RSA & Co. in der Schule

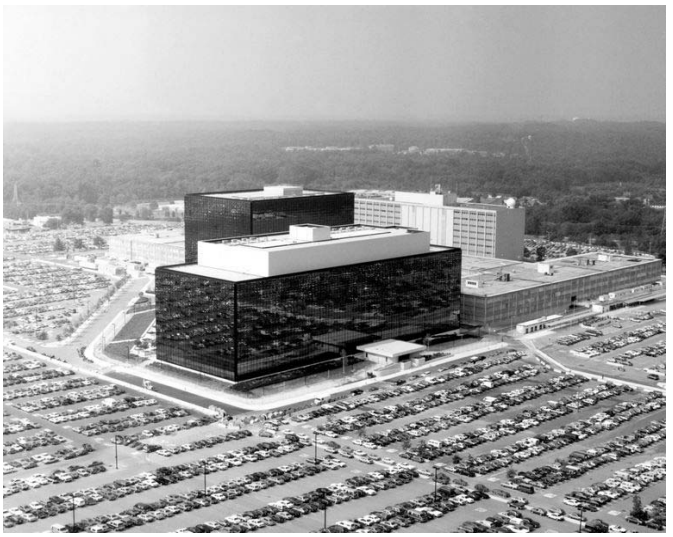
Moderne Kryptologie, alte Mathematik, raffinierte Protokolle

Neue Folge – Teil 3: RSA und die elementare Zahlentheorie

von Helmut Witten und Ralph-Hardo Schulz

Algorithmen der Zahlentheorie sind ein zentraler Bestandteil der modernen Kryptografie, ohne sie könnten keine sicheren Internet-Verbindungen hergestellt werden. Heute schützen täglich weit über eine Million Primzahlen die Welt des elektronischen Handels (du Sautoy, [2]2004, S. 22).

Für den bekannten Mathematiker Godfrey Harold Hardy lag dagegen der Zauber der Zahlentheorie darin, dass sie nicht von Anwendungen beschmutzt sei: „Gauß und viele weniger begabte Mathematiker konnten sich vermutlich zu Recht darüber freuen, dass es auf jeden Fall eine Wissenschaft gibt [die Zahlentheorie], die aufgrund ihrer Entfernung von den gewöhnlichen Tätigkeiten des Menschen für immer ehrlich und sauber bleiben sollte.“ Im Jahr 1940 schrieb er sogar: „Wirkliche Mathematik spielt für den Krieg keine Rolle. Bislang hat niemand einen kriegerischen Nutzen der Zahlentheorie entdeckt“ (zitiert nach http://de.wikiquote.org/wiki/Godfrey_Harold_Hardy).



http://de.wikipedia.org/wiki/Bild:National_Security_Agency_headquarters%2C_Fort_Meade%2C_Maryland.jpg

Bild 1: Das Hauptquartier der NSA ist Fort Meade in Maryland. Die Fenster des Hauptquartiers und Operationszentrums von Crypto City, wie Fort Meade auch genannt wird, bestehen unter der schwarzen Glasfassade aus einer Schutzschirmtechnik mit Kupfer, damit keine elektromagnetischen Signale nach außen dringen.

Wir wissen heute, dass sich kaum ein Mathematiker in der Frage der Anwendungen der Zahlentheorie so sehr getäuscht hat wie Hardy. Die NSA (***N**ational **S**ecurity **A**gency*), der größte und finanziell am besten ausgestattete Geheimdienst der Welt, ist nicht nur die Organisation, die das meiste Geld für Computer-Hardware ausgibt, sondern auch die, die weltweit die meisten Mathematiker (und Zahlentheoretiker) beschäftigt (siehe Bild 1; vgl. Bramford, [3]2001).

Durch die Vernetzung im Zahlungsverkehr entstand in den 60er- und 70er-Jahren des letzten Jahrhunderts ein großer Bedarf an Verschlüsselungstechnologien auch im zivilen Bereich. Im Jahr 1976 veröffentlichten Whitfield Diffie und Martin Hellman ihren bahnbrechenden Artikel „New Directions in Cryptography“ (Diffie/Hellman, 1976). Darin wurde nicht nur erstmals die Erfindung der asymmetrischen Kryptografie für die Öffentlichkeit beschrieben; dieses Papier ist auch ein Meilenstein der von Geheimdiensten unabhängigen kryptologischen Forschung (Singh, 2000b, S. 306 ff.).

Der Schlüsseltausch nach Diffie-Hellman, der im selben Papier präsentiert wurde, ist aber noch kein asymmetrisches Kryptosystem im engeren Sinn, da es keinen öffentlichen Schlüssel gibt (das Diffie-Hellman-Verfahren zählt zur Klasse der Schlüsselaustauschprotokolle). Nach dieser Veröffentlichung begann ein Wettlauf um das erste praxistaugliche asymmetrische Kryptosystem, aus dem bekanntlich das RSA-Kryptosystem als Sieger hervorging (Rivest/Shamir/Adleman, 1978, s. auch Witten/Schulz, 2006b). Erst im Jahr 1985 entwickelte der Amerikaner Taher ElGamal auf der Basis des Diffie-Hellman-Verfahrens das nach ihm benannte asymmetrische Kryptosystem. Die Zeiten, in denen Primzahlen lediglich Spielsteine in einem nutzlosen Spiel der Mathematiker waren, sind mit diesen Erfindungen endgültig vorbei.

Ein genetischer Weg zum RSA-Kryptosystem

Wie in den anderen Beiträgen der letzten Folgen von „RSA & Co. in der Schule“ geht es um die mathematischen Grundlagen, die zu einem tieferen Verständnis von RSA benötigt werden. Auch in dieser Folge soll der Artikel nicht wie z. B. bei Bartholomé u. a. (^{2}1996) als Kurs zur elementaren Zahlentheorie mit RSA als krönendem Abschluss aufgebaut werden, jeder Schritt wird vielmehr mit einer kryptologischen Fragestellung verknüpft. Wir haben uns dabei besonders von Hermann Puhlmanns Artikel „Kryptografie verstehen – Ein schülergerechter Zugang zum RSA-Verfahren“ anregen lassen (Puhlmann, 1998).

Im Teil „RSA für Einsteiger“ haben wir einen „experimentellen“ Zugang zu RSA beschrieben, der Schülerinnen und Schüler die Probleme beim Einsatz von RSA „am eigenen Leibe“ spüren lässt. Bei diesem Zugang mussten wir den RSA-Algorithmus als gegeben voraussetzen (Witten/Schulz, 2006a). Unter dem Gesichtspunkt der Zahlentheorie wurden dort folgende Themen behandelt (siehe auch Mindmap zum Thema „RSA“ in Bild 2):

- ▷ Eine Einführung in das modulare Rechnen,
- ▷ das Sieb des Eratosthenes zur Bestimmung von Primzahlen bzw. zur Faktorisierung von Zahlen, die nicht prim sind, und schließlich
- ▷ ein Verfahren zum schnellen modularen Potenzieren („square and multiply“).

Bild 2: Mindmap zum Thema RSA.

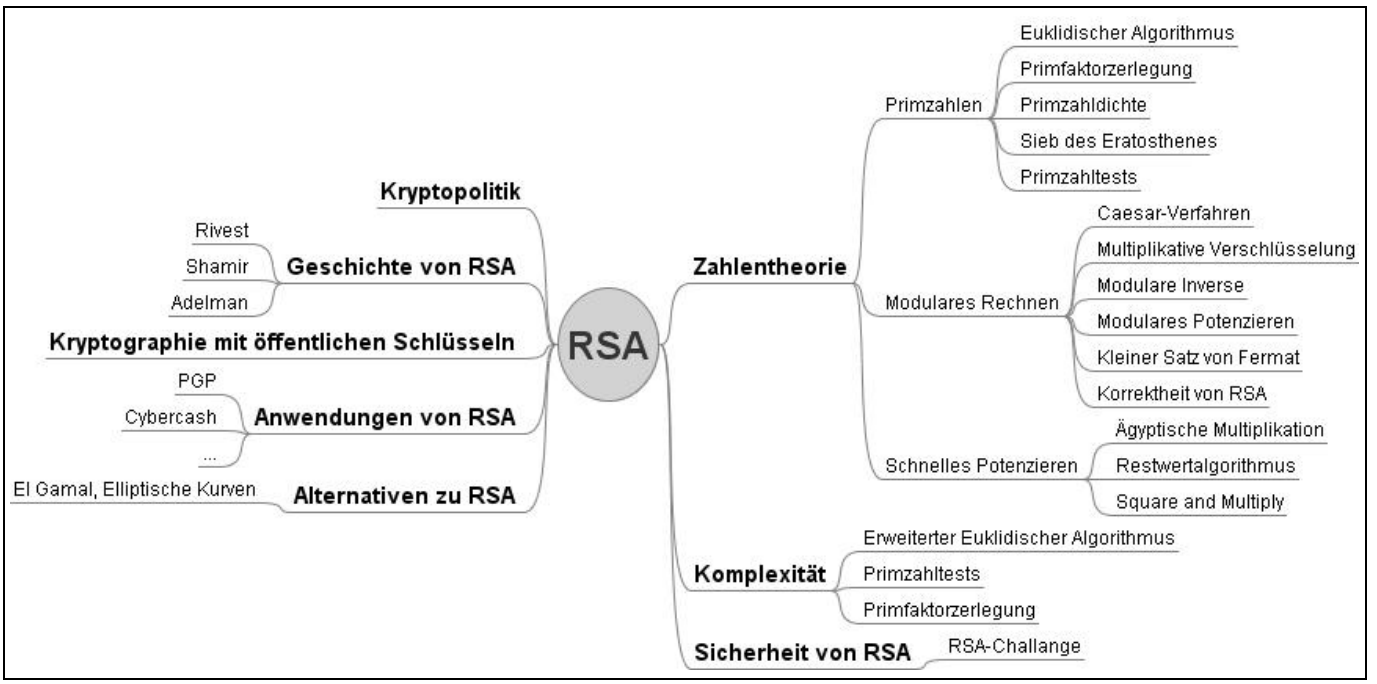
In der sich anschließenden Folge haben wir mit einem genetischen Weg zu RSA begonnen. Dabei standen der erweiterte euklidische Algorithmus (EEA) zur Berechnung der modularen Inversen und das Rechnen mit sehr großen Zahlen („Monsterzahlen“) im Zentrum (Witten/Schulz, 2006b). Als Beispiel wurde die modulare multiplikative Verschlüsselung näher untersucht. Es zeigte sich aber, dass diese Art der Verschlüsselung nicht für ein asymmetrisches Kryptosystem geeignet ist.

Im Sinn eines genetischen Vorgehens liegt es nahe, in einem nächsten Schritt die Verschlüsselung durch modulares Potenzieren zu untersuchen. Für die Entschlüsselung wird dann das modulare Wurzelziehen benötigt. Dieser Unterrichtsgang ermöglicht es den Lernenden, den sogenannten kleinen Satz von Fermat (siehe Kasten „Der Große und der Kleine Fermat“, nächste Seite) und eine verbesserte Version des RSA-Algorithmus selbstständig zu entdecken.

Verschlüsseln mit Pohlig-Hellman

Martin E. Hellman (http://www-ee.stanford.edu/~hellman/), einer der Erfinder der asymmetrischen Kryptografie, ist inzwischen emeritierter Professor für Elektrotechnik an der berühmten Stanford-Universität in der Nähe des Silicon Valley. Diese Universität hat viele bedeutende Informatiker und Ingenieure hervorgebracht, u. a. Donald Knuth sowie die Gründer von HP, Sun und Google. Das Motto der Universität lautet im Übrigen „Die Luft der Freiheit weht“ (Ulrich von Hutten) und ziert Stanfords Wappen in deutscher Sprache (s. Wikipedia, Stichwort „Stanford-Universität“).

Dieses Motto passt gut zu der von Diffie und Hellman unabhängig von der NSA entwickelten asymmetrischen



Der Große und der Kleine Fermat

Die Geschichte von Pierre de Fermat und seinem letzten Satz (dem sogenannten *Großen Satz von Fermat*) wird sehr anschaulich in dem schönen Buch von Simon Singh „Fermats letzter Satz – Die abenteuerliche Geschichte eines mathematischen Rätsels“ (Singh, 2000a) beschrieben. Dieses Buch erschien 1997 in Großbritannien und wurde das erste Buch über Mathematik, das es in die Bestsellerliste schaffte. Auch das zweite Buch von Simon Singh zur Geschichte der Kryptografie (Singh, 2000b) ist sehr empfehlenswert; nach unserer Ansicht gehört es zu den bislang besten populärwissenschaftlichen Büchern zu diesem Thema.

http://commons.wikimedia.org/wiki/Pierre_de_Fermat

Pierre de Fermat (1607/08–1665).

Pierre de Fermat wurde zum Jahreswechsel 1607/08 als Sohn des wohlhabenden Lederhändlers Dominique Fermat in der kleinen südfranzösischen Stadt Beaumont-de-Lomagne in der Nähe von Toulouse geboren – das häufig angenommene Geburtsdatum 17.08. 1601 hat sich inzwischen als falsch erwiesen. Er studierte auf Drängen seiner Familie Rechtswissenschaften und war bis zu seinem Tod am 12. Januar 1665 in verschiedenen hohen Ämtern tätig, zuletzt am obersten Strafgericht.

Die Mathematik betrieb er aus Leidenschaft und zur Entspannung. Man kann sich vorstellen, dass er bei langweiligen Gerichtsverhandlungen über mathematische Probleme nachdachte. Er hatte meist die Ausgabe von Diophants *Arithmetica* bei sich, die Claude Gaspar Bachet de Méziriac 1621 übersetzt und veröffentlicht hatte (Bachet ist uns bereits in der letzten Folge dieser Serie begegnet, s. Witten/Schulz, 2006b, S. 54).

In Diophants Buch war das zahlentheoretische Wissen der Antike zusammengefasst (s. Bild 3, nächste Seite). Häufig ging es dabei um Gleichungen, deren ganzzahlige Lösungen gesucht wurden. Solche Gleichung werden in der Mathematik heute *diophantische Gleichungen* genannt. Fermat dachte über die in dem Buch von Diophant zusammengetragenen zahlentheoretischen Probleme nach und suchte sie zu verallgemeinern. Glücklicherweise hatte die Ausgabe einen breiten Rand, auf dem Fermat seine Überlegungen und Gedanken festhielt.

Fermat hatte nie die Absicht, ein mathematisches Werk zu publizieren. Er hat zwar mit den wenigen Mathematikern seiner Zeit korrespondiert und ihnen einige seiner Ergebnisse übermittelt – allerdings meist ohne Beweis, sondern eher als Denksportaufgabe. Viele seiner Vermutungen wären nie öffentlich bekannt geworden, wenn nicht Fermats ältester Sohn Clément-Samuel die Bedeutung der Arbeiten seines Vaters erkannt hätte. Er verbrachte fünf Jahre damit, die Aufzeichnungen, Briefe und Randbemerkungen zu Diophants *Arithmetica* zu entziffern und veröffentlichte sie 1670 (Singh, 2000a, S. 88 ff.).

Da bei fast allen Sätzen die Beweise entweder nur angedeutet waren oder ganz fehlten, bildeten Fermats Vermutungen eine Herausforderung für die nachfolgenden Mathematikergenerationen.

Am schwierigsten war die Suche nach einem Beweis bei dem großen Fermat’schen Satz. Die Aussage dieses Satzes ist so einfach, dass sie jeder verstehen kann, der schon einmal vom Satz des Pythagoras gehört hat. Ganzzahlige Lösungen der Gleichung $a^2 + b^2 = c^2$ wie z. B. 3, 4 und 5 nennt man *pythagoreische Zahlentripel*. Man weiß seit dem Altertum, dass es davon unendlich viele gibt. Fermat behauptete in einer seiner Randbemerkungen, dass es für die Gleichung $a^n + b^n = c^n$ mit $n > 2$ überhaupt keine ganzzahligen Lösungen gibt. Darunter schrieb er: „Ich habe hierfür einen wahrhaft wunderbaren Beweis gefunden, doch ist der Rand hier zu schmal, um ihn zu fassen“.

Es hat über 300 Jahre gedauert, bis dieser Satz in den 90er-Jahren des letzten Jahrhunderts bewiesen werden konnte. Der Beweis von Andrew Wiles ist aber so kompliziert, dass ihn selbst viele der heutigen Zahlentheoretiker schwer oder gar nicht verstehen. Fermat konnte ihn unmöglich gefunden haben. Man nimmt an, dass Fermat den Satz nur für den Spezialfall $n = 4$ bewiesen hat und irrtümlich glaubte, ihn auf die anderen natürlichen Zahlen erweitern zu können – aber vielleicht ist Fermats wunderbarer Beweis bislang einfach noch nicht gefunden worden (vgl. Singh, 2000a; du Sautoy, ^{2}2004).

Während man die Aussage des großen Satzes von Fermat leicht nachvollziehen kann, benötigt man ein elementares Verständnis des modularen Potenzierens, um den kleinen Satz von Fermat zu verstehen. Der Satz lautet in moderner Schreibweise $a^p \equiv a \pmod p$ oder $a^{p-1} \equiv 1 \pmod p$, falls zusätzlich $\mathrm{ggT}(a, p) = 1$ gilt. Hierbei ist a eine natürliche Zahl und p eine Primzahl.

Zur Erinnerung: $a \equiv b \pmod n$, gesprochen „a kongruent b modulo n“ bedeutet, dass a und b bei der Division durch n den gleichen Rest haben. Der von C. F. Gauß in die Zahlentheorie eingeführte Begriff der Kongruenz verallgemeinert also die Gleichheit. Der kleine Satz von Fermat besagt somit: Wenn ich a p-mal mit sich selbst multipliziere (d. h. mit p potenziere), so ergibt sich jeweils der gleiche Rest bei Division von a^p und a durch p, sofern p eine Primzahl ist.

Beim Beweis des kleinen Satzes von Fermat sind die Verhältnisse einfacher als beim großen Satz. Im Jahr 1640 schrieb Fermat einen Brief an seinen Freund Bernard Frénicle de Bessy, in dem er seinen Satz erläuterte und erklärte, einen Beweis dafür gefunden zu haben. Trotz des Versprechens, Frénicle den Beweis bei nächster Gelegenheit zu schicken, behielt Fermat ihn für sich. Aber in diesem Fall dauerte es weniger als 100 Jahre, bis er im Jahr 1736 von Leonard Euler wiederentdeckt wurde (du Sautoy, ^{2}2004, S. 286 f.). Euler konnte ihn dabei sogar noch verallgemeinern, sodass er häufig als *Satz von Euler-Fermat* bezeichnet wird. Wie wir sehen werden, ist ein Beweis zumindest des kleinen Satzes von Fermat auch den Schülerinnen und Schülern der Sekundarstufe II zugänglich.

Die Bezeichnungen „groß“ bzw. „klein“ beziehen sich also auf die Schwierigkeit des jeweiligen Beweises. Allerdings sind bislang keine relevanten Anwendungen des „großen“ Satzes bekannt, es handelt sich somit um klassische Zahlentheorie im Sinne Hardys. Der „kleine“ Fermat ist dagegen zentral für die moderne Kryptologie und somit äußerst wichtig für die angewandte Zahlentheorie.


DIOPHANTI
ALEXANDRINI
ARITHMETICORVM
LIBRI SEX.
ET DE NVMERIS MVLTANGVLIS
LIBER VNVS.
Nunc primum Græcè & Latinè editi, atque absolutissimis Commentariis illustrati.
AVCTORE CLAVDIO GASPARE BACHETO
MEZIRIACO SEBVSIANO, V. C.
LVTETIAE PARISIORVM,
Sumptibus SEBASTIANI CRAMOISY, via Iacobæa, sub Ciconiis.
M. DC. XXI.
CVM PRIVILEGIO REGIS

http://upload.wikimedia.org/wikipedia/commons/6/60/Diophantus-cover.jpg

Bild 3: Titelbild der Ausgabe von Diophant, die von Fermat verwendet wurde.

Kryptografie. Auf seiner Homepage beschreibt Hellman – der sich seit Anfang der 80er-Jahre des letzten Jahrhunderts auch für nukleare Abrüstung einsetzt – sein Engagement für die Kryptologie, aber auch den Druck, der von interessierten staatlichen Stellen auf die Pioniere der zivilen Kryptografie ausgeübt wurde. Dieser Druck reichte bis zur Androhung einer Anklage wegen Verstoßes gegen die „International Traffic in Arms Regulations (ITAR)", also wegen illegalen Waffenexports.

Bei den beiden Papieren im Visier der staatlichen Stellen handelt es sich um Pohlig/Hellman 1978 (s. Schneier, 1997, S. 306 f. und S. 541) und Merkle/Hellman 1978 (s. Baumann, 2000, und Schneier, 1997, S. 526 ff.). Hellman schreibt dazu (http://www-ee.stanford.edu/~hellman/crypto.html):

> On the advice of Stanford's general counsel, I even presented two papers at a 1977 symposium at Cornell University, instead of my usual practice of having the student co-authors do the presentations. The attorney told me that if the ITAR were interpreted broadly enough to include our papers, he believed they were unconstitutional. But a court case could drag on for years, severely hindering a new Ph.D.'s career (especially if the attorney's belief was not shared by the jury), whereas I was already a tenured professor.
>
> I presented these thoughts to Ralph Merkle and Steve Pohlig, the students in question, but left the final decision to them. Initially they wanted to take the risk and give the papers, but eventually concern from their parents won out. Fortunately, the presentations went off without incident, though it was dramatic having Ralph and Steve stand mute by the podium, so they would get the recognition they deserved, as I gave the papers.

In der letzten Folge waren wir nach der Darstellung des Scheiterns der multiplikativen Verschlüsselung als asymmetrisches Kryptosystem zu dem Punkt gekommen, als nächstes eine Verschlüsselung durch Potenzieren zu untersuchen. Von der Verschlüsselung mit der modularen Multiplikation wissen die Schülerinnen und Schüler, dass die Verhältnisse besonders einfach werden, wenn der Modul eine Primzahl ist. Also bietet es sich an, das im Papier von Pohlig/Hellman 1978 vorgestellte einfache Verschlüsselungsverfahren durch modulares Potenzieren zu untersuchen (vgl. auch Bartholomé u. a., ^{2}1996, S. 132 f.):

$$(1) \qquad c \equiv m^e \pmod{p}$$

wobei p eine Primzahl, m die Botschaft (message) und c die Chiffre ist mit $m, c \in \{1, \ldots, p-1\}$. Der Exponent e ist der Schlüssel und ebenfalls eine natürliche Zahl $< p$. Zusätzlich muss $\mathrm{ggT}(e, p-1) = 1$ gelten, damit der inverse Schlüssel berechnet werden kann (s. u.).

Dieses Schema ähnelt dem RSA-System, allerdings ist der Modul in diesem Fall eine Primzahl. Es wird sich bei näherer Betrachtung herausstellen, dass Pohlig-Hellman zwar ein funktionierendes Kryptosystem ist, sich aber nicht zur asymmetrischen Verschlüsselung eignet, da es keinen öffentlichen Schlüssel gibt. Aus didaktischen Gründen wird das den Lernenden an dieser Stelle noch nicht verraten. Sie können aber anhand dieses Algorithmus viel über das modulare Wurzelziehen und den kleinen Satz von Fermat lernen.

Wir sind also gewappnet, die folgende Frage zu untersuchen: Wie kann man die durch (1) verschlüsselte Botschaft wieder entschlüsseln? Wie bei den reellen Zahlen gibt es zwei Umkehrungen des Potenzierens.

Falls der Exponent fest bleibt, muss man die e-te Wurzel ziehen, um wieder an die Botschaft m zu kommen. Das Verfahren von Pohlig-Hellman führt also genauso wie das RSA-System auf die Notwendigkeit des *modularen Wurzelziehens*.

Falls dagegen die Basis fest bleibt und der Exponent variabel ist, muss der *diskrete Logarithmus* bestimmt werden. Dies trifft auf den Schlüsseltausch nach Diffie-Hellman, die asymmetrische Verschlüsselung nach ElGamal (s. z. B. Schulz, ^{2}2003, S. 208 f.) und auf die Kryptografie mit elliptischen Kurven (s. Schulz, ^{2}2003, S. 214 ff.) zu – Verfahren, die wir in einer weiteren Folge dieser Artikelserie besprechen werden. Nebenbei bemerkt widmet sich auch das erwähnte Papier von Pohlig/Hellman dem Problem des diskreten Logarithmus unter der Fragestellung, wie man bei bekanntem m und c den Schlüssel e bestimmen kann. Damit handelt es sich um einen kryptanalytischen Angriff bei bekanntem Klartext (*known plaintext*).

http://www-ee.stanford.edu/~hellman/

Bild 4: Martin E. Hellman (geb. 1945).

Entschlüsseln mit Fermat

Beim Verfahren nach Pohlig-Hellman (siehe (1), S. 63) wird also durch modulares Potenzieren verschlüsselt; zum Entschlüsseln müsste, wie wir bereits angemerkt haben, die e-te Wurzel gezogen werden. Wenn man mithilfe der modularen Addition (Caesar) bzw. der modularen Multiplikation verschlüsselt, kann man durch Addition bzw. Multiplikation des jeweils inversen Elements entschlüsseln (vgl. Witten/Letzner/Schulz, 1998, und Witten/Schulz, 2006b). Warum sollte es beim modularen Potenzieren also nicht möglich sein, das Wurzelziehen durch ein erneutes Potenzieren zu erreichen?

Wir müssten dafür eine Zahl d finden, sodass $c^d = m^{ed} \equiv m \pmod p$ gilt. Falls m teilerfremd zu p ist, ist diese Forderung zu $m^{ed-1} \equiv 1 \pmod p$ äquivalent. Uns interessieren daher Zahlen k mit $m^k \equiv 1 \pmod p$ bzw. $m^{k+1} \equiv m \pmod p$.

Um diese Zahlen zu finden, bietet sich ein experimentelles Vorgehen an; wir erstellen dazu einige Potenztabellen für Primzahl-Moduln. Wir fassen dabei das Potenzieren wie in der Mittelstufe üblich als fortgesetzte Multiplikation auf, wobei das Produkt nach jedem Schritt modulo p reduziert wird. Wir benötigen dabei eine einfache Regel des modularen Rechnens: Modulo-Bildung und Multiplikation können vertauscht werden (s. Witten/Schulz 2006a).

In der unten angegebenen Tabelle könnte z. B. die dritte Zeile wie folgt berechnet werden: $3 \to 3 \cdot 3 \equiv \mathbf{2} \pmod 7 \to 3 \cdot 2 \equiv \mathbf{6} \pmod 7 \to 3 \cdot 6 \equiv \mathbf{4} \pmod 7 \to 3 \cdot 4 \equiv \mathbf{5} \pmod 7 \to 3 \cdot 5 \equiv \mathbf{1} \pmod 7$ usw.

Auf der anderen Seite würde man die Werte auch direkt durch Potenzieren erhalten: $3^1 \equiv \mathbf{3} \pmod 7$, $3^2 \equiv \mathbf{2} \pmod 7$, $3^3 \equiv \mathbf{6} \pmod 7$, $3^4 \equiv \mathbf{4} \pmod 7$, $3^5 \equiv \mathbf{5} \pmod 7$, $3^6 \equiv \mathbf{1} \pmod 7$ usw.

Für kleine Werte lassen sich die Werte dieser Tabelle per Hand berechnen. Zur Bequemlichkeit und für größere Werte von p kann man ein kleines Programm schreiben, das die Werte in der Tabelle mit einer geschachtelten Schleife berechnet.

Für $p = 7$ erhalten wir:

Exponent->	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	2	4	1	2	4	1	2
3	3	2	6	4	5	1	3
4	4	2	1	4	2	1	4
5	5	4	6	2	3	1	5
6	6	1	6	1	6	1	6

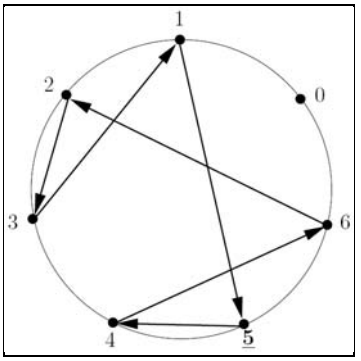


Bild 5: Zyklus von 5 für den Modul 7. (Ändert man die Orientierung aller Pfeile, so erhält man den Zyklus von 3.)

Da beim Potenzieren modulo 7 nur die sechs Zahlen von 1 bis 6 zur Verfügung stehen (die Null wird zur Vereinfachung zunächst ausgenommen) ergibt sich immer ein sogenannter Zyklus; die auftretenden Zahlen müssen sich beim fortgesetzten Multiplizieren ab einer gewissen Stelle wiederholen. Dieser Zyklus lässt sich bei kleinen Zahlen auch zeichnerisch darstellen (siehe Bild 5; vgl. auch Puhlmann, 1998).

Als weiteres Beispiel betrachten wir $p = 11$:

Exp->	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	5	10	9	7	3	6	1	2
3	3	9	5	4	1	3	9	5	4	1	3
4	4	5	9	3	1	4	5	9	3	1	4
5	5	3	4	9	1	5	3	4	9	1	5
6	6	3	7	9	10	5	8	4	2	1	6
7	7	5	2	3	10	4	6	9	8	1	7
8	8	9	6	4	10	3	2	5	7	1	8
9	9	4	3	5	1	9	4	3	5	1	9
10	10	1	10	1	10	1	10	1	10	1	10

Schon bei diesen beiden einfachen Beispielen fällt auf, dass in der Spalte 6 (für $p = 7$) bzw. in der Spalte 10 (für $p = 11$) nur die 1 steht. Wenn man das oben erwähnte kleine Programm erstellen lässt oder zur Verfügung stellt, kann man diese Beobachtung ohne großen Aufwand auch für weitere Primzahlen bestätigen.

Insgesamt erhält man damit als Vermutung den kleinen Satz von Fermat (siehe Kasten „Der Große und der Kleine Fermat“, vorhergehende Seite): $a^{p-1} \equiv 1 \pmod p$ oder $a^p \equiv a \pmod p$.

Im ersten Fall muss $\mathrm{ggT}(a, p) = 1$ gelten, also p kein Teiler von a sein.

Den Grundgedanken des Beweises kann man sich klar machen, wenn man sich noch einmal die Multiplikationstabellen modulo p ansieht: Dort kommen in jeder Zeile alle Zahlen von 1 bis p–1 genau einmal vor, nur die Reihenfolge ist permutiert. Insofern liefern alle Zeilenprodukte den gleichen Wert (vgl. Witten/Schulz, 2006b, und siehe Kasten „Beweis des kleinen Satzes von Fermat“, nächste Seite).

Mit dem kleinen Satz von Fermat kann jeder verschlüsselte Text nach Pohlig-Hellman auch wieder entschlüsselt werden. Als Beispiel wählen wir $p = 11$ und $e = 7$ (damit ist auch die Bedingung erfüllt, dass e und p–1 = 10 teilerfremd sind). Die Verschlüsselung der verschiedenen Möglichkeiten für m kann in der 7. Spalte der Potenztabelle abgelesen werden. So wird z. B. aus der Botschaft $m = 3$ die Chiffre $c = 9$. Gesucht ist jetzt die Zahl d, mit der jede dieser verschlüsselten Botschaften entschlüsselt werden kann.

Man überlegt sich, dass e und d inverse Schlüssel sind, wenn

$$(2) \qquad e \cdot d \equiv 1 \pmod \lambda$$

gilt, wobei λ die Länge des jeweiligen Zyklus ist, in dem m liegt (d. h., e und d sind modular invers bzgl. λ). In diesem Fall gilt nämlich $(m^e)^d = m^{\lambda + \lambda \ldots + \lambda + 1} \equiv m^1 \pmod p$, da $m^\lambda \equiv 1 \pmod p$ gilt und ein Zyklus beliebig oft durchlaufen werden kann, ohne das Ergebnis zu ändern. Für das Beispiel aus dem vorigen Absatz mit $m = 3$ liest man $\lambda = 5$ aus der Tabelle ab. Damit erhält man

Beweis des kleinen Satzes von Fermat

Für eine Primzahl p und alle zu p teilerfremden ganzen Zahlen a gilt

$$a^{p-1} \equiv 1 \pmod{p}.$$

Beweis (durch Betrachtung der Restklassen $\overline{a}$, d. h. der Mengen derjenigen ganzen Zahlen, die jeweils den gleichen Rest a bei Division durch p haben, für $a \in \{1, 2, \ldots, p-1\}$): Die möglichen Reste ungleich 0 bei Division einer Zahl durch p sind 1, 2, 3, …, p–1. Die Zahlen $1 \cdot a$, $2 \cdot a$, $3 \cdot a$, …, $(p-1) \cdot a$ sind ebenfalls p–1 verschiedene Zahlen; je zwei von ihnen haben verschiedene Reste (ungleich 0) modulo p, da wegen $\mathrm{ggT}(a, p) = 1$ gilt:

$$k \cdot a \equiv l \cdot a \pmod{p} \Rightarrow k \equiv l \pmod{p},$$

was wegen der Größe dieser Zahlen nur für $k = l$ möglich ist. Also ist jede der Zahlen $i \cdot a$ kongruent zu genau einem Element aus $\{1, 2, 3, \ldots, p-1\}$, und es gilt

$$\{\overline{1 \cdot a}, \overline{2 \cdot a}, \overline{3 \cdot a}, \ldots, \overline{(p-1) \cdot a}\} = \{1, 2, 3, \ldots, \overline{(p-1)}\}$$

und damit

$$\overline{1} \cdot \overline{2} \cdot \overline{3} \cdot \ldots \cdot \overline{(p-1)} = \overline{1 \cdot a} \cdot \overline{2 \cdot a} \cdot \overline{3 \cdot a} \cdot \ldots \cdot \overline{(p-1) \cdot a},$$

also

$$1 \cdot 2 \cdot 3 \cdot \ldots \cdot (p-1) \equiv [1 \cdot 2 \cdot 3 \cdot \ldots \cdot (p-1)] \cdot a^{p-1} \pmod{p};$$

hieraus folgt durch Division $a^{p-1} \equiv 1 \pmod{p}$.

Anmerkung 1: Das heißt für die Multiplikationstafel von $\mathbb{Z}_p^*$, dass das Produkt der Elemente der ersten Zeile und das der Elemente der a-ten Zeile übereinstimmen; also sind die Zeilenprodukte konstant. Bei $\mathbb{Z}_5^*$ z. B. erhält man

$\odot_5$	1 2 3 4	Zeilenprodukt
1	1 2 3 4	$1 \odot 2 \odot 3 \odot 4 = 4$
2	2 4 1 3	$2 \odot 4 \odot 1 \odot 3 = 4$
3	3 1 4 2	$3 \odot 1 \odot 4 \odot 2 = 4$
4	4 3 2 1	$4 \odot 3 \odot 2 \odot 1 = 4$

Anmerkung 2: Wie wir an den Beispielen sehen, ist umgekehrt p–1 nicht immer das kleinste x mit $a^x \equiv 1 \pmod{p}$. Zum Beispiel genügt wegen $3 \equiv 7^4 \pmod{11}$ ein kleinerer Exponent x als 10, um zu $3^x \equiv 1$ zu gelangen, nämlich 5:

$$3^5 \equiv 7^{20} \equiv (7^{10})^2 \equiv 1 \pmod{11}.$$

Andererseits kann man zeigen, dass bei gegebenem p immer ein a existiert, für das p–1 der kleinste Exponent t ungleich 0 mit $a^t \equiv 1 \pmod{p}$ ist. (Die multiplikative Gruppe $\mathbb{Z}_p^*$ von $\mathbb{Z}/p\mathbb{Z} = \mathbb{Z}_p$ ist zyklisch, also von einem Element, einer sogenannte Primitivwurzel des Körpers $\mathbb{Z}_p$, erzeugt. Daher gibt es Elemente der Ordnung $p-1 = |\mathbb{Z}_p^*|$.) Für den Nachweis der Korrektheit der Entschlüsselung beim RSA-Verfahren benötigen wir diese Aussage aber nicht, sondern „nur“ den kleinen Satz von Fermat.

Anmerkung 3: Weitere Beweise des kleinen Satzes von Fermat findet man beispielsweise im Beweisarchiv der Wikipedia (siehe Literatur und Internetquellen).

z. B. $d = 3$, da $3 \cdot 7 = 21 \equiv 1 \pmod{5}$. In der Tabelle kann man überprüfen, dass sich tatsächlich $c^d = 9^3 \equiv 3 \pmod{11}$ ergibt.

Für $m = 6$ gilt dagegen $\lambda = p-1 = 10$. Auch hier bietet sich 3 als Schlüssel zum Entziffern an, da ebenfalls $3 \cdot 7 = 21 \equiv 1 \pmod{10}$ gilt. Dies ergibt sich aber daraus, dass die Zahl der möglichen Schlüssel bei diesem Beispiel wegen des kleinen Moduls 11 sehr begrenzt ist. Deshalb soll noch ein Beispiel mit etwas größeren Zahlen folgen.

Wir wählen $p = 101$. Als Schlüsselzahlen kommen dann nur solche in Frage, die teilerfremd zu p–1 = 100 sind, also z. B. $e = 23$. Jetzt und im Folgenden rechnen wir mit Langzahlarithmetik, z. B. mit einem beliebigen CAS-System wie DERIVE oder mit dem PYTHON-Interpreter (http://www.python.org/; Arnhold, 2001). Damit ergibt sich für die Botschaft $m = 43$ die Chiffre $c = 13$.

Wenn man nun durch Probieren oder ein kleines Programm die Länge des Zyklus für m bestimmt, erhält man 50, da für diese Zahl erstmals $43^{50} \equiv 1 \pmod{101}$ gilt (die Zykluslängen müssen immer ein Teiler von p–1 sein, siehe Kasten „Beweis des kleinen Satzes von Fermat“). Den Exponenten d zum Entschlüsseln erhält man dann als modulare Inverse zu e, also als Lösung der Kongruenz (2). Bei Moduln dieser Größenordnung kann man die modulare Inverse zwar noch durch Probieren bestimmen; bequemer ist es aber, ein Programm mit dem erweiterten euklidischen Algorithmus zu verwenden (Witten/Schulz, 2006b).

So oder so erhält man $d = 37$ und bestätigt damit $13^{37} \equiv 43 \pmod{101}$. Nun wäre es aber unpraktisch, für jede Botschaft die Zykluslänge bestimmen zu müssen. Man nimmt einfach immer die maximale Zykluslänge p–1 – der Zyklus wird damit ggf. nur häufiger durchlaufen. Natürlich ergibt sich bei veränderter Zykluslänge in der Regel auch eine andere Zahl für d, in unserem Beispiel $d = 87$. Man bestätigt aber auch hiermit $13^{87} \equiv 43 \pmod{101}$. Wir halten fest, dass bei einem gegebenen Schlüssel e der inverse Schlüssel d von der jeweiligen Zykluslänge λ abhängt und nicht eindeutig bestimmt ist.

Leider ist das untersuchte Verfahren nicht für ein asymmetrisches Kryptosystem zu gebrauchen, da jeder zu einem gegebenen „öffentlichen“ Schlüssel, bestehend aus e und p, einen passenden „geheimen“ Schlüssel d nach dem kleinen Satz von Fermat mit dem eben beschriebenen Verfahren selbst berechnen kann. Das Papier von Pohlig-Hellman beschreibt also kein asymmetrisches Kryptosystem – das hatten die Autoren allerdings auch nicht behauptet.

Wir verbessern das RSA-Kryptosystem

Wenn eine Primzahl als Modul – wie bei Pohlig-Hellman – kein asymmetrisches Kryptosystem liefert, bietet es sich an, als Modul ein Produkt aus zwei verschiedenen Primzahlen zu verwenden. In der Tat wird beim RSA-Verfahren $n = p \cdot q$ $(p \neq q)$ gewählt. Ver- und

Exp->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	1	2	4	8	1	2	4	8	1	2	4	8
3	3	9	12	6	3	9	12	6	3	9	12	6	3	9	12
4	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4
5	5	10	5	10	5	10	5	10	5	10	5	10	5	10	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	4	13	1	7	4	13	1	7	4	13	1	7	4	13
8	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2
9	9	6	9	6	9	6	9	6	9	6	9	6	9	6	9
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
11	11	1	11	1	11	1	11	1	11	1	11	1	11	1	11
12	12	9	3	6	12	9	3	6	12	9	3	6	12	9	3
13	13	4	7	1	13	4	7	1	13	4	7	1	13	4	7
14	14	1	14	1	14	1	14	1	14	1	14	1	14	1	14

Exp->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	1	2	4	8	1	2	4	8	1	2	4	8
4	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4
7	7	4	13	1	7	4	13	1	7	4	13	1	7	4	13
8	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2
11	11	1	11	1	11	1	11	1	11	1	11	1	11	1	11
13	13	4	7	1	13	4	7	1	13	4	7	1	13	4	7
14	14	1	14	1	14	1	14	1	14	1	14	1	14	1	14

Tabellen 1 und 2.

Entschlüsseln funktionieren damit genau so wie bei Pohlig-Hellman; wesentliche Unterschiede gibt es bei der Konstruktion der Schlüssel e und d.

Auch in diesem Fall experimentieren wir zunächst mit Potenztabellen, um damit die Geheimnisse des Wurzelziehens modulo n zu ergründen. Wir wählen $n = 15 = 3 \cdot 5$ (siehe Tabelle 1).

Im Vergleich zu den bisher betrachteten Primzahlmoduln sieht diese Tabelle viel unübersichtlicher aus. Weiter oben haben wir überlegt (2), dass für die inversen Schlüssel e und d $e \cdot d \equiv 1 \pmod{\lambda}$ gelten muss, wobei λ die Länge des jeweiligen Zyklus ist. Diese Überlegung bleibt auch gültig, wenn der Modul nicht länger eine Primzahl ist. Interessant sind für uns daher nur diejenigen Zeilen, in denen eine 1 vorkommt.

Dies sind aber genau die Zeilen, deren Nummer teilerfremd zum Modul 15 sind, nämlich 1, 2, 4, 7, 8, 11, 13 und 14. Dies sind auch die Zahlen, die bezüglich dieses Moduls invertierbar sind (siehe Hilfssatz im Kasten „Die Sätze von Euler und Carmichael“, nächste Seite).

Wir betrachten die oben angegebene Potenztabelle nochmals, dabei sollen aber die Zeilen, die keine 1 enthalten, gestrichen werden (siehe Tabelle 2).

Die Tabelle ist damit erheblich übersichtlicher geworden, dies gilt besonders für die Zykluslängen.

Offenbar gilt $a^4 \equiv 1 \pmod{15}$, falls $ggT(a, 15) = 1$ ist. Für diejenigen, die das RSA-Verfahren bereits kennen, ist dieses Ergebnis überraschend, da sie eher den Wert der Euler'schen Funktion $\varphi(15) = (3–1)(5–1) = 8$ erwarten würden (vgl. Witten/Schulz, 2006a, und siehe Kasten „Die Sätze von Euler und von Carmichael“, nächste Seite). Tatsächlich gilt für alle a mit $ggT(a, 15) = 1$ ebenfalls $a^8 \equiv 1 \pmod{15}$, nur ist der Wert 8 nicht optimal. Wie wir im vorigen Abschnitt gesehen haben, führt dies möglicherweise zu größeren geheimen Schlüsseln d.

An dieser Stelle soll die *Carmichael-Funktion* $\lambda(n)$ eingeführt werden: $\lambda(n) = k$ ist der kleinste Exponent, für den $a^k \equiv 1 \pmod{n}$ für alle zu n teilerfremden Zahlen gilt. $\lambda(n)$ gibt damit die optimale Zykluslänge an und ist immer ein Teiler von $\varphi(n)$; für unser Beispiel mit $n = 15$ gilt – wie wir gesehen haben – $\lambda(15) = 4$, aber $\varphi(15) = 8$.

Für das Produkt von verschiedenen Primzahlen p, q kann man sich einfach überlegen, dass die *Euler'sche Funktion* φ den Wert $\varphi(p \cdot q) = (p–1)(q–1)$ hat (vgl. z.B. Reiß/Schmieder, [2]2007, S.287): Die Euler'sche Funktion $\varphi(n)$ gibt die Zahl der zu n teilerfremden Zahlen kleiner gleich n (ohne 0) an (siehe Kasten „Die Sätze von Euler und Carmichael“, nächste Seite). Die Zahlen, die ***nicht*** teilerfremd zu $p \cdot q$ sind, kann man wie folgt bestimmen: $p, 2p, \ldots, q \cdot p$ und entsprechend $q, 2q, \ldots, p \cdot q$. Da $p \neq q$ gilt, kommt in diesen Aufzählungen nur $p \cdot q$ doppelt vor, somit haben wir $q+p–1$ verschiedene ***nicht*** teilerfremde Zahlen zu $p \cdot q$. Den gesuchten Wert für die Euler-Funktion erhält man dann durch $\varphi(p \cdot q) = p \cdot q – (q+p–1) = (p–1)(q–1)$.

Uns interessiert nun aber der Wert von $\lambda(p \cdot q)$, da wir erwarten können, damit kleinere Schlüssel d und somit eine optimierte Variante des RSA-Systems zu erhalten. Es wird sich zeigen, dass $\lambda(p \cdot q) = kgV(p–1, q–1)$ gilt (Spezialfall des Satzes von Carmichael, siehe Kasten nächste Seite). Damit man den Schülerinnen und Schülern diese Tatsache nicht einfach mitteilen muss, bietet sich in Anlehnung an Puhlmann (1998) wiederum ein experimentelles Vorgehen an, um eine entsprechende Vermutung zu gewinnen. Dazu soll ein Programm geschrieben werden, das die Werte für die Funktionen von Euler und Carmichael berechnen kann. Je nach Leistungsstand

n	p	q	φ(n)	λ(n)
15	3	5	8	4
21	3	7	12	6
33	3	11	20	10
35	5	7	24	12
51	3	17	32	16
55	5	11	40	20
57	3	19	36	18
77	7	11	60	30
69	3	23	44	22
85	5	17	64	16
95	5	19	72	36
115	5	23	88	44
119	7	17	96	48
133	7	19	108	18
161	7	23	132	66
187	11	17	160	80

Tabelle 3: Wertetabelle für die Euler-Funktion $\varphi(n)$ und die Carmichael-Funktion $\lambda(n)$.

Die Sätze von Euler und Carmichael

Hilfssatz

Wir beginnen mit dem folgenden Hilfssatz:

Folgende Aussagen für natürliche Zahlen n, m mit $1 < m < n$ sind äquivalent:

(i) Es existiert ein t mit $m^t \equiv 1 \pmod{n}$.
(ii) m besitzt eine Inverse (mod n), ist also Einheit.
(iii) $\mathrm{ggT}(m, n) = 1$.

Beweisskizze

(i) $\Rightarrow$ (ii): Bei den Werten m mit $m^t \equiv 1$ erhält man durch $m^{-1} = m^{t-1}$ eine Inverse; diese Elemente sind daher definitionsgemäß Einheiten.
(ii) $\Rightarrow$ (iii): Wäre $1 \neq d = \mathrm{ggT}(m, n)$, so $m \cdot \frac{n}{d} = \frac{m}{d} n \equiv 0 \pmod{n}$ und damit m Nullteiler und keine Einheit (sonst wäre $0 = m^{-1} \cdot 0 \equiv m^{-1} m \frac{n}{d} = \frac{n}{d}$).
(iii) $\Rightarrow$ (ii): Nach dem Lemma von Bachet (vgl. Witten/Schulz, 2006b, S. 54) existieren ganze Zahlen r, s mit $r \cdot m + s \cdot n = \mathrm{ggT}(m, n) = 1$. Modulo n gilt daher: Es existiert ein r_1 mit $r_1 \cdot m \equiv 1 \pmod{n}$; es ist daher m eine Einheit.
(ii) $\Rightarrow$ (i): Die Menge $\{m^s \pmod{n} | s \in \mathbb{N}\}$ ist als Teilmenge von $\mathbb{Z}_n$ endlich. Daher gibt es Zahlen u und $v > u$ mit $m^u \equiv m^v$, woraus wegen der Existenz der Inversen von m die Kongruenz $m^{v-u} \equiv 1 \pmod{n}$ folgt.

Wir betrachten daher zunächst nicht mehr beliebige ganze Zahlen m, sondern nur solche ganze Zahlen $m = a$, die teilerfremd zu n sind. Wieder fragen wir uns, welche t die Aussage $a^t \equiv 1 \pmod{n}$ erfüllen. Eine Antwort gibt der folgende Satz von Euler, auch *Satz von Euler-Fermat* genannt.

Satz von Euler

Für jede ganze Zahl a, die zu $n \in \mathbb{N}$ teilerfremd ist, gilt

$$a^{\varphi(n)} \equiv 1 \pmod{n}.$$

Hierbei bezeichnet φ die *Euler'sche Phi-Funktion*: $\varphi(n)$ ist definiert als die Anzahl der zu n teilerfremden Zahlen aus der Menge $\{1, \ldots, n-1\}$; z. B. ist $\varphi(p) = p-1$ für jede Primzahl p (sodass sich dann der kleine Satz von Fermat ergibt) und

$$\varphi(p \cdot q) = (p-1)(q-1)$$

für voneinander verschiedene Primzahlen p und q. (Da wir den Satz von Euler nicht für das RSA-Verfahren benötigen, verweisen wir bezüglich des Beweises auf die Literatur, z. B. Bartholomé u. a., ^{2}1996, und Reiss/Schneider, ^{2}2007.)

Carmichael-Funktion

Definition der Carmichael-Funktion: Zu jeder natürlichen Zahl n sei $\lambda(n)$ die kleinste natürliche Zahl s ($s \neq 0$), sodass

$$a^s \equiv 1 \pmod{n}$$

für jedes zu n teilerfremde a gilt.

Die Abbildung $\mathbb{N} \to \mathbb{N}$ mit $n \to \lambda(n)$ heißt *Carmichael-Funktion* nach dem amerikanischen Mathematiker Robert Daniel Carmichael (1879–1967).

Anmerkung: In gruppentheoretischer Sprache ist $\lambda(n)$ der Exponent der Restklassengruppe $(\mathbb{Z}/n\mathbb{Z})^*$, das kleinste gemeinsame Vielfache aller Ordnungen von Elementen dieser Gruppe. Man kann $\lambda(p) = p-1$ für Primzahlen p zeigen.

Wir wollen nun $\lambda(p \cdot q)$ für Primzahlen p, q mit $p \neq q$ bestimmen. Seien also $n = p \cdot q$ und a teilerfremd zu n.

Aus $a^{\lambda(n)} \equiv 1 \pmod{n}$ folgt, dass $p \cdot q$ und damit p Teiler von $a^{\lambda(n)}-1$ ist; also gilt auch $a^{\lambda(n)} \equiv 1 \pmod{p}$. Man wählt nun $a = a_1$ als ein Element, für das ${a_1}^s$ für alle $s \in \{1, \ldots, p-2\}$ inkongruent 1 modulo p ist (s. o.), also $p-1$ der kleinste nicht triviale Exponent s mit ${a_1}^s \equiv 1$; ist $\lambda(n) = u(p-1)+r$ mit $r < p-1$ (Division mit Rest), so erhält man aus dem Satz von Fermat und aus

$${a_1}^{p-1} \equiv 1 \equiv {a_1}^{\lambda(pq)} \equiv ({a_1}^{p-1})^u \cdot {a_1}^r \equiv {a_1}^r \pmod{p},$$

dass $r = 0$ ist. Es folgt:

$p-1$ teilt $\lambda(p \cdot q)$.

Analog ergibt sich: $q-1$ teilt $\lambda(p \cdot q)$. Insgesamt zeigt dies

(☆) $\lambda(p \cdot q) \geq \mathrm{kgV}(p-1, q-1)$.

Andererseits folgt aus $s = \mathrm{kgV}(p-1, q-1)$ zunächst $s = v(p-1) = w(q-1)$ für geeignete Zahlen v und w und daraus mit dem Satz von Fermat

$$a^{\mathrm{kgV}(p-1, q-1)} = a^s = a^{(p-1)v} \equiv 1 \pmod{p}$$

für alle a mit $\mathrm{ggT}(a, p) = 1$.

p und analog q teilen also a^s-1 für alle zu p und q teilerfremde Zahlen. Es ist damit $a^s \equiv 1 \pmod{p \cdot q}$. Zusammen mit (☆) folgt

$$\lambda(p \cdot q) = \mathrm{kgV}(p-1, q-1).$$

Dies beinhaltet den folgenden ***Spezialfall des Satzes von Carmichael***: Sind p und q verschiedene Primzahlen, dann gilt

$$a^{\mathrm{kgV}(p-1, q-1)} \equiv 1 \pmod{p \cdot q}$$

für alle zu $p \cdot q$ teilerfremden Zahlen a; für feste Primzahlen p und q (und variables a) ist $\lambda = \mathrm{kgV}(p-1, q-1)$ der kleinste Exponent ungleich 0 mit $a^\lambda \equiv 1 \pmod{p \cdot q}$ für alle zu $p \cdot q$ teilerfremden a.

Anmerkung: Der Satz von Carmichael besagt allgemeiner:

$$\lambda({p_1}^{a_1}, \ldots, {p_k}^{a_k}) = \mathrm{kgV}(\lambda({p_1}^{a_1}), \ldots, \lambda({p_k}^{a_k}))$$

und

$$\lambda({p_i}^{a_i}) = \begin{cases} 2^{a_i-2} & \text{für } p_i = 2 \text{ und } a_i > 2 \\ {p_i}^{(a_i-1)(p-1)} & \text{sonst} \end{cases}.$$

PYTHON-Programm zur Erstellung einer Wertetabelle für die Euler- und die Carmichael-Funktion

```
# Mit dem Programm können die Werte der Euler-
# und der Carmichael-Funktion aus der
# elementaren Zahlentheorie bestimmt werden.

def ggT (a,b):
    '''
    Rekursive Berechnung des ggT
    mit dem euklidischen Algorithmus.'
    '''
    if b == 0: return a
    return ggT(b, a%b)

def invertierbar(n):
    '''
    Liefert eine Liste mit alle Zahlen 0<x<n,
    die relativ prim zu n sind.'
    '''
    return [x for x in range(n)
                    if ggT(x,n) == 1]

def phi(n):
    '''
    Liefert den Wert
    der Euler-Funktion phi(n)
    '''
    return len(invertierbar(n))

def ordnung(a, n):
    '''
    Liefert die kleinste Zahl o mit 0 < o < n,
    für die a^o % n = 1 gilt.
    Es muss ggT(a,n) = 1 gelten,
    ansonsten wird -1 zurückgegeben.
    '''
    if ggT(a,n) == 1:
        return [a**o % n for o in
                    range(1,n)].index(1) + 1
    # index(1) liefert die erste Stelle der
    # Liste, an der die 1 auftritt
    else: return -1

def car(n):
    '''
    Liefert den Wert
    der Carmichael-Funktion lambda(n)
    '''
    if n < 3: return 1
    else: return max([ordnung(a,n)
                      for a in invertierbar(n)])
```

und Interesse der Lerngruppe kann man dieses Programm vorgeben oder selbst entwickeln lassen (siehe Kasten „PYTHON-Programm zur Erstellung einer Wertetabelle für die Euler- und die Carmichael-Funktion“). Die im Kasten angegebene PYTHON-Implementierung ist angelehnt an den funktionalen Programmierstil und wurde mit einem Leistungskurs Informatik entwickelt. Im LOG-IN-Service findet sich zusätzlich ein PYTHON-Programm nach dem klassischen imperativen Paradigma, das sich leicht in andere Programmiersprachen übertragen lässt. Darüber hinaus gibt es im Service zwei Implementierungen in den Programmiersprachen HASKELL und JAVA, die uns freundlicherweise Walter Gussmann zur Verfügung gestellt hat.

Korrektheit der Entschlüsselung beim RSA-Verfahren

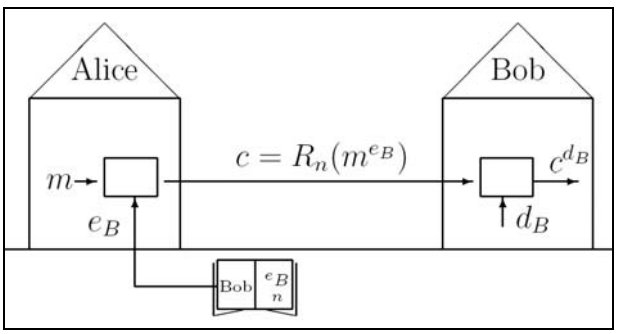
Beim RSA-System mit öffentlichen Schlüsseln n, e und geheimem Schlüssel d sind $n = p \cdot q$ für zwei verschiedene Primzahlen p, q und

$$d \cdot e \equiv 1 \pmod{(p-1)(q-1)}$$

gewählt.
Um Rechenleistung einzusparen, kann man stattdessen auch d und e so wählen, dass gilt:

$$d \cdot e \equiv 1 \pmod{\mathrm{kgV}(p-1, q-1)}.$$

Wir untersuchen hier die Korrektheit der Entschlüsselung im letzteren Fall (für den anderen Fall siehe Schulz, [2]2003, S. 208): Es gelte also für die Schlüssel $e = e_B$ und $d = d_B$ des Teilnehmers Bob: $e \cdot d \equiv 1 \pmod{\lambda}$ mit $\lambda = \mathrm{kgV}(p-1, q-1)$. Da λ dann $e \cdot d-1$ teilt, existiert ein t mit $e \cdot d = \lambda t+1$. Als Vielfaches von $(p-1)$ erfüllt λ ferner die Gleichung $\lambda = (p-1)s$ für eine geeignete natürliche Zahl s.
Für Zahlen m kleiner n (als Nachrichten) wird die Chiffrierung gemäß der folgenden Abbildung vorgenommen. Wir zeigen, dass c^{d_B} den Klartext ergibt.



Schema der Chiffrierung und Dechiffrierung beim RSA-System.

(*Zur Erinnerung:* $R_n(b)$ ist der Rest zwischen 0 und $n-1$ von b bei Division durch n.)
Aus $c = R_n(m^e)$ folgt $c \equiv m^e \pmod{n}$ und damit $c^d \equiv \mathrm{m^{ed}} \pmod{n}$. Ist m zu p teilerfremd, so gilt

$$m^{ed} = m^{\lambda t+1} = (m^{(p-1)s})^t \cdot m$$

und daher nach dem Satz von Fermat $m^{ed} \equiv 1^{st} \cdot m \equiv m \pmod{p}$.
Ist m nicht zu p teilerfremd, so folgt $m^{ed} \equiv 0 \equiv m \pmod{p}$.
Also erhält man in beiden Fällen $m^{ed} \equiv m \pmod{p}$.
Analog ergibt sich $m^{ed} \equiv m \pmod{q}$. Insgesamt folgt, dass sowohl p als auch q und damit auch ihr Produkt $p \cdot q$ Teiler von $m^{ed}-m$ sind. Also gilt $m^{ed}-m \equiv 0 \pmod{pq}$ und folglich $R_n(c^d) = m$.

Für diejenigen, die gar nicht programmieren wollen, findet sich auf der Seite http://www.math-it.org/Mathematik/Zahlentheorie/Carmichael_de.html ein Applet zur Berechnung der Werte der Euler- und der Carmichael-Funktion.

Nach diesen Vorbereitungen kann man z. B. die Tabelle 3 erstellen (siehe Seite 66; vgl. Puhlmann, 1998).

Die Tabelle liefert unmittelbar die (weiter oben bereits bewiesene) Vermutung $\varphi(p \cdot q) = (p-1)(q-1)$. Komplizierter ist die Auswertung für die Carmichael-Funktion. Bei oberflächlicher Betrachtung ergibt sich für $\lambda(p \cdot q)$ häufig die Hälfte von $\varphi(p \cdot q)$, die „Ausreißer" bei 85 und 133 sollten aber eine Herausforderung sein, den Dingen genauer auf den Grund zu gehen. Diese Herausforderung kann noch dadurch gesteigert werden, dass die Euler-Funktion im Unterricht gar nicht thematisiert wird (für das verbesserte RSA-Verfahren wird sie ja auch nicht benötigt). Mit dem Hinweis, die Primfaktorzerlegung von $p-1$ und $q-1$ zu untersuchen, sollte es möglich sein, die erwartete Vermutung zu finden.

Im Kasten „Korrektheit der Entschlüsselung beim RSA-Verfahren" (siehe vorige Seite) findet sich der Nachweis, dass auch das verbesserte RSA-Kryptosystem korrekt entschlüsselt.

Die optimierte Version des RSA-Verfahrens wird in der Literatur mitunter anstelle des „normalen" Verfahrens verwendet (siehe z. B. Bauer, 1995, S. 160; Wiesenbauer, 1999). Hermann Puhlmann (Puhlmann, 1998, S. 10) weist mit Recht darauf hin, dass die Lernenden bei dem geschilderten Unterrichtsgang – modulares Wurzelziehen durch Bestimmung der Zykluslänge – notwendigerweise auf die Carmichael-Funktion und nicht auf die Euler-Funktion stoßen.

Warum wird dann aber in der Literatur ganz überwiegend das RSA-Verfahren in der „herkömmlichen" Version mit $\varphi(p \cdot q) = (p-1)(q-1)$ verwendet? Hermann Puhlmann argumentiert, dass bei großen Zahlen die Berechnung vom $\mathrm{kgV}(p-1, q-1)$ zu rechenaufwendig wird. Dieses Argument kann aber wegen der Beziehung $\mathrm{ggT}(a, b) \cdot \mathrm{kgV}(a, b) = a \cdot b$ (siehe z. B. Bartholomé u. a., ^{2}1996, S. 51 oder Reiss/Schmieder, ^{2}2007, S. 141) nicht überzeugen, da der ggT mit dem euklidischen Algorithmus sehr effizient berechnet werden kann (vgl. Witten/Schulz 2006b). Anschließend muss nur noch das Produkt von a und b durch den ggT geteilt werden, auch hierfür ist der Rechenaufwand nicht allzu hoch, wie ein kleines Experiment zeigt.

Um realistische Zahlen zu verwenden, wählen wir wie schon in der letzten Folge die berühmte Zahl RSA129, eine Dezimalzahl mit 129 Stellen (vgl. Witten/Schulz, 2006b; Wiesenbauer, 1999). Die Primfaktorzerlegung dieser Zahl in p und q ergab Primzahlen mit 64 bzw. 65 Stellen, die hier nicht nochmals abgedruckt werden sollen. Es gilt $\mathrm{ggT}(p-1, q-1) = 4$; die Berechnung dieser Zahl mit dem PYTHON-Interpreter erfolgt auf einem normalen PC in praktisch unmessbarer Zeit, ebenso wie die Multiplikation $p-1$ mit $q-1$ und die anschließende Division durch 4.

Die Frage, warum die Praktiker – einschließlich Bruce Schneier (Schneier, 1997, S. 531 ff.) – in der Regel die Version mit der Euler-Funktion bevorzugen, ist damit also nicht beantwortet. Hier einige Vermutungen:

- In der Originalarbeit (Rivest/Shamir/Adleman, 1978) ist nur von dem Satz von Euler die Rede.
- Der Satz von Euler ist viel bekannter als der von Carmichael.
- Bei der Carmichael-Funktion muss zusätzlich das kgV bestimmt werden. Der numerische Unterschied zwischen dem Produkt und dem kgV von $p-1$ und $q-1$ und damit zwischen den Werten der Euler- und der Carmichael-Funktion ist allerdings nicht allzu groß, häufig nur 2.
- Das hat zur Folge, dass sich auch der Gewinn an Rechenzeit in engen Grenzen hält.

Nachtrag und Ausblick

In der letzten Folge (Witten/Schulz, 2006b) berichteten wir über den Wettbewerb „RSA Factoring Challenge", den die Firma RSA Security 1991 ausgeschrieben hatte. Dieser Wettbewerb wurde leider inzwischen beendet. Als Grund wird angegeben, dass sich das Verständnis der Sicherheit des RSA-Kryptosystems, die vor allem durch die Faktorisierung des Moduls n gefährdet ist, deutlich verbessert habe (s. http://www.rsa.com/rsalabs/node.asp?id=2094).

Das Argument der Firma RSA klingt nicht sehr überzeugend, wahrscheinlich handelt es sich eher um eine Sparmaßnahme, denn von den ursprünglich ausgelobten Preisgeldern in Höhe von 635 100 $ wurden lediglich 30 100 $ ausgezahlt (s. Wikipedia, Stichwort „RSA Factoring Challenge"). Möglicherweise hat diese Einsparung auch mit dem Verkauf der Firma RSA Security an den Speicherriesen EMC zu tun.

Ein weiteres Motiv für den Abbruch des Faktorisierungswettbewerbs könnte die Rekordzerlegung in Primfaktoren sein, die den „üblichen Verdächtigen" von der Universität Bonn für eine Zahl mit 1017 Binärstellen gelungen ist (s. *heise online* vom 21.05.2007).

In der nächsten Folge werden wir uns näher mit der Praxis des RSA-Verfahrens beschäftigen:

- Wie kann man mit vernünftigem Rechenaufwand große Primzahlen finden?
- Gibt es überhaupt genügend Primzahlen in der gesuchten Größenordnung?
- Wie sicher ist das RSA-Kryptosystem jetzt und in Zukunft?
- Was sind die wichtigsten Anwendungen von RSA?
- Welche Auswirkungen hatte der Patentschutz für das RSA-Verfahren?

Prof. Dr. Ralph-Hardo Schulz
Freie Universität Berlin
Fachbereich Mathematik und Informatik
Institut für Mathematik
Arnimallee 3
14195 Berlin

E-Mail: schulz@math.fu-berlin.de

StD Helmut Witten
Fachseminar für Informatik
1. Schulpraktisches Seminar
Charlottenburg-Wilmersdorf
Walther-Rathenau-Schule (Gymnasium)
Herbertstraße 4
14193 Berlin

E-Mail: helmut@witten-berlin.de

Im ***LOG-IN-Service*** (siehe S. 88) stehen die im Beitrag genannten Programme zur Erstellung einer Wertetabelle für die Euler- und Carmichael-Funktionen zum Herunterladen zur Verfügung. Darüber hinaus kann dort ein Verzeichnis der interessantesten Quellen aus der freien Enzyklopädie *Wikipedia* heruntergeladen werden.

Literatur und Internetquellen

Arnhold, W.: Lieben Sie PYTHON? In: LOG IN, 21. Jg. (2001), H. 2, S. 18–24.

Bartholomé, A.; Rung, J.; Kern, H.: Zahlentheorie für Einsteiger. Braunschweig; Wiesbaden: Vieweg, ²1996.

Bauer, F. L.: Entzifferte Geheimnisse – Methoden und Maximen der Kryptologie. Berlin; Heidelberg: Springer, 1995.

Baumann, R.: Das Rucksackproblem – Informatische und kryptologische Aspekte. In: LOG IN, 20. Jg. (2000), H. 2, S. 47–52.

Bramford, J.: NSA – die Anatomie des mächtigsten Geheimdienstes der Welt. München: Bertelsmann, ³2001.

Diffie, W.; Hellman, M. E.: New Directions in Cryptography. In: IEEE Trans. on Info. Theory, Vol. IT-22, November 1976, S. 644–654.
http://www-ee.stanford.edu/%7Ehellman/publications/24.pdf.
[zuletzt geprüft: 27. Juni 2008]

heise online: Zahlenknacker melden Rekordzerlegung in Primfaktoren. 21.05.2007.
http://www.heise.de/newsticker/Zahlenknacker-melden-Rekordzerlegung-in-Primfaktoren--/meldung/89996/
[zuletzt geprüft: 27. Juni 2008]

Merkle, R. C.; Hellman, M. E.: Hiding Information and Signatures in Trap Door Knapsacks. In: IEEE Trans. on Info. Theory, Vol. IT-24, September 1978, S. 525–536.
http://www-ee.stanford.edu/%7Ehellman/publications/30.pdf
[zuletzt geprüft: 27. Juni 2008]

Pohlig, S. C.; Hellman, M. E.: An Improved Algorithm for Computing Logarithms Over GF(p) and its Cryptographic Significance. In: IEEE Trans. on Info. Theory, Vol. IT-24, January 1978, S. 106–110.
http://www-ee.stanford.edu/%7Ehellman/publications/28.pdf
[zuletzt geprüft: 27. Juni 2008]

Puhlmann, H.: Kryptographie verstehen – Ein schülergerechter Zugang zum RSA-Verfahren. TU Darmstadt (1998).
Als gezippte PostScript-Datei online verfügbar unter:
http://wwwbib.mathematik.tu-darmstadt.de/Math-Net/Preprints/Listen/files/2000.ps.gz
Eine PDF-Version ist zu erhalten bei:
http://www.matheraetsel.de/texte/Kryptographie.pdf
[zuletzt geprüft: 27. Juni 2008]

Reiss, K.; Schmieder, G.: Basiswissen Zahlentheorie – Eine Einführung in Zahlen und Zahlbereiche. Berlin; Heidelberg; New York: Springer, ²2007.

Rivest, R.; Shamir, A.; Adleman, L.: A Method for Obtaining Digital Signatures and Public Key Cryptosystems. In: Comm. ACM, Vol. 21, Nr. 2 (1978), S. 120–126.
http://theory.lcs.mit.edu/~rivest/rsapaper.pdf
[zuletzt geprüft: 27. Juni 2008]

du Sautoy, M.: Die Musik der Primzahlen – Auf den Spuren des größten Rätsels der Mathematik. München: C. H. Beck, ²2004.

Schneier, B.: Angewandte Kryptographie – Protokolle, Algorithmen und Sourcecode in C. Bonn; Reading (MA, USA): Addison-Wesley, 1997.

Schulz, R.-H.: Codierungstheorie – Eine Einführung. Wiesbaden: Vieweg, ²2003.

Singh, S.: Fermats letzter Satz – Die abenteuerliche Geschichte eines mathematischen Rätsels. München: dtv, 2000a.

Singh, S.: Geheime Botschaften – Die Kunst der Verschlüsselung von der Antike bis in die Zeiten des Internet. München; Wien: Hanser 2000b.

Wiesenbauer, J.: Public Key Kryptosysteme in Theorie und Programmierung. Schriftenreihe zur Didaktik der MG, Heft 30, 1999, S. 144–159.

Witten, H.; Letzner, I.; Schulz, R.-H.: RSA & Co. in der Schule – Teil 2: Von Cäsar über Vigenère zu Friedman. In: LOG IN, 18. Jg. (1998), H. 5, S. 31–39.

Witten, H.; Schulz, R.-H.: RSA & Co. in der Schule – Neue Folge Teil 1: RSA für Einsteiger. In: LOG IN, 26. Jg. (2006a), H. 140, S. 45–54.

Witten, H.; Schulz, R.-H.: RSA & Co. in der Schule – Neue Folge Teil 2: RSA für große Zahlen. In: LOG IN, 26. Jg. (2006b), H. 143, S. 50–58.