingung:

$$pr_i * g + f * m_i \in (-\frac{q}{2}, \frac{q}{2}]^N \quad \text{(Entschlüsselungsbedingung)}.$$

(4) Wir nehmen für das Folgende an, dass die Entschlüsselungsbedingung

$$pr_i * g + f * m_i \in (-\frac{q}{2}, \frac{q}{2}]^N$$

gilt. Dann folgt aus

$$a_i \in (-\frac{q}{2}, \frac{q}{2}]^N \quad \text{und} \quad a_i \equiv pr_i * g + f * m_i \bmod q$$

sofort die Gleichheit der Vektoren in $\mathbb{Z}^N$:

$$a_i = pr_i * g + f * m_i.$$

(5) Nun können wir modulo p weiterrechnen:

$$b_i \equiv f_p * a_i \equiv f_p * (pr_i * g + f * m_i) \equiv f_p * f * m_i \equiv m_i \bmod p.$$

Da sowohl b_i als auch m_i nur Einträge aus dem Intervall $(-\frac{p}{2}, \frac{p}{2}]$ haben, folgt die Gleichheit

$$b_i = m_i \quad \text{in } \mathbb{Z}^N.$$

Damit haben wir die Ausgangsfolge m_i erhalten.

(6) Die Erfinder von NTRU behaupten, dass bei geeigneter Parameterwahl die Entschlüsselung mit extrem hoher Wahrscheinlichkeit funktioniert.

Bemerkung: Was passiert, wenn man die NTRU-Parameter so wählt, dass $p \mid q$ gilt? Dann folgt aus $a \equiv b \bmod q$ auch $a \equiv b \bmod p$. Somit liefert die Gleichung $h \equiv pf_q * g \bmod q$ für den öffentlichen Schlüssel $h \equiv 0 \bmod p$. Die Verschlüsselungsgleichung $e \equiv r * h + m \bmod q$ wird zu $e \equiv m \bmod p$, d.h. aus e kann man direkt m erschließen. Das darf natürlich nicht sein. Deswegen setzt man sogar voraus, dass $\mathrm{ggT}(p, q) = 1$ gilt.

Das nächste Beispiel zeigt, dass die richtige Entschlüsselung nicht selbstverständlich ist.

Beispiel: Wir haben 33 mal „KRYPTOGRAPHIE UND GITTER“aneinandergehängt, dann mit obigem NTRU-Schlüssel ($N = 11$, $p = 3$, $q = 32$)

$$h = (7, 25, 9, 17, 4, 9, 31, 26, 8, 11, 13)$$

und $d_r = 3$ verschlüsselt, anschließend mit dem zugehörigen privaten Schlüssel

$$f = (0, 1, -1, 1, 1, -1, 0, 0, -1, 1, 0), \quad f_p = (2, 1, 1, 2, 0, 0, 1, 0, 1, 0, 2)$$

entschlüsselt. (Hier ist $g = (-1, 0, -1, 1, 0, 0, 1, 0, -1, 0, 1)$.) Als Ergebnis erhielten wir:

```
KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GITTER
KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GITTER
KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GQBJFR KRYPTOGRAPHIE UND GITTER
KRYPTOGRAPHIE UNDQPBWTER KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GITTER
KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GITTER
KRYPTOGRAPXZNRUND GITTER KRYPTOGRAPGCBSCND GITTER KRYPTOGRAPHIE UND GITTER
KRYPTOGRAPHIE UND GKUODR KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GITTER
KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND AGHWER KRYPTOGRAPHIE UND GITTER
KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GITTER
KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GITTER KRYPTTSXSPHIE UND GITTER
KRYPTOGRAPHIE UND GITTER KRYPTOGRAPHIE UND GIRSBERKRYPTOGEVMZIE UND GITTER
```

Es gab 9 Fehler, die wir hier aufgelistet haben. Dabei ist $\widetilde{a} = pr_i * g + f * m_i$ und $\widetilde{m}_i$ das Ergebnis der Entschlüsselung. Ist alles in Ordnung, so ist $\widetilde{a}_i = a_i$.

- $m_{54} = (-1, -1, 1, -1, 1, 1, -1, 1, -1, 0, 1)$, $r_{54} = (-1, 0, 1, 0, -1, -1, 0, 0, 1, 0, 1)$,
  $\widetilde{a}_{54} = (7, -3, 8, 1, -10, 11, 0, -4, 7, 0, -17)$,
  $a_{54} = (7, -3, 8, 1, -10, 11, 0, -4, 7, 0, 15)$, $\min(\widetilde{a}_{54}) = -17$, $\max(\widetilde{a}_{54}) = 11$,
  $\widetilde{m}_{54} = (1, 1, -1, -1, 1, 0, -1, 0, -1, 1, -1)$.
- $m_{67} = (-1, -1, -1, -1, 1, 0, 0, -1, -1, 1, -1)$, $r_{67} = (1, -1, -1, 1, 0, -1, 0, 0, 0, 0, 1)$,
  $\widetilde{a}_{67} = (-16, -4, 11, 0, -3, 1, 2, 0, -8, 8, 4)$,
  $a_{67} = (16, -4, 11, 0, -3, 1, 2, 0, -8, 8, 4)$, $\min(\widetilde{a}_{67}) = -16$, $\max(\widetilde{a}_{67}) = 11$,
  $\widetilde{m}_{67} = (0, 1, 1, 0, 1, 0, -1, -1, 1, 1, 0)$.
- $m_{106} = (-1, 1, 1, 0, -1, -1, -1, 0, 1, -1, -1)$, $r_{106} = (-1, 1, 0, 0, -1, 0, 1, 1, -1, 0, 0)$,
  $\widetilde{a}_{106} = (6, 4, 5, -16, 2, 7, 2, -8, 2, -5, -2)$,
  $a_{106} = (6, 4, 5, 16, 2, 7, 2, -8, 2, -5, -2)$, $\min(\widetilde{a}_{106}) = -16$, $\max(\widetilde{a}_{106}) = 7$,
  $\widetilde{m}_{106} = (1, 1, -1, 1, 1, 1, 0, 0, 1, 1, -1)$.
- $m_{113} = (1, 0, -1, -1, -1, 0, 1, -1, -1, -1, 1)$, $r_{113} = (-1, 0, 1, 0, -1, 0, 0, -1, 1, 0, 1)$,
  $\widetilde{a}_{113} = (5, 2, -1, -2, 5, 1, -4, -2, 1, 9, -17)$,
  $a_{113} = (5, 2, -1, -2, 5, 1, -4, -2, 1, 9, 15)$, $\min(\widetilde{a}_{113}) = -17$, $\max(\widetilde{a}_{113}) = 9$,
  $\widetilde{m}_{113} = (0, -1, 0, -1, -1, -1, 1, 1, -1, 0, -1)$.
- $m_{129} = (-1, -1, 1, -1, 1, 1, -1, 1, -1, 0, 1)$, $r_{129} = (0, -1, 1, -1, 0, 0, 0, 1, 1, 0, -1)$,
  $\widetilde{a}_{129} = (7, 3, -4, 13, -16, 5, 0, -4, 7, -9, -2)$,
  $a_{129} = (7, 3, -4, 13, 16, 5, 0, -4, 7, -9, -2)$, $\min(\widetilde{a}_{129}) = -16$, $\max(\widetilde{a}_{129}) = 13$,
  $\widetilde{m}_{129} = (-1, 1, 1, 0, -1, 0, 1, -1, -1, 0, 0)$.
- $m_{156} = (1, 0, 0, -1, -1, 1, -1, 1, 1, -1, 1)$, $r_{156} = (0, -1, -1, -1, 1, 1, 0, 0, 1, 0, 0)$,
  $\widetilde{a}_{156} = (12, -4, -7, 17, -3, -7, -3, -6, 0, 2, 0)$,
  $a_{156} = (12, -4, -7, -15, -3, -7, -3, -6, 0, 2, 0)$, $\min(\widetilde{a}_{156}) = -7$, $\max(\widetilde{a}_{156}) = 17$,
  $\widetilde{m}_{156} = (-1, 0, -1, 1, 0, -1, 1, 1, 1, 0, 1)$.

- $m_{200} = (1,0,1,-1,-1,1,0,1,-1,-1,-1)$, $r_{200} = (1,0,0,-1,-1,1,1,-1,0,0,0)$,
  $\widetilde{a}_{200} = (5,5,-16,3,8,5,-1,-7,-3,4,-4)$,
  $a_{200} = (5,5,16,3,8,5,-1,-7,-3,4,-4)$, $\min(\widetilde{a}_{200}) = -16$, $\max(\widetilde{a}_{200}) = 8$,
  $\widetilde{m}_{200} = (1,1,-1,1,1,-1,0,1,1,-1,1)$.
- $m_{218} = (1,1,-1,1,-1,0,1,1,-1,-1,-1)$, $r_{218} = (1,1,-1,-1,0,-1,1,0,0,0,0)$,
  $\widetilde{a}_{218} = (-3,-2,-2,-2,8,5,-3,8,-16,-3,10)$,
  $a_{218} = (-3,-2,-2,-2,8,5,-3,8,16,-3,10)$, $\min(\widetilde{a}_{218}) = -16$, $\max(\widetilde{a}_{218}) = 10$,
  $\widetilde{m}_{218} = (-1,1,-1,0,-1,-1,1,-1,0,1,1)$.
- $m_{221} = (0,1,-1,-1,-1,-1,0,0,1,0,-1)$, $r_{221} = (1,-1,1,-1,0,1,0,-1,0,0,0)$,
  $\widetilde{a}_{221} = (0,8,-16,5,2,2,-5,-2,1,3,-1)$,
  $a_{221} = (0,8,16,5,2,2,-5,-2,1,3,-1)$, $\min(\widetilde{a}_{221}) = -16$, $\max(\widetilde{a}_{221}) = 8$,
  $\widetilde{m}_{221} = (0,-1,0,1,1,0,0,0,0,0,1)$.

(Bei der Wahl von $d_r = 2$ trat (bei zufällig gewählten r_i kein Fehler auf.)

4. Entschlüsselungsfehler

Zu NTRU-Parametern (N,p,q,d_f,d_g,d_r) seien wie üblich Schlüssel (f,f_p) und h über die Gleichungen

$$f * f_p \equiv 1 \bmod p, \quad f * f_q \equiv 1 \bmod q, \quad h \equiv p f_q * g \bmod q$$

definiert. Wurde eine Nachricht $m \in [-1,1]^N$ mit $r \in L_N(d_r,d_r)$ zu $e \equiv r * h + m \bmod q$ verschlüsselt, so funktioniert die Entschlüsselung, wenn die Entschlüsselungsbedingung

$$pr * g + f * m \in (-\frac{q}{2}, \frac{q}{2}]^N$$

erfüllt ist.

Lemma. *Seien (N,p,q,d_f,d_g,d_r) NTRU-Parameter mit $d_r \le d_g$ und $f \in L_N(d_f, d_f - 1)$, $g \in L_N(d_g,d_g)$.*

(1) *Für $r \in L_N(d_r,d_r)$ und $m \in [-1,1]^N$ gilt dann*
$$pr * g + f * m \in [-(2pd_r + 2d_f - 1), (2pd_r + 2d_f - 1)]^N.$$

(2) *Gilt*
$$2pd_r + 2d_f - 1 < \frac{q}{2},$$
so ist die Entschlüsselungsbedingung in jedem Fall (mit diesen Parametern) erfüllt und es wird richtig entschlüsselt.

(3) *Zu f und g gibt es $r \in L_N(d_r,d_r)$ und $m \in [-1,1]^N$ mit*
$$pr * g + f * m = (-(2pd_r + 2d_f - 1), *, \ldots, *).$$

(4) *Gilt*
$$2pd_r + 2d_f - 1 \ge \frac{q}{2},$$
so gibt es zu f und g Vektoren $r \in L_N(d_r,d_r)$ und $m \in [-1,1]^N$, für die die Entschlüsselungsbedingung verletzt ist, d.h. es gilt
$$pr * g + f * m \notin (-\frac{q}{2}, \frac{q}{2}]^N.$$

Beweis:

(1) Sei $f = (f_0, \ldots, f_{N-1})$, $g = (g_0, \ldots, g_{N-1})$, $r = (r_0, \ldots, r_{N-1})$ und $m = (m_0, \ldots, m_{N-1})$. Ist $r * g = (c_0, c_1, \ldots, c_{N-1})$, so gilt wegen $|g_j| \le 1$
$$c_0 = r_0 g_0 + \sum_{i=1}^{N-1} r_i g_{N-i}, \quad \text{also} \quad |c_0| \le |r_0| + \sum_{i=1}^{N-1} |r_i| = \sum_{i=0}^{N-1} |r_i| = 2d_r.$$
Da die Abschätzung natürlich auch für die anderen Koeffizienten gilt, folgt
$$\|pr * g\|_\infty \le 2pd_r.$$

Analog gilt für $f * m = (d_0, \dots)$

$$d_0 = f_0 m_0 + \sum_{i=1}^{N-1} f_i m_{N-i}, \quad \text{also} \quad |d_0| \le |f_0| + \sum_{i=1}^{N-1} |f_i| = \sum_{i=0}^{N-1} |f_i| = 2d_f - 1.$$

Auch diese Abschätzung gilt genauso für die anderen Koeffizienten, was dann

$$\|f * m\|_\infty \le 2d_f - 1$$

liefert. Mit der Dreiecksungleichung erhält man jetzt

$$\|pr * g + f * m\|_\infty \le 2pd_r + 2d_f - 1,$$

also die Behauptung.

(2) Die Voraussetzung $2pd_r + 2d_f - 1 < \frac{q}{2}$ liefert mit (1) sofort

$$pr * g + f * m \in [-(2pd_r + 2d_f - 1), (2pd_r + 2d_f - 1)]^N \subseteq (-\frac{q}{2}, \frac{q}{2})^N \subseteq (-\frac{q}{2}, \frac{q}{2}]^N,$$

sodass die Entschlüsselungsbedingung erfüllt ist. Also funktioniert in diesem Fall die Entschlüsselung.

(3) Wir skizzieren eine Möglichkeit. Wir betrachten den Fall $g_0 = 0$. Sei

$$g_{i_1} = g_{i_2} = \dots = g_{i_{d_g}} = 1, \quad g_{j_1} = g_{j_2} = \dots = g_{j_{d_g}} = -1.$$

Wir definieren $r = (r_0, \dots, r_{N-1})$ durch

$$r_{N-i_1} = \dots = r_{N-i_{d_r}} = -1, \quad r_{N-j_1} = \dots = r_{N-j_{d_r}} = 1 \quad \text{und} \quad r_k = 0 \text{ sonst.}$$

Wegen $d_r \le d_g$ ist dann $r \in L_N(d_r, d_r)$ und es gilt

$$r * g = (-2d_r, \dots).$$

Zu f definieren wir m durch

$$m_0 = -f_0, \quad m_i = -f_{N-i} \text{ für } i \ge 1.$$

Dann ist natürlich $m \in \{-1, 0, 1\}^N$ und wir erhalten mit obiger Formel sofort

$$f * m = (-(2d_f - 1), \dots),$$

weil f genau $2d_f - 1$ von 0 verschiedene Enträge hat. Addition ergibt nun

$$pr * g + f * m = (-(2pd_r + 2d_f - 1), \dots),$$

wie behauptet.

(4) Wählt man r und m wie in (3), so gilt unter der Voraussetung $2pd_r + 2d_f - 1 \ge \frac{q}{2}$ zunächst $-(2pd_r + 2d_f - 1) \le -\frac{q}{2}$ und damit dann

$$pr * g + f * m = (-(2pr_r + 2d_f - 1), \dots) \notin (-\frac{q}{2}, \frac{q}{2}]^N,$$

was die Behauptung beweist. ■

Bemerkung: Wir betrachten die von NTRU früher vorgeschlagenen Parameter und vergleichen $2pd_r + d_f - 1$ und $\frac{q}{2}$:

N	p	q	d_f	d_g	d_r	$2pd_r + 2d_f - 1$	$\frac{q}{2}$
11	3	32	4	3	3	25	16
107	3	64	15	12	5	59	32
167	3	128	61	20	18	229	64
263	3	128	50	24	16	195	64
503	3	256	216	72	55	761	128

Die Bedingung $2pd_r + d_f - 1 < \frac{q}{2}$ für eine in jedem Fall richtige Entschlüsselung ist also bei allen angegebenen Parameterwerten verletzt. Warum wird der Parameter q nicht größer gewählt? Stattdessen gibt es Überlegungen, wie wahrscheinlich Fehler auftreten und wie man mit ihnen umgehen kann.

Bemerkung: Wir gehen nochmals den Entschlüsselungsvorgang durch, mit den dort verwendeten Bezeichnungen. Es ist

$$a \equiv f * e \equiv f * (r * h + m) \equiv r * f * h + f * m \bmod q.$$

Mit $f * h \equiv pg \bmod q$ folgt

$$a \equiv pr * g + f * m \bmod q.$$

Also gibt es einen Vektor $v \in \mathbb{Z}^N$ mit

$$a = pr * g + f * m + qv.$$

Weiter wird gebildet

$$b \equiv f_p * a \equiv f_p * (pr * g + f * m + qv) \equiv f_p * f * m + qf_p * v \equiv m + qf_p * v \bmod p.$$

Im Fall $p = 3$ und bei zentrierter Wahl von b ist $b \in [-1, 1]^N$, sieht also nach einer Nachricht aus, auch wenn $qf_p * v \bmod p$ als Störterm auftritt.

5. Das NTRU-Gitter

Hat man zu den NTRU-Parametern (N, p, q, d_f, d_g, d_r) einen privaten Schlüssel f, g, so ist der öffentliche Schlüssel $h \equiv pf_q * g \bmod q$, wenn $f_q * f \equiv 1 \bmod q$ gilt. (Außerdem sollte f invertierbar modulo p sein.) Dann man den privaten Schlüssel f, g aus dem öffentlichen Schlüssel berechnen? Die Schlüsselgleichung ist nach Multiplikation mit f äquivalent zu

$$f * h \equiv pg \bmod q.$$

Wir suchen also Lösungen der Gleichung

$$x * h \equiv py \bmod q \text{ mit } x, y \in \mathbb{Z}^N.$$

Wir definieren

$$\Lambda_h = \{(x, y) \in \mathbb{Z}^{2N} : x * h \equiv py \bmod q\}.$$

Offensichtlich ist Λ_h ein Gitter, das wir als das zum öffentlichen Schlüssel h und den Parametern (N, p, q) (oder (N, p, q, d_f, d_g, d_r)) gehörige NTRU-Gitter bezeichnen.

Bemerkungen:

(1) Ist $(x, y) \in \Lambda_h$, d.h. gilt $x * h \equiv py \bmod q$, so gilt für $u \in \mathbb{Z}^N$ auch $u * x * h \equiv pu * y \bmod q$, d.h. es ist $(u * x, u * y) \in \Lambda_h$.

(2) Wir hatten gesehen, dass Multiplikation mit $s = (0, 1, 0, \ldots, 0)$ eine zyklische Rechtsverschiebung bewirkt, d.h. $s * (x_0, x_1, \ldots, x_{N-1}) = (x_{N-1}, x_0, x_1, \ldots, x_{N-2})$. Daher folgt

$$((x_0, x_1, \ldots, x_{N-1}), (y_0, y_1, \ldots, y_{N-1})) \in \Lambda_h \implies ((x_{N-1}, x_0, x_1, \ldots, x_{N-2}), (y_{N-1}, y_0, y_1, \ldots, y_{N-2})) \in \Lambda_h.$$

LEMMA. *Die Nachricht m werde mit r zu $e \equiv r * h + m \bmod q$ verschlüsselt. Sei $(\widetilde{f}, \widetilde{g}) \in \Lambda_h$, sodass $\widetilde{f}$ invertierbar modulo p ist mit Inversem $\widetilde{f}_p$. Gilt jetzt*

$$pr * \widetilde{g} + \widetilde{f} * m \in (-\frac{q}{2}, \frac{q}{2}]^N,$$

so kann man e auch mit $\widetilde{f}$ entschlüsseln, d.h. berechnet man

$$\widetilde{a} \equiv \widetilde{f} * e \bmod q \text{ mit } \widetilde{a} \in (-\frac{q}{2}, \frac{q}{2}]^N,$$

berechnet man

$$\widetilde{b} \equiv \widetilde{f}_p * \widetilde{a} \bmod p \text{ mit } \widetilde{b} \in (-\frac{p}{2}, \frac{p}{2}]^N,$$

so gilt $\widetilde{b} = m$.

Beweis: Wegen $(\widetilde{f}, \widetilde{g}) \in \Lambda_h$ gilt $\widetilde{f} * h \equiv p\widetilde{g} \bmod q$. Es folgt

$$\widetilde{a} \equiv \widetilde{f} * (r * h + m) \equiv pr * \widetilde{g} + \widetilde{f} * m \bmod q.$$

Die Voraussetzung an $pr * \widetilde{g} + \widetilde{f} * m$ und $\widetilde{a}$ liefert dann

$$\widetilde{a} = pr * \widetilde{g} + \widetilde{f} * m.$$

Nun rechnen wir modulo p:

$$\widetilde{b} \equiv \widetilde{f}_p * (pr * \widetilde{g} + \widetilde{f} * m) \equiv \widetilde{f}_p * \widetilde{f} * m \equiv m \bmod p.$$

Wegen $\widetilde{b} \in (-\frac{p}{2}, \frac{p}{2}]^N$ und $m \in \{-1, 0, 1\}^N \subseteq (-\frac{p}{2}, \frac{p}{2}]^N$ folgt $\widetilde{b} = m$, wie behauptet. ■

Bemerkungen:

(1) Wurde der öffentliche Schlüssel h aus $f \in L_N(d_f, d_f - 1)$ und $g \in L_N(d_g, d_g)$ mittels der Gleichung $f * h \equiv pg \bmod q$ konstruiert, so gilt genauso mit $s = (0, 1, 0, \ldots, 0)$

$$(s * f) * h \equiv p(s * g) \bmod q \quad \text{und} \quad s * f \in L_N(d_f, d_f - 1), \quad s * g \in L_N(d_g, d_g).$$

Das letzte Lemma zeigt, dass man auch mit $s * f = (f_{N-1}, f_0, \ldots, f_{N-2})$ entschlüsseln kann. Der private Schlüssel ist also nicht eindeutig durch h bestimmt. In jedem Fall leisten die Vektoren

$$\pm s^i * f \text{ für } i = 0, \ldots, N-1$$

Ähnliches. (Die Vektoren $s^i * f$, $i = 0, \ldots, N-1$, sind auch linear unabhängig, da die beschreibende Matrix einfach $M(f)$ ist.)

(2) Das letzte Lemma ermutigt dazu, auch mit anderen Gitterelementen $(\widetilde{f}, \widetilde{g}) \in \Lambda_h$ das Entschlüsseln zu probieren. Natürlich sollte $\widetilde{f}$ und $\widetilde{g}$ nicht zu groß sein, damit die Entschlüsselungsbedingung

$$pr * \widetilde{g} + \widetilde{f} * m \in (-\frac{q}{2}, \frac{q}{2}]^N$$

wahrscheinlich erfüllt ist. (Da man r und m aus Außenstehender nicht kennt, kann man die Bedingung natürlich nicht direkt überprüfen. Das folgende Lemma gibt aber einen Hinweis.)

LEMMA. *Ist $(\widetilde{f}, \widetilde{g}) \in \Lambda_h$, ist $\widetilde{f}$ invertierbar modulo p und gilt*

$$2pd_r\|\widetilde{g}\|_\infty + \|\widetilde{f}\|_1 < \frac{q}{2},$$

so kann man alle Nachrichten auch mit $\widetilde{f}$ entschlüsseln.

Beweis: Wir wollen $\|pr * \widetilde{g} + \widetilde{f} * m\|_\infty$ abschätzen. Der erste Eintrag von $pr * \widetilde{g}$ lässt sich betragsmäßig durch

$$|pr_0\widetilde{g}_0 + p\sum_{i=1}^{N-1} r_i\widetilde{g}_{N-i}| \le p\|\widetilde{g}\|_\infty \sum_{i=0}^{N-1} |r_i| = p\|\widetilde{g}\|_\infty \cdot 2d_r$$

abschätzen, genauso wie die anderen. Also folgt

$$\|pr * \widetilde{g}\|_\infty \le 2pd_r\|\widetilde{g}\|_\infty.$$

Wegen $m \in \{-1, 0, 1\}^N$ gilt für den ersten Eintrag von $\widetilde{f} * m$:

$$|\widetilde{f}_0 m_0 + \sum_{i=1}^{N-1} \widetilde{f}_i m_{N-i}| \le \sum_{i=0}^{N-1} |\widetilde{f}_i| = \|\widetilde{f}\|_1.$$

Da sich die anderen Einträge genauso abschätzen lassen, folgt

$$\|\widetilde{f} * m\|_\infty \le \|\widetilde{f}\|_1.$$

Damit erhalten wir

$$\|pr * \widetilde{g} + \widetilde{f} * m\|_\infty \le 2pd_r\|\widetilde{g}\|_\infty + \|\widetilde{f}\|_1.$$

Die Voraussetzung des Lemmas liefert dann

$$\|pr * \widetilde{g} + \widetilde{f} * m\|_\infty < \frac{q}{2}, \quad \text{also} \quad pr * \widetilde{g} + \widetilde{f} * m \in (-\frac{q}{2}, \frac{q}{2}]^N,$$

was nach dem vorangegangenen Lemma zeigt, dass m mit $\widetilde{f}$ richtig entschlüsselt wird. Da m beliebig war, folgt die Behauptung. ■

Bemerkung: Damit NTRU sicher ist, muss man die Parameter so wählen, dass die Voraussetzungen des Lemmas praktisch nicht erfüllbar sind.

SATZ. *Sei h ein öffentlicher NTRU-Schlüssel zu den NTRU-Parametern (N, p, q, d_f, d_g, d_r) (mit $\mathrm{ggT}(p, q) = 1$). Sei $\widetilde{h} \equiv \frac{1}{p}h \bmod q$.*

(1) *Das NTRU-Gitter lässt sich schreiben als*

$$\Lambda_h = \{(x,y) \in \mathbb{Z}^{2N} : x * \widetilde{h} \equiv y \bmod q\}.$$

(2) *Die Zeilen der Matrix*

$$\begin{pmatrix} M(1) & M(\widetilde{h}) \\ 0 & M(q) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 & \widetilde{h}_0 & \widetilde{h}_1 & \dots & \widetilde{h}_{N-1} \\ 0 & 1 & \dots & 0 & \widetilde{h}_{N-1} & \widetilde{h}_0 & \dots & \widetilde{h}_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 & \widetilde{h}_1 & \widetilde{h}_2 & \dots & \widetilde{h}_0 \\ 0 & 0 & \dots & 0 & q & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & q \end{pmatrix}$$

bilden eine Gitterbasis des NTRU-Gitters Λ_h.

(3) *Λ_h ist ein Gitter vom Rang $2N$ mit Determinante q^N.*

(4) *Ist $f \in L_N(d_f, d_f - 1)$ ein privater Schüssel und $g \in L_N(d_g, d_g)$ mit $f * \widetilde{h} \equiv g \bmod q$, ist $s = (0,1,0,\dots,0)$, so sind die N Vektoren $(s^i * f, s^i * g) \in \Lambda_h$ (für $i = 0,\dots,N-1$) linear unabhängig und haben Länge*

$$\sqrt{2d_f + 2d_g - 1}.$$

(5) *$(1,1,\dots,1,0,0,\dots,0)$ (mit N Einsen) ist ein Gittervektor und hat Länge $\sqrt{N}$.*

Beweis:

(1) Das Gitter Λ_h war definiert also $\Lambda_h = \{(x,y) \in \mathbb{Z}^{2N} : x * h \equiv py \bmod q\}$. Multipliziert man die Gleichung $x * h \equiv py \bmod q$ mit dem Inversen von p modulo q, so erhält man die äquivalente Formulierung $x * \widetilde{h} \equiv y \bmod q$, also

$$\Lambda_h = \{(x,y) \in \mathbb{Z}^{2N} : x * \widetilde{h} \equiv y \bmod q\}.$$

(2) (a) Wir zeigen zunächst, dass die Zeilen obiger Matrix im Gitter liegen. Ist $s = (0,1,0,\dots,0)$, so lässt sich für $1 \le i \le N$ die i-te Zeile obiger Matrix schreiben als

$$(s^{i-1} * 1, s^{i-1} * \widetilde{h}).$$

Trivialerweise gilt dann $(s^{i-1} * 1, s^{i-1} * \widetilde{h}) \in \Lambda_h$.
Für $N+1 \le i \le 2N$ ist die i-te Zeile obiger Matrix 0 modulo q, also ist die Zeile trivialerweise in Λ_h.

(b) Wir zeigen nun, dass sich jedes Gitterelement als ganzzahlige Linearkombination obiger Zeilenvektoren schreiben lässt. Für $x, y \in \mathbb{Z}^N$ gilt:

$$\begin{aligned} (x,y) \in \Lambda_h \quad &\iff \quad x * \widetilde{h} \equiv y \bmod q \quad \iff \quad x * \widetilde{h} = y + qz \text{ für ein } z \in \mathbb{Z}^N \quad \iff \\ &\iff \quad xM(\widetilde{h}) - zM(q) = y \text{ für ein } z \in \mathbb{Z}^N \quad \iff \\ &\iff \quad (x, -z) \begin{pmatrix} M(1) & M(\widetilde{h}) \\ 0 & M(q) \end{pmatrix} = (x,y) \text{ für ein } z \in \mathbb{Z}^N. \end{aligned}$$

Ist also $(x,y) \in \Lambda_h$, so ist (x,y) ganzzahlige Linearkombination der Zeilen obiger Matrix.

(c) Damit ist klar, dass Λ_h von den Zeilen obiger Matrix aufgespannt wird. Da die Determinante der Matrix $\neq 0$ ist, sind die Zeilenvektoren linear unabhängig, bilden also eine Gitterbasis.

(3) Die angegebene Matrix hat Determinante q^N, also hat das Gitter Rang $2N$ und Determinante q^N.

(4) Schreibt man die Vektoren $(s^i * f, s^i * g)$ für $i = 0,\dots,N-1$ in eine Matrix, so erhält man die aus $M(f)$, $M(g)$ zusammengesetzte Matrix. Da f invertierbar ist, ist $\det M(f) \neq 0$, also sind die N Vektoren linear unabhängig. Die Länge berechnet sich so:

$$\|(f,g)\| = \sqrt{\sum_{i=0}^{N-1} |f_i|^2 + \sum_{i=0}^{N-1} |g_i|^2} = \sqrt{2d_f + 2d_g - 1}.$$

(5) Da g genausoviele $+1$-Einträge wie -1-Einträge hat, gilt $(1,1,\ldots,1) * g = 0$. Wegen $\widetilde{h} = f_q * g \bmod q$ folgt dann $(1,1,\ldots,1) * \widetilde{h} = 0$, sodass $(1,1,\ldots,1,0,0,\ldots,0)$ trivialerweise ein Gittervektor ist. Die Länge ist $\sqrt{N}$. ■

Bemerkungen:

(1) Der letzte Satz sagt, dass private Schlüsselvektoren (f,g) kurze Gittervektoren im NTRU-Gitter Λ_h sind. Könnte man die kurzen Vektoren leicht finden, so könnte man auch einen privaten Schlüssel schnell entdecken.
(2) Die Sicherheit beruht also darauf, dass man die Gittervektoren mit einer Länge $\leq \sqrt{2d_f + 2d_g - 1}$ praktisch nicht finden kann.
(3) Eine Methode, eine einigermaßen schöne Gitterbasis zu finden, war die LLL-Reduktion. Ist $b_1,\ldots,b_{2N}$ eine LLL-reduzierte Gitterbasis von Λ_h, so gilt für den ersten Gittervektor die Abschätzung $(\|b_1\| \leq 2^{(m-1)/4}(\det\Lambda)^{1/m})$

$$\|b_1\| \leq 2^{\frac{2N-1}{4}}(q^N)^{\frac{1}{2N}} = 2^{\frac{2N-1}{4}}\sqrt{q}.$$

(4)

N	p	q	d_f	d_g	d_r	$\sqrt{2d_f+2d_g-1}$	$2^{(2N-1)/4}\sqrt{q}$
11	3	32	4	4	3	3.87	≈ 215
107	3	64	15	15	5	7.68	$\approx 8.57\cdot 10^{16}$
167	3	128	61	61	18	15.59	$\approx 1.30\cdot 10^{26}$
263	3	128	50	50	16	14.11	$\approx 3.66\cdot 10^{40}$
503	3	256	216	216	55	29.38	$\approx 6.88\cdot 10^{76}$

Beispiel: Wir betrachten ein Beispiel mit den NTRU-Parametern $(N,p,q,d_f,d_g,d_r) = (11,3,32,4,3,3)$ und den Schüsselvektoren

$$\begin{aligned} f &= (0,0,1,0,1,-1,0,1,1,-1,-1),\\ f_p &= (1,1,0,2,1,0,1,0,2,0,2),\\ f_q &= (18,6,21,30,18,9,22,11,24,23,11),\\ g &= (-1,0,1,-1,0,0,0,1,-1,1,0),\\ h &= (2,9,19,28,0,30,30,30,28,29,19),\\ \widetilde{h} &= (22,3,17,20,0,10,10,10,20,31,17), \end{aligned}$$

wobei also h der öffentliche Schlüssel ist und $p\widetilde{h} \equiv h \bmod q$ gilt. Das zugehörige NTRU-Gitter ist

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & 3 & 17 & 20 & 0 & 10 & 10 & 10 & 20 & 31 & 17 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 17 & 22 & 3 & 17 & 20 & 0 & 10 & 10 & 10 & 20 & 31 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 31 & 17 & 22 & 3 & 17 & 20 & 0 & 10 & 10 & 10 & 20 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 31 & 17 & 22 & 3 & 17 & 20 & 0 & 10 & 10 & 10 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 20 & 31 & 17 & 22 & 3 & 17 & 20 & 0 & 10 & 10 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 10 & 10 & 20 & 31 & 17 & 22 & 3 & 17 & 20 & 0 & 10 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 10 & 10 & 10 & 20 & 31 & 17 & 22 & 3 & 17 & 20 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 10 & 10 & 10 & 20 & 31 & 17 & 22 & 3 & 17 & 20 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 20 & 0 & 10 & 10 & 10 & 20 & 31 & 17 & 22 & 3 & 17 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 17 & 20 & 0 & 10 & 10 & 10 & 20 & 31 & 17 & 22 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 17 & 20 & 0 & 10 & 10 & 10 & 20 & 31 & 17 & 22 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32
\end{pmatrix}.$$

LLL-Reduktion liefert

$$\begin{pmatrix}
-1 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 1 & 1 & -1 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & 0 & -1 & 0 & -1 & 1 & 0 & -1 & 1 & -1 & 0 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & -1 \\
1 & 1 & -1 & 0 & -2 & 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & -1 & 1 & -2 & 1 \\
-1 & -1 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 1 & 1 & 1 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & -1 \\
1 & 0 & 1 & -1 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & -1 & 0 \\
1 & 1 & -1 & -1 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 1 & -1 & 1 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & -1 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & -1 \\
0 & 1 & 1 & -1 & -1 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 1 & -1 & 1 & 0 & -1 & 0 & 1 & -1 & 0 & 0 \\
1 & -1 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & -1 & 0 & 1 & -1 \\
-1 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & -1 & 1 & 0 & -1 & 0 & 1 & -1 & 0 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 2 & 1 & 0 & -3 & 0 & -1 & -3 & -1 & 2 & -1 & -4 & -4 & -3 & -6 & -4 & -5 & -1 & -2 & 1 & 1 & -5 \\
-1 & -3 & 3 & 1 & 4 & -1 & -1 & 3 & -1 & -6 & 1 & 5 & 0 & -1 & 1 & 2 & -1 & -3 & -3 & -3 & -2 & 5 \\
3 & -2 & 1 & -1 & -3 & 3 & -1 & -1 & 2 & 2 & -4 & -1 & 1 & 1 & -5 & 4 & -2 & 3 & 2 & 2 & 0 & -5 \\
2 & -4 & 3 & -2 & 1 & -1 & -3 & 3 & -1 & -1 & 2 & 0 & -5 & -1 & 1 & 1 & -5 & 4 & -2 & 3 & 2 & 2 \\
0 & 3 & 0 & 1 & 3 & 1 & -2 & 1 & -4 & -2 & -1 & 6 & 4 & 5 & 1 & 2 & -1 & -1 & 5 & 4 & 4 & 3 \\
2 & 1 & 3 & 0 & -1 & 2 & -2 & -5 & 2 & -1 & -3 & -2 & 2 & 2 & -1 & -3 & -4 & -2 & -3 & 5 & 6 & 0 \\
1 & 1 & -3 & 1 & 6 & -1 & 1 & 3 & -3 & -1 & -4 & 1 & 3 & 3 & 3 & 2 & -5 & -5 & 0 & 1 & -1 & -2 \\
-3 & 1 & 0 & -2 & 2 & -2 & -1 & 2 & 3 & -4 & 4 & 1 & 1 & -4 & 3 & -1 & 3 & 1 & 2 & 1 & -6 & -1 \\
5 & 0 & 1 & 2 & -4 & 0 & -3 & 1 & 1 & -4 & 1 & 2 & -5 & -5 & -1 & 2 & -2 & -2 & 2 & 3 & 2 & 4 \\
3 & -3 & 1 & 1 & -2 & -2 & 4 & -3 & 2 & -1 & 1 & -4 & 2 & -3 & -2 & -2 & 0 & 5 & 1 & -1 & -1 & 5 \\
-1 & -2 & 0 & 3 & -1 & 3 & 0 & 1 & 1 & -1 & 1 & -2 & -1 & 1 & 2 & -4 & -5 & -4 & -3 & -7 & -4 & -5
\end{pmatrix}.$$

i	f_i	$\|f_i\|_\infty$	$\|(f_i, g_i)\|$	
1	$(-1,0,0,1,0,1,-1,0,1,1,-1)$	1	$\sqrt{13} = 3.61$	$f_1 = s^1 * f$
2	$(-1,1,1,0,0,-1,0,-1,1,0,-1)$	1	$\sqrt{13} = 3.61$	$f_2 = -s^3 * f$
3	$(1,1,-1,0,-2,1,-1,0,-1,0,0)$	2	$\sqrt{20} = 4.47$	
4	$(-1,-1,0,0,1,0,1,-1,0,1,1)$	1	$\sqrt{13} = 3.61$	$f_4 = s^2 * f$
5	$(1,0,1,-1,0,1,1,-1,-1,0,0)$	1	$\sqrt{13} = 3.61$	$f_5 = s^9 * f$
6	$(1,1,-1,-1,0,0,1,0,1,-1,0)$	1	$\sqrt{13} = 3.61$	$f_6 = s^4 * f$
7	$(0,1,0,1,-1,0,1,1,-1,-1,0)$	1	$\sqrt{13} = 3.61$	$f_7 = s^{10} * f$
8	$(0,1,1,-1,-1,0,0,1,0,1,-1)$	1	$\sqrt{13} = 3.61$	$f_8 = s^5 * f$
9	$(1,-1,0,1,1,-1,-1,0,0,1,0)$	1	$\sqrt{13} = 3.61$	$f_9 = s^7 * f$
10	$(-1,0,1,1,-1,-1,0,0,1,0,1)$	1	$\sqrt{13} = 3.61$	$f_{10} = s^6 * f$
11	$(-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1)$	1	$\sqrt{11} = 3.32$	
12	$(4,2,1,0,-3,0,-1,-3,-1,2,-1)$	4	$\sqrt{196} = 14.00$	
13	$(-1,-3,3,1,4,-1,-1,3,-1,-6,1)$	6	$\sqrt{173} = 13.15$	
14	$(3,-2,1,-1,-3,3,-1,-1,2,2,-4)$	4	$\sqrt{149} = 12.21$	
15	$(2,-4,3,-2,1,-1,-3,3,-1,-1,2)$	4	$\sqrt{149} = 12.21$	
16	$(0,3,0,1,3,1,-2,1,-4,-2,-1)$	4	$\sqrt{196} = 14.00$	
17	$(2,1,3,0,-1,2,-2,-5,2,-1,-3)$	5	$\sqrt{174} = 13.19$	
18	$(1,1,-3,1,6,-1,1,3,-3,-1,-4)$	6	$\sqrt{173} = 13.15$	
19	$(-3,1,0,-2,2,-2,-1,2,3,-4,4)$	4	$\sqrt{148} = 12.17$	
20	$(5,0,1,2,-4,0,-3,1,1,-4,1)$	5	$\sqrt{174} = 13.19$	
21	$(3,-3,1,1,-2,-2,4,-3,2,-1,1)$	4	$\sqrt{149} = 12.21$	
22	$(-1,-2,0,3,-1,3,0,1,1,-1,1)$	3	$\sqrt{194} = 13.93$	

Wir haben jetzt „MORGENSTUND HAT GOLD IM MUND“ mit h zu

$$e_1 = (15,10,19,27,3,27,21,5,18,22,24), \quad e_2 = (9,27,9,2,5,16,7,0,4,15,4),$$
$$e_3 = (3,23,27,31,20,0,23,11,15,3,4), \quad e_4 = (10,1,20,2,17,7,26,14,3,11,13),$$
$$e_5 = (26,18,5,19,19,16,5,2,29,22,29), \quad e_6 = (23,1,21,15,28,19,19,27,8,13,11),$$
$$e_7 = (2,30,14,20,3,9,29,16,19,7,9), \quad e_8 = (15,8,15,31,20,31,14,19,4,26,4)$$

verschlüsselt. Nun versuchen wir mit den Zeilen der reduzierten Matrix zu entschlüsseln:

- $f_1 = (-1,0,0,1,0,1,-1,0,1,1,-1)$: „MORGENSTUND HAT GOLD IM MUND “
- $f_2 = (-1,1,1,0,0,-1,0,-1,1,0,-1)$: „MORGENSTUND HAT GOLD IM MUND “
- $f_3 = (1,1,-1,0,-2,1,-1,0,-1,0,0)$: „MORGENSTUND HAT GOQUFKM MUND “
- $f_4 = (-1,-1,0,0,1,0,1,-1,0,1,1)$: „MORGENSTUND HAT GOLD IM MUND “
- $f_5 = (1,0,1,-1,0,1,1,-1,-1,0,0)$: „MORGENSTUND HAT GOLD IM MUND “
- $f_6 = (1,1,-1,-1,0,0,1,0,1,-1,0)$: „MORGENSTUND HAT GOLD IM MUND “
- $f_7 = (0,1,0,1,-1,0,1,1,-1,-1,0)$: „MORGENSTUND HAT GOLD IM MUND “
- $f_8 = (0,1,1,-1,-1,0,0,1,0,1,-1)$: „MORGENSTUND HAT GOLD IM MUND “
- $f_9 = (1,-1,0,1,1,-1,-1,0,0,1,0)$: „MORGENSTUND HAT GOLD IM MUND “
- $f_{10} = (-1,0,1,1,-1,-1,0,0,1,0,1)$: „MORGENSTUND HAT GOLD IM MUND “
- $f_{11} = (-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1)$: f ist nicht invertierbar.
- $f_{12} = (4,2,1,0,-3,0,-1,-3,-1,2,-1)$: f ist nicht invertierbar.
- $f_{13} = (-1,-3,3,1,4,-1,-1,3,-1,-6,1)$: „DVUGXM YESGWISSLRPXUHMYV ABFT“
- $f_{14} = (3,-2,1,-1,-3,3,-1,-1,2,2,-4)$: „JWXQDFYGODERKPWFT KVYRNMLLTRV“
- $f_{15} = (2,-4,3,-2,1,-1,-3,3,-1,-1,2)$: „JWXQDFYGODERKPWFT KVYRNMLLTRV“
- $f_{16} = (0,3,0,1,3,1,-2,1,-4,-2,-1)$: f ist nicht invertierbar.
- $f_{17} = (2,1,3,0,-1,2,-2,-5,2,-1,-3)$: „DLHMXIDUVKOOVGZHESSTWSUDNHOHU“
- $f_{18} = (1,1,-3,1,6,-1,1,3,-3,-1,-4)$: „DVUGXM WVOJWISSLRPXUHMMSEDBFT“
- $f_{19} = (-3,1,0,-2,2,-2,-1,2,3,-4,4)$: f ist nicht invertierbar.
- $f_{20} = (5,0,1,2,-4,0,-3,1,1,-4,1)$: f ist nicht invertierbar.
- $f_{21} = (3,-3,1,1,-2,-2,4,-3,2,-1,1)$: „JWXQDFYBWMMRKPUKNCBVYRNMLLTRV“

- $f_{22} = (-1,-2,0,3,-1,3,0,1,1,-1,1)$: „FRZFVQJUMMLXBYWRQMVWOC HEVKK“

Beispiel: Wir wählen die Parameter $(N,p,q,d_f,d_g,d_r) = (107,3,64,15,12,5)$ und zufällig $f \in L_N(d_f, d_f-1)$ und $g \in L_N(d_g,d_g)$, sodass f invertierbar modulo p und q ist.

$$
\begin{aligned}
f &= (0,0,0,0,1,0,0,0,0,0,0,-1,0,0,1,-1,0,1,0,0,1,0,0,0,0,0,1,1,0,1,0,0,0,0,0,0,0,0,\\
&\quad 0,0,0,0,1,0,-1,0,0,0,-1,0,0,1,-1,0,0,0,0,0,0,-1,0,1,1,0,0,0,0,0,0,-1,0,0,0,0,0,\\
&\quad 0,0,0,0,0,0,-1,0,1,0,0,0,0,0,1,0,-1,0,-1,0,0,-1,-1,1,0,0,0,0,0,0,0,1,-1,0,0,-1),\\
g &= (1,-1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,-1,0,0,0,0,0,0,1,1,0,-1,0,0,0,0,0,0,0,0,\\
&\quad 0,1,1,0,-1,0,0,-1,0,-1,0,0,0,0,0,-1,0,0,0,-1,0,0,0,0,0,-1,0,0,0,-1,0,0,0,0,0,-1,\\
&\quad 0,1,0,-1,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0),\\
h &= (45,58,38,32,58,63,19,19,19,52,17,37,60,62,50,33,25,62,28,30,46,34,42,13,48,8,3,\\
&\quad 60,8,26,37,23,6,45,33,28,36,62,61,7,33,35,61,60,36,15,44,17,59,44,53,37,53,6,\\
&\quad 10,2,27,3,35,52,40,6,40,2,40,22,51,47,24,3,8,27,39,3,62,58,61,12,7,1,28,\\
&\quad 52,17,9,45,47,49,0,33,56,58,36,21,53,56,14,9,12,11,45,60,47,34,34,56,27,7).
\end{aligned}
$$

Wir bilden das NTRU-Gitter und führen LLL-Reduktion durch (LLL-Reduktion in 2:20 (Minuten:Sekunden)). Die Normen der entstehenden Basisvektoren $b_1,\ldots,b_{214}$ sind

64.00	64.00	64.00	64.00	64.00	64.00	64.00	64.00	64.00	64.00
64.00	64.00	64.00	269.83	268.88	269.11	262.45	265.36	266.16	266.16
281.23	280.51	274.45	272.11	257.56	257.56	259.55	262.74	268.61	269.08
270.72	286.18	287.52	256.23	260.94	275.99	271.13	270.65	261.07	264.96
266.23	266.23	271.33	274.14	284.55	295.48	261.89	262.38	270.81	280.02
265.54	254.22	254.22	274.50	247.75	250.06	253.50	253.50	254.00	259.77
267.77	282.82	294.81	249.18	249.18	245.56	290.61	262.64	248.41	248.41
248.41	248.41	273.92	276.01	286.36	246.32	250.96	257.38	262.55	427.76
283.45	419.11	260.68	403.12	408.72	397.46	400.92	408.63	422.33	392.09
387.98	392.66	390.43	407.83	412.73	384.74	386.68	434.10	409.87	426.33
396.41	412.14	430.15	412.85	388.43	405.93	362.08	419.45	396.57	437.63
380.52	385.62	414.14	389.33	360.81	409.68	417.11	372.26	374.38	419.72
399.31	378.43	385.03	398.45	396.69	396.61	354.79	418.68	390.97	395.09
367.59	370.20	417.95	399.12	418.45	374.23	406.31	356.29	391.95	371.28
379.43	380.45	406.60	372.52	390.91	362.47	374.51	419.39	411.72	414.90
398.86	430.99	421.88	366.33	377.54	386.24	405.97	392.56	409.56	410.19
395.96	384.55	385.36	373.13	431.62	385.81	400.88	403.97	417.38	410.56
413.91	435.45	412.27	360.68	408.52	370.15	378.78	382.81	395.05	380.34
443.54	390.15	395.37	400.80	417.34	404.95	401.48	431.13	384.40	398.97
338.59	392.50	432.32	423.94	404.71	381.76	410.94	427.04	426.72	368.21
380.33	386.59	393.28	427.54	369.52	373.87	405.96	390.60	380.10	390.68
426.44	367.10	397.63	367.43						

Wir können ein paar Vektoren identifizieren, wenn M die zum Gitter Λ_h gehörige Matrix ist: $b_1 = M_{108}$, $b_2 = M_{109}$, $b_3 = M_{110}$, $b_4 = M_{111}$, $b_5 = M_{112}$, $b_6 = M_{113}$, $b_7 = M_{114}$, $b_8 = M_{115}$, $b_9 = M_{116}$, $b_{10} = M_{117}$, $b_{11} = M_{118}$, $b_{12} = M_{119}$, $b_{13} = M_{123}$. Die ersten 13 f_i's sind 0. Für folgende Indizes ist f_i nicht 0, aber über $\mathbb{Q}$ nicht invertierbar:

```
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, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55,
56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76,
77, 78, 79, 80, 81, 82, 83, 86, 87, 89, 90, 97, 104, 105, 110, 114, 115, 117, 118,
120, 124, 135, 138, 139, 143, 147, 153, 155, 159, 164, 175, 179, 183, 185, 208, 212.
```

Für die folgenden Indizes ist f_i über $\mathbb{Q}$ invertierbar, nicht jedoch modulo $p = 3$:

```
145, 146, 151, 166, 168, 170, 180, 186, 188, 193, 197, 207.
```

Mit den restlichen f_i's haben wir die Entschlüsselung versucht:

- f_{84}: „FAEPEMHISYVNRJCGEUE ECMHXGLBUTDKTGP“
- f_{85}: „RSCGP XVXUASKP WIYMGCXBLTXOAWHJQAVI“
- f_{88}: „THYUKVFJFXXHXOTPJRROUOZLATHJBERXGIV“
- f_{91}: „BDBHXTPWVAEJDWTTSJR IARQNBKGXCQYTTK“
- f_{92}: „PSNTNXMDZKNNUVOYGPRAGYAIAPUWKTJPIPC“
- f_{93}: „DUZYNPNELI JELFGCAADZHISSPWU KEKXRY“
- f_{94}: „JBTJEYJLSWJ EIWMJAHKBK QWXJJCYMGAYB“
- f_{95}: „AJWXHVSPHVFYHPUWFWBOBNHDWS T OORMPU“
- f_{96}: „CJKAPRIQXTONLNQVYF XORFNFCXZGJPBLAC“
- f_{98}: „WKRHHVGGPDXDVJLXXIPMMX EZXWJRULQOZP“
- f_{99}: „HSCTIM ZOVEVFSHNWWOSW KQLDAS ICRUCE“
- f_{100}: „ZFZHZ VRSYVSGTNFNABKBHVAQJLHIMKKHBC“
- f_{101}: „LRYBRINVSJYKUIUJRMMVTMMBXMBRWFSOKFW“
- f_{102}: „EVZNNVYOTRRTNRGCFNPREWZHGOMXMSUIQLL“
- f_{103}: „LJTFQIGM YFKOADITXNSDQMXGSQTJEJ YDL“
- f_{106}: „ZVHITVASLRXLIHWQLEZSOBYPTOKXALUQYES“
- f_{107}: „WMVKEJUTVDAKDWZPCEBRDGGPMVGPQCUACHK“
- f_{108}: „VS PBPJ IITCZWZN ZLVATGFPQCJWBCGYPH“
- f_{109}: „KWPYTTQIPJEWPGHPWGHRXYEMWOGEMMTMQNQ“
- f_{111}: „IWYMEA PTXZWRRFIAX LHGAOHFGOPYQOQZR“
- f_{112}: „JZRHSDOEDDBOISUPXBOEHFVOARBJGIWXFWC“
- f_{113}: „BXLKJOUHEIUS GGTVPWDJLDFPFZNVFMCGQ “
- f_{116}: „WSHUXPBVGCYFAUTZOAZZKGBTSYHFO PUHCW“
- f_{119}: „JNRKYYDMC COCHJHYG VYTBC SQHALLEMAG“
- f_{121}: „MFHHHVXIFVZXSLXLZSPLKQL TGNPROGGIKQ“
- f_{122}: „USJRZGRJFNCGBYVHMIBEPNNEILRNHTFFHZ “
- f_{123}: „XMQ FSBWELBGZYOJERQBUFWLOC IEQB TYX“
- f_{125}: „SAFXJSQHLWHHTYKBEREAGGCOFVLGJGJ TFY“
- f_{126}: „YBKOKLLJQOIHXMVHPCQBPJVJOIOFLUKWOID“
- f_{127}: „VHUNMDZCYOZLEOAAWNDAACL CRBTEDGKUXW“
- f_{128}: „VXECVZVAAQSNDVJ XSYTMPXLMSCN BDFICJ“
- f_{129}: „XTEVF JSEJUOCMABSQ NJNQITUQTGHLSUYN“
- f_{130}: „AHDQNZI LLSK LMOSGLBSJWLFXCXUNNKTK “
- f_{131}: „KBBVEDDNJATRHESJUNAS QFYIVPCNIQWBQA“
- f_{132}: „VZSRPJOUYVLFZPIRQOYYWBTQXQQOAQKCROF“
- f_{133}: „FWQKIZVAIVWZTSRTCNUOZFD ISAFJAVRIXU“
- f_{134}: „WOZAFUBYLLGEEPWFOHRDEAYBZDAR IJAPTP“
- f_{136}: „COYSCK UMKMWNXXMRPUWESMHKN HGTHLDQ“
- f_{137}: „WYLPPAVSTVEBFNXHWYXILNEQSXZGOECWJIH“
- f_{140}: „OSAAXCTOGHJEFIUBYIMZDRIELKNGIUMGKJM“
- f_{141}: „EVSRZUBYZEGCOEDWCXGVSWECSHOSJGNWISA“
- f_{142}: „FFX TERUJYWMCAYYNNHLTFKOOVIAQNKBYNX“
- f_{144}: „TGAQH AODWTNYRFVB YSRZTXTOMSHATQETA“
- f_{148}: „SPUKHCDKPZEP PGWDVMSIBVD PBABSILRQC“
- f_{149}: „QFCSMUSHFRRIACYIPPSZECBLXVFKZDUZH R“
- f_{150}: „BTQCZZMDALWFQQLAILPOYHSFK WGQSHXAXZ“
- f_{152}: „IVQP JM IAQRFRMLTXYTABCJWHHVWOOXIF“
- f_{154}: „IQ RHHISEYIIHAIIGJEUHNGZYNFNKZWYNYB“
- f_{156}: „JTYQWPQAZIYDDKUEEOMOQ EY ZBX LMUBX“
- f_{157}: „LM WUXBVXLAXQOHMQONNPMILHKC WZKQBJ“
- f_{158}: „GJXVZZJJKR DWCAMMDXTL HGXTNKTHWLIQG“
- f_{160}: „VRDSJQXJNYNNDDAEFODSVNXIIOVNC JSLHB“
- f_{161}: „RRZVJTMSEZXMEJTSBY ODOHLD XAZWVAFBK“
- f_{162}: „HFQWRMVDPBRXLPL YZULSTEFHBCWOFPXKDS“

- f_{163}: „YKKAUOIOKZVETHYQY IRIFOLVLMXNTRHOXP“
- f_{165}: „VUZGR BTZUVBSTUIAACAYTYYEIXVFYRSDX“
- f_{167}: „XSUHXEPIJLDAMTDOHZLQBFZWY AUVEJBOPJ“
- f_{169}: „MIXQIAKNUSHKLMQPFAERTFMSRFUNOTOZSCF“
- f_{171}: „XRDZDAIFDVJCEWLYGOVIUNLEIWOQSTPFGEG“
- f_{172}: „LFVOFAURXQUDOWDXLICSFHKDNMLIQVT VIA“
- f_{173}: „AVIYHVCBDTXBEVUN YIJSRIVFJRSKLQFJNP“
- f_{174}: „MFXVMLHZTRVNMMVDIDCEDGJFPPTHQRMLOLO“
- f_{176}: „KSQMOKKNYB ADXGHLXZLHOSGKLYFEXCUDPF“
- f_{177}: „RMOAFWWHDQIIVDNYHWBKQNRHTUBRAATQWF“
- f_{178}: „VNNOJDCXMFGKBUAO YLOXSEAHLPPCJYY QT“
- f_{181}: „SDNBE HHM VDXAAUONCPWRBWBKYPMYLLDKZ“
- f_{182}: „XRWVWMSTRCBLGBMXHEMWFBO BDWXLAHTOPI“
- f_{184}: „IYDYLIVDVYWGFCI HYSHYVPXHZXWDYZOVDS“
- f_{187}: „ASSHKFEOJUUPGYIYSMT WUAPEQRGZNBIQHY“
- f_{189}: „HTGCOZFZFGSRKH UHUQWLCIJGCZMOWXADPR“
- f_{190}: „ODHAAVPWWXLOOYGOJHCSOQRGVOVRNBZOASC“
- f_{191}: „SBOQPGRJLNYGULSJLZKT MHVBEFMRDE NWF“
- f_{192}: „LWFIFJEKWA RCFYILGHMAHLS RLQXLFPS“
- f_{194}: „ORSXSCZQWFDM WDPJMDTICGDKRMGFCANLKE“
- f_{195}: „RKEBPGSXTCLFTLXJKUVCBY WM KYECZOI E“
- f_{196}: „FBAAAONBVQCDKXAXPUKBBDBYZZUSUXL QAF“
- f_{198}: „LRRAIVJYNULPUTNUMBRUFOZNKDTFSHOMIEF“
- f_{199}: „GHLZJPZQQPZFXSXMQCKHBHKWFMGKXJLXTWF“
- f_{200}: „YKZWBTJVEPXKD XEFMZSKPUWUIB JEZW OB“
- f_{201}: „FVJCZPGUGEFHFGEJMAXGGLULLKQQDYDE P“
- f_{202}: „JMLIWKPL T GNOBIPJXCDOYLMGM DHKJ AH“
- f_{203}: „N DWZCBRBIA QDYZUIUBFQUSEQLVIDPZWLA“
- f_{204}: „CTTFRVEYAMCEH KKBORJRGMWYZEMWGE OOA“
- f_{205}: „JL YETCZCGMWTKXVBI TSWAKPWENWDFUNKR“
- f_{206}: „PCUKGTIURBBF NDFWMXOARYD URMXFFQGBY“
- f_{209}: „WGRWLKBTOXUFPU VDAADYAMIDUKOWOIIWCN“
- f_{210}: „RMFYBAR YPSLEWM O GDO SZUZYRJZTIWK “
- f_{211}: „UQPMQYBIVSJDVMTVDBC XMICF SFTVENBYD“
- f_{213}: „YMZEUXOQZBZCXL FQXCYHQLI KLYGLL GTX“
- f_{214}: „MAQ RQE WSLEIGZJLYOCKRZMDCECJST GT“

(1) Schöne Gittervektoren sind (f, g) und $(s^i * f, s^i *)$ für $i = 1, \ldots, N-1$. Sie haben Länge

$$\|(f,g)\| = \sqrt{d_f + (d_f - 1) + d_g + d_g} = \sqrt{2d_f + 2d_g - 1} = \sqrt{53} = 7.28.$$

(2) Außerdem gibt es noch den Vektor $(1, 1, \ldots, 1, 0, 0, \ldots, 0)$ der Länge $\sqrt{N} = \sqrt{107} = 10.34$.
(3) Leider hilft hier auch die LLL-Abschätzung

$$\|b_1\| \leq 2^{\frac{2N-1}{2}} \|v\| \text{ für jeden Gittervektor } v \neq 0$$

nicht weiter, denn

$$2^{\frac{2N-1}{2}} = 2^{106.5} \approx 0.11 \cdot 10^{33}.$$

Wir haben früher die hinreichende Entschlüsselungsbedingung

$$2pd_r\|\widetilde{g}\|_\infty + \|\widetilde{f}\|_1 < \frac{q}{2}$$

für die Entschlüsselung mit $\widetilde{f}$ hergeleitet, wenn $\widetilde{f}$ modulo p invertierbar ist. Für die modulo p invertierbaren f_i haben wir

$$Q = \frac{2}{q} \cdot (2pd_r\|g_i\|_\infty + \|f_i\|_1)$$

gebildet und der Reihe nach sortiert, mit folgendem Ergebnis:

```
83.84 (i=96), 84.25 (i=214), 86.34 (i=140), 87.72 (i=127), 90.91 (i=130), 91.16 (i=134),
91.81 (i=200), 91.97 (i=174), 92.62 (i=101), 93.91 (i=154), 94.16 (i=100), 94.47 (i=191),
95.22 (i=192), 95.47 (i=123), 95.69 (i=206), 96.47 (i=178), 96.66 (i=106), 96.88 (i=131),
97.41 (i=109), 97.78 (i=85), 97.97 (i=205), 98.38 (i=122), 99.41 (i=136), 100.09 (i=196),
100.44 (i=126), 101.19 (i=189), 101.44 (i=142), 101.66 (i=111), 101.88 (i=92), 102.19 (i=181),
102.66 (i=121), 102.88 (i=160), 102.88 (i=91), 103.66 (i=161), 103.66 (i=94), 103.72 (i=177),
103.84 (i=125), 103.88 (i=141), 104.28 (i=210), 104.53 (i=156), 105.81 (i=107), 105.81 (i=211),
106.50 (i=157), 107.22 (i=137), 107.84 (i=148), 108.66 (i=144), 108.66 (i=190), 109.88 (i=203),
110.66 (i=195), 110.72 (i=213), 110.78 (i=209), 110.91 (i=112), 111.06 (i=108), 111.97 (i=128),
112.09 (i=201), 112.69 (i=165), 113.25 (i=173), 114.03 (i=93), 114.31 (i=132), 115.50 (i=204),
115.59 (i=129), 116.38 (i=95), 116.59 (i=202), 116.91 (i=149), 117.06 (i=116), 117.72 (i=199),
117.78 (i=176), 118.28 (i=182), 119.56 (i=162), 120.84 (i=198), 121.12 (i=167), 121.25 (i=99),
121.34 (i=169), 121.97 (i=102), 122.59 (i=187), 123.25 (i=119), 123.91 (i=158), 123.94 (i=84),
123.94 (i=98), 125.16 (i=113), 126.38 (i=171), 127.72 (i=163), 128.34 (i=133), 129.12 (i=172),
129.91 (i=184), 130.84 (i=152), 135.78 (i=194), 137.84 (i=103), 145.75 (i=150), 150.25 (i=88).
```

Für unsere Ausgangsvektoren f und g ist $Q = 1.84$.

Bei gleichen Startvektoren f und g haben wir nun q variiert, und zwar $q = 2^n$. Dazu dann $h \equiv f_q * g \bmod q$ berechnet, das Gitter aufgestellt, LLL-reduziert. Die Tabellen geben die Normen der Basisvektoren und den Q-Wert an.

Das erste Mal hatten wir bei $q = 65536 = 2^{16}$ Erfolg:

452.20	0.15	412.32	0.14	364.49	0.12	350.05	0.11	349.24	0.13
337.52	0.12	399.01	0.13	292.68	0.10	385.24	0.13	363.20	0.12
321.02	0.11	258.73	0.08	292.62	0.10	286.72	0.10	385.88	0.14
350.51	0.12	284.41	0.10	312.60	0.10	235.03	0.07	343.53	0.13
258.00	0.09	318.72	0.11	281.36	0.11	358.14	0.13	294.32	0.10
294.27	0.10	316.40	0.10	342.68	0.11	283.75	0.10	313.15	0.11
355.27	0.12	241.65	0.09	324.84	0.11	348.46	0.12	320.21	0.11
352.34	0.11	353.22	0.12	249.81	0.09	367.05	0.12	298.89	0.10
230.91	0.08	306.61	0.11	357.86	0.12	259.01	0.10	318.73	0.11
343.36	0.12	244.97	0.08	200.88	0.07	342.39	0.13	224.25	0.07
333.04	0.11	239.59	0.09	309.75	0.10	311.85	0.12	327.69	0.11
186.55	0.07	246.19	0.10	313.76	0.10	321.12	0.12	287.87	0.09
345.63	0.12	265.31	0.09	243.67	0.09	212.97	0.07	292.63	0.10
283.89	0.10	310.55	0.10	248.45	0.09	369.13	0.14	323.74	0.11
337.47	0.12	314.86	0.11	278.41	0.10	265.41	0.09	299.30	0.10
306.11	0.10	240.84	0.09	263.62	0.09	311.67	0.12	311.50	0.11
318.93	0.11	234.71	0.08	281.67	0.09	288.33	0.10	272.69	0.09
380.18	0.13	295.53	0.12	244.96	0.08	334.40	0.12	274.81	0.10
312.05	0.10	293.05	0.10	341.83	0.11	328.25	0.11	298.04	0.11
317.69	0.12	245.39	0.08	339.32	0.12	349.20	0.12	285.64	0.09
270.67	0.10	324.61	0.11	287.13	0.10	280.20	0.10	286.91	0.10
309.04	0.10	340.44	0.13	48574.70	39.64	48574.64	39.67	48574.73	39.62
202329.37	64.96	202329.31	64.96	213373.18	71.08	217542.92	73.82	191624.93	65.34
191624.77	65.35	198239.33	62.28	198239.37	62.25	204234.24	63.27	213459.51	69.58
218733.97	71.54	215984.30	70.94	171594.17	63.66	198740.82	68.76	182301.68	59.75
194169.78	65.59	207602.97	62.58	231140.16	69.74	191033.07	68.78	219720.10	75.25
216700.84	79.56	214453.89	79.99	203265.41	71.64	199664.59	71.91	194175.82	63.15
174323.49	68.10	204382.30	71.22	195092.22	70.21	180981.26	67.66	233981.26	79.07
182055.53	62.14	188118.41	74.75	189479.49	62.54	209223.90	74.89	174806.39	65.22
207367.75	80.00	201462.22	66.90	212939.28	75.27	177161.85	58.93	213092.11	75.59
183093.10	59.25	200939.74	61.91	196607.90	65.14	171790.21	64.56	208543.19	75.08
208402.75	69.26	170394.55	61.02	188697.73	63.71	236616.45	81.16	201384.31	67.28
188650.46	71.37	181577.73	63.77	188717.88	67.05	210874.38	69.03	164097.19	54.48
200826.08	65.93	192905.28	66.23	227685.62	74.39	179902.15	63.81	202593.58	68.89
171733.44	57.45	209286.36	84.36	150108.03	50.71	214574.48	68.60	203910.89	66.90
202495.56	73.53	210082.58	77.34	221588.52	73.21	227961.13	73.39	186579.26	58.20
181230.98	60.90	160773.68	55.24	193068.42	66.52	164465.79	53.85	200687.33	67.46
215988.27	73.18	210989.66	74.41	209726.59	65.34	161280.62	51.57	217898.27	79.67
198615.63	81.74	211812.83	77.51	199868.50	62.15	200167.03	69.95	209586.04	69.73
154261.57	52.74	212373.22	77.68	185594.68	58.59	207833.02	71.07	186610.00	73.44
176919.96	59.74	217584.77	78.17	179187.23	57.11	191481.39	65.54	190493.29	59.93
165525.86	61.82	209320.94	65.55	195357.53	66.08	208550.63	67.61	166645.24	58.74
198310.63	77.28	198260.12	61.63	184933.47	67.03	187950.56	63.68		

Hier war der Durchbruch. Wir versuchen zu entschlüsseln:

- f_1 ist nicht invertierbar modulo p.
- f_2 ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_3 ($Q = 0.12$, $\|(f_3, g_3)\| = 364.49$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_4 ($Q = 0.11$, $\|(f_4, g_4)\| = 350.05$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_5 ($Q = 0.13$, $\|(f_5, g_5)\| = 349.24$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.

- f_6 ist nicht invertierbar.
- Entschlüsselungsversuch mit f_7 ($Q = 0.13$, $\|(f_7, g_7)\| = 399.01$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_8 ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_9 ($Q = 0.13$, $\|(f_9, g_9)\| = 385.24$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{10} ($Q = 0.12$, $\|(f_{10}, g_{10})\| = 363.20$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{11} ($Q = 0.11$, $\|(f_{11}, g_{11})\| = 321.02$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{12} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{13} ($Q = 0.10$, $\|(f_{13}, g_{13})\| = 292.62$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{14} ($Q = 0.10$, $\|(f_{14}, g_{14})\| = 286.72$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{15} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{16} ($Q = 0.12$, $\|(f_{16}, g_{16})\| = 350.51$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{17} ($Q = 0.10$, $\|(f_{17}, g_{17})\| = 284.41$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{18} ($Q = 0.10$, $\|(f_{18}, g_{18})\| = 312.60$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{19} ($Q = 0.07$, $\|(f_{19}, g_{19})\| = 235.03$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{20} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{21} ($Q = 0.09$, $\|(f_{21}, g_{21})\| = 258.00$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{22} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{23} ($Q = 0.11$, $\|(f_{23}, g_{23})\| = 281.36$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{24} ($Q = 0.13$, $\|(f_{24}, g_{24})\| = 358.14$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{25} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{26} ($Q = 0.10$, $\|(f_{26}, g_{26})\| = 294.27$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{27} ($Q = 0.10$, $\|(f_{27}, g_{27})\| = 316.40$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{28} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{29} ($Q = 0.10$, $\|(f_{29}, g_{29})\| = 283.75$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{30} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{31} ($Q = 0.12$, $\|(f_{31}, g_{31})\| = 355.27$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{32} ist nicht invertierbar modulo p.
- f_{33} ist nicht invertierbar modulo p.
- f_{34} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{35} ($Q = 0.11$, $\|(f_{35}, g_{35})\| = 320.21$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{36} ($Q = 0.11$, $\|(f_{36}, g_{36})\| = 352.34$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{37} ($Q = 0.12$, $\|(f_{37}, g_{37})\| = 353.22$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{38} ist nicht invertierbar modulo p.

- Entschlüsselungsversuch mit f_{39} ($Q = 0.12$, $\|(f_{39}, g_{39})\| = 367.05$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{40} ($Q = 0.10$, $\|(f_{40}, g_{40})\| = 298.89$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{41} ist nicht invertierbar modulo p.
- f_{42} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{43} ($Q = 0.12$, $\|(f_{43}, g_{43})\| = 357.86$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{44} ist nicht invertierbar.
- f_{45} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{46} ($Q = 0.12$, $\|(f_{46}, g_{46})\| = 343.36$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{47} ist nicht invertierbar.
- Entschlüsselungsversuch mit f_{48} ($Q = 0.07$, $\|(f_{48}, g_{48})\| = 200.88$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{49} ($Q = 0.13$, $\|(f_{49}, g_{49})\| = 342.39$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{50} ist nicht invertierbar modulo p.
- f_{51} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{52} ($Q = 0.09$, $\|(f_{52}, g_{52})\| = 239.59$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{53} ($Q = 0.10$, $\|(f_{53}, g_{53})\| = 309.75$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{54} ($Q = 0.12$, $\|(f_{54}, g_{54})\| = 311.85$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{55} ($Q = 0.11$, $\|(f_{55}, g_{55})\| = 327.69$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{56} ($Q = 0.07$, $\|(f_{56}, g_{56})\| = 186.55$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{57} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{58} ($Q = 0.10$, $\|(f_{58}, g_{58})\| = 313.76$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{59} ($Q = 0.12$, $\|(f_{59}, g_{59})\| = 321.12$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{60} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{61} ($Q = 0.12$, $\|(f_{61}, g_{61})\| = 345.63$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{62} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{63} ($Q = 0.09$, $\|(f_{63}, g_{63})\| = 243.67$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{64} ($Q = 0.07$, $\|(f_{64}, g_{64})\| = 212.97$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{65} ($Q = 0.10$, $\|(f_{65}, g_{65})\| = 292.63$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{66} ($Q = 0.10$, $\|(f_{66}, g_{66})\| = 283.89$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{67} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{68} ($Q = 0.09$, $\|(f_{68}, g_{68})\| = 248.45$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{69} ($Q = 0.14$, $\|(f_{69}, g_{69})\| = 369.13$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{70} ($Q = 0.11$, $\|(f_{70}, g_{70})\| = 323.74$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{71} ist nicht invertierbar modulo p.

- f_{72} ist nicht invertierbar.
- Entschlüsselungsversuch mit f_{73} ($Q = 0.10$, $\|(f_{73}, g_{73})\| = 278.41$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{74} ($Q = 0.09$, $\|(f_{74}, g_{74})\| = 265.41$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{75} ($Q = 0.10$, $\|(f_{75}, g_{75})\| = 299.30$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{76} ($Q = 0.10$, $\|(f_{76}, g_{76})\| = 306.11$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{77} ($Q = 0.09$, $\|(f_{77}, g_{77})\| = 240.84$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{78} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{79} ($Q = 0.12$, $\|(f_{79}, g_{79})\| = 311.67$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{80} ($Q = 0.11$, $\|(f_{80}, g_{80})\| = 311.50$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{81} ($Q = 0.11$, $\|(f_{81}, g_{81})\| = 318.93$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{82} ($Q = 0.08$, $\|(f_{82}, g_{82})\| = 234.71$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{83} ($Q = 0.09$, $\|(f_{83}, g_{83})\| = 281.67$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{84} ($Q = 0.10$, $\|(f_{84}, g_{84})\| = 288.33$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{85} ($Q = 0.09$, $\|(f_{85}, g_{85})\| = 272.69$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{86} ($Q = 0.13$, $\|(f_{86}, g_{86})\| = 380.18$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{87} ($Q = 0.12$, $\|(f_{87}, g_{87})\| = 295.53$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{88} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{89} ($Q = 0.12$, $\|(f_{89}, g_{89})\| = 334.40$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{90} ($Q = 0.10$, $\|(f_{90}, g_{90})\| = 274.81$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{91} ist nicht invertierbar modulo p.
- f_{92} ist nicht invertierbar modulo p.
- f_{93} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{94} ($Q = 0.11$, $\|(f_{94}, g_{94})\| = 328.25$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{95} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{96} ($Q = 0.12$, $\|(f_{96}, g_{96})\| = 317.69$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{97} ($Q = 0.08$, $\|(f_{97}, g_{97})\| = 245.39$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{98} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{99} ($Q = 0.12$, $\|(f_{99}, g_{99})\| = 349.20$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{100} ($Q = 0.09$, $\|(f_{100}, g_{100})\| = 285.64$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{101} ($Q = 0.10$, $\|(f_{101}, g_{101})\| = 270.67$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- Entschlüsselungsversuch mit f_{102} ($Q = 0.11$, $\|(f_{102}, g_{102})\| = 324.61$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.

- f_{103} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{104} ($Q = 0.10$, $\|(f_{104}, g_{104})\| = 280.20$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{105} ist nicht invertierbar modulo p.
- f_{106} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{107} ($Q = 0.13$, $\|(f_{107}, g_{107})\| = 340.44$): „WIE FINDET MAN KURZE GITTERVEKTOREN“.
- f_{108} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{109} ($Q = 39.67$, $\|(f_{109}, g_{109})\| = 48574.64$): „EUKHFMWFNAZVZBSGI LYBNRIEKRPIJBSOLG“.
- Entschlüsselungsversuch mit f_{110} ($Q = 39.62$, $\|(f_{110}, g_{110})\| = 48574.73$): „VIPZWIWENOPEOVUYYILNRLHWNYEV JFJOOU“.
- Entschlüsselungsversuch mit f_{111} ($Q = 64.96$, $\|(f_{111}, g_{111})\| = 202329.37$): „ICPCWMHOHEAJWAZBNQBOSJBLFYLIRPKKZST“.
- Entschlüsselungsversuch mit f_{112} ($Q = 64.96$, $\|(f_{112}, g_{112})\| = 202329.31$): „RHHUZFEBIVVBCYXTG WZHNY NLHROHOIQSM“.
- Entschlüsselungsversuch mit f_{113} ($Q = 71.08$, $\|(f_{113}, g_{113})\| = 213373.18$): „IGBUZUQJEEKXNWISDWRNIHTAZGZHOMJCPLR“.
- f_{114} ist nicht invertierbar modulo p.
- f_{115} ist nicht invertierbar modulo p.
- f_{116} ist nicht invertierbar.
- Entschlüsselungsversuch mit f_{117} ($Q = 62.28$, $\|(f_{117}, g_{117})\| = 198239.33$): „NPUXHOLHHFAXIVEENVLCHQTXVFIZJBRVFZQ“.
- f_{118} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{119} ($Q = 63.27$, $\|(f_{119}, g_{119})\| = 204234.24$): „GGNMLWQ ZJYEVANFFAXTOITJLQNUEGLQEQH“.
- Entschlüsselungsversuch mit f_{120} ($Q = 69.58$, $\|(f_{120}, g_{120})\| = 213459.51$): „PFZHNUY RYHCYTTBKAOOOAHEZTISBZNNHJ“.
- f_{121} ist nicht invertierbar modulo p.
- f_{122} ist nicht invertierbar modulo p.
- f_{123} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{124} ($Q = 68.76$, $\|(f_{124}, g_{124})\| = 198740.82$): „UGYCYTYBTAEOWAVGRAAKDRVDJVWRMERPGHK“.
- Entschlüsselungsversuch mit f_{125} ($Q = 59.75$, $\|(f_{125}, g_{125})\| = 182301.68$): „Q CAVPTGZKXNHHJDKMCWPIJGFBCWRWFKRRI“.
- Entschlüsselungsversuch mit f_{126} ($Q = 65.59$, $\|(f_{126}, g_{126})\| = 194169.78$): „UUJLBQGTONJXJHZ VIMVXTQYPOKD UWMNDX“.
- Entschlüsselungsversuch mit f_{127} ($Q = 62.58$, $\|(f_{127}, g_{127})\| = 207602.97$): „MLWRABIP UGSRXBHUPJFKLUXCNGOYODBP E“.
- Entschlüsselungsversuch mit f_{128} ($Q = 69.74$, $\|(f_{128}, g_{128})\| = 231140.16$): „EVPTNKOJRMVBIYCJGTXCYX QOBRPMCRCKGK“.
- f_{129} ist nicht invertierbar modulo p.
- f_{130} ist nicht invertierbar.
- Entschlüsselungsversuch mit f_{131} ($Q = 79.56$, $\|(f_{131}, g_{131})\| = 216700.84$): „EPWRRZHM JLRYSS IUXDGPLEIHWVMFBIDXT“.
- f_{132} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{133} ($Q = 71.64$, $\|(f_{133}, g_{133})\| = 203265.41$): „KIUIGJOLRMYZSSXLOCPANFHBAVJCYKHASDJ“.
- Entschlüsselungsversuch mit f_{134} ($Q = 71.91$, $\|(f_{134}, g_{134})\| = 199664.59$): „AGYBJEWPMYOQINIGAFSZWDOFOJSBNCYH UB“.
- Entschlüsselungsversuch mit f_{135} ($Q = 63.15$, $\|(f_{135}, g_{135})\| = 194175.82$): „BEIXMMTXKWRFQFOQTETLKPAY JJX THGZZJ“.
- Entschlüsselungsversuch mit f_{136} ($Q = 68.10$, $\|(f_{136}, g_{136})\| = 174323.49$): „BXDWSOFIHTGFODBXUAAHSLWPJJEZXDAACOB“.

- f_{137} ist nicht invertierbar modulo p.
- f_{138} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{139} ($Q = 67.66$, $\|(f_{139}, g_{139})\| = 180981.26$): „LZJMFZDQ BFXLIEZSE VRFVPGYO HTXYKTP“.
- Entschlüsselungsversuch mit f_{140} ($Q = 79.07$, $\|(f_{140}, g_{140})\| = 233981.26$): „MUHPVNCNNWS RTFNTFKOXCOIDQWUKNYROYA“.
- f_{141} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{142} ($Q = 74.75$, $\|(f_{142}, g_{142})\| = 188118.41$): „SOBJTPCDLITIEOP PKORZEKQBMNJEFZUXEB“.
- Entschlüsselungsversuch mit f_{143} ($Q = 62.54$, $\|(f_{143}, g_{143})\| = 189479.49$): „CJPDYJJJVDGJD CJCVKBNLQTRXGQCLUBNWX“.
- f_{144} ist nicht invertierbar modulo p.
- f_{145} ist nicht invertierbar modulo p.
- f_{146} ist nicht invertierbar modulo p.
- f_{147} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{148} ($Q = 75.27$, $\|(f_{148}, g_{148})\| = 212939.28$): „KLSNYATICGURFYUHXYHYSVHJYNYQSFYEWB “.
- f_{149} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{150} ($Q = 75.59$, $\|(f_{150}, g_{150})\| = 213092.11$): „XFZINTSSAGAEAQFWXYAYOREDYR MSTWHLOJ“.
- f_{151} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{152} ($Q = 61.91$, $\|(f_{152}, g_{152})\| = 200939.74$): „FHUVMQNMJRKDDWY LQDZTZ GJXXXFOFLIRB“.
- Entschlüsselungsversuch mit f_{153} ($Q = 65.14$, $\|(f_{153}, g_{153})\| = 196607.90$): „LMJILAO D ZCKYTBELVRURCRNVMHLKJMDR“.
- Entschlüsselungsversuch mit f_{154} ($Q = 64.56$, $\|(f_{154}, g_{154})\| = 171790.21$): „BMFO ZGYYPZJSKXQTMLNIGRZBBCEPPJHEFE“.
- f_{155} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{156} ($Q = 69.26$, $\|(f_{156}, g_{156})\| = 208402.75$): „CGBWBKSXZYZGKGSRZ RQMZNZLQUXPNHQUOM“.
- Entschlüsselungsversuch mit f_{157} ($Q = 61.02$, $\|(f_{157}, g_{157})\| = 170394.55$): „NKRFWVUHIXRJPEBKGVUFCCPDXBEBQFKIALG“.
- f_{158} ist nicht invertierbar modulo p.
- f_{159} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{160} ($Q = 67.28$, $\|(f_{160}, g_{160})\| = 201384.31$): „EYPQTLJKLZRXIWESXRCTYGNVHALPZNQMIUJ“.
- Entschlüsselungsversuch mit f_{161} ($Q = 71.37$, $\|(f_{161}, g_{161})\| = 188650.46$): „XCMHTUEMYCN NCLOYFFMYGTBFPQXJR AVSU“.
- Entschlüsselungsversuch mit f_{162} ($Q = 63.77$, $\|(f_{162}, g_{162})\| = 181577.73$): „JXEJFBPRPRKSEH DSOBLBAR GVVXCM JJNH“.
- Entschlüsselungsversuch mit f_{163} ($Q = 67.05$, $\|(f_{163}, g_{163})\| = 188717.88$): „YLHAHGPSVHOOCTSJLOKWLQFFMXWBLJXGHJM“.
- f_{164} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{165} ($Q = 54.48$, $\|(f_{165}, g_{165})\| = 164097.19$): „OBEFOCOJRJNTUUAOXDNPQMMQWAYKEMSWKGC“.
- Entschlüsselungsversuch mit f_{166} ($Q = 65.93$, $\|(f_{166}, g_{166})\| = 200826.08$): „WHNDJHAPHVLFTMGVZGCOGDNBPIUTIQOAZ U“.
- Entschlüsselungsversuch mit f_{167} ($Q = 66.23$, $\|(f_{167}, g_{167})\| = 192905.28$): „CBSKGLYCWUGYZCBHCNQKMFWSBD YJXLQEFN“.
- Entschlüsselungsversuch mit f_{168} ($Q = 74.39$, $\|(f_{168}, g_{168})\| = 227685.62$): „IRSXBTJEOCP VSIOJJHGOQUUEQMWQBOIINJ“.
- Entschlüsselungsversuch mit f_{169} ($Q = 63.81$, $\|(f_{169}, g_{169})\| = 179902.15$): „YNDDFNJQYBLEBSIHCRXICSU NVVKOEDQKSJ“.

- Entschlüsselungsversuch mit f_{170} ($Q = 68.89$, $\|(f_{170}, g_{170})\| = 202593.58$): „TBAVQRFUBKJXLSKEIHIHPKSIONLFSUVRASB“.
- Entschlüsselungsversuch mit f_{171} ($Q = 57.45$, $\|(f_{171}, g_{171})\| = 171733.44$): „KTTEUHSFG GKNZINNUMNAOETKXIZAGYEILU“.
- Entschlüsselungsversuch mit f_{172} ($Q = 84.36$, $\|(f_{172}, g_{172})\| = 209286.36$): „WRBRTOD CJ IZ WHCNOTKJMW GETNUBBED“.
- f_{173} ist nicht invertierbar modulo p.
- f_{174} ist nicht invertierbar modulo p.
- f_{175} ist nicht invertierbar modulo p.
- f_{176} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{177} ($Q = 77.34$, $\|(f_{177}, g_{177})\| = 210082.58$): „ZBSNFXXTCHXOOJXFMMBAFJZBVPVHJMQTNFZ“.
- Entschlüsselungsversuch mit f_{178} ($Q = 73.21$, $\|(f_{178}, g_{178})\| = 221588.52$): „PSEXNSAZGNCGYDQQEIIOPLBOPHQNFZJDQOJ“.
- f_{179} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{180} ($Q = 58.20$, $\|(f_{180}, g_{180})\| = 186579.26$): „LFV ARWTXSALSGAPZXYAVGZEIQRQGCQUVIR“.
- Entschlüsselungsversuch mit f_{181} ($Q = 60.90$, $\|(f_{181}, g_{181})\| = 181230.98$): „NCWQPV KYRYHUAKIEGZPQMIGTPURPJQKDIB“.
- f_{182} ist nicht invertierbar.
- Entschlüsselungsversuch mit f_{183} ($Q = 66.52$, $\|(f_{183}, g_{183})\| = 193068.42$): „VJCQLTGTRAZZZILDSAERQYQFCHIOWIVBVEF“.
- Entschlüsselungsversuch mit f_{184} ($Q = 53.85$, $\|(f_{184}, g_{184})\| = 164465.79$): „WUVFXSEIAIGOJREM OQKGZWKNEBIMUZWHF“.
- f_{185} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{186} ($Q = 73.18$, $\|(f_{186}, g_{186})\| = 215988.27$): „GUM OSQUVIOZQWZPAXKMDHVARDUJYTBTAEP“.
- f_{187} ist nicht invertierbar.
- f_{188} ist nicht invertierbar modulo p.
- f_{189} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{190} ($Q = 79.67$, $\|(f_{190}, g_{190})\| = 217898.27$): „R SUCLSIVVYLGSMYFBOWEYNSXEAXSPFXTDE“.
- Entschlüsselungsversuch mit f_{191} ($Q = 81.74$, $\|(f_{191}, g_{191})\| = 198615.63$): „GZM KSDEIPKHWTESPZMBGKMUF QX CNKVDU“.
- Entschlüsselungsversuch mit f_{192} ($Q = 77.51$, $\|(f_{192}, g_{192})\| = 211812.83$): „LUVCVSWNNZJQVYODKYXRBLSUCDDEBBGXVZX“.
- Entschlüsselungsversuch mit f_{193} ($Q = 62.15$, $\|(f_{193}, g_{193})\| = 199868.50$): „DIYSPMJPVT KUWIJHLCWFBKAUKLTOCQORAS“.
- Entschlüsselungsversuch mit f_{194} ($Q = 69.95$, $\|(f_{194}, g_{194})\| = 200167.03$): „AZTVWHQXY NXDYUPDBSTSQIZFNXZCFDQNCH“.
- Entschlüsselungsversuch mit f_{195} ($Q = 69.73$, $\|(f_{195}, g_{195})\| = 209586.04$): „QXCGGAQCSBAHATMD NDMQQBYSXF OAMTI S“.
- Entschlüsselungsversuch mit f_{196} ($Q = 52.74$, $\|(f_{196}, g_{196})\| = 154261.57$): „BZIYQFESRHJ F EEXTBIVZKOJLGWZTSJS N“.
- f_{197} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{198} ($Q = 58.59$, $\|(f_{198}, g_{198})\| = 185594.68$): „KJRCTSABCLHXDVKQTZWYWQCJQGGADGPTMXS“.
- Entschlüsselungsversuch mit f_{199} ($Q = 71.07$, $\|(f_{199}, g_{199})\| = 207833.02$): „RNSDGQKTN ZQSYQVWPMGMRGHAQTQQAHFTEO“.
- Entschlüsselungsversuch mit f_{200} ($Q = 73.44$, $\|(f_{200}, g_{200})\| = 186610.00$): „KZHHETYPITMPDA QVQLJWDZNKMRWXYZCDIO“.
- f_{201} ist nicht invertierbar.
- Entschlüsselungsversuch mit f_{202} ($Q = 78.17$, $\|(f_{202}, g_{202})\| = 217584.77$): „HBKVJ THFJGOAQDFBMLWUXZFXZFZFSRNDKH“.

- Entschlüsselungsversuch mit f_{203} ($Q = 57.11$, $\|(f_{203}, g_{203})\| = 179187.23$): „EIHL EZMENXIGFZRBUONYVOMUNLXOOUAROY“.
- Entschlüsselungsversuch mit f_{204} ($Q = 65.54$, $\|(f_{204}, g_{204})\| = 191481.39$): „VKDNJANGOLMIICXTM RXLT TLESYEY DOXY“.
- f_{205} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{206} ($Q = 61.82$, $\|(f_{206}, g_{206})\| = 165525.86$): „KJRGRQFRZVCODZPTUOY RYPP QGQWUQWEJD“.
- f_{207} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{208} ($Q = 66.08$, $\|(f_{208}, g_{208})\| = 195357.53$): „PSMBTNCHIUTEAKEDHXYNDOUMBQKYGMMGGJL“.
- f_{209} ist nicht invertierbar modulo p.
- f_{210} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{211} ($Q = 77.28$, $\|(f_{211}, g_{211})\| = 198310.63$): „PWNTEYSEHIUDHHBLKOUOTPNTIMIAANUARPW“.
- f_{212} ist nicht invertierbar modulo p.
- Entschlüsselungsversuch mit f_{213} ($Q = 67.03$, $\|(f_{213}, g_{213})\| = 184933.47$): „TAWZVEKXMNXYALHAUMZDFIIDRUXGCMOXFH “.
- f_{214} ist nicht invertierbar modulo p.

Wir formulieren nochmals ein Kriterium:

SATZ. *Gilt für die NTRU-Parameter die Ungleichung*

$$(2pd_r + \sqrt{N})^2 \cdot 2^{N+\frac{3}{2}} < q,$$

ist $(\widetilde{f}, \widetilde{g})$ der erste Vektor einer LLL-reduzierten Basis von Λ_h und ist $\widetilde{f}$ invertierbar modulo p, so kann man mit $\widetilde{f}$ jede Nachricht (richtig) entschlüsseln.

Beweis: Die LLL-Abschätzungen liefern für den ersten Vektor einer LLL-reduzierten Gitterbasis

$$\|(\widetilde{f}, \widetilde{g})\| \leq 2^{\frac{2N-1}{4}} (\det \Lambda_h)^{\frac{1}{2N}} = 2^{\frac{2N-1}{4}} \sqrt{q}.$$

In einem früheren Lemma haben wir gezeigt: Ist $2pd_r\|\widetilde{d}\|_\infty + \|\widetilde{f}\|_1 < \frac{q}{2}$, so kann man mit $\widetilde{f}$ alle Nachrichten entschlüsseln. Nun gilt

$$\|\widetilde{g}\|_\infty \leq \|\widetilde{g}\| \leq \|(\widetilde{f}, \widetilde{g})\|$$

und

$$\|\widetilde{f}\|_1 \leq \sqrt{N}\|\widetilde{f}\| \leq \sqrt{N}\|(\widetilde{f}, \widetilde{g})\|,$$

womit jetzt folgt

$$\begin{aligned} 2pd_r\|\widetilde{g}\|_\infty + \|\widetilde{f}\|_1 &\leq (2pd_r + \sqrt{N})\|(\widetilde{f}, \widetilde{g})\| \leq (2pd_r + \sqrt{N}) \cdot 2^{\frac{2N-1}{4}} \sqrt{q} = \\ &= (2pd_r + \sqrt{N}) \cdot 2^{\frac{N}{2}+\frac{3}{4}} \cdot \frac{1}{2}\sqrt{q} = \sqrt{(2pd_r + \sqrt{N}) \cdot 2^{N+\frac{3}{2}}} \cdot \frac{1}{2}\sqrt{q} < \\ &< \sqrt{q} \cdot \frac{1}{2}\sqrt{q} = \frac{1}{2}q, \end{aligned}$$

sodass die Voraussetzungen des genannten Lemmas erfüllt sind. Daher kann man mit $\widetilde{f}$ entschlüsseln. ■

Bemerkung: Es ist klar, dass die NTRU-Parameter so gewählt werden müssen, dass die Voraussetzungen des letzten Satzes nicht erfüllt sind.

6. Ein weiterer Gittereinsatz

LEMMA. *Sei h öffentlicher NTRU-Schlüssel zu den Parametern (N, p, q, d_f, d_g, d_r) und $m \in [-1, 1]^N$ eine mit $r \in L_N(d_r, d_r)$ zu $e \equiv r * h + m \bmod q$ verschlüsselte Nachricht. Sei $\Lambda_h = \{(x, y) \in \mathbb{Z}^{2N} : x * h \equiv py \bmod q\}$ das zugehörige NTRU-Gitter. Dann ist*

$$v = (pr, e - m) \in \Lambda_h$$

und

$$\|(0, e) - v\| \leq \sqrt{2p^2d_r + N}.$$

Beweis: Aus der Gleichung $e \equiv r * h + m \bmod q$ folgt durch Multiplikation mit p

$$pr * h \equiv p(e - m) \bmod q,$$

was nach Definition $(pr, e - m) \in \Lambda_h$ impliziert. Damit folgt

$$\|(0, e) - (pr, e - m)\| = \|(-pr, m)\| = \sqrt{\sum_{i=0}^{N-1} p^2r_i^2 + \sum_{i=0}^{N-1} m_i^2} \leq \sqrt{p^2 \cdot 2d_r + N},$$

was sofort die Behauptung liefert. ■

Bemerkungen:

(1) Könnten wir zu $(0, e)$ alle Gitterpunkte v bestimmen, die höchstens Abstand $\sqrt{2p^2d_r + N}$ von $(0, e)$ haben, so könnten wir die Ausgangsnachricht m bestimmen.

(2)

N	p	q	d_f	d_g	d_r	$\sqrt{2d_f + 2d_g - 1}$	$\sqrt{2p^2 d_r + N}$
11	3	32	4	4	3	3.87	8.06
107	3	64	15	15	5	7.68	14.04
167	3	128	61	61	18	15.59	22.16
263	3	128	50	50	16	14.11	23.47
503	3	256	216	216	55	29.38	38.64

(3) Es ist für große N kein schnelles Verfahren bekannt, das obige Aufgabe löst. (CVP - Closest Vector Problem)