Kryptographie verstehen
Ein schülergerechter Zugang zum RSA-Verfahren

Dr. Hermann Puhlmann
Arbeitsgruppe Fachdidaktik
Fachbereich Mathematik
Technische Universität Darmstadt
puhlmann@mathematik.tu-darmstadt.de

August 1998

1 Einführung

Die Bedeutung der Kryptographie wurde schon von verschiedenen Autoren hervorgehoben. Dabei betonten sie ihre Bedeutung in Hinblick auf den globalen Datenaustausch durch das Internet und schilderten die Vorzüge asymmetrischer Kryptosysteme wie des RSA-Verfahrens oder des Verfahrens von ElGamal [BB96, Bau96, Sch96].

Der Kern dieser Verfahren besteht im Auffinden zweier zueinander inverser Funktionen f und g, die sich zwar mit jedem Computer problemlos auswerten lassen, die jedoch die Eigenschaft haben, dass sich auch bei Kenntnis der einen Funktion die dazu inverse praktisch nur mit Hilfe einer geheimen Zusatzinformation ermitteln lässt. Ohne diese Zusatzinformation erfordert das Auffinden der Umkehrfunktion dagegen einen so hohen Rechenaufwand, dass selbst die schnellsten verfügbaren Computer an dieser Aufgabe scheitern.

Dies ist zunächst eine erstaunliche Eigenschaft der Funktionen f und g, zumal ihre Definitionsbereiche Restklassenringe $\mathbb{Z}_n$, also endliche Mengen sind.

Dieser Beitrag stellt einen Zugang zur Mathematik der Kryptographie vor, der an die Kenntnisse der Schülerinnen und Schüler über die Rechengesetze im Bereich der reellen Zahlen anknüpft und so ein besseres Verständnis für das Rechnen in Restklassenringen wecken soll. Dabei wird nur auf mathematische Inhalte der Sekundarstufe I aufgebaut. Im Rahmen eines mathematisch-informatischen Projektes, das sich an die Potenz- und Wurzelrechnung in Klasse 9/10 anschließt, könnte der Begriff des Potenzierens in einem ganz anderen Zusammenhang noch einmal beleuchtet werden, und die Schülerinnen und Schüler könnten in der hier gezeigten Weise wesentliche Teile des RSA-Verfahrens selbst entdecken.

2 Cäsar-Chiffre: Rechnen modulo 26

Wir beginnen mit einer sogenannten Cäsar-Chiffre. Dabei werden alle Buchstaben des Alphabets um eine bestimmte Stellenzahl s weitergeschoben, d. h. beim

Verschlüsseln wird jeder Buchstabe des Originaltextes durch den Buchstaben ersetzt, der um s Positionen weiter hinten im Alphabet steht (wobei man nach dem Buchstaben „z“ wieder mit „a“ beginnt).

Natürlich kennen die Schüler diese Art der Verschlüsselung längst, sie dient aber der Einübung von Sprechweisen und der Einführung des Rechnens modulo n. Ordnen wir nämlich den Buchstaben die Zahlen von 0 (für „a“) bis 25 (für „z“) zu, so lässt sich die Verschlüsselung als Funktion auf der Menge $\mathbb{Z}_{26}$ der Zahlen von 0 bis 25 schreiben:

$$f(x) = x + s \bmod 26 \ .$$

Die Funktion f ordnet der Position x eines Buchstabens die um s verschobene Position des verschlüsselten Buchstabens zu. Die Rechnung modulo 26 bewirkt, dass nach dem „z“ mit dem „a“ fortgefahren wird. Es ist klar, dass man nur die Zahl s kennen muss, um jeden Text ver- und entschlüsseln zu können. Die Zahl s ist deshalb der *Schlüssel* des Kryptoverfahrens. Zum Entschlüsseln muss man von jeder Positionsnummer die Zahl s subtrahieren. Dies lässt sich unmittelbar ausführen.

Im Weiteren wird es wichtig sein, zueinander inverse Funktionen über nur *eine* Rechenart zu beschreiben. Daher sollte man auch aufzeigen, dass es zu s ein additiv Inverses gibt, also eine Zahl $t \in \{0, \ldots, 25\}$ mit $s + t = 0 \bmod 26$, das heißt $t = -s \bmod 26$. Zum Beispiel ist für $s = 7$ das additiv Inverse $t = 19$. Damit erhalten wir als Entschlüsselungsfunktion

$$g(y) = y + t \bmod 26 \ .$$

Die Verschlüsselungsfunktion f wird auf jeden Buchstaben eines Textes angewandt. So wird der Text „geheimtext“ mit dem Schlüssel $s = 7$ zu „nlolptalea“ codiert, und mit $t = 19$ entsteht wieder der ursprüngliche Text. Anhand dieses kleinen Beispiels erkennt man, dass die Cäsar-Verschlüsselung leicht zu knacken ist, und zwar durch eine Häufigkeitsanalyse der Buchstabenverteilung. Da in „nlolptalea“ der Buchstabe „l“ dreimal vorkommt, kann man raten, dass er für den im Deutschen häufig vorkommenden Buchstaben „e“ im Originaltext steht. Da das „l“ im Alphabet sieben Buchstaben nach dem „e“ kommt, ist damit der verwendete Schlüssel schon gefunden. (In der Praxis braucht man für gute Häufigkeitsanalysen natürlich längere Textpassagen.)

3 Verschlüsseln durch Multiplikation

Statt nun gleich zum RSA-Verfahren überzugehen, bei dem im Restklassenring potenziert wird, schlage ich vor, zunächst die Verschlüsselung durch Multiplikation zu betrachten. Dabei lassen wir die Einschränkung auf 26 verschiedene Buchstaben fallen. Statt dessen betrachten wir die n möglichen Botschaften von 0 bis $n - 1$, die wir durch Multiplikation mit einem Schlüssel s codieren (vgl. hierzu das Beispiel auf Seite 4). Eine Botschaft b wird damit verschlüsselt zu

$$f(b) = b \cdot s \bmod n \ .$$

Die Schülerinnen und Schüler schlagen sofort vor, zum Entschlüsseln durch s zu dividieren. Wie jedoch eine Division durch s durchzuführen ist, ist keineswegs klar. Damit haben wir eine erste Funktion, deren Umkehrung (zunächst) nicht leicht zu finden ist.

Um die Aufgabenstellung zu verdeutlichen, schreiben wir sie noch einmal anders auf. Gesucht ist eine Funktion g, die $f(b)$ wieder in b überführt, also

$$g(f(b)) = g(b \cdot s) = b \bmod n \quad .$$

Zunächst könnte man schreiben $g(y) = \frac{y}{s} \bmod n$, doch was ist $\frac{y}{s}$ in $\mathbb{Z}_n$? Ein Rückgriff auf die Formulierung bei der Cäsar-Chiffre bringt uns der Lösung näher. Statt zu subtrahieren, haben wir dort das additiv Inverse addiert. Im neuen Fall ist also mit dem multiplikativ Inversen s^{-1} von s zu multiplizieren, für das wir bewusst $\frac{1}{s}$ schreiben:

$$g(y) = y \cdot \frac{1}{s} \bmod n \quad .$$

Was aber ist $\frac{1}{s}$? Nun, $t = \frac{1}{s}$, falls $s \cdot t = 1$. Wir können also in einer Multiplikationstabelle nachschauen, welche Zahl t mit der Zahl s das Produkt 1 bildet. Abbildung 1 zeigt als Beispiele die Tabellen zur Multiplikation modulo 5 bzw. 6.

$\cdot$ mod 5	0	1	2	3	4
0	0	0	0	0	0
1	0	**1**	2	3	4
2	0	2	4	**1**	3
3	0	3	**1**	4	2
4	0	4	3	2	**1**

$\cdot$ mod 6	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	**1**	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	**1**

Abbildung 1: Die Multiplikationstabellen modulo 5 bzw. 6

Aus der mod 5-Tabelle lesen wir ab, dass $3 \cdot 2 = 1 \bmod 5$ ist. Zu $s = 3$ ergibt sich daher $\frac{1}{s} = \frac{1}{3} = 2 \bmod 5$. Zu $s = 0$ können wir kein multiplikativ Inverses finden, aber dies wird die Schülerinnen und Schüler nicht überraschen, da man durch Null nicht dividieren darf. Erstaunlicher ist die zweite Tabelle. Denn hier zeigt sich, dass es außer der Null noch weitere Zahlen gibt, durch die nicht dividiert werden kann. Ob man an dieser Stelle mitteilt, begründet oder die Schülerinnen und Schüler selbst herausfinden lässt, dass $\mathrm{ggT}(s, n) = 1$ sein muss, damit das gewünschte t mit $s \cdot t = 1 \bmod n$ existiert, hängt von der Unterrichtssituation ab. Wichtig ist jedoch die Erkenntnis, dass nicht alle Zahlen in $\mathbb{Z}_n$ ein multiplikativ Inverses haben und hierauf bei der Wahl des Schlüssels Rücksicht zu nehmen ist.

An dieser Stelle sei noch einmal auf die Sprachregelung eingegangen. Um die Analogie zum Rechnen mit reellen Zahlen herauszuarbeiten, ist es günstig, tatsächlich von $\frac{1}{s}$, also von einem „s-tel“ zu reden. Außerdem ist es sinnvoll, die Schreibweise $y \cdot \frac{1}{s}$ und nicht $\frac{y}{s}$ zu wählen, denn hierin kommt zum Ausdruck, dass der Schlüssel $\frac{1}{s}$ zum Dechiffrieren nur vom Chiffrierschlüssel s (und der Zahl n) und nicht von der verschlüsselten Botschaft y abhängt.

Um die Verwendung von öffentlichem und privatem Schlüssel in der Kryptographie zu illustrieren, plädiere ich dafür, nun die Chiffrierung durch Multiplikation tatsächlich in der Lerngruppe anzuwenden. Jeder Teilnehmer legt dazu zwei Zahlen n und s als öffentlichen Schlüssel fest und merkt sich seinen privaten Schlüssel $\frac{1}{s} \bmod n$. Jetzt können die verschiedenen Einsatzweisen der Chiffrierung, wie das Senden geheimer Nachrichten, die elektronische Unterschrift oder der Authentisierungsmechanismus mit konkreten Beispielen eingeübt werden. Aufgrund der Bekanntgabe von n und s können die Schüler mit der Multiplikationstabelle oder durch Probieren auch leicht gegenseitig ihre Codes knacken. Trotzdem entsteht ein Eindruck davon, dass die Inverse einer Funktion nicht immer leicht bzw. nur mit mathematischem Wissen zu finden ist.

Der folgende Abschnitt illustriert das Vorgehen beim multiplikativen Verschlüsseln an Beispielen für kleine und mittelgroße Moduln n. Dabei werden auch die erwähnten Einsatzweisen der Chiffrierung vorgestellt, die in [Bau96, Sch96] ausführlicher dargestellt sind.

Multiplikative Verschlüsselung am Beispiel

Wir verschlüsseln wieder den Text „geheimtext“. Zuerst rechnen wir modulo 27. Damit können wir wieder buchstabenweise verschlüsseln, indem die Buchstaben die Nummern 1 für „a“ bis 26 für „z“ erhalten. Hinzu nehmen wir „–“ als nulltes Zeichen (da man dies besser als ein Leerzeichen sieht). Der Schlüssel sei $s = 10$. Die Nummer des „g“ ist 7, diese wird zu $7 \cdot 10 = 16 \bmod 27$ verschlüsselt, was dem Buchstaben „p“ entspricht. Auf diese Weise wird „geheimtext“ zu „pwzwivkwxk“. Der inverse Schlüssel zu $s = 10$ ist $t = 19$. Dies überprüft man leicht, denn $10 \cdot 19 = 190 = 7 \cdot 27 + 1 = 1 \bmod 27$. Die Multiplikation mit 19 überführt „pwzwivkwxk“ also wieder in „geheimtext“.

Auch in diesem Beispiel läßt sich der Schlüssel leicht herausfinden. Man erkennt „w“ als häufigsten Buchstaben und vermutet, dass er dem „e“ im Originaltext entspricht. Nun hat „e“ die Nummer 5 und „w“ die 23. Es ist s daher so zu bestimmen, dass $s \cdot 5 = 23 \bmod 27$. Hierzu findet man, z. B. mit einer Multiplikationstafel, die Lösung $s = 10$.

Das multiplikative Verschlüsseln modulo 27 ist also nicht sicherer als die Cäsar-Chiffre. Während bei dieser das Alphabet „weitergeschoben“ wird, wird es bei jenem in einer gewissen Weise permutiert. Die Kenntnis eines Originalbuchstabens zusammen mit seiner Verschlüsselung genügt aber zum Knacken des Codes.

Die Häufigkeitsanalyse wird aufwendiger, wenn jeweils eine ganze Gruppe von Buchstaben durch eine codierte Gruppe ersetzt wird. Zur Illustration verwenden wir Gruppen aus drei Buchstaben. Mit den oben angegeben Nummern der Buchstaben können wir eine solche Gruppe als Zahl im 27-er System lesen. Also

$$\text{„}geh\text{“} \mathrel{\hat{=}} 7 \cdot 27^2 + 5 \cdot 27 + 8 = 5246 \ .$$

Rechnen wir modulo $27^3 = 19683$, so wird das Ergebnis jeder Rechnung sich auch wieder als eine Gruppe von drei Buchstaben interpretieren lassen. Mit dem Schlüssel $s = 1000$ wird aus „geh“ damit $5246 \cdot 1000 = 10322 \bmod 19683$, was der Buchstabenfolge „ndh“ entspricht. Wir füllen „geheimtext“ noch mit Leerzeichen auf, damit die Buchstabenanzahl ein Vielfaches von drei ist. Dann wird „geheimtext−−“ mit diesem Verfahren zu „ndhedmvbxt−−“ verschlüsselt.

Der inverse Schlüssel zu $s = 1000$ ist $t = 12853$. Dies herauszufinden, dürfte mit einer Multiplikationstabelle bereits mühsam sein. Die Überprüfung des Ergebnisses ist aber wieder einfach, denn $1000 \cdot 12853 = 1 \bmod 19683$.

Zum Schluss dieses Beispiels sei noch dargestellt, wie die beiden Schlüssel s und t als öffentlicher und privater Schlüssel eingesetzt werden können. Wir nehmen an, Schüler A hat die beiden Schlüssel bestimmt. Er gibt nun den Schlüssel $s = 1000$ öffentlich bekannt (die Zahl $n = 19683$ verwendet die ganze Lerngruppe, d. h. n ist auch bekannt). Damit kann eine Schülerin B einen Text verschlüsseln und geheim an A schicken, z. B. den Text „geheimtext−−“ in der Form „ndhedmvbxt−−“. Schüler A verwendet seinen privaten Schlüssel $t = 12853$ und entschlüsselt damit die geheime Botschaft.

Im übrigen kann A nicht sicher sein, dass die empfangene Botschaft tatsächlich von B stammt. Da der Schlüssel s öffentlich bekannt ist, könnte auch der Mitschüler C eine Botschaft an A geschickt haben, in der er behauptet, B zu sein. Hier kommt die Verwendung des privaten Schlüssels zur digitalen Unterschrift ins Spiel. Wir formulieren dies für den Fall, dass A eine Nachricht „unterschreiben“ will. Dazu verschlüsselt A die Botschaft mit seinem privaten Schlüssel t. Der Text „geheimtext−−“ würde dann nicht zu „ndhedmvbxt−−“, sondern zu „qfhinmohxt−−“. Jeder andere kann nun den öffentlichen Schlüssel s verwenden, um die Botschaft zu entschlüsseln. Die Botschaft ist also nicht geheim. Es ist aber garantiert, dass sie nur von A kommen kann, denn niemand sonst konnte sie durch Verschlüsselung mit t „unterschreiben“.

Möchte A nun eine geheime Botschaft an B senden, die zudem unterschrieben ist, so kann er dies folgendermaßen tun: Er verwendet einen öffentlichen Schlüssel s_B von B, um die Botschaft zu verschlüsseln. Damit ist sichergestellt, dass nur B die Botschaft lesen kann. Diese verschlüsselte Botschaft wird von A nun aber ein weiteres mal verschlüsselt, nämlich mit seinem privaten Schlüssel t_A. Das stellt sicher, dass die Botschaft nur von A kommen kann. Die Empfängerin B verwendet nun nacheinander den öffentlichen Schlüssel s_A von A und ihren eigenen privaten Schlüssel t_B zum Entschlüsseln.

4 Das RSA-Verfahren

In diesem Abschnitt wird ein für Schüler gangbarer Weg zum „Nacherfinden“ des RSA-Verfahrens vorgestellt. Dabei wird es wichtig sein, viele Zahlenbeispiele zu betrachten: Zuerst solche mit kleinen Zahlen, dann auch „realistischere“ mit größeren Zahlen. Im Rahmen dieses Artikels kann jeweils nur ein Beispiel angegeben werden. Es sei aber empfohlen, die angegebenen Beispiele beim Lesen selbst nachzurechnen und weitere zu erfinden.

4.1 Verschlüsseln durch Potenzieren

Das RSA-Verfahren (benannt nach den Erfindern R. L. **R**ivest, A. **S**hamir und L. **A**dleman [RSA78]) schließt sich an die bis jetzt behandelten Chiffriertechniken gut an: Nach Addieren und Multiplizieren wird jetzt durch Potenzieren verschlüsselt. Betrachten wir dies, ohne bereits auf RSA-Details einzugehen. Mit einem Schlüssel s wird in $\mathbb{Z}_n$ eine Botschaft b zu

$$f(b) = b^s \bmod n$$

codiert. Indem man das Potenzieren als wiederholtes Multiplizieren begreift, ist dies auch praktisch leicht durchführbar. Behalten wir die Analogie zum Rechnen im Bereich der reellen Zahlen bei, so ist klar, dass mit

$$g(y) = \sqrt[s]{y} \bmod n$$

entschlüsselt wird.

Wie in $\mathbb{Z}_n$ Wurzeln zu ziehen sind, kann nur durch eine neue Formulierung erkannt werden. So wie wir oben statt der Subtraktion und der Division eine Addition und eine Multiplikation mit dem jeweiligen Inversen durchgeführt haben, führen wir das Wurzelziehen nun auf das Potenzieren zurück. Im Reellen könnten wir schreiben

$$g(f(b)) = (b^s)^{\frac{1}{s}} = b^{(s \cdot \frac{1}{s})} = b^1 = b \ .$$

Die Vermutung liegt nahe, dass nur das multiplikativ Inverse des Schlüssels s zu bestimmen und mit ihm zu potenzieren ist. Doch Vorsicht! Rechnen wir mit Restklassen, so dürfen wir nicht einfach auch im Exponenten die Modulo-Rechnung durchführen. Als einfaches Gegenbeispiel sei $2^{2\cdot 3} \bmod 5$ genannt. Obwohl $2 \cdot 3 = 1 \bmod 5$ ist, gilt $2^{2\cdot 3} = 64 = 4 \bmod 5 \neq 2$.

4.2 Wurzelziehen durch Potenzieren

Es ist also ein anderer Weg zum „Wurzelziehen“ in $\mathbb{Z}_n$ zu suchen. Er wird bereitet durch den Satz von Fermat-Euler, für dessen verschiedene Formulierungen wir auf [RU95, S. 186ff.] verweisen.[1] Es ist jedoch nicht nötig, diese Aussage als Rechenrezept vorzugeben. Statt dessen können die Schülerinnen und Schüler das benötigte Ergebnis an Beispielen selbst erarbeiten. Dabei wird auf ein intuitives Verständnis Wert gelegt, aber nicht auf einen formalen Beweis.

Nach dem oben Gesagten ist klar: Zu einer Botschaft b und einem Schlüssel s ist der „inverse Schlüssel“ t zu bestimmen, so dass gilt $(b^s)^t = b \bmod n$. Für die Zahl n gibt man, den Konventionen des RSA-Verfahrens folgend, das Produkt zweier Primzahlen p und q vor. Um zu erkennen, wie t gefunden wird, ordnet man die Zahlen von 1 bis $(n-1)$ in einem Kreis, in dem die Potenzen einer Zahl $b \in \mathbb{Z}_n \setminus \{0\}$ markiert werden. Dabei verbindet man die Potenzen von b durch Pfeile, die die Multiplikation mit b bedeuten. Folgt man den Pfeilen, so durchläuft man auf diese Weise, ausgehend von $1 = b^0$, die Potenzen von b. Abbildung 2 illustriert dieses Vorgehen. Dort ist $n = 15 = 3 \cdot 5 = p \cdot q$, und die durchgezogenen Pfeile zeigen die Potenzen von 2. Dies sind 1, 2, 4, 8 und wieder 1. Das heisst, dass die Zweierpotenzen einen Zyklus der Länge 4 durchlaufen. Dies kann man beim Entschlüsseln ausnutzen: Ist die Botschaft $b = 2$ und der Schlüssel $s =$

[1] In der hier benötigten Form geben wir den Satz in Abschnitt 5 an.

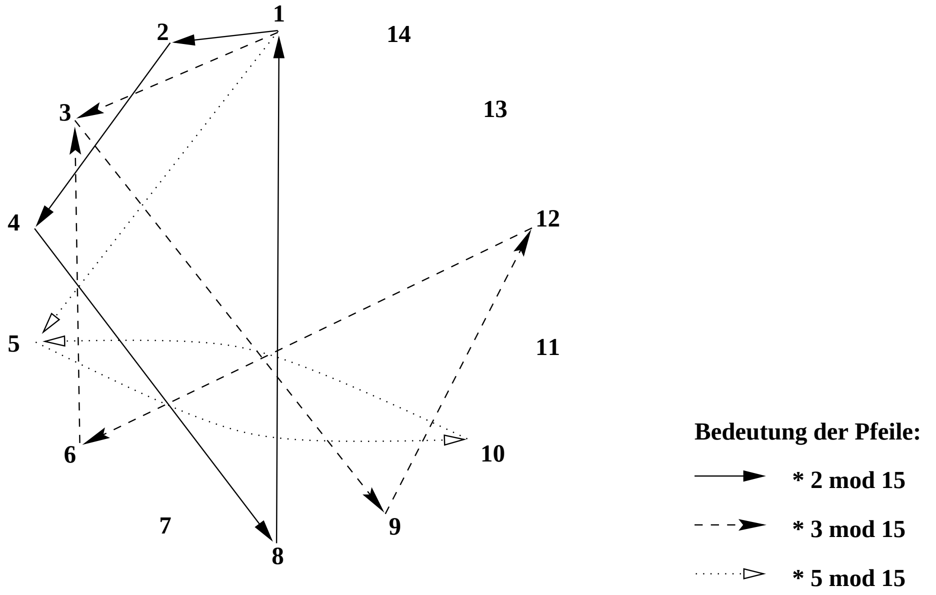

Abbildung 2: Die Potenzen von 2, 3 und 5 in $\mathbb{Z}_{15}$

3, so durchlaufen nämlich auch die Potenzen der codierten Nachricht $b^s = 8$ einen Zyklus. Abbildung 3 zeigt, wie dieser in den Zyklus der Zweierpotenzen eingebettet ist. Man erkennt auch, dass $8^3 = 2$, dass also $t = 3$ ein inverser Schlüssel zu $s = 3$ ist.

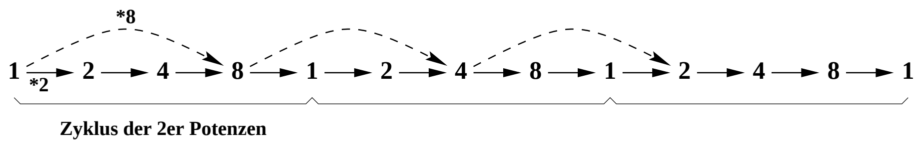

Abbildung 3: Einbettung der Potenzen von 8 in den Zyklus der Potenzen von 2

Die Verallgemeinerung der Situation liefert:

Beobachtung: Durchlaufen die Potenzen einer Zahl b in $\mathbb{Z}_n$ einen Zyklus der Länge l, so sind s und t inverse Schlüssel, falls $s \cdot t = 1 \bmod l$, denn dann gilt

$$(b^s)^t \quad = \quad b^{l+\ldots+l+1} = b^1 \quad . \tag{1}$$

Kennt man l, so kann man — wie beim Verschlüsseln durch Multiplikation — zu vorgegebenem s die Zahl t in der Multiplikationstabelle zur Rechnung modulo l finden. Dabei bemerkt man zugleich, dass s so zu wählen ist, dass $\mathrm{ggT}(s, l) = 1$, denn sonst hat s kein multiplikativ Inverses t in $\mathbb{Z}_l$. Auch hier kann die Parallele zur multiplikativen Verschlüsselung gezogen werden.

Um ein allgemeines Verfahren formulieren zu können, müssen wir noch einsehen, dass die Potenzen einer Zahl in $\mathbb{Z}_n$ stets einen Zyklus durchlaufen, und wir müssen eine Aussage über dessen Länge l gewinnen.

Die erste Aussage ist leicht einzusehen. Da $\mathbb{Z}_n$ nur n Elemente hat, muß in der unendlichen Folge $b^1, b^2, b^3, \ldots$ der Potenzen von b das Folgenglied b^{n+1} (oder ein davor gelegenes) einen Wert annehmen, der bereits zuvor als Folgenglied vorkommt. Von dem ersten Vorkommen eines „doppelten" Folgengliedes an setzt sich die Folge zyklisch fort. Ohne Begründung sei hinzugefügt, dass das erste „doppelte" Folgenglied gleich 1 ist, falls $\mathrm{ggT}(b, n) = 1$. Andernfalls ist es gleich b, so dass die Zahl b in jedem Fall Teil des Zyklus ist.

Die Länge der Zyklen wird für verschiedene Zahlen im Allgemeinen unterschiedlich sein. Dies zeigt bereits Abbildung 2. Dennoch ist es möglich, eine gemeinsame Aussage für alle Zahlen in $\mathbb{Z}_n$ zu machen. Mit einer Zykluslänge l ist nämlich auch jedes Vielfache von l die Länge eines Zyklus, wenngleich dieser dann Unterzyklen enthält. Es zeigt sich, dass alle in $\mathbb{Z}_n$ auftretenden Längen unterzyklenfreier Zyklen Teiler der maximalen solchen Länge sind, die im Folgenden mit ℓ bezeichnet wird.

Bei der Berechnung von ℓ kommt zum Tragen, dass n als Produkt zweier Primzahlen p und q gewählt wird. Anhand von Beispielen (vgl. dazu Tabelle 1) kann man die Vermutung gewinnen, dass unter diesen Voraussetzungen

$$\ell = \mathrm{kgV}(p-1, q-1)$$

ist. Zum Beweis dieser Beziehung sei auf Bücher zur Zahlentheorie, etwa [RU95] verwiesen.

Die so bestimmte Zahl ℓ kann anstelle von l in (1) verwendet werden, um Paare zueinander inverser Schlüssel zu bestimmen, die unabhängig von der versandten Botschaft b sind.

$n = p \cdot q$	p	q	Zyklenlänge ℓ
15	3	5	4
21	3	7	6
35	5	7	12
77	7	11	30
85	5	17	16

Tabelle 1: Maximale Zyklenlänge der Potenzen in $\mathbb{Z}_n$

4.3 Formulierung des RSA-Verfahrens

RSA-Kryptoverfahren: Ein Teilnehmer wählt zwei Primzahlen p und q. Hieraus berechnet er das Produkt $n = p \cdot q$ und die Zahl $\ell = \mathrm{kgV}(p-1, q-1)$. Ferner wählt er einen Schlüssel s mit $\mathrm{ggT}(s, \ell) = 1$ und einen inversen Schlüssel t derart, dass $s \cdot t = 1 \bmod \ell$.

Die Zahlen n und t gibt der Teilnehmer öffentlich bekannt (öffentlicher Schlüssel). Alle übrigen Zahlen hält er geheim (privater Schlüssel). Die Funktionen zum Ver- und Entschlüsseln sind dann

$$f(b) = b^s \bmod n \qquad \text{und} \qquad g(y) = y^t \bmod n \;.$$

Als Beispiel seien $p = 7$ und $q = 11$. Damit ist $n = 77$, $\ell = 30$, und man kann $s = 7$ und $t = 13$ wählen. Die Botschaft $b = 8$ wird verschlüsselt zu $f(8) = 8^7 = 57 \bmod 77$ und $y = 57$ wird entschlüsselt zu $g(57) = 57^{13} = 8 \bmod 77$.

4.4 Das RSA-Verfahren am Beispiel

Beim RSA-Verfahren ist die Zahl n, bezüglich derer man die Restklassen bildet, das Produkt der beiden Primzahlen p und q, die man als Grundlage für ein Schlüsselpaar (s, t) festlegt. Es kann daher kein einheitliches n für alle Teilnehmer gewählt werden. Statt dessen wird man ein Intervall angeben, innerhalb dessen jeder Teilnehmer seinen Modul n festlegen muss. Wir betrachten dies anhand eines Beispiels.

Die Zahl n möge dabei stets zwischen $27^2 = 729$ und $27^3 = 19683$ liegen. Dann kann jeder Zweierblock von Buchstaben (mit der Nummerierung aus dem Beispiel der multiplikativen Codierung) als eine Zahl dargestellt werden, die kleiner als n ist. Beispielsweise entspricht „ge" der Zahl $7 \cdot 27 + 5 = 196$. Diese Zahl kann nun — modulo n — mit einem Schlüssel s potenziert werden. Das Ergebnis ist möglicherweise größer als 729 und kann daher nicht mehr mit zwei Buchstaben dargestellt werden. Es ist aber kleiner als 27^3 und kann daher mit drei Buchstaben wiedergegeben werden.

Wir führen dies für $p = 101$ und $q = 103$ durch. Dann ist $n = 10403$, und wir können als Schlüssel $s = 77$ wählen. Dann wird „ge" verschlüsselt zu den drei Buchstaben, die der Zahl $196^{77} \bmod 10403$ entsprechen. Dies ist die Zahl 9662, also erhält man die Buchstabenfolge „mfw". Auf diese Weise wird „geheimtext" zu „mfwbvzebylqzhxy". Es ist bei dieser Art der Verschlüsselung also eine Verlängerung des ursprünglichen Textes in Kauf zu nehmen.

Zum Entschlüsseln stellt man Folgen von drei Buchstaben als Zahl dar. Da man nur spezielle Folgen von Buchstaben entschlüsselt, nämlich solche, die durch eine Verschlüsselung entstanden sind, erhält man stets eine Zahl, die kleiner als n ist. In unserem Beispiel kommen also keine Buchstabenfolgen vor, deren Wert größer oder gleich 10403 ist. Nun potenziert man mit dem inversen Schlüssel t, der in unserem Fall gleich 8213 ist. Zur Entschlüsselung von „mfw" berechnet man also $9662^{8213} \bmod 10403 = 196$. Das Ergebnis wird sogar stets eine Zahl sein, die kleiner als 729 ist und daher wieder die ursprünglichen zwei Buchstaben darstellt. So wird mit dem inversen Schlüssel $t = 8213$ die Folge „bvz" zu „he", und „eby" wird zu „im". Fährt man so fort, so erhält man insgesamt aus „mfwbvzebylqzhxy" wieder „geheimtext".

Ist s (zusammen mit n) der öffentliche Schlüssel von A, so ist das gerade gezeigte Verfahren geeignet, eine geheime Botschaft an A zu senden. Will umgekehrt A digital unterschreiben, so wird er seinen privaten Schlüssel $t = 8213$ verwenden, um damit Zweiergruppen von Buchstaben zu codieren. Damit wird „ge" zu „c–q", „he" zu „–ld" und insgesamt „geheimtext" zu „c–q–ldmllhwi–hr". Mit dem öffentlichen Schlüssel $s = 77$ kann sich nun jeder überzeugen, dass der codierte Text tatsächlich von A stammt. Dazu werden wieder jeweils drei Buchstaben zusammengefasst, die ihnen entsprechende Zahl wird modulo n mit 77 potenziert, und das Ergebnis wird mit zwei Buchstaben ausgedrückt, so dass insgesamt wieder „geheimtext" entsteht.

5 Ausblick

Bei realen Anwendungen des RSA-Verfahrens verwendet man sehr große Primzahlen (z. B. 100-stellige), so dass man einen sehr langen Text auf einmal verschlüsselt (und nicht nur Gruppen von zwei oder drei Buchstaben). Die Codierung von „Zweierpäckchen“, die hier im Beispiel verwendet wurde, läßt sich mit Häufigkeitsanalysen noch sehr gut knacken. Verschlüsselt man dagegen sehr lange Texte in einem Schritt, so ist dies nicht mehr praktikabel. Ein Angriff auf eine RSA-Verschlüsselung, also das Finden des privaten Schlüssels t mit Hilfe der bekannten Zahlen n und s, erfordert das Zerlegen von n in seine Primfaktoren p und q (was als eine schwierige Aufgabe gilt). Kennt man diese, so läßt sich t aus $(p-1)$, $(q-1)$ und s leicht mit Hilfe des euklidischen Algorithmus berechnen.

Abschließend sei darauf hingewiesen, dass die Formulierung des RSA-Verfahrens in der Literatur leicht variiert. Der wesentliche Unterschied zu unserer Beschreibung besteht in der Wahl der Zahl ℓ. Während wir $\mathrm{kgV}(p-1, q-1)$ verwenden, wird dort das Produkt $(p-1)\cdot(q-1)$ benutzt, das stets ein Vielfaches des kgV ist. Dies liegt daran, dass zur Begründung des RSA-Verfahrens der Satz von Euler benutzt wird (vgl. [Beu94, S. 124] oder [RU95, S. 186]), nach dem

$$b^x = b \bmod p\cdot q, \qquad \text{falls } x = 1 \bmod (p-1)\cdot(q-1) \ .$$

Dieses Ergebnis kann von den Schülerinnen und Schülern nicht auf experimentellem Weg gefunden werden, da die oben entwickelte schärfere Aussage gilt. Ich plädiere deshalb dafür, für Zwecke des entdeckenden Lernens das RSA-Verfahren unter Verwendung des kgV zu formulieren und erst dann, wenn die Berechnung des kgV zu aufwendig wird (nämlich bei großen Zahlen), das Produkt $(p-1)\cdot(q-1)$ zu verwenden.

Erweiterungen für die Sekundarstufe II

In allen hier gegebenen Beispielen lassen sich die multiplikativen Inversen mit Hilfe einer Multiplikationstafel bestimmen. Bei großen Zahlen (wie 27^3) ist das Aufstellen einer kompletten Tafel zwar nicht mehr praktikabel, es genügt aber eine Spalte der Tafel, etwa die der Vielfachen von 1000 modulo 27^3, wenn man $\frac{1}{1000} \bmod 27^3$ finden will. Dies läßt sich mit einem kleinen Computerprogramm schnell ermitteln.

Es wurde schon darauf hingewiesen, dass man zum Auffinden multiplikativer Inverser bei großen Zahlen den euklidischen Algorithmus verwendet [Beu94, RU95, Ihr94]. Seine Behandlung und das Hineinschnuppern in die zahlentheoretischen Grundlagen des RSA-Verfahrens stellen gute Erweiterungsmöglichkeiten für die Sekundarstufe II dar.

Literatur

[Bau96] Rüdeger Baumann. Informationssicherheit durch kryptologische Verfahren. *LogIn*, 16(5/6):52–61, 1996.

[BB96] Klaus Becker und Albrecht Beutelspacher. Datenverschlüsselung. *Log-In*, 16(5/6):16–21, 1996.

[Beu94] Albrecht Beutelspacher. *Kryptologie*. Vieweg Verlag, 4. Auflage, 1994.

[Ihr94] Thomas Ihringer. *Diskrete Mathematik*. Teubner Verlag, 1994.

[RSA78] R. L. Rivest, A. Shamir und L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Comm. A.C.M.*, 21:120–126, 1978.

[RU95] Reinhold Remmert und Peter Ullrich. *Elementare Zahlentheorie*. Birkhäuser Verlag, 1995.

[Sch96] Sigrid Schubert. Basismechanismen der Informationssicherheit. *LogIn*, 16(5/6):10–15, 1996.