RSA & Co. in der Schule

Moderne Kryptologie, alte Mathematik, raffinierte Protokolle

Neue Folge Teil 6: Das Faktorisierungsproblem oder: Wie sicher ist RSA?

von Helmut Witten und Ralph-Hardo Schulz

RSA-Schlüssel geknackt?

Am 14. Februar 2012 meldete die *New York Times*, dass die Sicherheit des Online-Handels durch eine neu aufgedeckte Schwachstelle in ernste Gefahr geraten sei (Markow, 2012):

> The importance in ensuring that encryption systems do not have undetected flaws cannot be overstated. The modern world's online commerce system rests entirely on the secrecy afforded by the public key cryptographic infrastructure.

Diese Schwachstelle wurde bei im Internet für jedermann zugänglichen öffentlichen RSA-Schlüsseln gefunden. Eine Gruppe von Wissenschaftlern um den renommierten Zahlentheoretiker Arjen K. Lenstra hatte seit 2009 im Internet mehrere Millionen öffentliche Schlüssel gesammelt und deren Sicherheit überprüft. Sie konnten über 12 000 Schlüssel brechen (vgl. Lenstra u. a., 2012)!

Öffentliche RSA-Schlüssel enthalten einen Modul, der ein Produkt von zwei sehr großen Primzahlen ist – eine sogenannte Semiprimzahl. Wenn es gelingt, diesen Modul in seine Primfaktoren zu zerlegen, hat man das RSA-System für diese Zahlen geknackt.

Wie konnte das passieren? Die meisten der von den Forschern gefundenen Schlüssel haben eine Länge von 1024 Bit. Noch nie ist eine *harte* 1024-Bit-Semiprimzahl (so bezeichnen wir im Folgenden ein schwer zu faktorisierendes Produkt aus zwei Primzahlen) mit den heute bekannten Verfahren in ihre beiden Primfaktoren zerlegt worden, obwohl dies nach den bisherigen Ergebnissen etwa für das Jahr 2020 erwartet wird. (Deshalb wird schon jetzt empfohlen, nur noch 2048 Bit lange Schlüssel zu verwenden.) Aber selbst wenn es gelingen wird, *eine* 1024-Bit-Zahl zu zerlegen, wird dies einen unglaublich hohen Rechenaufwand erfordern. Niemand wird dann gleich Tausende von diesen Zahlen faktorisieren können. Auf die ebenso einfache wie geniale Methode der Forscher um Arjen Lenstra, mit der es ihnen dennoch gelungen ist, Tausende 1024-Bit-Schlüssel schon jetzt zu knacken, kommen wir am Ende dieses Beitrags zurück.

In diesem Artikel wollen wir zunächst die „klassischen" Untersuchungen zur Sicherheit des RSA-Verfahrens wiedergeben, die in erster Linie mit der Faktorisierung des Moduls und der inzwischen beendeten RSA-Challenge (vgl. RSA Laboratories, 2012) verbunden sind. (Inzwischen sind alle noch offenen RSA-Challenges auch im Krypto-Wettbewerb *MysteryTwister C3* zu finden; vgl. Internetquellen am Ende des Beitrags.) Praktische Experimente zur Faktorisierung haben wir in Schulz/Witten (2010) wiedergegeben; in diesem Heft beschreiben wir ausführlich das Quadratische Sieb, das zurzeit schnellste Verfahren für harte Semiprimzahlen bis ca. 100 Stellen im Dezimalsystem (siehe Seite 70ff.). Für noch größere Zahlen wird das Quadratische Sieb vom Zahlkörpersieb (*Number Field Sieve*) übertroffen. Für eine genauere Beschreibung dieses seit zwanzig Jahren schnellsten Verfahrens müssen wir aber auf die Literatur verweisen. Ein guter Einstieg ist dabei *A Tale of Two Sieves* von Carl Pomerance, dem Erfinder des Quadrati-

http://www.win.tue.nl/~klenstra/

Bild 1: Arjen Klaas Lenstra (geb. 2. März 1956 in Groningen).

Der niederländische Mathematiker Arjen K. Lenstra war an der Entwicklung des zurzeit schnellsten Faktorisierungsalgorithmus beteiligt, dem Zahlkörpersieb (*Number Field Sieve*). Ihm sind zahlreiche Faktorisierungen von harten Semiprimzahlen aus der RSA-Challenge gelungen, u. a. RSA-100 (die allererste Faktorisierung überhaupt bei diesem berühmten Wettbewerb), RSA-129 (das erste verteilte Faktorisierungsprojekt im Internet) sowie der aktuelle Faktorisierungsrekord (RSA-768). Er arbeitet als Leiter des Laboratory for Cryptologic Algorithms (LACAL) an der École Polytechnique Fédéral de Lausanne (EPFL). Kürzlich gelang einer Arbeitsgruppe an diesem Institut das Knacken von tausenden RSA-Schlüsseln.

http://www.usc.edu/dept/molecular-science/RSA-2003.htm

Bild 2:
Rivest, Shamir und Adleman (v.l.n.r.) im Jahr 2003.

schen Siebs und Mit-Erfinder des Zahlkörpersiebs (vgl. Pomerance, 1996).

Warum RSA nicht ARS heißt

Im Herbst 1976 trafen sich drei junge Forscher am MIT: Ronald („Ron“) Rivest und Adi Shamir waren Computer-Wissenschaftler mit dem Spezialgebiet Kryptografie, Leonard („Len“) Adleman Mathematiker mit dem Spezialgebiet Zahlentheorie. Rivest war fasziniert von dem Artikel *New Directions in Cryptography* von Whitfield („Whit“) Diffie und Martin Hellman (vgl. Diffie/Hellman, 1976). Zu der Zeit war es noch unklar, ob es ein solches, von Diffie und Hellman vorgeschlagenes asymmetrisches Kryptosystem überhaupt geben konnte. (Übrigens dient der in dem genannten Papier ebenfalls veröffentlichte Diffie-Hellman-Algorithmus dem Schlüsseltausch über einen unsicheren Kanal. Es dauerte danach fast zehn Jahre, bis es Taher Elgamal, einem Schüler von Martin Hellman, 1985 gelang, daraus ein asymmetrisches Kryptosystem zu entwickeln, das sowohl zur Verschlüsslung wie zum Schlüsselaustausch verwendet werden kann. Es hat im Zusammenhang mit elliptischen Kurven noch zusätzliche Bedeutung erlangt. Wir werden in der nächsten Folge dieser Beitragsreihe genauer auf diese Alternativen zu RSA eingehen).

Rivest und Shamir entwickelten 1976/77 viele Ideen für ein asymmetrisches Kryptosystem. Die Aufgabe von Adleman war, mögliche Schwachpunkte zu finden. Len Adleman war lange Zeit erfolgreich, erst als Ron Rivest im Frühjahr 1977 das heute als RSA bekannte Verfahren erfunden hatte, musste Adleman seine Niederlage eingestehen. Rivest schrieb ein Papier über das neu entdeckte Verfahren und listete die Autoren wie üblich alphabetisch auf. Adleman protestierte, er habe nicht genug getan, um als Autor genannt zu werden. Man einigte sich schließlich darauf, dass sein Name als letzter genannt würde.

Die erste Veröffentlichung des Verfahrens erfolgte im Herbst 1977 in einer Kolumne von Martin Gardner in der populärwissenschaftlichen Zeitschrift *Scientific American* (die deutsche Ausgabe heißt *Spektrum der Wissenschaft*) und erregte gleich großes Aufsehen (siehe Abschnitt „Beispiel 2: RSA-129“, nächste Seite). Im Jahr 1978 wurde das wissenschaftliche Papier in den *Communications of the ACM* veröffentlicht (vgl. Rivest/Shamir/Adleman, 1978). 25 Jahre nach der Erfindung erhielten Rivest, Shamir und Adleman 2002 den Turing-Preis der ACM. Seitdem witzelt Adleman gerne: „ARS sounds better and better to me now.“ In dem Beitrag von Sara Robinson anlässlich der Verleihung des Turing-Preises (vgl. Robinson, 2003) wird die Geschichte von RSA ausführlich beschrieben.

Brechen des RSA-Kryptosystems durch Faktorisierung des Moduls

Wie kann das RSA-System gebrochen werden, wenn es gelingt, den Modul n, der Bestandteil des öffentlichen Schlüssels ist, zu faktorisieren? Wir wollen dafür kurz rekapitulieren, welche Algorithmen bei der Schlüsselkonstruktion im RSA-Kryptosystem benötigt werden (vgl. dazu auch Witten/Schulz, 2006a und 2006b). Als Hilfsmittel zum Rechnen benutzen wir das Open-Source Computer-Algebra-System (CAS) SAGE, das online mit dem SAGE Cell Server so einfach wie ein leistungsfähiger Taschenrechner im Browser genutzt werden kann. Wir werden sehen, dass dieser „Taschenrechner“ mit den von uns benötigten zahlentheoretischen Funktionen ausgestattet ist (siehe auch Bild 3, Seite 62).

Das RSA-Verfahren

Wir bezeichnen mit m den Klartext und mit c die chiffrierte Botschaft, beides sind natürliche Zahlen. (Wenn man Texte übermitteln will, benötigt man eine zusätzliche Codierung, die Texte in Zahlen und umgekehrt umwandelt. Ein einfaches Verfahren dafür findet man im zweiten Beispiel.) Der öffentliche Schlüssel besteht aus den Zahlen e und n, der private Schlüssel wird mit d bezeichnet. Die Zahl n ist dabei eine Semiprimzahl, d.h. ein Produkt aus zwei Primzahlen.

Schlüsselerzeugung

Insgesamt muss gelten:
$n = p \cdot q$ (p, q prim)
$phi = (p-1) \cdot (q-1)$ (*phi* ist der Wert der Eulerschen Funktion $\varphi(n)$)
$\mathrm{mod}(d \cdot e, phi) = 1$ (d und e müssen modular invers bezüglich *phi* sein).
Damit können der öffentliche (n, e) und der private Schlüssel (d) erzeugt werden.
Danach müssen *p*, *q*, *d* und *phi* geheim gehalten werden!

Ver- bzw. Entschlüsselung

Zur Verschlüsselung benötigt man e und n, zur Entschlüsselung d und n.
Es gelten die Beziehungen:
$c = \mathrm{mod}(m^e, n)$ bzw. $m = \mathrm{mod}(c^d, n)$

Beispiel 1: Ein kleiner Modul

Wir wählen $n = 55$ und $e = 17$ als öffentlichen Schlüssel, die verschlüsselte Botschaft sei 18. Wie lautet der Klartext?

Lösung:

In diesem Fall ist die Faktorisierung offenbar schon für Grundschülerinnen und Grundschüler ohne Rechnereinsatz möglich: $55 = 5 \cdot 11$.

Man gewinnt aus den Primzahlen 5 und 11 den Modul zur Schlüsselberechnung
$phi = (5 - 1) \cdot (11 - 1) = 4 \cdot 10 = 40$.

Nun suchen wir einen privaten Schlüssel d, für den $\mathrm{mod}(17 \cdot d, 40) = 1$ gelten muss, d.h. d und 17 müssen *modular invers* bezüglich des Moduls 40 sein.

Bei so kleinen Zahlen kann man den gesuchten Schlüssel d durch systematisches Probieren finden, z.B. arbeitsteilig per Hand oder durch ein kleines SAGE-Programm (wenn man SAGE nicht auf seinem Rechner installieren will, kann man mit http://www.sagemath.org/eval.html den SAGE Cell Server direkt online aufrufen. Den voreingestellten Beispiel-Code sollte man löschen und danach seine Eingaben machen. Die eigentliche Berechnung wird dann durch die Schaltfläche „Evaluate“ angestoßen):

```
d = 1
while mod(17*d,40) <> 1:
    d = d+1
print d
```

Die Ausgabe dieses Programms ist 33 – und wirklich ergibt $17 \cdot 33 = 561$ bei Division durch 40 den Rest 1.

Somit ist $d = 33$ der gesuchte private Schlüssel, mit dem wir die „Geheimbotschaft“ 18 entschlüsseln können. Dies geht wiederum sehr einfach mit SAGE:

`mod(18^33,55)` liefert den gesuchten „Klartext“ 13.

Zur Kontrolle verschlüsseln wir noch einmal 13 und erhalten mit

`mod(13^17,55)` die vorgegebene „verschlüsselte Botschaft“ 18 zurück.

Man sieht, dass das RSA-Kryptosystem mit diesem kleinen Modul keinerlei Sicherheit bietet.

Beispiel 2: RSA-129

Im August 1977 erschien in der Zeitschrift *Scientific American* eine Kolumne von Martin Gardner mit dem Titel *A New Kind of Cipher That Would Take Millions of Years to Break* (vgl. Gardner, 1977). In diesem Artikel wurde ein Rätsel präsentiert. Die Autoren dieses Rätsels waren Ron Rivest, Adi Shamir und Len Adleman, die sich mit Fragen der gerade neu entdeckten asymmetrischen Kryptografie beschäftigten; die Veröffentlichung des RSA-Verfahrens stand noch bevor (vgl. Rivest/Shamir/Adleman, 1978; vgl. auch Witten/Schulz, 2006b).

Alice veröffentlicht ihren Modul n und ihren öffentlichen Exponenten e, dabei ist $e = 9007$ und
n = 114381625757888867669235779976146612010218296721242362562561842935706935245733897830597123563958705058989075147599290026879543541.

Eve empfängt den chiffrierten Text
c = 968696137546220614771409222543558829057599911245743198746951209308162982251457083569314766228839896280133919905518299451578151 54.
Wie lautet der Klartext, wenn A = 01, B = 02, …, Z = 26 und das Leerzeichen mit 00 codiert wird?

Quelle (ins Deutsche übertragen):
http://stat.math.uregina.ca/~kozdron/Teaching/Regina/124Winter09/Handouts/RSA129.pdf

Anmerkung:
Martin Gardner zitierte in seinem Artikel Ron Rivest mit der Schätzung, dass „40 quadrillion years“, also 40 Billiarden ($40 \cdot 10^{15}$) Jahre Rechenzeit benötigt würden, um die Semiprimzahl n in ihre Faktoren p und q zu zerlegen. Diese Zeitspanne ist weitaus größer als das Alter der Erde (ca. $4{,}55 \cdot 10^9$ Jahre). Wenn Rivests damalige Schätzung zugetroffen hätte, wäre eine Entzifferung unmöglich gewesen.

Lösung:

Tatsächlich gelang es bereits im April 1994 einer Gruppe von Wissenschaftlern um Arjen K. Lenstra zusammen mit 600 Freiwilligen, die er über das Internet gefunden hatte, die Zahl n in ihre Primfaktoren zu zerlegen:

n = 3490529510847650949147849619903898133417764638493387843990820577 · 32769132993266709549961988190834461413177642967992942539798288533

Anmerkung:

Die Zahl n erhielt später den Namen RSA-129, weil die Dezimaldarstellung 129 Stellen hat. Zur Faktorisierung wurde eine Variante des Quadratischen Siebs verwendet (vgl. Schulz/Witten, S. 70 ff. in diesem Heft).

Mit der Faktorisierung kann man nun die Botschaft entschlüsseln: Die Primfaktoren von n werden wie üblich mit p und q bezeichnet und zusammen mit dem öffentlichen Schlüssel $e = 9007$ in SAGE eingegeben, sodass sie über ihre jeweiligen Variablennamen angesprochen werden können (siehe Bild 3, nächste Seite).

Als nächstes wird der Wert für *phi* berechnet:

```
phi = (p - 1)*(q - 1)
```

Bei der Größe der hier verwendeten Zahlen ist es natürlich unmöglich geworden, d durch simples Probieren (wie im ersten Beispiel) zu finden. Man errechnet die modulare Inverse (auch für sehr große Zahlen) vielmehr sehr effizient mit dem erweiterten Euklidischen Algorithmus (vgl. Witten/Schulz, 2006b), dies erledigt SAGE mit der inverse_mod-Funktion unsichtbar im Hintergrund.

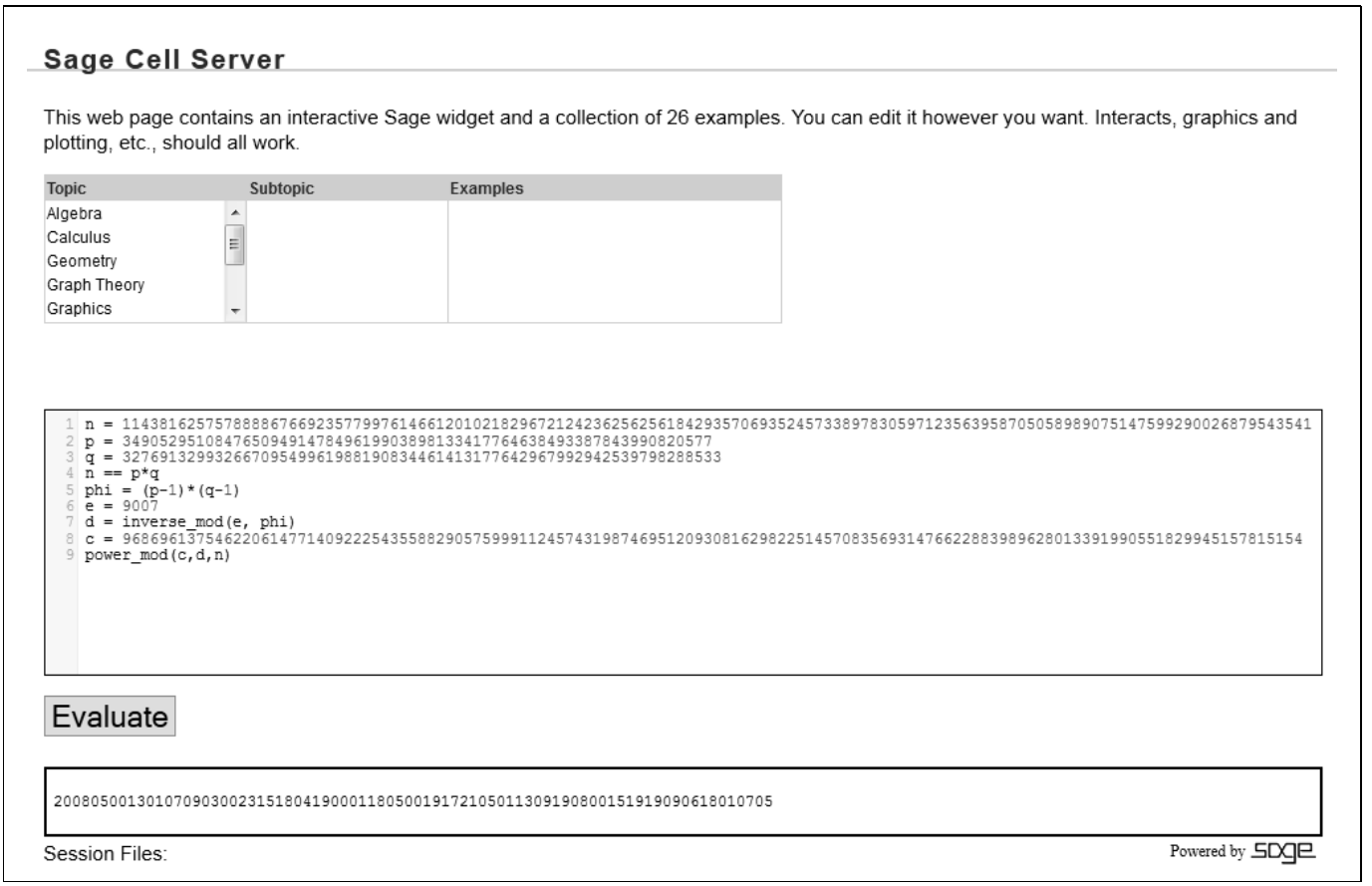


Bild 3: Der SAGE Cell Server, eine Software-Komponente auf der SAGE-Homepage, mit dem das Computer-Algebra-System direkt online genutzt werden kann.

Nach der Eingabe im oberen Fenster, die auch über mehrere Zeilen gehen kann, muss die Schaltfläche „Evaluate“ gedrückt werden, um das Ergebnis im unteren Fenster zu erhalten. Die vorhergehenden Ergebnisse werden dabei überschrieben, wenn die Ausgabe nicht explizit mit den print-Befehlen erzwungen wird. In dem Bildschirmfoto sind die Werte zu dem RSA-129-Beispiel zu sehen.

Man kann daher mit dem SAGE-Befehl

```
d = inverse_mod(e,phi)
```

ohne Weiteres den geheimen Schlüssel von Alice rekonstruieren (natürlich nur, weil p und q bekannt waren).

Mit der SAGE-Funktion power_mod(c,d,n) erhält man die entschlüsselte Botschaft:

200805001301070903002315180419000118050019172105011309190800151919090618010705.

Auch hier arbeitet ein sehr effizienter Algorithmus im Hintergrund: Potenzieren durch Quadrieren und Multiplizieren (*square and multiply*; vgl. Witten/Schulz, 2006a).

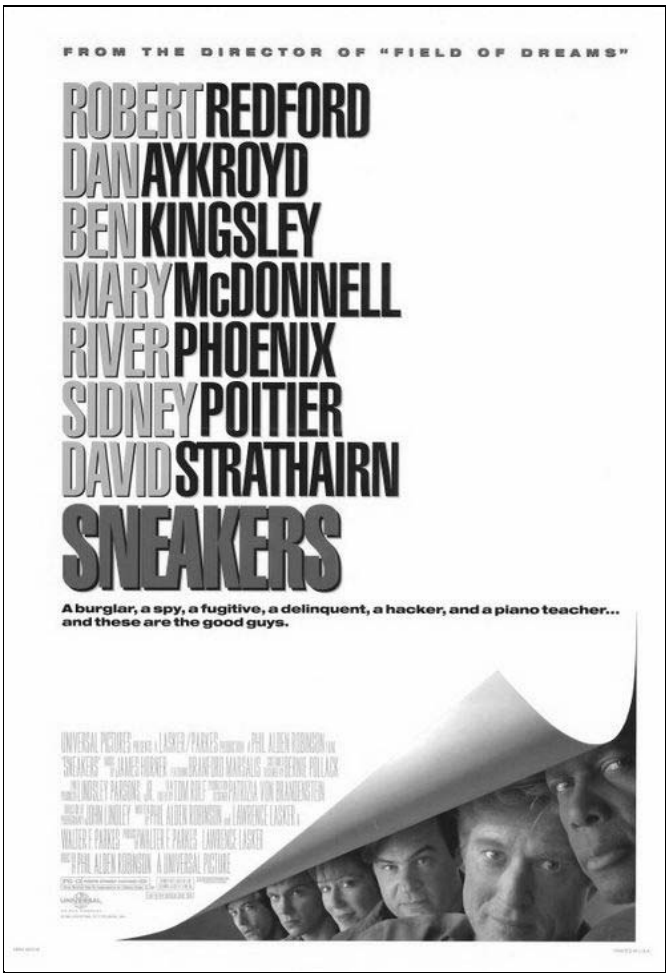
Diese Botschaft ergibt mit der oben erwähnten Codierung (je zwei Ziffern ergeben einen Buchstaben) den Satz „THE MAGIC WORDS ARE SQEAMISH OSSIFRAGE“ (vgl. Wikipedia, Stichwort „The Magic Words are Squeamish Ossifrage“; vgl. auch Witten/Schulz, 2006b).

Das Faktorisierungsproblem

Jede natürliche Zahl lässt sich in eindeutiger Weise in ein Produkt von Primfaktoren zerlegen – dieser Fundamentalsatz der Arithmetik war bereits Euklid bekannt, wurde aber erstmals von Carl Friedrich Gauß 1798 in seiner Dissertation, die 1801 unter dem Titel *Disquisitiones Arithmeticae* in Leipzig erschien, vollständig und korrekt bewiesen.

Die tatsächliche Zerlegung einer großen Zahl in ihre Primfaktoren war laut Carl Pomerance (s.o.) lange Zeit für die meisten Mathematiker uninteressant – es genügte ihnen die Tatsache, dass der Fundamentalsatz der Arithmetik garantiert, dass das Faktorisierungsproblem „im Prinzip“ für alle Zahlen lösbar ist. Algorithmen zur Faktorisierung und Primzahltests waren höchstens als Vergleichswert (Benchmark) für die Leistungsfähigkeit der seit etwa 1950 an den Universitäten immer stärker verbreiteten Computer interessant.

Das änderte sich ein Vierteljahrhundert später schlagartig beim Aufkommen der asymmetrischen Kryptografie im Allgemeinen und RSA im Besonderen. Während die Multiplikation auch sehr großer Zahlen ein algorithmisch leichtes Problem ist, ist bis heute kein vergleichbar effizientes Verfahren zur Faktorisierung bekannt. Diese Asymmetrie garantiert (bislang) die Sicherheit der elektronischen Kommunikation z.B. beim E-Commerce. Wir wer-



Quelle: LOG-IN-Archiv

Bild 4: Plakat zum Film Sneakers (1992).

den weiter unten sehen, ob diese Sicherheit durch die anfangs zitierten neuen Untersuchungsergebnisse von Arjen Lenstra und seinen Mitarbeitern erschüttert wurde.

Untersuchungen zur Primfaktorzerlegung sind laut Carl Pomerance eher untypisch für die mathematische Forschung. Es werden keine Theoreme formuliert und bewiesen, man geht wie eine Experimentalwissenschaft vor. Die Rolle der Natur übernehmen in diesem Fall die ganzen Zahlen. Ob eine Zahl ein Primfaktor der untersuchten Zahl n ist oder nicht, kann dadurch entschieden werden, dass die Division mit dem vermuteten Primfaktor aufgeht oder nicht. So kann man z. B. die oben angegebene Zerlegung der Zahl RSA-129 mit jedem System überprüfen, das die Langzahlarithmetik beherrscht (z. B. SAGE).

Die erste Methode zur Faktorisierung ist bis heute die Probedivision, die besonders effizient mit dem Euklidischen Algorithmus durchgeführt werden kann. Ist der größte gemeinsame Teiler (ggT) gleich 1, sind die untersuchten Zahlen (n, k) zueinander teilerfremd. Man sagt dann auch, dass sie *relativ prim* sind. Ansonsten findet der Euklidische Algorithmus die größte Zahl, die n und k als gemeinsamen Faktor haben.

Man verwendet für die Probedivisionen aus Effizienzgründen nur Primzahlen, um keine Faktoren eventuell doppelt zu prüfen. Mit dieser Methode findet man ziemlich schnell heraus, ob die untersuchte Zahl kleine Primfaktoren hat (die meisten Zahlen haben kleine Primfaktoren). Man kann sogar mehrere Primfaktoren auf einen Schlag bestimmen, wenn man z. B. das Produkt der Primzahlen kleiner 50 bildet, also

$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot \ldots \cdot 47 = 614889782588491410$

berechnet und den ggT von dieser Zahl mit der zu untersuchenden Zahl n bildet.

Ein kleines „Zufallsexperiment“ mit SAGE kann diese Aussagen illustrieren. Die durch „zufälliges“ Tippen erzeugte Zahl n lautet z. B.

651653632032165461302003213846313230207 11,

der SAGE-Befehl

```
gcd(614889782588491410,n)
```

liefert 3 (gcd = *greatest common divisor* = ggT).

Nur wenn man 1 erhält, ist man sicher, dass n keinen Primfaktor enthält, der kleiner als 50 ist.

Neugierig geworden, ob diese Zahl n wirklich nur den einen Primfaktor 3 kleiner als 50 hat, kann man sich mit `factor(n)` die vollständige Primfaktorzerlegung anzeigen lassen:

$n = 3 \cdot 107 \cdot 2030073620037898633339573874910633115 91.$

Der dritte Primfaktor ist dann eine „zufällig“ erzeugte Primzahl mit immerhin 39 Stellen.

Wichtig für die Faktorisierung sind auch die Primzahltests, die wir in der letzten Folge besprochen haben (vgl. Witten/Schulz, 2010b). So kann man mit dem SAGE-Befehl

```
is_prime(203007362003789863333957387491063311591)
```

überprüfen, ob diese 39-stellige Zahl tatsächlich *prim* ist. Da der Rückgabewert `True` ist, können wir uns beruhigt zurücklehnen: Die oben angegebene Primfaktorzerlegung ist vollständig!

Inzwischen ist bewiesen, dass Primzahltests in der Klasse P liegen, also effizient durchgeführt werden können (vgl. Witten/Schulz, 2010b). Von allen bislang bekannten Faktorisierungsverfahren (außer dem Shor-Algorithmus für Quantencomputer, siehe weiter unten) weiß man dagegen nur, dass sie in der Klasse NP liegen, da die Korrektheit einer bereits gefundenen Faktorisierung durch einfache Multiplikation(en) in polynomialer Zeit überprüft werden kann. Allerdings ist es nicht bewiesen, dass auch in Zukunft kein schneller Algorithmus zur Faktorisierung gefunden werden kann, ebenso wenig ist nachgewiesen, dass die Faktorisierung NP-vollständig ist. Trotzdem zeigen Experimente zur Faktorisierung, dass bei den bekannten Verfahren der Aufwand in der Regel ungefähr exponentiell mit der Stellenzahl wächst (vgl. Schulz/Witten, 2010), sodass man die Sicherheit von RSA durch die Länge der verwendeten Schlüssel steuern kann (siehe weiter unten).

Neue Faktorisierungsverfahren?

Insofern stand (und steht) die Sicherheit von RSA von Anfang an und bis heute auf schwankender Grund-

lage: Warum sollte es nicht möglich sein, dass ein Genie einen Algorithmus findet, der eine Semiprimzahl in polynomialer Zeit zerlegt? Diese Schwachstelle war den RSA-Erfindern von Anfang an bewusst; sie wurde sogar 1992 Gegenstand des Hollywood-Films *Sneakers* (deutscher Untertitel *Die Lautlosen*; vgl. IMDb – Stichwort „Sneakers …"; siehe auch Bild 4, vorige Seite) mit Starbesetzung – u.a. Robert Redford, Ben Kingsley und Sidney Poitier. Len Adleman wirkte als mathematischer Berater mit (siehe Kasten „Len Adleman und der Film *Sneakers*"). In dem Film gibt es einen russischen Mathematiker, der einen neuen, schnellen Algorithmus zur Primfaktorzerlegung erfunden und dummerweise in einem Chip implementiert hat. Er wird ermordet, und dieser Chip gelangt in die Hände einer Mafia-ähnlichen Organisation, die daraufhin alle Codes brechen kann. „Die Lautlosen" (ein sogenanntes Tiger-Team) unter der Führung von Martin Bishop (Robert Redford) stehen danach vor der Aufgabe, diesen Chip aus den Händen der Mafia zurückzuholen.

Der Film *Sneakers* war mit einem Erlös von über 105 Millionen Dollar (bei Produktionskosten von 35 Millionen Dollar) ein kommerzieller Erfolg; er ist seit vielen Jahren als DVD erhältlich und immer noch spannend anzusehen, auch weil manche dort gezeigten Geräte (z.B. ein Akustikkoppler für die Datenfernübertragung) zwar nur noch von historischem Interesse sind, das Faktorisierungsproblem aber immer noch hochaktuell ist.

Tatsächlich ist bislang kein neues Faktorisierungsverfahren gefunden worden, das für große Zahlen (ab 100 Stellen im Dezimalsystem) schneller als das Zahlkörpersieb ist. Der aktuelle Rekord ist die Zerlegung der Zahl RSA-768 (mit 232 Stellen im Dezimalsystem; vgl. Wikipedia – Stichwort „RSA-768") am 12. Dezember 2009 durch eine Gruppe von Forschern um den Bonner Mathematiker Thorsten Kleinjung, der schon an der Universität Bonn an der Zerlegung einiger Semiprimzahlen aus der RSA-Challenge beteiligt war und zz. Mitarbeiter von Arjen Lenstra am LACAL in Lausanne ist (vgl. Kleinjung u.a., 2010; vgl. auch Witten/Schulz, 2010a).

Das zweitschnellste Verfahren (nach wie vor das schnellste Verfahren für Zahlen mit bis zu 100 Dezimalstellen), das *Quadratische Sieb* (siehe Schulz/Witten, Seite 70 ff. in diesem Heft), ist inzwischen stark verbessert worden. Die Implementierung in *CrypTool 2* (CT 2) schafft auf einem aktuellen normalen PC die Zerlegung von RSA-100 (eine Semiprimzahl mit 100 dezimalen Stellen) in ca. 2½ Stunden. Für die erste Zerlegung dieser Zahl benötigte Arjen Lenstra auf einem MasPar-Supercomputer im Jahr 1991 noch mehrere Tage (vgl. Wikipedia – Stichwort „RSA-100"); Ron Rivest hatte im originalen RSA-Artikel von 1978 die hierfür benötigte Zeit auf 74 Jahre geschätzt.

Auch RSA-129 ist in die Reichweite handelsüblicher PCs gekommen. Die Implementierung des Quadratischen Siebs in CT 2 ermöglicht nämlich die Zerlegung dieser Zahl im Schulnetz mit vielleicht 30 Rechnern über ein Wochenende (siehe Bild 5, nächste Seite), während im Jahr 1994 Arjen Lenstras Gruppe mit 1600 Computern über acht Monate benötigte (allerdings nicht durchgehend, meist wurden die Workstations mit

Len Adleman und der Film *Sneakers*

The story of Sneakers, the movie and Len Adleman the mathematician is as follows:

Larry Lasker was one of the writers of the 1983 hit movie War Games. Based on that success, he started to produce his own movies. [...]

A short while later [1990], Larry again made contact. This time he was well on his way to making Sneakers, starring Robert Redford, Sidney Poitier, Mary McDonnell, Dan Aykroyd and River Phoenix. He told me that there would be a scene wherein a researcher would lecture on his mathematical work regarding a breakthrough in factoring – and hence in cryptography. Larry asked if I would prepare the slides and words for that scene. I liked Larry and his desire for verisimilitude, so I agreed. Larry offered money, but I countered with Robert Redford – I would do the scene if my wife Lori could meet Redford.

I worked hard on the scene. The "number field sieve" (the fastest factoring algorithm currently known) is mentioned along with a fantasy about towers of number fields and Artin maps. I was tempted to name the new breakthrough the "function field sieve" – since I was actually working on a paper at the time which would later appear with that title – but I decided against it, for reasons that escape me now.

I made beautiful slides on my Mac. This took a great deal of time (graphics programs were not as user friendly as they are now) but I wanted the stuff to look impressive. As it turns out, Larry had them redrawn by hand by some guy on his crew – he said that hand drawn slides looked more realistic. Of course he was right – but I could have saved a lot computer time had I known in the first place.

The lecture scene was actually shot at a small college in LA. Larry told me that some physics professor there saw the slides and said that they did not show math at all. He offered to redraw them for a small fee – Larry declined.

Lori and I were there when the scene was shot. I was most pleased with my phrase "a breakthrough of Gaussian proportions", – the Prince of Mathematics could use a plug in a major motion picture. We were introduced to Redford and chatted with him for about five minutes – that is Bob and I chatted – Lori said hello and then apparently was too star struck to add more. I was given credit at the end of the movie as (in my recollection) "mathematical consultant". Anyway the Academy snubbed me – since apparently the mathematical consultant Oscar for that year went to someone else.

Len Adleman, 18 Jan 1998

Quelle: http://www.usc.edu/dept/molecular-science/fm-sneakers.htm

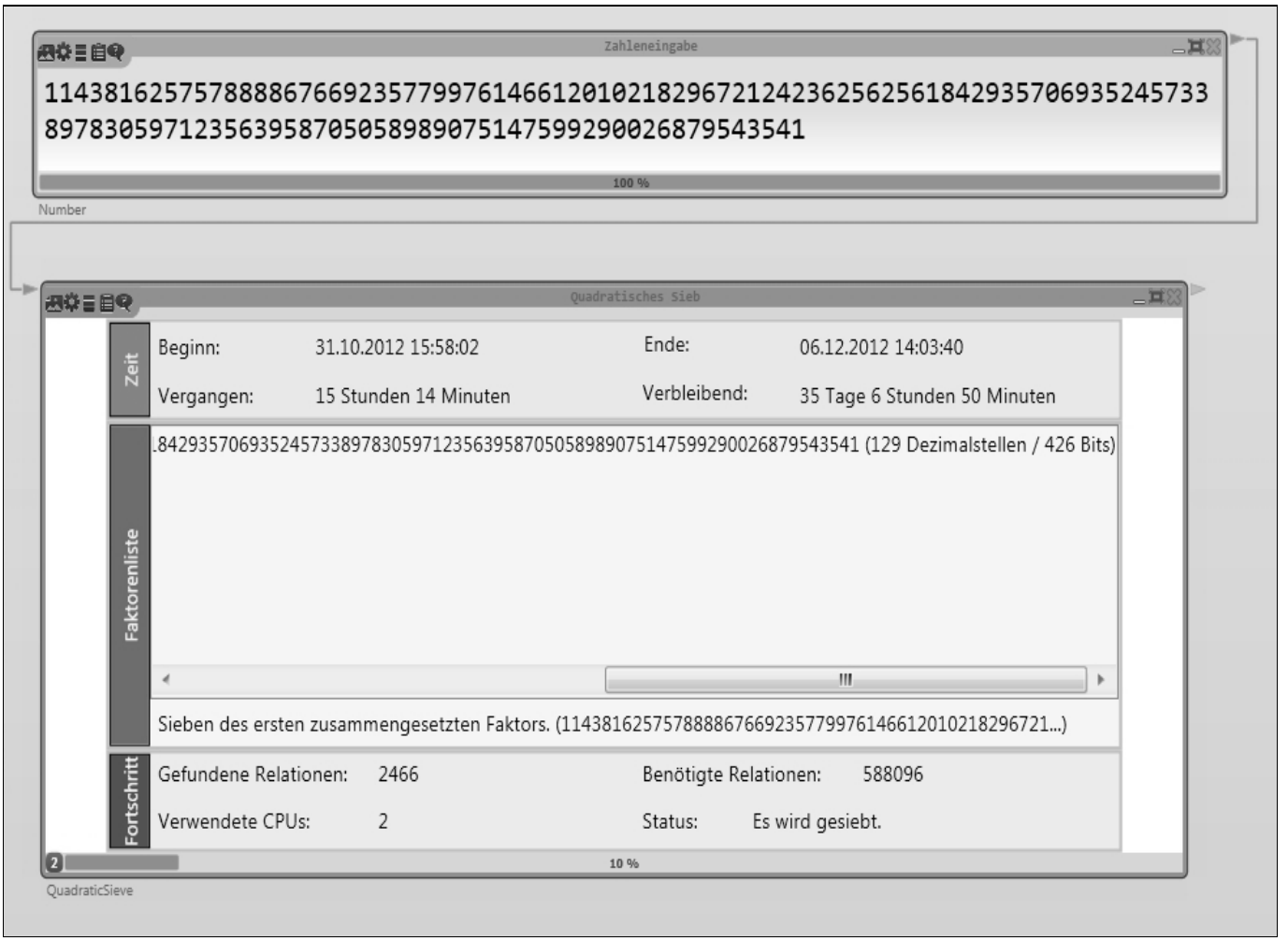


Bild 5: Bildschirm-Foto des Quadratischen Siebs nach gut 15 Stunden des Siebens von RSA-129 mit einem handelsüblichen PC (Intel Core i7 mit 3,4 GHz).

Im oberen Fenster sieht man RSA-129 in voller Länge, im unteren Fenster sieht man oben das Ende und weiter unten den Anfang dieser Zahl. Mit CrypTool 2 wird geschätzt, dass die Zerlegung in 35 weiteren Tagen fertig sein könnte. Die Schätzungen schwanken zu Beginn um einige Tage (± 10 d). Da die Rechnung bei CT 2 auch auf mehrere vernetzte PCs aufgeteilt werden kann, wäre die Zerlegung im Netz einer Schule über ein Wochenende möglich!

der Faktorisierung beschäftigt, wenn sie nicht im Lehrbetrieb an den verschiedenen Universitäten benötigt wurden; vgl. Janeba, 1994).

Wie findet man „harte“ Semiprimzahlen?

Die RSA-Challenge (s.o.) bietet eine Fülle von noch nicht zerlegten, schwer zu faktorisierenden Semiprimzahlen, die wir hier als *hart* bezeichnen. Diese Zahlen wurden in den 1990er-Jahren von Mitarbeitern der Firma *RSA-Security* auf einem Laptop erzeugt, dessen Festplatte später physikalisch zerstört wurde, damit wirklich niemand mehr Zugang zu den erzeugenden Primfaktoren haben konnte.

Wenn man selbst solche Primzahlen generieren will, braucht man einen guten Zufallsgenerator, der bei den Zahlen der gewünschten Größenordnung (z.B. 1024 Bit für 2048-Bit-Schlüssel) so lange sucht, bis eine Primzahl gefunden ist. Der Primzahltest der Wahl ist der Miller-Rabin-Test, der seit vielen Jahren den Test von Solovay-Strassen abgelöst hat (vgl. Witten/Schulz, 2010b) und z.B. in *CrypTool 1* (CT 1) implementiert ist. Mit dem Gauß'schen Primzahlsatz kann man zeigen, dass es mehr als genügend Primzahlen der gewünschten Größenordnung gibt, und zwar $1{,}26 \cdot 10^{305}$ (vgl. Witten/Schulz, 2010a). Man kann es sich also erlauben, die Bedingungen zu verschärfen!

Damit die Semiprimzahlen nämlich nicht mit weiteren, in diesem Artikel nicht erwähnten älteren und in Spezialfällen schnelleren Faktorisierungsverfahren zerlegt werden können, müssen u.a. folgende Bedingungen erfüllt sein (vgl.. z.B. Karpfinger/Kiechle, 2010, S.126):

- Die Größenordnung von p und q sollte jeweils 1024 (oder 2048) Bit sein, die beiden Primfaktoren dürfen aber nicht zu nahe beieinander liegen (→ Faktorisierungsverfahren von Fermat; siehe Schulz/Witten, Seite 70 ff. in diesem Heft);
- $p - 1$ sollte mindestens einen großen Primteiler r haben und $q - 1$ mindestens einen großen Primteiler s;
- $p + 1$ und $q + 1$ sollten nur große Primteiler haben;

▷ Die Zahlen $r - 1$ und $s - 1$ sollten ebenfalls nur große Primteiler haben.

Auch die Botschaft m sollte bestimmten Vorgaben genügen und eventuell vorverarbeitet werden (vgl. Robinson, 2003).

Einen Überblick über die älteren Faktorisierungsverfahren erhält man übrigens in dem erwähnten Buch von Kapfinger/Kiechle (2010, Kap. 11, S. 191 ff.). Weitere Einzelheiten zu den o. a. Bedingungen erfährt man im Programm CT 1 im Fenster Einzelverfahren → RSA-Kryptosystem → Faktorisieren einer Zahl, wenn man dort die Hilfe-Taste (F1) drückt.

RSA brechen mit TWINKLE und TWIRL?

Von Adi Shamir stammt der Vorschlag, die Faktorisierung durch spezielle Geräte zu beschleunigen. Shamir wies in seinem Papier darauf hin (vgl. Shamir, 1999, S. 11), dass diese Idee bereits in den 20er-Jahren des vorigen Jahrhunderts von Derrick Norman Lehmer und seinem Sohn Derrick Henry Lehmer (beide nacheinander Professoren für Mathematik in Berkeley) realisiert wurde, z. B. mit Zahnrädern und unterschiedlich langen Fahrradketten oder mit gelochten 16-mm-Filmen. Diese Geräte konnten u. a. zur Faktorisierung verwendet werden (siehe Bild 6).

Das erste von Shamir 1999 vorgeschlagene Gerät TWINKLE soll opto-elektronisch funktionieren. Der Name ist einerseits ein Akronym (***T**he **W**eizmann **In**stitute **K**ey **L**ocating **E**ngine*; Adi Shamir arbeitet seit vielen Jahren am Weizmann-Institut in Tel Aviv), andererseits ein Wortspiel wegen der blinkenden LEDs, die in diesem Gerät verwendet werden. Es wurde im Hinblick auf das Quadratische Sieb und auf das Zahlkörpersieb entwickelt und sollte bei beiden Verfahren den Siebschritt bei 512-Bit-Semiprimzahlen dramatisch beschleunigen. Es ist nicht bekannt, ob ein TWINKLE tatsächlich gebaut wurde (vgl. Wikipedia – Stichwort „TWINKLE“).

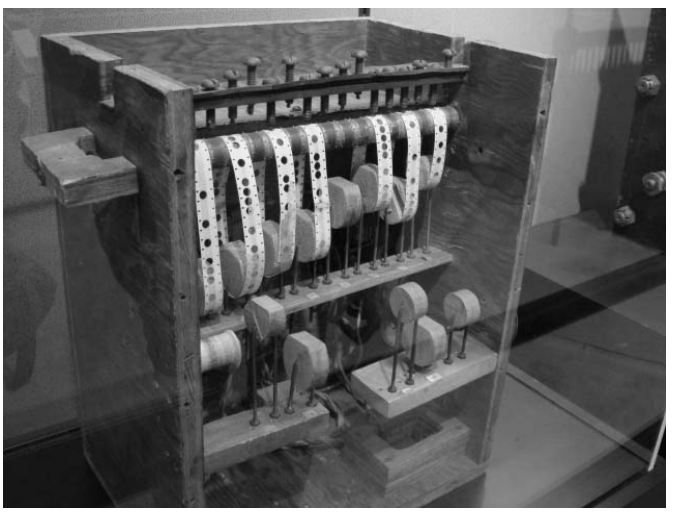


Foto: LOG-IN-Archiv (aus dem Computer History Museum, Mountain View, CA, USA)

Bild 6: Ein Lehmer-Sieb von 1936 – ein primitiver Digitalcomputer, um Primzahlen zu finden oder diophantische Gleichungen zu lösen.

Das zweite, ebenfalls hypothetische Gerät, folgte 2003 und erhielt den schönen Namen TWIRL, natürlich wieder ein Akronym (***T**he **W**eizmann **I**nstitute **R**elation **L**ocator*). Shamir hatte das Gerät zusammen mit seinem Mitarbeiter Eran Tromer erarbeitet. Die Funktionsweise wird in der PowerPoint-XP-Präsentation mit animierten Illustrationen demonstriert, die auf der Konferenz *Crypto 2003* vorgestellt wurde (vgl. Shamir/Tromer, 2003). Wenn man diese Animationen betrachtet, sieht man u. a. einen Prozessor, der um einen zyklischen Computer-Speicher rotiert. TWIRL kann man als Wirbel oder Drehung übersetzen, sodass wir auch bei diesem Gerät nicht auf ein Wortspiel verzichten müssen.

TWIRL sollte diesmal in Silizium-Technik ausgeführt werden und gegenüber herkömmlicher Technik einen erheblichen Kostenvorteil durch dramatische Zeit-Einsparung bringen, gebaut wurde es bislang noch nicht. Man kann natürlich nicht ausschließen, dass z. B. die NSA Tausende von Dollars in den Bau solcher Geräte investiert hat und der Öffentlichkeit nichts davon verrät – aber auch nach der Schätzung von Shamir benötigt man mit 160 TWIRL-Clustern mindestens ein Jahr zur Zerlegung eines 1024-Bit-Schlüssels.

Kann RSA mit einem Quantencomputer geknackt werden?

Um auf diese Frage direkt zu antworten: Im Prinzip ja, aber …

Quantencomputer basieren auf den Prinzipien der Quantenmechanik, insbesondere der Superposition (Überlagerung) und der Verschränkung, die zu tatsächlich oder vermeintlich absurden Konsequenzen führt. Von Erwin Schrödinger stammt das berühmte Beispiel mit einer Katze in einer Kiste, die gleichzeitig tot und lebendig ist – beide Zustände sind überlagert, erst beim Öffnen der Kiste (das entspricht der Messung in der Quantenphysik) entscheidet sich, welchen Zustand die Katze annimmt! Die Verschränkung ist die Grundlage für die von Einstein sogenannte „spukhafte Fernwirkung“ (vgl. Bussemer, 2010a, S. 70 ff.), die heute für die Quantenkryptografie bereits praktisch genutzt werden kann.

Quantencomputer schienen bis Mitte der 1990er-Jahre eine eher esoterische Möglichkeit zu sein, bis der Mathematiker Peter Shor 1993 am Bell Lab in Murray Hill, New Jersey, in den USA einen schnellen (polynomialen) Algorithmus zur Faktorisierung von ganzen Zahlen erfand. So hat beispielsweise eine Forschungsgruppe der IBM im Jahr 2001 einen Quantencomputer mit sieben Qubits eingesetzt, um mithilfe des Shor-Algorithmus die Zahl 15 in die Faktoren 5 und 3 zu zerlegen (vgl. Vandersypen u. a., 2001) – auch die heute primär eingesetzten Computer haben in ihrer Entwicklung so klein angefangen (siehe aber weiter unten). Der Shor-Algorithmus gewinnt seine Schnelligkeit vor allem nur durch die Anwendung der Quanten-Fouriertransformation, die wir an dieser Stelle nicht erklären können. Wir verweisen stattdessen auf die Literatur,

besonders auf das hervorragende Buch von Matthias Homeister (^{2}2008) und – als preiswerte Alternative – auf das schmale Bändchen *Quanteninformation* von Dagmar Bruß (2003). Grundlagen zur Quanteninformatik wurden auch in der Beitragsreihe *Können Quanten rechnen?* von Peter Bussemer in LOG IN vorgestellt (vgl. Bussemer, 2009, 2010a und 2010b).

Durch die Möglichkeit, für Faktorisierungen einen Algorithmus mit polynomialer Laufzeit zur Verfügung zu haben (und damit alle RSA-Schlüssel knacken zu können), war die Öffentlichkeit elektrisiert. Peter Shor, heute Professor für Mathematik am MIT, erhielt für den nach ihm benannten Algorithmus auf dem Internationalen Mathematiker-Kongress 1998 in Berlin den *Nevanlinna-Preis für theoretische Informatik* (vgl. Bussemer, 2010b, S.119), eine der höchsten Auszeichnungen, die auf diesem Gebiet vergeben werden können. Auf dem Kongress wurde Shor von Volker Strassen vorgestellt, der in einem Limerick auf das praktische Problem hinwies, dass Quantencomputer zu dieser Zeit noch nicht einmal 12 oder 15 faktorisieren konnten:

> To read our E-mail, how mean
> of the spies and their quantum machine;
> Be comforted though,
> they do not yet know
> how to factorize twelve or fifteen.
>
> Quelle: http://www-math.mit.edu/~shor/notapoet.html

Volker Strassen brachte zum Ausdruck, warum die praktischen Kryptologen trotz des Shor-Algorithmus noch gut schlafen können. Obwohl es – wie oben kurz erwähnt – im Jahr 2001 Forschern bei IBM gelungen ist, die Zahl 15 mit diesem Algorithmus und einem Quantencomputer zu faktorisieren, ist es offenbar noch ein sehr weiter Weg, bis diese Art von Faktorisierung eine echte Bedrohung für die Sicherheit von RSA wird – eine weitere Faktorisierung oder gar ein Durchbruch auf diesem Wege ist in den letzten 10 Jahren nicht bekannt geworden.

Ist RSA durch die neue Untersuchung von Arjen Lenstra & Co. geschwächt oder gar geknackt?

Auch bezüglich der eingangs erwähnten Untersuchung können wir uns entspannt zurücklehnen. In einem Interview erklärte Arjen Lenstra: „If properly implemented, RSA is fine“ (Messmer, 2012).

Der geniale Gedanke bei dieser Untersuchung war, dass nicht versucht wurde, *einzelne* Schlüssel zu faktorisieren. Es wurde vielmehr nach dem größten gemeinsamen Teiler von jeweils zwei Schüsseln gesucht.

Wir wollen das an zwei selbst konstruierten Schlüsseln durchspielen, die – entgegen den Regeln – einen gemeinsamen Primfaktor haben. Dazu erzeugen wir drei 512-Bit-Primzahlen p, q_1 und q_2 – der Einfachheit halber mit CT 1:

→ Einzelverfahren → RSA-Kryptosystem
→ Primzahlen generieren

jeweils mit der Untergrenze 2^{512} und der Obergrenze 2^{513}. Die Aufgabe erledigt der in CT 1 implementierte Miller-Rabin-Test in wenigen Sekunden.

Die Variablen werden dann mit ihren Werten in SAGE eingegeben:

```
p =
13931663684190457165277107136142136655125628604926985024762644970078828945971890597371796778946014456758400919423920723383879937983455115635208370374785889

q1 =
24930443029311467022497097701056772781154209695498885203813653610102894423264830034304018220511004510032933943214365018892528392690992121888850400999640119

q2 =
21904250581779901389352671867606049603736858403969936266584703126654816361834039316048068206416998425597306465403175363493295528939944089609255988688176269
```

Daraus erzeugen wir zwei 1024-Bit-Moduln mit $n_1 = p \cdot q_1$ sowie $n_2 = p \cdot q_2$ und testen den ggT(n_1,n_2):

```
n1 = p*q1
n2 = p*q2
time gcd(n1,n2) == p
```

(Zur Erinnerung: gcd = *greatest common divisor*, also ggT. Das vorangestellte time bewirkt, dass bei der Berechnung die dafür benötigte Zeit gemessen wird.) Die Ausgabe liefert erwartungsgemäß True; für die benötigte Zeit wird ausgegeben:

```
Time: CPU 0.00 s, Wall: 0.00 s
```

Wir haben durch gcd(n1, n2) die *zwei* 1024-Bit-Schlüssel n_1 und n_2 in 0,00 s faktorisiert, denn die anderen Faktoren q_1 und q_2 finden wir danach durch einfache Division! Das gelingt aber nur, weil die beiden Schlüssel (nach Konstruktion) einen gemeinsamen Faktor p haben.

Bei allen Problemen, die bislang im Zusammenhang mit dem RSA-Kryptosystem aufgetreten sind, handelt es sich um Probleme der Implementierung oder des Schlüssel-Managements – da macht die Entdeckung der Möglichkeit, RSA-Schlüssel mit dem Euklidischen Algorithmus zu knacken, offenbar keine Ausnahme. Wenn RSA korrekt implementiert wird, ist es (noch) sicher. Dazu gehört die Vorschrift, die Primzahlen zufällig zu wählen. Bei der riesigen Anzahl der zur Verfügung stehenden Zahlen (s.o.) ist die Wahrscheinlichkeit, eine Primzahl doppelt zu verwenden, verschwindend gering, wenn ein vernünftiger Zufallsgenerator verwendet wird (vgl. Esslinger u.a., 2012). Tatsächlich wurden aber unvertretbar viele Primfaktoren doppelt genutzt und konnten daher mit der eben beschriebenen Methode gefunden werden.

Ron was wrong, Whit is right

Dies ist der provokative Titel der nun schon mehrfach erwähnten Arbeit der Gruppe um Arjen Lenstra am LACAL in Lausanne (vgl. Lenstra u.a., 2012). Speziell die in diesem Titel enthaltene Behauptung hat verschiedene Krypto-Experten zu heftigem Widerspruch herausgefordert.

Zunächst einmal besteht Einigkeit unter den Experten, dass RSA bei korrekter Anwendung nach wie vor

sicher ist. Aber in dem Titel steckt die Behauptung, dass das von Ron Rivest entwickelte RSA-Kryptosystem prinzipiell doch etwas unsicherer ist als die auf der Basis von Whit Diffies und Martin Hellmans Schlüsseltausch entwickelten Verfahren (ElGamal-Kryptosystem, Digital Signature Algorithm DSA, Elliptische-Kurven-Kryptographie ECC). Die Sicherheit dieser Verfahren beruht auf der Schwierigkeit, den diskreten Logarithmus zu bestimmen. Da wir in der nächsten Folge diese Alternativen zum RSA-Verfahren ohnehin vorstellen wollen, müssen wir eine tiefer gehende Beantwortung dieser Frage noch etwas zurückstellen.

Hier nur so viel: Die Kritik von Arjen Lenstra und seiner Gruppe beruht auf der Tatsache, dass im RSA-Modul zwei Geheimnisse enthalten sind (nämlich die Primfaktoren p und q) – wenn man eins der Geheimnisse kennt, ist RSA geknackt. Die auf dem diskreten Logarithmus beruhenden Verfahren kommen mit *einem* Geheimnis aus (vgl. Wikipedia – Stichwort „Decisional-Diffie-Hellman-Problem“).

Dass die Firma RSA dem widersprechen würde, verwundert nicht. Aber auch andere Krypto-Experten wie Dan Kaminsky und Nadia Heninger lobten einerseits die Arbeit, kritisierten aber andererseits die im Titel enthaltene These. Dan Kaminsky schrieb in seinem Blog unter anderem:

> Key Management — as Whit Diffie himself has said — is The Hard Problem now for cryptography. Whether you use RSA or DSA or ECDSA [das heißt DSA mit elliptischen Kurven (*Anm. der Autoren*)], that differential risk is utterly dwarfed by our problems with key management.
>
> Quelle: http://dankaminsky.com/2012/02/14/ronwhit/

Weitere Informationen zum „Broken Moduli Bug“ finden sich bei Nadia Heninger, die ebenfalls umfangreiche empirische Untersuchungen zu der Verletzlichkeit von öffentlichen RSA-Schlüsseln unternommen hat (vgl. Heninger, 2012). Sie fand sogar heraus, dass sich fast 1 Prozent der privaten Schlüssel berechnen ließen – die meisten davon stammten aus eingebetteten Systemen wie Routern oder Firewalls.

Die Ergebnisse der Studie von Nadia Heninger sowie der von Arjen Lenstra u.a. wurden nochmals von Dan Kaminsky zusammengefasst und bewertet (vgl. Kaminsky, 2012). Dieser letztgenannte Artikel enthält weitere Empfehlungen für sichere Implementierungen von RSA.

In der Zeitschrift <*kes*> vom April 2012 erschien ein Artikel von Esslinger u.a., in dem die unterschiedlichen Gründe für „shared primes“ in verschiedenen Moduln aufgeschlüsselt, aktuelle Gegenmaßnahmen formuliert und zusätzlich zu den von Lenstra untersuchten öffentlichen Schlüsseln die Qualität nicht-öffentlicher Unternehmensschlüssel untersucht werden. Ein dazu benutztes Testprogramm findet sich auf der *CrypTool*-Webseite

http://www.cryptool.org/de/ctp-dokumentation-de/361-ctp-paper-rsa-moduli

Das Programm benutzt PYTHON und SAGE und ist sinnvoll, um rund eine Million Moduln in einem Tag zu untersuchen. Eine schnellere in C++ geschriebene Variante für größere Mengen von Moduln kann man vom *CrypTool*-Projekt auf Nachfrage erhalten.

Fazit und Ausblick

RSA wurde vor nunmehr 35 Jahren erfunden und hat bislang allen Attacken standgehalten. Allerdings wurde auch deutlich, dass die Implementierung viele nicht-triviale Fallen enthält, sodass man dieses Geschäft den Profis überlassen sollte. Die Sicherheit von RSA beruht in erster Linie auf der Schwierigkeit, harte Semiprimzahlen zu faktorisieren. Die bisherigen dramatischen Verbesserungen bei der für die Faktorisierung benötigten Zeit konnten leicht durch längere Schlüssel ausgeglichen werden. Ob die Faktorisierung auch in Zukunft schwierig sein wird, ob mit dem Zahlkörpersieb bereits das theoretisch und praktisch mögliche Optimum erreicht wurde oder ob ein bislang nicht entdecktes sehr viel schnelleres Verfahren existieren könnte oder ob dereinst Quantencomputer RSA den Garaus machen werden – alles das ist nach wie vor nicht bewiesen. Die Experten für Computer-Sicherheit jedenfalls glauben (zumindest für die nächsten 65 Jahre) an RSA.

In der nächsten Folge werden wir uns mit den Alternativen zu RSA auseinandersetzen.

(wird fortgesetzt)

Helmut Witten
Brandenburgische Straße 23
10707 Berlin

E-Mail: helmut@witten-berlin.de

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

Literatur und Internetquellen

Bruß, D.: Quanteninformation. Reihe „Fischer Kompakt“, Band 15563. Frankfurt: Fischer Taschenbuch Verlag, 2003.

Bussemer, P.: Können Quanten rechnen? Quanteninformatik – Einführung in die Grundprinzipien. Teil 1: Grundbegriffe der Quantenphysik. In: LOG IN, 29. Jg. (2009), Heft 160/161, S. 98–102.

Bussemer, P.: Können Quanten rechnen? Quanteninformatik – Einführung in die Grundprinzipien. Teil 2: Komponenten von Quantencomputern. In: LOG IN, 30. Jg. (2010a), Heft 162, S. 65–72 (einschließlich R. Baumann: Exkurs: Schrödingers Katze und Bertlmanns Socken).

Bussemer, P.: Können Quanten rechnen? Quanteninformatik – Einführung in die Grundprinzipien. Teil 3: Algorithmen der Quanteninformatik – Überblick. In: LOG IN, 30. Jg. (2010b), Heft 163/164, S. 116–121.

CrypTool 1 (CT 1):
http://www.cryptool.org/de/cryptool1

CrypTool 2 (CT 2):
http://www.cryptool.org/de/cryptool2

Diffie, W.; Hellman, M. E.: New Directions in Cryptography. In: IEEE Transactions on Information Theory. 22. Jg. (1978), Nr. 6, S. 644–654.
http://www.cs.jhu.edu/~rubin/courses/sp03/papers/diffie.hellman.pdf

Esslinger, B.; Simon, V.; Schneider, J.: RSA-Sicherheit in der Praxis – Fehler in der Anwendung des RSA-Algorithmus. In: <kes> – Die Zeitschrift für Informations-Sicherheit, 28. Jg. (2012), Heft 2, S. 22–27.
http://www.cryptool.org/images/ctp/documents/kes_2012_RSA_Sicherheit.pdf

Gardner, M.: A New Kind of Cipher That Would Take Millions of Years to Break. In: Scientific American, Band 237 (1977), H. 8, S. 120–124.

Heninger, N.: New research: There's no need to panic over factorable keys – just mind your Ps and Qs. 15. Februar 2012.
https://freedom-to-tinker.com/blog/nadiah/new-research-theres-no-need-panic-over-factorable-keys-just-mind-your-ps-and-qs/

Homeister, M.: Quantum Computing verstehen – Grundlagen, Anwendungen, Perspektiven. Reihe „Computational Intelligence". Wiesbaden: Vieweg, ²2008.

IMDb – Stichwort „Sneakers – Die Lautlosen":
http://www.imdb.com/title/tt0105435/

Janeba, M.: Factoring Challenge Conquered – With a Little Help From Willamette. 1994.
http://www.willamette.edu/~mjaneba/rsa129.html

Kaminsky, D.: Primal Fear – Demuddling The Broken Moduli Bug. 17. Februar 2012.
http://dankaminsky.com/2012/02/17/primalfear/

Karpfinger, Chr.; Kiechle, H: Kryptologie – Algebraische Methoden und Algorithmen. Wiesbaden: Vieweg + Teubner, 2010.

Kleinjung, Th.; Aoki, K.; Franke, J.; Lenstra, A. K.; Thomé, E.; Bos, J. W.; Gaudry, P.; Kruppa, A.; Montgomery, P. L.; Osvik, D. A.; Rile, H. te; Timofeev, A.; Zimmermann, P.: Factorization of a 768-bit RSA modulus – version 1.4, February 18, 2010.
http://eprint.iacr.org/2010/006.pdf

Lenstra, A. K.; Hughes, J. P.; Augier, M.; Bos, J. W.; Kleinjung, Th.; Wachter, Chr.: Ron was wrong, Whit is right. Cryptology ePrint Archive – Report 2012/064. 14./17. Februar 2012.
http://eprint.iacr.org/2012/064
http://eprint.iacr.org/2012/064.pdf

Markoff, J.: Flaw Found in an Online Encryption Method. In: The New York Times, 14. Februar 2012.
http://www.nytimes.com/2012/02/15/technology/researchers-find-flaw-in-an-online-encryption-method.html?_r=1

Messmer, E.: Crypto researcher Arjen Lenstra shares thoughts on paper blasting RSA cryptosystem. In: Network World, 17. Februar 2012.
http://www.networkworld.com/news/2012/021712-rsa-lenstra-256309.html

MysteryTwister C3 – The Crypto Challenge Contest:
http://www.mysterytwisterc3.org/en/
http://www.mysterytwisterc3.org/de/

Pomerance, C.: A Tale of Two Sieves. In: Notices of the American Mathematical Society, 43. Jg. (1996), Nr. 12, S. 1473–1458.
http://www.ams.org/notices/199612/pomerance.pdf

Rivest, R. L.; Shamir, A.; Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. In: Communications of the ACM, 21. Jg. (1978), Nr. 2, S. 120–126.
http://people.csail.mit.edu/rivest/Rsapaper.pdf

Robinson, S.: Still Guarding Secrets after Years of Attacks, RSA Earns Accolades for its Founders. In: SIAM News, 36. Jg. (2003), Nr. 5.
http://www.siam.org/pdf/news/326.pdf

RSA Laboratories: The RSA Factoring Challenge FAQ. 2012.
http://www.rsa.com/rsalabs/node.asp?id=2094

SAGE (freies Computer-Algebra-System):
http://www.sagemath.org/

SAGE Cell Server:
http://www.sagemath.org/eval.html

Shamir, A.: Factoring Large Numbers with the TWINKLE Device (Extended Abstract). In: Ç. K. Koç und Chr. Paar (Hrsg.): Cryptographic Hardware and Embedded Systems – First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999. Reihe „Lecture Notes in Computer Science", Band 1717. Berlin; Heidelberg: Springer, 1999, S. 2–12.
http://unina.stidue.net/Universita'%20di%20Genova/MoraF/ALTRO/twinkle.pdf

Shamir, A.; Tromer, E.: Factoring Large Numbers with the TWIRL Device. In: D. Boneh (Hrsg.): Advances in Cryptology – CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003. Reihe „Lecture Notes in Computer Science", Band 2729. Berlin; Heidelberg: Springer, 2003, S. 1–26.
Veröffentlichung:
http://cs.tau.ac.il/~tromer/papers/twirl.pdf
Animierte PowerPoint-Präsentation:
http://tau.ac.il/~tromer/slides/crypto03-twirl.ppt
PowerPoint-Präsentation ohne Animation (PDF):
http://tau.ac.il/~tromer/slides/crypto03-twirl.pdf

Schulz, R.-H.; Witten, H.: Zeit-Experimente zur Faktorisierung – Ein Beitrag zur Didaktik der Kryptologie. In: LOG IN, 30. Jg. (2010), Heft 166/167, S. 107–114.

Schulz, R.-H.; Witten, H.: Faktorisieren mit dem Quadratischen Sieb – Ein Beitrag zur Didaktik der Algebra und Kryptologie. In: LOG IN, 31. Jg. (2011/2012), Heft 171/172, S. 70–78
(in diesem Heft).

Vandersypen, L. M. K.; Steffen, M.; Breyta, G.; Yannoni, C. S.; Sherwood, M. H.; Chuang, I. L.: Experimental realization of Shor's factorizing algorithm using nuclear magnetic resonance. In: letters to nature, Bd. 414 (2001), 20./27. Dezember 2001, S. 883–887.
http://cryptome.org/shor-nature.pdf

Wikipedia – Stichwort „Decisional-Diffie-Hellman-Problem":
http://de.wikipedia.org/wiki/Decisional-Diffie-Hellman-Problem

Wikipedia – Stichwort „RSA-100":
http://en.wikipedia.org/wiki/RSA_numbers#RSA-100

Wikipedia – Stichwort „RSA-768":
http://en.wikipedia.org/wiki/RSA_numbers#RSA-768

Wikipedia – Stichwort „The Magic Words are Squeamish Ossifrage":
http://en.wikipedia.org/wiki/The_Magic_Words_are_Squeamish_Ossifrage

Wikipedia – Stichwort „TWINKLE":
http://en.wikipedia.org/wiki/TWINKLE

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

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

Witten, H.; Schulz, R.-H.: RSA & Co. in der Schule – Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge – Teil 3: RSA und die elementare Zahlentheorie. In: LOG IN, 28. Jg. (2008), Heft 152, S. 60–70.

Witten, H.; Schulz, R.-H.: RSA & Co. in der Schule – Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge – Teil 4: Gibt es genügend Primzahlen für RSA? In: LOG IN, 30. Jg. (2010a), Heft 163/164, S. 97–103.

Witten, H.; Schulz, R.-H.: RSA & Co. in der Schule – Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge – Teil 5: Der Miller-Rabin-Primzahltest oder: Falltüren für RSA mit Primzahlen aus Monte Carlo In: LOG IN, 30. Jg. (2010b), Heft 166/167, S. 92–106.

Alle Internetquellen wurden zuletzt am 31. August 2012 geprüft.