Institut für Formale Methoden der Informatik
Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Diplomarbeit Nr.3286

Varianten des NTRU-Kryptosystems

Patrick Bächtle

Studiengang :	Mathematik
Prüfer :	Prof. Dr. Volker Diekert
Betreuer :	Dr. Manfred Kufleitner
begonnen am :	15.12.2011
beendet am :	13.09.2012
CR – Klassifikation :	F.4.3, G.1.3, G.2.1, G.3, J.2

Vorwort

Die vorliegende Arbeit behandelt das NTRU-Kryptosystem, welches zu den Public-Key-Kryptosystemen geört. Bei den Public-Key-Kryptosystemen gibt es jeweils einen öffentlichen Schlüsselanteil zum verschlüsseln und einen passenden geheimen Schlüsselanteil, der zum Entschlüsseln benötigt wird.
Wie die Bezeichnung bereits vermuten lässt, ist verschlüsseln für jeden möglich, da dieser Schlüsselanteil öffentlich zugänglich ist. Einem Angreifer eines Kryptosystems darf es also nicht möglich sein aus der Kenntnis des öffentlichen Schlüssels die ursprüngliche Nachricht zu rekonstruieren. Um dies zu gewährleisten greifen Kryptosysteme auf schwer zu lösende mathematische Probleme zurück.
Das RSA-Verfahren aus dem Jahr 1977 benutzt hierzu die Schwierigkeit große Zahlen zu faktorisieren. Wählt man die für das RSA-Verfahren benutzten Zahlen groß genug, so geht man davon aus, dass ein Angreifer das Verfahren nicht in kurzer Zeit brechen kann. Beweisbar ist diese Sicherheit nicht, auf Experimenten basierende Beobachtungen zeigen jedoch, dass bei geeigneter Parameterwahl die Laufzeit solcher Angriffe sehr viel Zeit beansprucht.
Ein anderes Public-Key-Kryptosystem, das Diffie-Hellman-System, basiert auf einem anderen schwer zu lösendem mathematischen Problem. Hierbei verlässt man sich auf die Schwierigkeit des diskreten Logarithmus in endlichen Gruppen. Es ist auch hier die Sicherheit des Verfahrens nicht beweisbar. Für genügend große Parameter liegt aber die gleiche Situation wie beim RSA-Verfahren vor.
Ein Verfahren mit beweisbarer Sicherheit ist das Vernom-One-Time-Pad. Dieses Verfahren zeichnet sich einerseits dadurch aus, dass die Schlüssel die gleiche Länge wie die zu verschlüsselnden Nachrichten haben müssen. Bei längeren Nachrichten führt dies zu einem großen Aufwand bei der Schlüsselerzeugung. Diese Ineffizienz ist der Grund, warum das Vernom-One-Time-Pad in der Praxis nur selten Anwendung findet.

In dieser Diplomarbeit wollen wir das NTRU-Verfahren genauer betrachten. Das NTRU-Kryptosystem wurde 1996 von Hoffstein, Pipher und Silverman im Rahmen der RUMP-Session auf der CRYPTO '96 vorgestellt. Erst kürzlich wurde es als Standard in der Finanzbranche zertifiziert. NTRU ist ein Public-Key-Kryptosystem, bei dem Ver- und Entschlüsselung Operationen im Quotienten eines Polynomrings (genauer: $R = \mathbb{Z}[x]/x^N - 1$) sind. Die Schlüssel, die für das Verfahren benutzt werden sind demnach Polynome aus diesem Polynomring. Wie auch bei den anderen Public-Key-Systemen kann die Sicherheit bei NTRU nicht bewiesen werden.
Die Sicherheit basiert auf der Annahme, dass faktorisieren eines Polynoms in R schwer ist. Wie auch schon bei RSA und Diffie-Hellman stützt sich diese Annahme auf experimentelle Ergebnisse, die auf eine exponentielle Laufzeit der Angriffsversuche hindeuten.

Im Rahmen dieser Diplomarbeit werden wir grundlegend das NTRU-Verfahren und einige Angriffe vorstellen. Anschließend wollen wir den algebraischen Grundraum R mit einer leicht modifizierten Version austauschen und so modifizierte Variante von NTRU genauer untersuchen. Im Vordergrund dieser Untersuchung steht die Funktionalität des Verfahrens, da schon beim Standard Verfahren Entschlüsselungsfehlern auftreten können. Des Weiteren wollen die modifizierten Varianten auf ihre theoretische Sicherheit hin überprüfen. Dabei werden wir insbesondere auf die sogenannten Gitterangriffe gegen ein NTRU-System eingehen.

Inhaltsverzeichnis

1 Grundlagen

1.1 Ring der Konvulotionspolynome

Der für das NTRU-Verfahren zugrunde liegende Algebraische Raum ist der Ring der Konvolutionspolynome $R := \mathbb{Z}/(x^N - 1), N \in \mathbb{N}$. Daraus ergibt sich direkt, dass gilt $x^N = 1$, es werden also nur Polynome vom Grad $< N$ betrachtet. Die Addition zweier Polynome $f, g \in R$ ist die übliche Komponentenweise Addition der Koeffizienten; für $f = f_0 + f_1 x + \ldots + f_{N-1} x^{N-1}$ und $g = g_0 + g_1 x + \ldots + g_{N-1} x^{N-1}$ ergibt sich

$$f + g = \sum_{i=0}^{N-1} (f_i + g_i) x^i$$

Die Multiplikation in R ist gegeben als das Konvolutionsprodukt.
Es ist $f \circledast g = h$ mit

$$h_k = \sum_{i=0}^{k} f_i \cdot g_{k-i} + \sum_{i=k+1}^{N-1} f_i \cdot g_{N+k-i} = \sum_{i+j \equiv k \ mod \ N} f_i \cdot g_j$$

Definition 1.1: Sei $F \in R$. Die Koeffizientenweite von F definieren wir als

$$|F|_\infty = \max_{1 \leq i \leq N} F_i - \min_{1 \leq i \leq N} F_i$$

Weiter sei die L^2 Norm auf R eines Polynoms gegeben als

$$|F|_2 = \left(\sum_{i=1}^{N} (F_i - \overline{F}_i) \right)^{\frac{1}{2}}, \qquad wobei \quad \overline{F} = \frac{1}{N} \sum_{i=1}^{N} F_i$$

Definition 1.2: Seien $d_1, d_2 \in \mathbb{N}$.
Wir definieren den Teilraum der Polynome $\mathcal{L}(d_1, d_2) \subset R$ als

$$\mathcal{L}(d_1, d_2) = \Big\{ F \in R \Big| \begin{array}{r} \text{F besitzt } d_1 \text{ Koeffizienten gleich 1,} \\ d_2 \text{ Koeffizienten gleich -1, der Rest gleich 0} \end{array} \Big\}$$

Bemerkung 1.3: Da beim NTRU-Verfahren die Polynome $f, g, \phi \in R_p$ mit $p = 3$ gewählt werden (siehe Abschnitt 2.1), so haben diese Koeffizienten aus $\{-1, 0, 1\}$ und sind damit in $\mathcal{L}(d_1, d_2)$ enthalten.
Wir definieren daher für die einzelnen Polynome:

$$\mathcal{L}_f := \mathcal{L}(d_f, d_f - 1), \qquad \mathcal{L}_g := \mathcal{L}(d_g, d_g), \qquad \mathcal{L}_\phi := \mathcal{L}(d, d)$$

Mit dieser Festlegung ergeben sich für die L^2-Normen der Polynome folgende Werte:

$$|f|_2 = \sqrt{2d_f - 1 - N^{-1}}, \qquad |g|_2 = \sqrt{2d_g}, \qquad |\phi|_2 = \sqrt{2d}$$

1.2 Gitter

Definition 1.4: Seien $b_1, b_2, \ldots, b_d \in \mathbb{R}^n$ linear unabhängige Vektoren, $d \leq n$.
Das von den b_i erzeugte Gitter $\mathcal{L}(b_1, \ldots, b_d)$ ist die Menge aller ganzzahligen Linearkombinationen

$$\mathcal{L}(b_1, \ldots, b_d) = \left\{ v \in \mathbb{R}^n : v = \sum_{i=1}^{d} a_i \cdot b_i; \quad a_i \in \mathbb{Z} \right\}$$

Die Menge $B = \{b_1, \ldots, b_d\}$ nennen wir Gitterbasis von $\mathcal{L}(B)$ und d die Dimenson $dim(\mathcal{L}(B))$ des vorliegenden Gitters. Für Gitterangriffe gegen ein NTRU System (vgl. Kapitel 4) betrachten wir nur Gitter mit vollem Rang, d.h. $d = n$.
Üblicherweise wird die Basis eines Gitters als Matrix dargestellt, wobei die einzelnen Vektoren b_i als Zeilen aufgefasst werden:

$$B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_d \end{pmatrix}$$

Bemerkung 1.5: Die Basis eines Gitters ist nicht eindeutig, wie wir mit den nächsten Definitionen sehen werden.

Definition 1.6: Eine unimodulare Matrix ist eine ganzzahlige Matrix T mit $det(T) = \pm 1$
$GL_n(\mathbb{Z})$ ist die Menge der unimodularen Matrizen.

Definition 1.7: Um eine Gitterbasis B in eine andere Basis $\widetilde{B}$ zu überführen wird B mit einer unimodularen Matrix $T \in GL_n(\mathbb{Z})$ multipliziert

$$\widetilde{B} = B \cdot T$$

Definition 1.8: Die Determinante eines Gitters $\mathcal{L}(B)$ mit vollem Rang ist definiert als die Determinante der zugehörigen Basismatrix

$$det(\mathcal{L}(B) = det(B))$$

Bemerkung 1.9: Es ist direkt zu sehen, das die Gitterdeterminante eine invariante Größe in einem Gitter ist,
denn seien $B, \widetilde{B}$ zwei verschiedene Gitterbasen eines Gitters, dann gilt $\widetilde{B} = B \cdot T$ für ein $T \in GL_n(\mathbb{Z})$ und damit

$$det(B) = det(\widetilde{B} \cdot T) = det(\widetilde{B}) \cdot det(T)$$

Definition 1.10: Sei $\mathcal{L}(b_1, \ldots, b_n)$ ein Gitter.
Die folgende Menge nennen wir Grundmasche zur Basis $(b_1, \ldots, b_n)$

$$G = \Big\{ \sum_{i=1}^{n} t_i \cdot b_i | 0 \leq t_i < 1 \quad , t \in \mathbb{N}, i \in [1, n] \Big\}$$

Bemerkung 1.11: Geometrisch lässt sich die Gitterdeterminante als Volumen der Grundmasche des Gitters auffassen.

1.3 Gitter Probleme

In einem gegebenem Gitter $\mathcal{L}(B)$ gibt es zwei fundamentale Probleme die man betrachten kann. Diese sind:

Shortest Vector Problem (SVP):
Finde zu einer gegebenen Gitterbasis B den kürzesten, von 0 verschiedenen, Vektor in $\mathcal{L}(B)$.

Closest Vector Problem (CVP):
Bei gegebener Gitterbasis B und Zielvektor t (es kann auch $t \notin \mathcal{L}(B)$ gelten, finde den Gitterpunkt $v \in \mathcal{L}(B)$ mit minimalen Abstand zu t.

Bemerkung 1.12: Eine kurze Überlegung zeigt, das kürzeste und nächste Vektoren nicht eindeutig sein müssen, denn zum Beispiel sind in $\mathbb{Z}^2$ die Vektoren

$$v = \begin{pmatrix} 0 \\ \pm 1 \end{pmatrix} \quad und \quad w = \begin{pmatrix} \pm 1 \\ 0 \end{pmatrix}$$

alle kürzeste Vektoren.
Mit steigender Gitterdimension wird es auch schwieriger SVP und CVP zu lösen.
Um ein NTRU System anzugreifen wird das Schlüsselrekonstruktionsproblem auf ein SVP zurückgeführt (siehe Kapitel 3), deswegen wollen wir etwas näher auf das SVP eingehen.
Das Shortest Vecotr Problem gehört zu den NP-vollständigen Problemen, d.h. es existiert kein bekannter Algorithmus des das Problem in Polynomialzeit löst.

Um nun eine genauere Aussage über die Länge eines kürzesten Vektors in einem gegebenen Gitter $\mathcal{L}$ zu machen benötigen wir

Definition 1.13: (Heuristik von Gauß)
Sei $\mathcal{L}$ ein beliebiges Gitter mit $dim(\mathcal{L}) = d$, dann erwartet man die Länge eines kürzesten Vektors v im Gitter mit

$$det(\mathcal{L})^{\frac{1}{d}} \sqrt{\frac{d}{2\pi e}} \leq |v|_2 \leq det(\mathcal{L})^{\frac{1}{d}} \sqrt{\frac{s}{\pi e}}$$

2 Das NTRU Verfahren

In diesem Kapitel wollen wir das ursprüngliche NTRU-Verfahren, welches von Jeffrey Hoffstein, Jill Pipher und Joseph H. Silverman 1996 eingeführt wurde, beschreiben.

2.1 Erzeugung des öffentlichen Schlüssels

Für die Erzeugung eines öffentlichen Schlüssels $h \in R_q$ werden zufällig zwei Polynome $f \in \mathcal{L}_f,\ g \in \mathcal{L}_g$ gewählt. Insbesondere muss f sowohl in R_q, als auch in R_p invertierbar sein, d.h. zu f existiert jeweils ein F_q, F_p so dass

$$F_q \circledast f \equiv 1\ (mod\ q) \qquad und \qquad F_p \circledast f \equiv 1\ (mod\ p) \tag{1}$$

gilt.
Da es sich bei R_q, R_p nicht notwendigerweise um Körper handelt (da $X^N - 1$ nicht irreduzibel sein muss), kann es nicht-Invertierbare Polynome f geben. Wir werden aber noch sehen, das f bei geeigneter Parameterwahl mit hoher Wahrscheinlichkeit Invertierbar ist.

Nun kann der öffentliche Schlüssel berechnet werden.
Dieser ergibt sich zu

$$h \equiv F_q \circledast g\ (mod\ q) \tag{2}$$

2.1.1 Rotationsinvarianz

Definition: Sei $v = (v_0, v_1, \ldots, v_{N-1}) \in \mathbb{Z}^N$ ein beliebiger Vektor.
Unter einer Rotation v_{r_l} von v um l Stellen versteht man einen zyklischen Rechtsshift derart, dass

$$v_{r_l} = (v_{0-l\ mod\ N}, v_{1-l\ mod\ N}, \ldots, v_{N-1-l\ mod\ N})$$

Folgerung: Sei $f \in R_p$ so gewählt, dass Gleichung (2) erfüllt ist. Die Rotation f_{r_l}, mit zugehörigem Inversen in R_p, ist ein gültiger privater Schlüssel, mit dem korrektes Entschlüsseln möglich ist.

Beweis : Sei f_{r_l} eine Rotation von f.
Betrachte

$$\sum_{i+j \equiv k\ mod\ N} f_{i-l} \cdot g_j \qquad \text{mod } q$$

$$\sum_{(i+l)+j \equiv k\ mod\ N} f_i \cdot g_j \qquad \text{mod } q$$

$$\sum_{i+j \equiv (k-l)\ mod\ N} f_i \cdot g_j \qquad \text{mod } q$$

Was genau dem Summand g_{k-l} entspricht. □

2.2 Verschlüsselung

Alice möchte nun eine Nachricht $m \in \mathcal{L}_m$ an Bob senden. Dazu wählt sie ein weiteres Zufallspolynom $\phi \in \mathcal{L}_\phi$, welches wir als Störpolynom bezeichnen, und benutzt Bob's öffentlichen Schlüssel h um

$$e \equiv p\phi \circledast h + m \ (mod\ q)$$

zu berechnen.
Diese verschlüsselte Nachricht e übermittelt Alice nun an Bob.

2.3 Entschlüsselung

Bob erhält die von Alice verschlüsselte Nachricht e und entschlüsselt diese mit seinem privaten Schlüsssel f.
Aus Effizienzgründen sollte bei der Erzeugung von h bereits F_q und F_p gespeichert werden.
Im ersten Schritt berechnet Bob

$$a \equiv f \circledast e \ (mod\ q)$$

Um nun die ursprüngliche Nachricht zu erhalten berechnet Bob im zweiten Schritt

$$m \equiv F_p \circledast a \ (mod\ p)$$

Korrektheit : Für das im ersten Schritt berechnete Polynom a gilt:

$$\begin{aligned} a \equiv f \circledast e &\equiv f \circledast p\phi \circledast h + f \circledast m \ (mod\ q) \\ &\underset{(4)}{=} f \circledast p\phi \circledast F_q \circledast g + f \circledast m \ (mod\ q) \\ &\underset{(3)}{=} p\phi \circledast g + f \circledast m \ (mod\ q) \end{aligned}$$

Bei geeigneter Parameterwahl (siehe Kapitel 2.4) ist es bei dem letzten Polynom immer möglich die Koeffizienten aus dem Intervall $[-q/2, q/2]$ zu wählen, was zu Gleichheit über $\mathbb{Z}[x]/(x^N - 1)$ (und nicht nur über R_q) führt. Reduzieren wir in diesem Polynom die Koeffizienten $mod\ p$ so erhält man direkt $f \circledast m$, bei dem wir durch einfache Multiplikation mit F_p die ursprüngliche Nachricht m wiederherstellen. □

2.4 Parameterwahl

Wie wir gesehen haben sind an die Parameter (N, p, q, d_f, d_g, d) eines NTRU-Systems einige Bedingungen zu stellen.
Zum einen muss der geheime Schlüssel $f \in \mathcal{L}_f$ invertierbar sei, zum anderen muss die verschlüsselte Nachricht, also das Polynom $a = f \circledast m + p \cdot \phi \circledast g$ genügend kleine Koeffizienten zum korrekten Entschlüseln haben, d.h. also

$$|a = f \circledast m + p \cdot \phi \circledast g| < q,$$

wie wir im Beweis in Abschnitt 2.4 gesehen haben.
Die Parameter müssen also so gewählt werden, dass Effizienz und Funktionalität des Verfahrens gewährleistet sind.
Auf der anderen Seite muss durch die Parameterwahl auch sichergestellt werden, dass Angriffe auf das Verfahren, wie zum Beispiel Gitterangriffe möglichst schwer und ineffizient durchführbar sind.

2.4.1 Invertierbarkeit von f

Wie bereits erwähnt ist nicht jedes Polynom $f \in R$ invertierbar, da $x^N - 1$ nicht notwendigerweise irreduzibel sein muss.
J. Silverman diskutiert in [4] die Wahrscheinlichkeit, das ein zufällig gewähltes Polynom in R_q invertierbar ist.
Die Idee dabei ist den Quotienten $\frac{|R_q^*|}{|R_q|}$ zu berechnen, wobei R_q^* die Gruppe der Einheiten, also der invertierbaren Elemente, in R_q ist.

Lemma 2.1: Seien N, p Primzahlen und $q = p^k, k \in \mathbb{N}$ und $n = ord_N(p)$ die Ordnung von p in $\mathbb{Z}/N\mathbb{Z}$, d.h.

$$p^n \equiv 1 \qquad (mod\ N)$$

Dann gilt

$$\frac{|R_q^*|}{|R_q|} = \left(1 - \frac{1}{p}\right)\left(1 - \frac{1}{p^n}\right)^{\frac{N-1}{n}}$$

Für den Beweis des Lemmas sei auf [4] verwiesen.

Um die Wahrscheinlichkeit für die Invertierbarkeit zu erhöhen schlägt Silverman vor, die Wahl für f einzuschränken, so dass gilt $ggt(f(1), q) = 1$.
Speziell schlägt er vor dafür $f(1) = 1$ zu wählen; die Ringe mit dieser neuen Eigenschaft wollen wir mit $R_q^*(1)$ und $R_q(1)$ notieren.

Dies führt auf die neue Wahrscheinlichkeit für die Invertierbarkeit eines Polynoms f mit $f(1) = 1$ von

$$\frac{|R_q^*(1)|}{|R_q(1)|} = \left(1 - \frac{1}{p}\right)^{-1} \frac{|R_q^*|}{|R_q|} = \left(1 - \frac{1}{p^n}\right)^{\frac{N-1}{n}} \tag{3}$$

Bemerkung 2.2: Aus (3) ersieht man direkt, dass man p und N so aufeinander abstimmen sollte, dass p eine möglichst große Ordnung in $\mathbb{Z}/N\mathbb{Z}$ aufweist.

Bemerkung 2.3: Ein möglicher Parametersatz für moderate Sicherheit eines NTRU-Systems ist $(N, p, q) = (107, 3, 64)$(vgl. dazu [1]).
Es zeigt sich, das schon für deutlich kleinere Werte für N die Invertierbarkeit sehr hoch ist.
Sei $(N, p, q) = (11, 3, 31)$ dann ist $ord_N(p) = 5$,
denn $p^5 = 3^5 = 243 = 11 \cdot 22 + 1 \equiv 1 \qquad (mod\ 11)$
Mit Mit (3)ergibt sich nun

$$\frac{|R_q^*(1)|}{|R_q(1)|} = \left(1 - \frac{1}{243}\right)^2 \approx 0,9918$$

also eine Wahrscheinlichkeit von über 99,1% für die Invertierbarkeit von f.

2.4.2 Entschlüsselungskriterien

Um eine korrekte Entschlüsselung zu gewährleisten haben wir gesehen, dass

$$|f \circledast m + p \cdot \phi \circledast g| < q$$

gelten muss.
Hoffstein, Pipher und Silverman schätzen im speziellen die Normen der einzelnen Polynome f, g, ϕ, m dafür über folgendes Lemma ab (vgl. [1]).

Lemma 2.4: Für ein $\epsilon > 0$ existieren konstante Werte $\gamma_1, \gamma_2 > 0$ so, dass für beliebige Polynome $F, G \in R$ die Wahrscheinlichkeit größer als $1 - \epsilon$ ist, das sie die folgende Ungleichung erfüllen:

$$\gamma_1 |F|_2 \cdot |G|_2 \leq |F \circledast G|_\infty \leq \gamma_2 |F|_2 \cdot |G|_2 \tag{4}$$

Die Autoren spalten die obige Bedingung noch weiter auf und schätzen ab zu

$$|f \circledast m| \leq \frac{q}{4} \qquad und \qquad |p \cdot \phi \circledast g| \leq \frac{q}{4}$$

Wendet man nun Lemma 2.4 auf diese Abschätzung an, so erhält man

$$|f|_2 \cdot |m|_2 \approx \frac{q}{4} \cdot \gamma_2 \qquad und \qquad |\phi|_2 \cdot |g|_2 \approx \frac{q}{4p} \cdot \gamma_2$$

und damit eine Bedingung um die Parameter für das NTRU Verfahren zu wählen.

Bemerkung 2.5: In zahlreichen Probeläufen mit dem Parametersatz $(N, p, q) = (11, 3, 31)$ und umgekehrter herangehensweise, d.h. Werte für d_f, d_g, d_ϕ, d_m festlegen und das Verhalten der γ_2 zu untersuchen, hat sich gezeigt, das korrektes Entschlüsseln möglich ist sobald die beiden γ_2 nahe beieinander liegen.
Dies wollen wir etwas konkreter verdeutlichen.
Wir wollen zuerst die beiden γ_2 voneinander unterscheiden:

$$|f|_2 \cdot |m|_2 \approx \frac{q}{4} \cdot \gamma_2 \qquad und \qquad |\phi|_2 \cdot |g|_2 \approx \frac{q}{4p} \cdot \widetilde{\gamma_2}$$

Setzen wir nun die Anzahl der Einsen in den Polynomen fest (vgl. Kapitel 1):
Sei $d_f = 5, d_g = 5, d_\phi = 4, d_m = 3$
(Für diese Wahl der Parameter traten bei den Testläufen Entschlüsselungsfehler auf)
Mit diesen Werten ergibt sich

$$|f|_2 \approx 2,984 \qquad |g|_2 \approx 3,16$$

$$|\phi|_2 \approx 2,82 \qquad |m|_2 \approx 2,44$$

und damit also

$$|f|_2 \cdot |m|_2 \approx \frac{q}{4} \cdot \gamma_2 \qquad \Longrightarrow \gamma_2 \approx 0,88$$

$$|\phi|_2 \cdot |g|_2 \approx \frac{q}{4p} \cdot \widetilde{\gamma_2} \qquad \Longrightarrow \widetilde{\gamma_2} \approx 0,375$$

In diesem Fall erhalten wir Entschlüsselungsfehler.
Passt man nun das Störpolynom ϕ an mit $d_\phi = 1$, d.h. also $|\phi|_2 = \sqrt{2}$, so ergibt sich eine Änderung für $\widetilde{\gamma_2}$ zu

$$\widetilde{\gamma_2} \approx 0,78$$

Bei Polynomen mit diesen Parametern war korrektes Entschlüsseln wieder möglich.

2.4.3 Alternative Parameterwahl

Wie gesehen ist die Bedingung für korrektes Entschlüsseln:

$$|a| = |p \cdot \phi \circledast g + f \circledast m| < q$$

Betrachten wir nun die beiden Summanden einzeln für sich:

Summand $s_1 := p \cdot \phi \circledast g$
O.B.d.A. sei $d_g > d_\phi$.
Dann ist der größte Koeffizient k_{s_1} von s_1 maximal $k_\phi = 2d_\phi \cdot p$,
da nicht mehr als d_ϕ viele Einsen auf d_g viele Einsen treffen können und analog die Minus Einsen auf Minus Einsen treffen.

Summand $s_2 := f \circledast m$
Analog zu oben ergibt sich $k_{s_2} = (2d_f + 1)\frac{1}{2}p$,
da für die Koeffizienten der Nachricht m gilt: $-\frac{1}{2}p \leq m \leq \frac{1}{2}p$.
Fügen wir nun die beiden Werte aus unserer Überlegung zusammen, so erhalten wir für den Größten Koeffizienten k_a in a

$$k_a = 2d_\phi \cdot p + (2d_f + 1)\frac{1}{2}p = p(2d_\phi + d_f + \frac{1}{2})$$

und also die Bedingung für korrektes Entschlüsseln mit

$$\frac{1}{2}q > p(2d_\phi + d_f + \frac{1}{2}) \tag{5}$$

Bemerkung 2.6: Vergleichen wir diese Bedingung mit vorigem Abschnitt, so sieht man direkt das korrektes Entschlüsseln nicht möglich ist, denn
sei $d_f = 5$ und $d_\phi = 4 \qquad (p = 3,\ q = 31)$

$$\Longrightarrow \frac{1}{2}q > p\Big(2d_\phi + d_f + \frac{1}{2}\Big) = 27$$

was uns direkt einen Widerspruch liefert.

3 Angriffe auf NTRU

3.1 Brute-force

Eve möchte als Angreifer den geheimen Schlüssel f angreifen, über eine Brute-Force Suche. hierfür muss sie Polynome f aus dem Schlüsselraum auswählen und testen ob $f \circledast h \mod q$ kleine Koeffizienten besitzt.
Nach vorigem Abschnitt ist $f \in \mathcal{L}(d_f, d_f - 1)$.
Somit ergibt sich die Anzahl aller möglichen f zu

$$|\mathcal{L}(d_f, d_f - 1)| = \binom{N}{d_f} \cdot \binom{N - d_f}{d_f}$$

$$= \frac{N!}{d_f!(n - d_f)!} \cdot \frac{(N - d_f)!}{d_f!(N - 2d_f)!} = \frac{N!}{(d_f!)^2(N - 2d_f)!}$$

Nun haben wir in Abschnitt 2.1.1 gesehen, dass zum Entschlüsseln nicht nur f, sondern auch alle N verschiedenen Rotationen von f gültig sind. Damit ergibt sich, dass es im schlimmsten Fall

$$\frac{|\mathcal{L}(d_f, d_f - 1)|}{N}$$

viele Versuche dauert bis Eve einen gültigen Schlüssel findet.

3.2 Gitter Angrif nach Coppersmith-Shamir

Wir wollen in diesem Abschnitt den Gitterbasierten Angriff auf ein NTRu-System beschreiben, welcher auf D. Coppersmith und A. Shamir zurückgeht (siehe [3]).
Das zugrunde liegende Gitter sei mit $\mathcal{L}^{CS}$ bezeichnet, mit der Basis

$$B^{CS} = \left(\begin{array}{c|c} \alpha E_N & H \\ \hline 0 & qE_N \end{array}\right) \in \mathbb{R}^{2N \times 2N}, \qquad H \in \mathbb{R}^{N \times N}$$

Betrachten wir nun die einzelnen Variablen H, q, α etwas näher.
Bei H handelt es sich um eine $N \times N$ Matrix deren Einträge sich aus den Koeffizienten des öffentlichen Schlüssels h folgendermaßen zusammensetzen:
Es ist die NTRU-Gleichung (2) umgeformt

$$g = f \circledast h \qquad \mod q$$

bzw. vektoriell geschrieben

$$g_i = \sum_{j+k \equiv i \mod N} f_i \cdot h_k \qquad \mod q \qquad \text{für } i = 0, 1, \ldots, N-1$$

Damit erhalten wir nun ein Gleichungssystem bestehend aus N Gleichungen:

$$\begin{array}{rcccccccccc}
g_0 & = & f_0h_0 & + & f_1h_{N-1} & + & \dots & + & f_{N-2}h_2 & + & f_{N-1}h_1 & \mod q \\
g_1 & = & f_0h_1 & + & f_1h_0 & + & \dots & + & f_{N-2}h_3 & + & f_{N-1}h_2 & \mod q \\
g_2 & = & f_0h_2 & + & f_1h_1 & + & \dots & + & f_{N-2}h_4 & + & f_{N-1}h_3 & \mod q \\
\vdots & & & & & & & & & & & \\
g_{N-1} & = & f_0h_{N-1} & + & f_1h_{N-2} & + & \dots & + & f_{N-2}h_1 & + & f_{N-1}h_0 & \mod q
\end{array}$$

Wandeln wir dieses Gleichungssystem nun in Matrix-schreibweise um, so erhalten wir:

$$(g_0, g_1, \dots, g_{N-1}) = (f_0, f_1, \dots, f_{N-1}) \underbrace{\begin{pmatrix} h_0 & h_1 & \cdots & h_{N-1} \\ h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_0 \end{pmatrix}}_{=:H} \mod q$$

Würden wir nur H als Gitterbasis auffassen, so erhielten wir nicht alle gültigen Schlüssel, da die Reduktion modulo q nicht berücksichtig wird.
Dieses Problem wird durch die q-fache Einheitsmatrix in der Gitterbasis gelöst.
Um nun wieder eine quadratische Gitterbasis zu erhalten wird die Matrix ergänzt zu

$$B^{CS} = \left(\begin{array}{cccc|cccc} \alpha & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & \alpha & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha & h_1 & h_2 & \cdots & h_0 \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right)$$

Damit erhalten wir nun Vektoren von der Form $\tau = (\alpha\tilde{f}, \tilde{g})$ die im Gitter enthalten sind, wobei $\tilde{f}, \tilde{g}$ jeweils Rotationen von f, g sind, die auch für korrektes Entschlüsslen benutzt werden können, wie wir in Kapitel 2 gesehen haben.
Da es sich bei f, g um Polynome mit Koeffizienten aus $\{-1, 0, 1\}$ handelt trifft man die Annahme, das es sich bei τ um einen nahezu kürzesten Vektor im vorliegenden Gitter handelt, also dass τ ein kürzester Vektor ist oder nur unwesentlich länger.
Nach der Heuristik von Gauß ist die untere Schranke für die Länge eines kürzesten Vektors gegeben durch

$$\sqrt[2N]{det(\mathcal{L}^{CS})} \cdot \sqrt{\frac{2N}{2\pi e}} = (q\alpha)^{\frac{1}{2}} \sqrt{\frac{N}{\pi e}} = \sqrt{\frac{Nq\alpha}{\pi e}} =: s \tag{6}$$

Um einen kürzesten Vektor zu finden wird ein Angreifer versuchen das Verhältnis $s/|\tau|_2$ zu maximieren.

Bemerkung 3.1: $|\tau|_2$ ist eine bekannte Größe, da d_f und d_g öffentliche Parameter um NTRU System sind.

In [1] erreichen die Autoren dies indem sie $\alpha = \frac{|g|_2}{|f|_2}$ wählen.
Somit dient die Variable α als Balancierungsfaktor um die Erfolgschancen eines Gitterangriffs zu steigern.
Damit ergibt sich

$$c_h^{-1} := \frac{s}{|\tau|_2} = \sqrt{\frac{Nq}{2\pi e|f|_2|g|_2}}$$

Bemerkung 3.2: Liegt c_h nahe bei 1, so ist die Länge des Zielvektors τ ungefähr die Länge eines kürzesten Vektors im Gitter und somit haben Angriffsversuche keine guten Erfolgsaussichten den privaten Schlüssel f oder eine Rotation davon zu finden.
Bei größer werdendem c_h sind die Aussichten auf Erfolg größer, allem Anschein nach (vgl. [1]) aber immer noch exponentiell in N, was bei großem N damit genügend Sicherheit bieten sollte.

4 Variante des NTRU-Verfahren über $\mathbb{Z}[x]/x^N + 1$

Im folgenden Kapitel beschreiben wir eine leicht modifizierte Variante des NTRU-Verfahrens und untersuchen dessen neue Eigenschaften.
Wir tauschen dafür den Algebraischen Raum $R = \mathbb{Z}[x]/x^N - 1$, in dem alle Rechenoperationen stattgefunden haben, aus mit

$$\widetilde{R} := \mathbb{Z}[x]/x^N + 1$$

Die Konsequenz hieraus ist, dass in einem Polynom gilt $x^N = -1$.
Dies führt dazu, dass sich die Multiplikation im Verfahren leicht verändert zu

$$f \tilde{\circledast} g = h \qquad\qquad f, g \in \tilde{R}$$

mit

$$h_k = \sum_{i=0}^{k} f_i \cdot g_{k-i} - \sum_{i=k+1}^{N-1} f_i \cdot g_{N+k-i} \tag{7}$$

4.1 Das Verfahren

Am NTRU-Verfahren ändert sich zum Ver- und Entschlüsseln im wesentlichen nichts.
Wir bestimmen zuerst ein Parametersatz (N, p, q) mit N prim und $\text{ggt}(p, q) = 1$.
Darüber hinaus legen wir die Form der Polynome fest, also

$$f \in \mathcal{L}(d_f, d_f - 1), \quad g \in \mathcal{L}(d_g, d_g), \quad \phi \in \mathcal{L}(d_\phi, d_\phi)$$

f wählen wir wieder so, dass es in $\widetilde{R}_q$ und $\widetilde{R}_p$ invertierbar ist, d.h. es existieren F_q, F_p mit

$$F_q \tilde{\circledast} f \equiv 1 \pmod{q} \qquad \textit{und} \qquad F_p \tilde{\circledast} f \equiv 1 \pmod{p}$$

Damit ergibt sich nun der öffentliche Schlüssel

$$h = F_q \tilde{\circledast} g \mod q$$

nun können wir eine Nachricht $m \in \mathcal{L}_m$ verschlüsseln.
Dazu wählen wir ein zufälliges Störpolynom $\phi \in \mathcal{L}_\phi$ aus und berechnen

$$e = p \cdot \phi \tilde{\circledast} h + m \mod q$$

Bei e handelt es sich um die verschlüsselte Nachricht.

Um nun die verschlüsselte Nachricht e wieder zu Entschlüsseln benötigen wir den geheimen Schlüssel f und sein multiplikatives Inverses F_q in $\widetilde{R}_q$.
Im ersten Schritt berechnen wir

$$a = f \tilde{\circledast} e \mod q$$

um im zweiten Schritt via

$$m = F_p \tilde{\circledast} a \mod q$$

die ursprüngliche Nachricht m zu erhalten.

4.2 Rotationsinvarianz

In Abschnitt 2.1.1 haben wir eine wichtige Eigenschaft eines NTRU-Systems kennengelernt, die Rotationsinvarianz. Dort haben wir gesehen, dass nicht nur der geheime Schlüssel f, sondern auch alle N-vielen Rotationen von f zum korrekten Entsclüsseln geeignet sind.
Wir wollen nun untersuchen wie sich diese Eigenschaft in unserer neuen NTRU-Variante verhält.

Sei $f \in \mathcal{L}_f$ der geheime Schlüssel. Wir identifizieren f mit seinem Koeffizientenvektor $f = (f_0, f_1, \ldots, f_{N-1})$.
Wir wollen nun eine modifizierte Version der Rotation eines Vektors angeben, die den negativen Faktor im Konvolutionsprodukt $\tilde{\circledast}$ berücksichtigt.

$$\Big[f_0, f_1, \ldots, f_{N-1} | -f_0, -f_1, \ldots, -f_{N-1} | f_0, f_1, \ldots\Big] \tag{8}$$

Um nun eine Rotation von f in unserer neuen NTRU-Variante zu erhalten wählen wir aus (8) nun eine N elementige zusammenhängende Teilmenge aus.

Beispiel 4.1: Sei f wie oben definiert. Nun wollen wir eine modifizierte Rotation von f betrachten. Sei f_{r_2} eine um 2 Stellen rotierte Version von f, dann hat f_{r_2} folgende Form

$$f_{r_2} = (f_2, f_3, \ldots, f_{N-1}, -f_1, -f_2)$$

bzw. als Polynom geschrieben

$$f_{r_2} = f_2 + f_3 x + f_4 x^2 + \ldots + f_{N-1} x^{N-3} - f_0 x^{N-2} - f_1 x^{N-1}$$

Lemma 4.2: Mit obigen Bezeichnungen ist für eine beliebige Rotation $f_{r_n}, \quad n \in [0, N-1]$ die NTRU-Gleichung (2) erfüllt.

Beweis :
Zur besseren Übersicht wollen wir passend zu Beispiel 4.1 das Lemma für f_{r_2} beweisen.
Sei also $f_{r_2} = (f_2, f_3, \ldots, f_{N-1}, -f_1, -f_2)$ und h der öffentliche Schlüssel aus (2).
Dann ist $f_{r_2} \tilde{\circledast} h =$

$$\begin{aligned}
&(f_2h_0 - \ldots - f_{N-1}h_3 + f_0h_2 \quad + f_1h_1) \\
&(f_2h_1 + \ldots - f_{N-1}h_4 + f_0h_3 \quad + f_1h_2)x \\
&\quad \vdots \\
&(f_2h_{N-3} + \ldots + f_{N-1}h_0 + f_0h_{N-1} + f_1h_{N-2})x^{N-3} \\
&(f_2h_{N-2} + \ldots + f_{N-1}h_1 - f_0h_0 \quad + f_1h_{N-1})x^{N-2} \\
&(f_2h_{N-1} + \ldots + f_{N-1}h_2 - f_0h_1 \quad - f_1h_0)x^{N-1} = \tilde{g}
\end{aligned}$$

Mit dieser Multiplikation erhalten wir ein Koeffizientenvektor $\tilde{g}$, welcher sich direkt aus den Koeffizienten des ursprünglich für NTRU genutzten Poylnoms g darstellen lässt:
Sei $g = (g_0, g_1, \ldots, g_{N-1})$
dann ergibt sich nach voriger Rechnung genau

$$\tilde{g} = (g_2, g_3, \ldots, g_{N-1}, -g_0, -g_1)$$

was genau der gleichen Rotationsvorschrift wie f genügt.
Analog beweisen wir diesen Sachverhalt für beliebige Rotationen $f_{r_n}, \quad n \in [0, N-1]$. $\square$

4.2.1 Parameterwahl

Für die Parameterwahl von NTRU im Polynomring $\tilde{R}$ verweisen wir auf Abschnitt 2.4.3, da der Faktor -1 im Konvulotionsprudukt keinen Einfluss auf den größtmöglichen Koeffizienten von a hat.
Obiger Faktor hat nur Einfluss darauf wie sich der größter Koeffizient letztendlich zusammensetzt.
Demnach ist also die Bedingung für korrektes Entschlüsseln also

$$\frac{1}{2}q > p(2d_\phi + d_f + \frac{1}{2})$$

4.3 Angriffe

4.3.1 Brute-Force Angriff

Um das neue NTRU-System via Brute-Force anzugreifen ist es wie im Standardverfahren nötig alle möglichen Schlüssel $f \in \mathcal{L}_f$ aufzuzählen und zu testen ob

$$f \tilde{\circledast} h \mod q$$

kleine Koeffizienten hat.
Da f wie gehabt aus der gleichen Polynomenge wie in Kapitel 2 gewählt wird bleibt die Anzahl der $f's$ gleich, also gilt

$$|\mathcal{L}_f| = \frac{N!}{(d_f!)^2(N-2d_f)!}$$

Wir wollen uns nun überlegen wie sich die neue Rotationsinvarianz nun auf einen Brute-Force Angriff auswirkt.
Wir haben gesehen, dass es nun $2N$ viele verschiedene Rotationen von f gibt, die zum Entschlüsseln geeignet sind, betrachten wir diese allerdings in Hinblick auf die Menge $\mathcal{L}_f$ so wird schnell klar, dass nicht alle diese Rotationen f_{r_n} auch in $\mathcal{L}_f$ liegen werden.
So liegt zum Beispiel die Rotation f_{r_2} wieder in $\mathcal{L}_f$ wenn genau eine 1 zu einer -1 wird und umgekehrt; in solch einem Fall wären damit sicherlich die Rotationen f_{r_1}, f_{r_3} nicht in $\mathcal{L}_f$ enthalten.
Führen wir diese Argumentation weiter, so ergibt sich, dass im besten Fall, nämlich wenn f abwechselnd positive und negative Koeffizienten aufweist,
$\frac{2N}{2} = N$ viele Rotationen in $\mathcal{L}_f$ enthalten sind.
Damit ergibt sich für einen Brute-Force Angriff im bestmöglichsten Fall ein Erfolg einen passenden geheimen Schlüssel zu finden nach $\frac{|\mathcal{L}_f|}{N}$ vielen Versuchen. Dies entspricht gerade der Laufzeit aus Abschnitt 3.1.
Da dies aber der beste Fall ist, der möglich ist, so können wir diesen mit dem Wissen leicht verhindern indem wir f nicht in der oben genannten zyklischen Form wählen.

Betrachten wir nun den schlechtmöglichsten Fall für einen Brute-Force Angriff.
Diesen erhalten wir, wenn wir f so annehmen, dass genau die ersten $\frac{d_f}{2}$ Einsen positives und die letzten $\frac{d_f}{2}$ Einsen negatives Vorzeichen besitzen oder umgekehrt. Dies führt dazu, dass genau 2 Rotationen in $\mathcal{L}_f$ enthalten sind, nämlich f selbst und f_{r_N}. Damit würde unsere Brute-Force Suche spätestens nach

$$\frac{|\mathcal{L}_f|}{2}$$

einen gültigen Schlüssel liefern, was deutlich länger dauern würde als die Suche in Kapitel 3.1.

4.3.2 Gitterangriff

Wir wollen den in Abschnitt 3.3 vorgestellten Gitterangriff nun auf unsere neue Variante des NTRU-Systems anwenden und untersuchen ob sich der Wechsel des algebraischen Raumes auf die Theoretische Sicherheit auswirkt.

Gitterbasis

Analog zu 3.3 wollen wir uns die neue Gitterbasis aus der NTRU-Gleichung (2) herleiten. Es ist

$$g = f \tilde{\circledast} h \qquad \mod q$$

also

$$g_k = \sum_{i=0}^{k} f_i \cdot h_{k-i} - \sum_{i=k+1}^{N-1} f_i \cdot h_{N+k-i}$$

Betrachten wir diese nun einzeln, so ergibt sich

$$\begin{array}{rcccccccccccl}
g_0 & = & f_0h_0 & - & f_1h_{N-1} & - & \dots & - & f_{N-2}h_2 & - & f_{N-1}h_1 & & \mod q \\
g_1 & = & f_0h_1 & + & f_1h_0 & - & \dots & - & f_{N-2}h_3 & - & f_{N-1}h_2 & & \mod q \\
g_2 & = & f_0h_2 & + & f_1h_1 & + & \dots & - & f_{N-2}h_4 & - & f_{N-1}h_3 & & \mod q \\
\vdots & & & & & & & & & & & & \\
g_{N-1} & = & f_0h_{N-1} & + & f_1h_{N-2} & + & \dots & + & f_{N-2}h_1 & + & f_{N-1}h_0 & & \mod q
\end{array}$$

In Matrix-Schreibweise sieht dies folgendermaßen aus:

$$(g_0, g_1, \dots, g_{N-1}) = (f_0, f_1, \dots, f_{N-1}) \underbrace{\begin{pmatrix} h_0 & h_1 & \cdots & h_{N-1} \\ -h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ -h_1 & -h_2 & \cdots & h_0 \end{pmatrix}}_{=:\widetilde{H}} \qquad \mod q$$

Wir ergänzen die Matrix $\widetilde{H}$ zu der Gitterbasis

$$\widetilde{B}^{CS} = \left(\begin{array}{cccc|cccc} \alpha & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & \alpha & \cdots & 0 & -h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha & -h_1 & -h_2 & \cdots & h_0 \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right)$$

Da wir analog zu 3.2 vorgegangen sind, haben wir sichergestellt, dass Vektoren der Form $\tilde{\tau} = (\alpha\tilde{f}, \tilde{g})$ im Gitter enthalten sind, wobei $\tilde{f}, \tilde{g}$ Rotationen von f, g sind.

Damit haben wir nun eine Gitterbasis, die uns einen Angriff auf das neue NTRU-System bietet.
Es sind $f, g \in \tilde{R}_p$, weshalb wir wieder annehmen können, dass f, g und deren Rotationen wieder mit zu den kürzesten Gittervektoren gehören. Nach der Gauß'schen Heuristik erwarten wir die Länge eines kürzesten Vektors mit

$$s = \sqrt{\frac{Nq\alpha}{\pi e}}$$

Wir sehen hier bereits, das die neue Eigenschaft der doppelten Anzahl an Rotationen von f, g keinen Einfluss auf einen Gitterangriff hat, da dieser nur von Dimension und Determinante des Gitter abhängt.
Um einen Gitterangriff also zu erschweren sollte, wie auch schon in Kapitel 3.2, das Verhältnis $s/\tilde{\tau}$ möglichst nahe bei 1 liegen.

5 Weiter Varianten von NTRU

Wie wir im vorigen Kapitel gesehen haben hat es keine wesentlichen Verbesserungen oder Verschlechterungen des NTRU-Verfahrens beim Tausch des algebraischen Grundraum gegeben.
Daher wollen wir nun versuchen noch einen Schritt weiter zu gehen und noch einen anderen algebraischen Grundraum für NTRU auswählen und dessen Auswirkungen darauf untersuchen.

5.1 Über $\mathbb{Z}[x]/x^N - 2$

Wir wollen nun das NTRU-Verfahren über dem Polynomring $\hat{R} := \mathbb{Z}[x]/x^N - 2$ betrachten.
Es gilt $x^N = 2$ für $N \in \mathbb{N}$, wobei N wie üblich als Primzahl anzusehen ist.
Analog zu Kapitel 4 ändert sich damit in erster Linie die Multiplikation der Polynome, die zum Ver- und Entschlüsslen benutzt werden.
Für $f, g \in \hat{R}$ gilt

$$f \hat{\circledast} g = h$$

mit

$$h_k = \sum_{i=0}^{k} f_i \cdot g_{k-i} + 2 \sum_{i=k+1}^{N-1} f_i \cdot g_{N+k-i}$$

wie man sich durch einfaches nachrechnen leicht klarmachen kann.
Die Koeffizienten der Polynome, die für das Verfahren benötigt werden, wählen wir wie üblich aus $\mathbb{Z}_q$ bzw. $\mathbb{Z}_p$ und notieren diese Polynomringe mit $\hat{R}_q$ bzw. $\hat{R}_p$.

5.2 Das Verfahren

Am NTRU-Verfahren und an der Vorgehensweise ändert sich, wie auch schon in Kapitel 4, im Vergleich zum Original Verfahren nichts.
Wir wählen wieder $f \in \mathcal{L}_f$, $g \in \mathcal{L}_g$, wobei wir wieder die Invertierbarkeit von f in $\hat{R}_q$ und $\hat{R}_p$ verlangen.
Damit erhalten wir den öffentlichen Schlüssel

$$h = F_q \hat{\circledast} g \mod q$$

Um nun eine Nachricht $m \in \mathcal{L}_m$ zu verschlüsseln, wählen wir ein Störpolynom $\phi \in \mathcal{L}_\phi$ und berechnen die verschlüsselte Nachricht

$$e = p \cdot \phi \hat{\circledast} h + m \mod q$$

Die Entschlüsselung funktioniert wie üblich über die folgenden zwei Schritte.
Zuerst berechnen wir mit dem geheimen Schlüssel f

$$a = f \hat{\circledast} e \mod q$$

und wählen die Koeffizienten von a im Intervall $[-q/2, q/2]$
um damit

$$m = F_p \hat{\circledast} a \mod q$$

die ursprüngliche Nachricht wiederherzustellen.
Für den Korrektheitsbeweis sei auf Kapitel 2.2 verwiesen, da dieser unabhängig von der Multiplikation geführt wird.
Zum korrekten Entschlüsseln kommt es wie in Kapitel 2 sehr auf die Parameterwahl des NTRU-Systems an.
Mehr dazu in Abschnitt 5.4.

5.3 Rotationsinvarianz

Zunächst wollen wir uns mit der für ein NTRU-System wichtigen Eigenschaft der Rotationsinvarianz befassen.
Beim Standard Verfahren aus Kapitel 2 ist jede Rotation des geheimen Schlüssels f zum korrekten Entschlüsseln geeignet.
In der Variante aus Kapitel 4 ist dies nicht mehr der Fall gewesen, wir haben dies allerdings umgangen, indem wir bei jedem Koeffizienten des rotierten Anteils von f das Vorzeichen gewechselt haben, also sei zum Beispiel $f = (1, 0, -1, 1, 0, 0)$
Dann ergab sich eine mögliche Rotation zu $f_{r_2} = (1, 1, 0, 0, -1, -0)$.
Damit haben wir beim Entschlüsseln mit f_{r_2} erreicht, dass wir genau ein auf die gleiche Weise rotiertes Polynom g_{r_2} erhalten und die Ntru-Gleichung erfüllt bleibt.
Das wurde dadurch erreicht, dass die im Vorzeichen geänderten Koeffizienten im Produkt $f \tilde{\circledast} g \mod q$ genau wieder zu einer 1 wurden, somit also mit dem Inversen der -1 multipliziert wurden. Da -1 selbstinvers ist ergab sich direkt obige Rotation.
Wenden wir diese Folgerung nun auf die NTRU-Variante in $\hat{R}$ an, so erhalten wir eine neue Rotationsvorschrift für f und $z \in \mathbb{Z}_q$ mit $z \cdot 2 = 1 \mod q$ über

$$\Big[f_0, f_1, \ldots, f_{N-1} | z \cdot f_0, z \cdot f_1, \ldots, z \cdot f_{N-1} | f_0, f_1, \ldots\Big] \tag{9}$$

Um nun eine Rotation $f_{r_n},\ n \in [0, N-1]$ von f zu erhalten wählen wir analog zu 4.2 eine N-elementige zusammenhängende Menge aus (9) aus.

Lemma 5.1: Mit obiger Rotationsvorschrift (9) ist für ein f_{r_n} die NTRU-Gleichung (2) erfüllt.

Beweis :
Analog zu Lemma 4.2 wollen wir uns auf die Rotation f_{r_2} zu besseren Übersichtlichkeit beschränken.
Sei also $f_{r_2} = (f_2, f_3, \ldots, f_{N-1}, zf_1, zf_2)$ und h der öffentliche Schlüssel aus (2).
Dann ist $f_{r_2} \hat{\circledast} h =$

$$
\begin{array}{l}
(f_2h_0 + 2f_3h_{N-1} + \ldots + 2f_{N-1}h_3 + \underbrace{2z}_{=1} f_0h_2 \quad + \underbrace{2z}_{=1} f_1h_1) \\
(f_2h_1 + f_3h_0 \quad + \ldots + 2f_{N-1}h_4 + \underbrace{2z}_{=1} f_0h_3 \quad + \underbrace{2z}_{=1} f_1h_2)x \\
\quad \vdots \\
(f_2h_{N-3} + f_3h_{N-4} \; + \ldots + f_{N-1}h_0 \; + \underbrace{2z}_{=1} f_0h_{N-1} + \underbrace{2z}_{=1} f_1h_{N-2})x^{N-3} \\
(f_2h_{N-2} + f_3h_{N-3} \; + \ldots + f_{N-1}h_1 \; + z \cdot f_0h_0 \quad + \underbrace{2z}_{=1} f_1h_{N-1})x^{N-2} \\
(f_2h_{N-1} + f_3h_{N-2} \; + \ldots + f_{N-1}h_2 \; + z \cdot f_0h_1 \quad + z \cdot f_1h_0)x^{N-1} = \hat{g}
\end{array}
$$

Mit dieser Multiplikation erhalten wir ein Koeffizientenvektor $\hat{g}$, welcher sich direkt aus den Koeffizienten des ursprünglich für NTRU genutzten Poylnoms g darstellen lässt:
Sei $g = (g_0, g_1, \ldots, g_{N-1})$
dann ergibt sich nach voriger Rechnung genau

$$\tilde{g} = (g_2, g_3, \ldots, g_{N-1}, zg_0, zg_1)$$

was genau der gleichen Rotationsvorschrift wie f genügt.
Analog beweisen wir diesen Sachverhalt für beliebige Rotationen $f_{r_n}, \quad n \in [0, N-1]$. $\square$

Bemerkung 5.2: Theoretisch ist mit obiger Rotation korrektes Entschlüsseln möglich, allerdings hat sich in Testläufen gezeigt, dass es fast ausschließlich zu Entschlüsselungsfehler, bei der Benutzung der Rotationen f_{r_n}, kam.
Der Grund hierfür ist leicht einzusehen, wenn man sich klarmacht welche Bedingungen an korrektes Entschlüsseln gestellt werden.
Im Korrektheitsbeweis sehen wir das $|a| = |p \cdot \phi \hat{\circledast} g + f \hat{\circledast} m|_\infty < q$ gelten muss.
Ersetzen wir nun f durch eine Rotation wie oben f_{r_n} so sehen wir schnell das im zweiten Summanden z als Koeffizient auftaucht; mit großer Wahrscheinlichkeit auch mehrmals; so das die Koeffizientenweite von a schnell größer als q wird, was die Eindeutigkeit über $\mathbb{Z}$ zerstört und damit korrektes Entschlüsseln nicht mehr funktionieren muss.

Beispiel 5.3: Sei $(N, p, q) = (11, 3, 31)$, $f = (f_0, f_1, \ldots, f_{N-1}) \in \hat{R}_p$.
Dann ist beispielsweise $f_{r_2} = (f_2, f_3, \ldots, f_{N-1}, 16f_0, 16f_1)$.
Treffen nun die letzten beiden Koeffizienten von f_{r_2} im Produkt mit m jeweils auf eine 1, so sieht man schnell, dass die Koeffizientenweite bereits größer q und damit korrektes Entschlüsseln nicht mehr gegeben ist. Bei höheren Rotationen treten im Produkt natürlich noch mehr Summanden mit dem Faktor 16 auf, was korrektes Entschlüsseln damit nahezu unmöglich macht.

5.4 Parameterwahl

Wir wollen uns hier an Abschnitt 2.4.3 halten und analog zu diesem argumentieren.
Die Bedingung, die unsere Polynome erfüllen müssen ist wie gehabt

$$|a| = |p \cdot \phi \hat{\circledast} g + f \hat{\circledast} m|_\infty < q$$

Summand $s_1 := p \cdot \phi \circledast g$
O.B.d.A. sei $d_g > d_\phi$.
Dann ist der größte Koeffizient k_{s_1} von s_1 maximal $k_\phi = 2 \cdot 2d_\phi \cdot p$,
analog zu 2.4.3, wobei zu beachten ist, dass durch die Reduktion via $x^N = 2$ der Faktor 2 in jedem Summand auftauchen kann und dieser somit auch den größten Koeffizient verdoppeln kann.

Summand $s_2 := f \circledast m$
Analog zu oben ergibt sich $k_{s_2} = 2 \cdot (2d_f + 1)\frac{1}{2}p$,
da für die Koeffizienten der Nachricht m gilt: $-\frac{1}{2}p \leq m \leq \frac{1}{2}p$.
Fügen wir nun die beiden Werte aus unserer Überlegung zusammen, so erhalten wir für den Größten Koeffizienten k_a in a

$$k_a = p(2d_\phi + 2d_f + 1)$$

und also die Bedingung für korrektes Entschlüsseln mit

$$\frac{1}{2}q > p(2d_\phi + 2d_f + 1) \tag{10}$$

Wie wir somit sehen ist bei der NTRU-Variante in $\hat{R}$ die Parameterwahl noch weiter eingeschränkt als beim Standardverfahren bzw. bei der Variante aus Kapitel 4.

6 Zusammenfassung

Wir haben im letzten Kapitel gesehen wie sich das NTRU-Verfahren bei einem weiteren Tausch des algebraischen Grundraumes verhält. Aufgrund der Eigenschaft, dass in einem Produkt zweier Polynome für den Faktor $x^N = 2$ gilt, wurden insgesamt gesehen die Koeffizienten der verschlüsselten Nachricht e größer. Dies hatte eine eingeschränktere Wahl der Parameter (d_f, d_g, d_ϕ) zur Folge hatte. Führen wir diesen Ansatz fort, indem wir beispielsweise den Polynomring $\mathbb{Z}[x]/x^N - e$, wobei e eine Einheit in $\mathbb{Z}_q$ ist, für unser NTRU-System benutzen. Mit dieser Annahme können wir eine noch weiter eingeschränkte Parameterwahl erwarten. Das kann bis hin zur Nicht-Funktionalität des Verfahrens führen, da die neu auftauchenden Faktoren $x^N = e$, je nach Wahl von e schnell in der Summe über dem festgelegten Parameter q liegen können.

Insgesamt betrachtet lässt sich festhalten, dass keine der beiden neuen NTRU-Varianten nennenswerte Vorteile aufweist. Einzig die Eigenschaft der Rotationsinvarianz, die sich in der Verdoppelung der zum Entschlüsseln geeigneten geheimen Schlüssel f bemerkbar macht hat einen Einfluss auf einen Brute-Force Angriff. Brute-Force-Angriffe werden dadurch mindestens um einen Faktor 2 erschwert. Da eine Brute-Force Angriff ein eher schwacher Angriff auf ein Kryptosystem ist, bleibt diese Änderung ohne größere Sicherheitstechnische Vorteile.

Der Nachteil der vorgestellten Varianten beginnt erst bei der Variante aus Kapitel 5, nämlich die eingeschränkte Parameterwahl, die durch das größere Polynom $x^N - 2$ zustande kommt. Um korrektes Entschlüsseln letztlich zu gewährleisten müssen wir die Parameterwahl einschränken (siehe 5.4). Durch diese Einschränkung machen wir das NTRU-System angreifbarer, da wir dazu mindestens ein Polynom einschränken müssen (siehe Bemerkung 5.2), und erleichtern damit bereits einen Brute-Force-Angriff.

7 Literaturverzeichnis

[1] Jeffrey Hoffstein, Jill Pipher und Joseph H. Silverman:
NTRU:A Ring-Based Public Key Cryptosystem.
In: ANTS, Band 1423 der Reihe
Lecture Notes in Computer Science , Seiten 267-288. Springer, 1998.

[2] Don Coppersmith, Adi Shamir:
Lattice Attacks on NTRU.
In: Lecture Notes in Computer Science, 1997, Band 1233, S.52-61.

[3] Skript zur Vorlesung von Prof. Dr. C.P. Schnorr:
Gittertheorie und algorithmische Geometrie
(online verfügbar) http://ismi.math.uni-frankfurt.de/schnorr/lecturenotes/
(gefunden Mai 2012)

[4] Joseph H. Silverman: *NTRU Cryptosystems Technical Report #009:*
Invertibility in Truncated Polynomial Rings
(online verfügbar) http://www.securityinnovation.com/uploads/
Crypto/NTRUTech009.pdf
(gefunden März 2012)

[5] Bronstein, Semendjajew, Musiol, Mühlig:
Taschenbuch der Mathematik
(Verlag Harri Deutsch, 2006)

[6] Joseph H. Silverman: *NTRU Cryptosystems Technical Report #014:*
Almost Inverses and fast NTRU- Key Creation
(online verfügbar) http://www.securityinnovation.com/uploads/
Crypto/NTRUTech014.pdf
(gefunden Januar 2012)

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst und nur die im Literaturverzeichnis angegebenen Quellen verwendet habe.

Patrick Bächtle Datum