Paul-Klee-Gymnasium

Facharbeit aus der Mathematik

Thema:

Asymmetrische Verschlüsselungsverfahren

am Beispiel des RSA-Kryptosystems

Verfasser	:	Martin Andreas Thoma
Kursleiter	:	Claudia Wenninger
Abgegeben am	:	20.01.2010 (verändert am 06.04.2010)

Erzielte Note	:	_____
Erzielte Punkteanzahl	:	_____

_____

Unterschrift der Kursleiterin

Inhaltsverzeichnis

1 Prinzip der asymmetrischen Verschlüsselung

Bereits im 5. Jahrhundert v. Chr. war ein Verfahren zur geheimen Weitergabe an Informationen bekannt: Die Skytale[1]. Man wickelte Papier spiralförmig um einen Stab, die Skytale, und schrieb die Nachricht längs der Skytale auf das Papier. Dann wurde das Papier dem Empfänger gebracht, der mit einer gleichen Skytale diese Nachricht entschlüsseln konnte.

Verfahren, die den gleichen Schlüssel zur Ver- als auch zur Entschlüsselung verwenden nennt man symmetrisch[2]. In dem Beispiel ist die Skytale der Schlüssel. Es ist unüblich die Skytale als symmetrische Verschlüsselung zu bezeichnen, normalerweise sind Block- oder Stromchiffren damit gemeint. Diese sind jedoch schwerer zu beschreiben.

Nach Kerckhoffs' Prinzip darf ein Verschlüsselungssystem keine Geheimhaltung erfordern[3], also muss der Schlüssel für die Sicherheit sorgen. Wollen allerdings zwei Personen miteinander geheim kommunizieren, so muss dieser Schlüssel übertragen werden. Bei der Übertragung könnte er abgefangen werden. Asymmetrische Verschlüsselungsverfahren benutzen einen öffentlichen Schlüssel zum verschlüsseln und einen privaten Schlüssel zum entschlüsseln. Will Alice eine geheime Nachricht von Bob empfangen, so schickt sie Bob ihren öffentlichen Schlüssel. Bob verschlüsselt seine Nachricht mit diesem Schlüssel und schickt die Nachricht an Alice. Der private Schlüssel wird nicht übertragen. In dieser Hinsicht sind asymmetrische Verschlüsselungsverfahren sicherer als symmetrische.

Mithilfe von asymmetrischen Verschlüsselungen kann man auch digitale Signaturen erstellen und sich damit authentifizieren. Im RSA-Verfahren sind privater und öffentlicher Schlüssel austauschbar. Das heißt, wenn eine Nachricht mit dem privaten Schlüssel verschlüsselt wird, kann sie mit dem öffentlichen Schlüssel entschlüsselt werden. Da jedoch nur der Besitzer des privaten Schlüssels eine Nachricht erstellen kann, die man mit dem öffentlichen Schlüssel entschlüsseln kann, ist es so möglich den Absender einer Nachricht zu authentifizieren.

[1] [Wrixon], S. 21f
[2] [Birthälmer], S. 4
[3] [Petitcolas]

2 Einwegfunktionen

Eine Einwegfunktion ist in der Mathematik eine Beziehung zwischen zwei Mengen, die „komplexitätstheoretisch „schwer“ umzukehren ist“[4]. Ein Beispiel für eine Einwegfunktion ist die Multiplikation zweier Zahlen. Die Laufzeit des Schönhage-Strassen-Algorithmus zur Multiplikation zweier n-stelliger ganzer Zahlen ist mit $\mathcal{O}(n \cdot \log(n) \cdot \log(log(n)))$[5] deutlich kleiner als die Laufzeit von des Zahlkörpersiebs $\mathcal{O}(e^{(1,92+o(1))\sqrt[3]{\ln n}\sqrt[3]{(\ln \ln n)^2}})$[6], das der Faktorisierung dient.

Die Sicherheit des RSA-Verfahrens zur asymmetrischen Verschlüsselung basiert auf der Annahme, dass die Faktorisierung einer großen Zahl deutlich länger dauert als das Multiplizieren der Primfaktoren. Falls es keinen besseren Algorithmus zur Faktorisierung als zur Multiplikation gibt, ist diese Annahme korrekt. Nach dem Stand von 2009 ist dies der Fall.

Weitere Hinweise zur Sicherheit des RSA-Kryptosystems sind in Kapitel 7.4 zu finden.

3 Restklassen

Teilt man eine ganze Zahl a durch eine ganze Zahl $m \neq 0$, so bleibt ein Rest $r \in \mathbb{N}_0$. Anhand aller möglichen Reste $0 \leq r < m$ teilt man nun alle Zahlen in $|m|$ Teilmengen ein. Diese Teilmengen nennt man Restklassen. Man sagt, alle Zahlen, die den selben Rest r beim Teilen durch m lassen, gehören der selben Restklasse modulo m an[7]. Ein Beispiel aus dem Alltag sind Zeitangaben. Man schreibt nicht 348 Minuten, sondern 5 Stunden und 48 Minuten. Es wird also modulo 60 gerechnet. Auch in der Grund-schule rechnet man mit Restklassen modulo 10, wenn man ganze Zahlen in Einer, Zehner und Hunderter unterteilt.
Ein weiteres Beispiel ist die Einteilung in gerade und ungerade Zahlen. Bleibt bei einer Zahl kein Rest beim Teilen durch zwei, so wird sie als „gerade“ bezeichnet und ist in einer Restklasse modulo 2 mit allen anderen geraden Zahlen.

[4] [Beutelspacher], S. 114
[5] [Pethö], S. 25
[6] [Rothe], S. 384
[7] [Forster], S. 45

4 Eulersche φ-Funktion

Die Eulersche φ-Funktion gibt für jede natürliche Zahl n an, wie viele positive ganze Zahlen $a \leq n$ zu ihr relativ prim sind[8]. a ist zu n relativ prim, wenn $ggT(a, n) = 1$ gilt, also wenn a und n keinen größeren gemeinsamen Teiler als 1 haben. Man sagt auch „a und b sind teilerfremd“.

$\varphi(n)$ ist zugleich die Ordnung der multiplikativen Gruppe $(\mathbb{Z}/n\mathbb{Z})^*$. $\varphi(n)$ gibt also an, wie viele Zahlen im Restklassenring modulo n ein multiplikativ Inverses haben. Mehr dazu in Kapitel 6

Für Primzahlen gilt $\varphi(p) = p - 1$, da eine Primzahl nur durch sich und eins teilbar ist. Sei A die multiplikative Gruppe einer Primzahl p, B die multiplikative Gruppe einer Primzahl q und C die multiplikative Gruppe von $p \cdot q$. Dann ist $|C| = |A| \cdot |B|$ und $\varphi(p \cdot q) = |C|$, $\varphi(p) = |A|$ sowie $\varphi(q) = |B|$.

Daraus folgt, dass $\varphi(pq) = \varphi(p) \cdot \varphi(q) = (p - 1) \cdot (q - 1)$ für zwei Primzahlen $p \neq q$ gilt.

5 Lineare Kongruenzen

5.1 Allgemeine Informationen

Zwei Zahlen $a, b \in \mathbb{Z}$ heißen kongruent modulo $m \in \mathbb{N}$, falls a und b bei der Division durch m den den gleichen Rest lassen. Man schreibt $a \equiv b \pmod{m}$[9].

Gilt $ax \equiv b \pmod{m}$, für $a, b, x \in \mathbb{Z}$ und $m \in \mathbb{N}$, dann bedeutet das, dass $m|(ax - b)$ für ein passendes x. Man nennt $ax \equiv b \pmod{m}$ ein lineares Kongruenzsystem.

[8] [Brill], S. 148
[9] [Reiss], S. 179f

5.2 Chinesischer Restsatz

Der Chinesische Restsatz sagt, ob lineare Kongruenzsysteme lösbar sind und wie diese Lösungen aussehen:

Seien $m_1, m_2, ..., m_n$ paarweise teilerfremde natürliche Zahlen und $a_1, a_2, \ldots, a_n$ ganze Zahlen.

Dann ist das System linearer Kongruenzen

$$x \equiv a_1 \pmod{m_1}, \quad x \equiv a_2 \pmod{m_2}, \quad \ldots, \quad x \equiv a_n \pmod{m_n}$$

lösbar. Alle Lösungen des Systems liegen in einer gemeinsamen Restklasse modulo $M = \prod_{i=1}^{n} m_i$

Beweis nach [Reiss], S. 221f:

(I) $M_j = \frac{M}{m_j}$ für $j = 1, \ldots, n$

(II) $y_j \cdot M_j \equiv 1 \pmod{m_j}$, y_j mit dem erweitertem Euklidischem Algorithmus bestimmen

(III) $a_j \cdot y_j \cdot M_j \equiv a_j \pmod{m_j}$ für $j = 1, \ldots, n$
Weil m_j für $i \neq j$ ein Teiler von M_i ist, gilt auch:

(IV) $a_i \cdot y_i \cdot M_i \equiv 0 \pmod{m_j}$ für alle $i, j = 1, \ldots, n$ mit $i \neq j$

Da alle Summanden bis auf Einen ($j = i$) gleich Null sind, stimmt dieser Ausdruck:

$$a_i \cdot y_i \cdot M_i \equiv \sum_{j=1}^{n} a_j \cdot y_j \cdot M_j \pmod{m_i}$$

$$a_i \equiv \sum_{j=1}^{n} a_j \cdot y_j \cdot M_j \pmod{m_i}, \text{ da } y_i \cdot M_i \equiv 1 \pmod{m_i}$$

a_i ist die Lösung des Kongruenzsystems. Alle Lösungen liegen in dieser Restklasse.

Beispielaufgabe

Folgende Aufgabe wurde [Berendt] entnommen:

> 17 chinesische Piraten erbeuten eine Truhe mit Goldstücken. Beim Versuch, diese gleichmäßig zu verteilen, bleiben 7 Goldstücke übrig. Um diese entbrennt

ein heftiger Streit, bei dem einer der Piraten das Leben lässt. Die verbleibenden 16 versuchen erneut, die Goldstücke gerecht zu verteilen, behalten jedoch elf Stücke übrig. Bei der folgenden Auseinandersetzung geht wieder einer der Streitenden über Bord. Den 15 Überlebenden gelingt dann die Teilung. Wie viele Goldstücke müssen es mindestens gewesen sein?

Restklassensystem

$$
\begin{aligned}
x &:= \text{Anzahl der Goldstücke} \\
x &\equiv 7 \pmod{17} \\
x &\equiv 11 \pmod{16} \\
x &\equiv 0 \pmod{15}
\end{aligned}
$$

Lösung

I Produkte

$$
\begin{aligned}
M &= 17 \cdot 16 \cdot 15 = 4080 \\
M_1 &= \frac{4080}{17} = 240 \\
M_2 &= \frac{4080}{16} = 255 \\
M_3 &= \frac{4080}{15} = 272
\end{aligned}
$$

II Multiplikativ Inverses der Restklassensysteme

$$
\begin{aligned}
9 \cdot 240 &\equiv 1 \pmod{17} \\
15 \cdot 255 &\equiv 1 \pmod{16} \\
8 \cdot 272 &\equiv 1 \pmod{15}
\end{aligned}
$$

III Multiplikation der Restklassensysteme mit a_j

$$
\begin{aligned}
7 \cdot 9 \cdot 240 &\equiv 7 \pmod{17} \\
11 \cdot 15 \cdot 255 &\equiv 11 \pmod{16} \\
8 \cdot 272 &\equiv 0 \pmod{15}
\end{aligned}
$$

IV Berechnung der Lösung des Restklassensystem

$$x = \sum_{j=1}^{3} a_j \cdot y_j \cdot M_j \pmod{15 \cdot 16 \cdot 17} = 7 \cdot 240 \cdot 9 + 11 \cdot 255 \cdot 15 = 57195$$

$$57195 \equiv 75 \pmod{4080}$$

75 ist die kleinste positive Lösung des Kongruenzsystems.

Antwort:

Die Anzahl der von den Piraten erbeuteten Goldstücken muss mindestens 75 betragen, kann aber auch $75 + 1 \cdot 4080$, $75 + 2 \cdot 4080$ oder ein beliebiger anderer positiver Vertreter dieser Restklasse $\pmod{4080}$ sein.

6 Multiplikativ inverses Element

6.1 Definition und Beispiele

Das multiplikativ inverse Element d von e ergibt bei der Multiplikation mit e das neutrale Element der Multiplikation, also die Eins: $d \cdot e = 1$

In $\mathbb{R} \setminus \{ 0 \}$ hat jedes Element ein multiplikativ Inverses, den Kehrbruch. In $\mathbb{Z}/7\mathbb{Z}$ ist das multiplikativ Inverse von zwei in der Restgruppe von vier, da $2 \cdot 4 = 8$ und $8 \equiv 1 \pmod 7$. Mit dem erweitertem euklidischem Algorithmus kann man das multiplikativ Inverse von a in $\mathbb{Z}/n\mathbb{Z}$ finden.

6.2 Erweiterter euklidischer Algorithmus

Sind zwei Zahlen $a > b$ gegeben und will deren größten gemeinsamen Teiler berechnen, so kann man den erweiterten euklidischen Algorithmus anwenden:

1. Größtmögliches q wählen, so dass gilt $a = q_1 \cdot b + r_1$
2. $b = q_2 \cdot r_1 + r_2$
3. $r_1 = q_3 \cdot r_2 + r_3$
4. ...
5. bis $r_{n-2} = q_n \cdot r_{n-1} + r_n$ mit $r_n = 0$

Dann ist $r_{n-1} = ggT(a, b)$

Mit diesem Algorithmus kann man nun das multiplikativ Inverse von a in $\mathbb{Z}/n\mathbb{Z}$ finden, wenn der größte gemeinsame Teiler von a und n gleich 1 ist. Da im vorletzten Schritt $r_{n-1} = 1$ ist, kann man 1 als Linearkombination der Reste von r_{n-3} und r_{n-2} darstellen. Diese Reste kann man wiederum als Linearkombination vorhergehender Reste darstellen. Dies setzt man so lange fort, bis man eine Linearkombination mit a und n von 1 hat. Da wir im Restklassenring n sind, muss man nur das Produkt mit a betrachten und kann das multiplikativ Inverse zu a im Restklassenring $\mathbb{Z}/n\mathbb{Z}$ ablesen.

Hier ein Beispiel zur Veranschaulichung:

Sei $a = (\text{Primzahl}_1 - 1) \cdot (\text{Primzahl}_2 - 1) = (3-1) \cdot (47-1) = 92$ und $b = 71$

Gesucht ist das multiplikativ Inverse $b \in \mathbb{Z}/a\mathbb{Z}$ von $x \cdot 71 \equiv 1 \pmod{92}$:

Schritt 1: euklidischer Algorithmus		**Schritt 2**: nach Rest auflösen
$91 = 1 \cdot 71 + 21$	$\rightarrow$	$21 = 92 - 71$
$71 = 3 \cdot 21 + 8$	$\rightarrow$	$8 = 71 - 3 \cdot 21$
$21 = 2 \cdot 8 + 5$	$\rightarrow$	$5 = 21 - 2 \cdot 8$
$8 = 1 \cdot 5 + 3$	$\rightarrow$	$3 = 8 - 1 \cdot 5$
$5 = 1 \cdot 3 + 2$	$\rightarrow$	$2 = 5 - 1 \cdot 3$
$3 = 1 \cdot 2 + 1$	$\rightarrow$	$1 = 3 - 1 \cdot 2$

Schritt 3: so lange Reste einsetzen, bis eine Linearkombination der Form $1 = x \cdot 92 + y \cdot 71$ gefunden ist:

$1 = 3 - (5 - 3)$	$= 2 \cdot 3 - 5$
$1 = 2 \cdot (8 - 5) - (21 - 2 \cdot 8)$	$= 4 \cdot 8 - 2 \cdot 5 - 21$
$1 = 4 \cdot 8 - 2 \cdot (21 - 2 \cdot 8) - 21$	$= 8 \cdot 8 - 3 \cdot 21$
$1 = 8 \cdot (71 - 3 \cdot 21) - 3 \cdot (92 - 71)$	$= 11 \cdot 71 - 24 \cdot 21 - 3 \cdot 92$
$1 = 11 \cdot 71 - 3 \cdot 92 - 24 \cdot (92 - 71)$	$= 35 \cdot 71 - 27 \cdot 92$

Das bedeutet 35 ist das multiplikativ Inverse zu 71 in $\mathbb{Z}/92\mathbb{Z}$ und erfüllt damit die Kongruenzgleichung $35 \cdot 71 \equiv 1 \pmod{92}$.

Zusätzlich hat man damit weitere multiplikativ Inverse gefunden:

- $-27 \cdot 92 \equiv 1 \pmod{71}$
- $-27 \cdot 92 \equiv 1 \pmod{35}$
- $35 \cdot 71 \equiv 1 \pmod{27}$

7 RSA-Verfahren

Das RSA-Verfahren ist ein asymmetrisches Kryptosystem, das von Ronald Linn **R**ivest, Adi **S**hamir und Leonard **A**dleman entwickelt und im August 1977 veröffentlicht wurde[10].

Da das RSA-Verfahren asymmetrisch ist, wird sowohl ein öffentlicher Schlüssel als auch ein privater Schlüssel benötigt. Der öffentliche Schlüssel dient der Verschlüsselung und besteht aus dem RSA-Modul N und dem Verschlüsselungsexponenten e. Der private Schlüssel besteht aus dem Entschlüsselungsexponenten d und dem selben RSA-Modul N.

Die Erzeugung dieser drei Zahlen N, e und d für das RSA-Verfahren kann, in Anlehnung an [wiki-RSA], in fünf Schritte unterteilt werden:

Schritt 1: Als erstes werden zwei große Primzahlen gewählt. Je größer diese Primzahlen sind, desto sicherer ist die Verschlüsselung. (vgl. Kapitel 7.4)

Schritt 2: In diesem Schritt wird RSA-Modul $N = p \cdot q$ berechnet, das ein Teil des öffentlichen Schlüssels ist.

Schritt 3: Schritt 3: Nun wird der Wert der Eulerschen φ-Funktion bei N berechnet. Da p und q Primzahlen sind, gilt $\varphi(N) = (p-1) \cdot (q-1)$ (vgl. Kapitel 4)

Schritt 4: Nachdem $\varphi(N)$ ermittelt wurde, kann der zweite Teil des öffentlichen Schlüssel, der Verschlüsselungsexponent e, ermittelt werden. Folgende Bedingungen müssen für e gelten: $1 < e < \varphi(N)$ und $ggT(e, \varphi(N)) = 1$

Schritt 5: Es folgt die Berechnung des Entschlüsselungsexponenten d als multiplikativ Inverses von e bezüglich des Modules $\varphi(N)$. Es soll also folgende Kongruenz gelten: $e \cdot d \equiv 1 \pmod{\varphi(N)}$

7.1 Verschlüsselung mit dem öffentlichen Schlüssel

Ein Text wird verschlüsselt, indem er in Zahlen umgewandelt wird. Für diese Umwandlung kann der ASCII-Code verwendet werden. Sobald man Zahlen hat, kann

[10] [msri], S. 2

man den unverschlüsselten Klartext K mit folgender Kongruenzgleichung in den verschlüsselten Geheimtext G umwandeln: $G \equiv K^e \pmod{N}$.
Die Zahl N sollte deutlich größer sein als K[11].

In folgendem Beispiel wird der Klartext „12“ verschlüsselt. Dazu werden als Erstes alle benötigten Variablen erzeugt oder berechnet:

Schritt 1: $p := 347$ und $q := 379$

Schritt 2: $N = p \cdot q = 347 \cdot 379 = 131513$

Schritt 3: $\varphi(N) = \varphi(131513) = (347 - 1) \cdot (379 - 1) = 346 \cdot 378 = 130788$

Schritt 4: $1 < (e := 877) < \varphi(N)$

Schritt 5: $877 \cdot d \equiv 1 \pmod{130788}$

Um ein d zu finden, das diese Kongruenz erfüllt, wird der erweiterte euklidische Algorithmus angewendet:

Schritt 1: euklidischer Algorithmus		**Schritt 2**: nach Rest auflösen
$130788 = 149 \cdot 877 + 115$	$\rightarrow$	$115 = 130788 - 149 \cdot 877$
$877 = 7 \cdot 115 + 72$	$\rightarrow$	$72 = 877 - 7 \cdot 115$
$115 = 1 \cdot 72 + 43$	$\rightarrow$	$43 = 115 - 72$
$72 = 1 \cdot 43 + 29$	$\rightarrow$	$29 = 72 - 43$
$43 = 1 \cdot 29 + 14$	$\rightarrow$	$14 = 43 - 29$
$29 = 2 \cdot 14 + 1$	$\rightarrow$	$1 = 29 - 2 \cdot 14$

Schritt 3: so lange Reste einsetzen, bis eine Linearkombination der Form $1 = x \cdot 877 + y \cdot 130788$ gefunden ist:

[11] vgl. [Reiss], S. 288

$$1 = 29 - 2 \cdot (43 - 29)$$

$$1 = 3 \cdot 29 - 2 \cdot (115 - 72)$$

$$1 = 3 \cdot (72 - 43) - 2(130788 - 149 \cdot 877) + 2 \cdot (877 - 7 \cdot 115)$$

$$1 = 3 \cdot (877 - 7 \cdot 115) - 3 \cdot (115 - 72) - 2 \cdot 130788 + 300 \cdot 877 - 14 \cdot 155$$

$$1 = 303 \cdot 877 - 2 \cdot 130788 - 38 \cdot 115 + 3 \cdot 72$$

$$1 = 303 \cdot 877 - 2 \cdot 130788 - 38 \cdot (130788 - 149 \cdot 877) + 3 \cdot (877 - 7 \cdot 115)$$

$$1 = 5968 \cdot 877 - 40 \cdot 130788 - 21 \cdot (130788 - 149 \cdot 877) = 9097 \cdot 877 - 61 \cdot 130788$$

$\rightarrow d = 9097$

Probe:

$$\frac{d \cdot e - 1}{\varphi(N)} = \frac{877 \cdot 9097 - 1}{130788} = 61$$

Nun wird der Geheimtext G mit dem Verschlüsselungsexponenten e aus dem Klartext K berechnet:

$$G \equiv K^e \pmod{N}$$

$$G \equiv 12^8 77 \equiv 12 \cdot (12^6)^{2 \cdot 73} \equiv 12 \cdot 92698^{2 \cdot 73} \equiv 12 \cdot 122810^{73} \pmod{131513}$$

$$G \equiv 12 \cdot 122810 \cdot 122810^{2 \cdot 2 \cdot 2 \cdot 3 \cdot 3} \equiv 27077 \cdot 122234^{2 \cdot 2 \cdot 3 \cdot 3} \equiv 27077 \cdot 90339^{2 \cdot 3 \cdot 3} \pmod{131513}$$

$$G \equiv 27077 \cdot 95706^9 \equiv 27077 \cdot 9189^3 \equiv 27077 \cdot 56590 \pmod{131513}$$

$$G \equiv 29467 \pmod{131513}$$

7.2 Entschlüsselung mit dem privaten Schlüssel

7.2.1 Entschlüsselung ohne den Chinesischen Restsatz

Der Geheimtext wird entschlüsselt, indem man ihn mit d potenziert und dann denn kleinsten positiven Repräsentanten der Restklasse N berechnet: $K \equiv G^d \pmod{N}$

Im Folgenden wird die Entschlüsselung auf das vorhergehende Beispiel angewendet:

$G = 29467 \quad N = 131513 \quad d = 9097$

$K \equiv G^d \pmod{N}$

$K \equiv 29467^{9097} \pmod{131513}$

$K \equiv 29467 \cdot (29467^2)^{2 \cdot 2 \cdot 3 \cdot 379} \equiv 29467 \cdot (55263^2)^{2 \cdot 3 \cdot 379} \equiv 29467 \cdot 4283^{2 \cdot 3 \cdot 379} \pmod{131513}$

$K \equiv 29467 \cdot 89937^{2 \cdot 3} \equiv 29467 \cdot 88417^3 \equiv 29467 \cdot 24587 \pmod{131513}$

$K \equiv 12 \pmod{131513}$

7.2.2 Entschlüsselung mit dem Chinesischen Restsatz

Falls der Entschlüsselungsexponent d sehr groß ist, kann das Potenzieren lange dauern. Dies kann mit dem Chinesischem Restsatz beschleunigt werden[12]. Diese Idee wurde [Paixão] entnommen:

Der Geheimtext G und der Exponent d wird folgendermaßen geteilt:

$$\begin{matrix} G_p \equiv G \pmod{p} \\ G_q \equiv G \pmod{q} \end{matrix} \quad \text{und} \quad \begin{matrix} d_p \equiv d \pmod{p-1} \\ d_q \equiv d \pmod{q-1} \end{matrix} \quad , \text{da } e \cdot d \equiv 1 \pmod{(p-1) \cdot (q-1)})$$

Dies funktioniert, da der folgende Hilfssatz gilt:

$$K^e d \equiv G \pmod{p \cdot q} \Leftrightarrow K^{ed} \equiv G \pmod{p} \wedge K^{ed} \equiv G \pmod{q}$$

Nun wird dieser kleinere Teil von G mit den kleineren Exponenten „entschlüsselt“:

$$K_p \equiv {G_p}^{d_p} \pmod{p}$$
$$K_q \equiv {G_q}^{d_q} \pmod{q}$$

Der Chinesische Restsatz ermöglicht es nun, eine Lösung K für die folgenden Kongruenzen zu finden:

[12] [Paixão], S. 1

$$K \equiv K_p \pmod{p}$$
$$K \equiv K_q \pmod{q}$$

Dazu müssen die multiplikativ inversen Elemente ermittelt werden:

$$y_p \cdot p \equiv 1 \pmod{q}$$
$$y_q \cdot q \equiv 1 \pmod{p}$$

Nun wird der Klartext zusammengesetzt:

$$K \equiv (K_p \cdot y_p \cdot q + K_q \cdot y_q \cdot p) \pmod{N}$$

Diese Methode ist etwa vier mal schneller als die Entschlüsselung mit d[13].

7.3 Entschlüsselung ohne den privaten Schlüssel

Ist der Entschlüsselungsexponent d unbekannt, so kann dieser durch Faktorisierung von N zurückgewonnen werden. Faktorisiert man N, so kann $\varphi(N)$ berechnet werden und die Kongruenz $e \cdot d \equiv 1 \pmod{\varphi(N)}$ gelöst werden.

Als praktische Arbeit wurde mir folgende Aufgabe gestellt:

```
N := 65937_24735_90338_11941_75738_06421_27758_89771;
e := 47;
# geheime Botschaft:
y := 65087_24329_19692_65260_21005_67465_76581_27499;
```

Wie lautet die Botschaft im Klartext?

Um diese Aufgabe zu lösen, versucht man den privaten Schlüssel mit Hilfe des öffentlichen Schlüssels wiederherzustellen.

Der private Schlüssel d muss die Kongruenzgleichung $e \cdot d \equiv 1 \pmod{\varphi(N)}$ erfüllen, allerdings sind $\varphi(N)$ und d nicht bekannt. Um $\varphi(N)$ zu berechnen, werden die

[13] [Paixão], S. 2

Primfaktoren p und q, aus denen N besteht, benötigt. N muss also faktorisiert werden. Sobald ein Faktor von N bekannt ist, kann man N durch diesen Faktor teilen und man erhält den zweiten Faktor. Dann kann man wie in Kapitel 7 vorgehen und d berechnen. Mit d kann man wie in Kapitel 7.2 beschrieben die Geheimbotschaft entschlüsseln.

Wird das Aribas-Skript im Anhang ausgeführt, das diese Schritte ausführt, erhält man den Klartext „**RSAribas**“.

7.4 Sicherheit des RSA-Algorithmus

Der RSA-Algorithmus kann, wie oben beschrieben, „geknackt“ werden, indem der öffentliche Schlüssel faktorisiert wird. Allerdings ist das bei großen Zahlen nahezu unmöglich, da die Algorithmen sehr lange Laufzeiten haben[14]. Wie lange die Faktorisierung dauert hängt einerseits vom Algorithmus, andererseits von der Hardware ab.

Um ein Gefühl dafür zu bekommen was „sehr lange“ bedeutet, kann man die „RSA Factoring Challenge“ als Beispiel betrachten. Eine 193-stellige Zahl sollte Faktorisiert werden. Für diese Aufgabe wurde am 18. März 1991 ein Preisgeld von 20.000 US-Dollar ausgeschrieben. Ein Team des Instituts für Experimentelle Mathematik in Essen hat am 2. November 2005 diese Aufgabe gelöst und dafür über fünf Monate benötigt. Der Rechenaufwand betrug etwa 30 2.2GHz-Opteron-CPU Jahre[15].

Das Faktorisieren wird zwar immer schneller, da man über bessere und billigere Computer sowie effizientere Algorithmen verfügt, allerdings steigt mit den Hardwareverbesserungen die Sicherheit des RSA-Algorithmus. Mit einer besseren Hardware können deutlich größere Primzahlen erzeugt werden und damit auch ein deutlich längerer öffentlicher Schlüssel generiert werden, der zur Verschlüsselung der Nachricht dient.

[14] [RSA-2190]
[15] [RSA-2964]

Ein verbesserter Faktorisierungsalgorithmus würde die Sicherheit des RSA-Kryptosystems gefährden. Es ist jedoch unwahrscheinlich, dass ein solcher Algorithmus gefunden wird, da das Problem als schwer angesehen wird.[16]

Ein weiterer möglicher Angriff auf das RSA-Kryptosystem wäre die e-te Wurzel (mod n) zu finden. Da $G \equiv K^e \pmod{N}$ ist, wäre $K \equiv \sqrt[e]{K^e} \pmod{N}$. Allerdings ist kein Angriff, der so funktioniert, bekannt4. Um zu veranschaulichen, dass bei einer Kongruenzgleichung nicht wie gewohnt die Wurzel gezogen werden kann, betrachte man folgendes Beispiel: $4 \equiv 16 \pmod{3}$, aber $\sqrt{4} \neq \sqrt{16} \pmod{3}$.

7.5 Warum der RSA-Algorithmus funktioniert

Der Geheimtext G wird durch potenzieren mit e gewonnen ($G \equiv K^e \pmod{N}$) und der Klartext durch potenzieren mit d ($K \equiv G^d \pmod{N}$). Zu zeigen ist, dass $K \equiv K^{e \cdot d} \pmod{N}$ mit $ed \equiv 1 \pmod{\varphi(N)}$ gilt.

Es kann folgende Umformung durchgeführt werden:

$$ed \equiv 1 \pmod{\varphi(N)}$$
$$ed = 1 + x \cdot \varphi(N) \text{mit } x \in \mathbb{Z}$$

Daraus folgt:

$$K^{e \cdot d} = K^{1 + x \cdot \varphi(N)} = K \cdot K^{x \cdot \varphi(N)}$$

Nun wird der **Satz von Fermat-Euler** als Hilfssatz eingeführt:

$K^{\varphi(N)} \equiv 1 \pmod{N}$ für $K, N \in \mathbb{N}$ und $ggT(N, K) = 1$

Beweis nach [Reiss], S. 218 f.: Es gibt $\varphi(N)$ verschiedene, zu N teilerfremde Reste. Da K zu N teilerfremd ist, muss auch jedes der Produkte $K \cdot r_1, K \cdot r_2, \ldots, K \cdot r_{\varphi(N)}$ zu N teilerfremd sein.

[16] [RSA-2191]

Es gilt:

$$\prod_{i=1}^{n=\varphi(N)} K \cdot r_i \equiv \prod_{i=1}^{n=\varphi(N)} r_i \pmod{N}$$

$$K^{\varphi(N)} \prod_{i=1}^{n=\varphi(N)} r_i \equiv \prod_{i=1}^{n=\varphi(N)} r_i \pmod{N}$$

Da alle r_i teilerfremd zu N sind, kann man durch das Produkt teilen. Daraus ergibt sich:

$$K^{\varphi(N)} \equiv 1 \pmod{N}$$

Das bedeutet für den RSA-Algorithmus:

$$K \cdot K^{x \cdot \varphi(N)} \equiv K \cdot (K^{\varphi(N)})^x \equiv K \cdot 1^x \equiv K \pmod{N}$$

8 Literaturverzeichnis

Beutelspacher, A., Neumann, H. B. und Schwarzpaul, T.:

Kryptografie in Theorie und Praxis.

Wiesbaden, Vieweg+Teubner Verlag, 2005.

Brill, M.:

Mathematik für Informatiker.

Wien, Hanser Verlag, 2004.

Forster, O.:

Algorithmische Zahlentheorie.

Wiesbaden, Vieweg Braunschweig/Wiesbaden, 1996.

Pethö, A. und Pohst, M.:

Algebraische Algorithmen.

Wiesbaden, Vieweg+Teubner Verlag, 1999.

Reiss, K. und Schmieder, G.:

Basiswissen Zahlentheorie.

Berlin Heidelberg, Springer-Verlag, 2007.

Rothe, J.:

Komplexitätstheorie und Kryptologie.

Berlin, Springer, 2008.

Wrixon, F. B.:

Geheimsprachen.

Königswinter, Tandem Verlag GmbH, 2006.

Internetadressen

[Berendt] Berendt, Gerhard.
Seminar über Zahlentheorie/Kryptographie. 10.04.2008
URL: http://userpage.fu-berlin.de/~berendt/lehre2008_neu.html
[Stand: 09.01.2010]

[Birthälmer] Birthälmer, Melita.
Kryptografie 13.04.2008
URL: http://www.birthaelmer.com/fileadmin/birthaelmer/portfolio/Kryptografie_web.pdf
[Stand: 09.01.2010]

[msri] SIAM News.
Still Guarding Secrets after Years of Attacks, RSA Earns Accolades for its Founders. 17.06.2003
URL: http://www.msri.org/people/members/sara/articles/rsa.pdf
[Stand: 09.01.2010]

[Paixão] Paixão, Cesar A. M..
An efficient variant of the RSA cryptosystem. 11.08.2009
URL: http://eprint.iacr.org/2003/159.pdf
[Stand: 26.11.2009]

[Petitcolas] Petitcolas, Fabien.
DESIDERATA DE LA CRYPTOGRAPHIE MILITAIRE. 20.06.2009
URL: http://www.petitcolas.net/fabien/kerckhoffs/la_cryptographie_militaire_i.htm#desider
[Stand: 09.01.2010]

[RSA-2190] RSA Laboratories.
What are the best factoring methods in use today? 03.11.2009
URL: http://www.rsa.com/rsalabs/node.asp?id=2190
[Stand: 09.01.2010]

[RSA-2191] RSA Laboratories.
What improvements are likely in factoring capability? 03.11.2009
URL: http://www.rsa.com/rsalabs/node.asp?id=2191
[Stand: 09.01.2010]

[RSA-2216] RSA Laboratories.
What would it take to break the RSA cryptosystem? 03.11.2009
URL: http://www.rsa.com/rsalabs/node.asp?id=2216
[Stand: 09.01.2010]

[RSA-2964] RSA Laboratories.
RSA-640 factored! 03.01.2010
URL: http://www.rsa.com/rsalabs/node.asp?id=2964
[Stand: 09.01.2010]

[wiki-RSA] Ladenthin, Bernhard.
RSA-Kryptosystem. 08.01.2010
URL: http://de.wikipedia.org/wiki/RSA-Kryptosystem#Erzeugung_des_öffentlichen_und_p
[Stand: 09.01.2010]

9 Anhang

```
# Gegebener Öffentlicher Schlüssel N = p*q und e sowie Geheimtext y
N := 659372473590338119417573806421277588977 1.
e := 47.
y := 650872432919692652602100567465765812749 9.
p := ec_factorize(N).          # 6890701303127411
q := N div p.                  # 956207997714884457630761
phi :=  (p-1)*(q-1).     #6588943695914805815679579653627015131600
d := mod_inverse(e, phi).
cc := byte_string(y).
z := y**d mod N.
string(byte_string(z)).
```

Ich erkläre, dass ich die Facharbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benützt habe.

Ort, Datum	Unterschrift