DuD
Datenschutz und Datensicherheit

Vorzüge und Grenzen des RSA-Verfahrens

Frank Bourseau, Dirk Fox, Christoph Thiel

Die Eignung des RSA-Verfahrens für digitale Signaturen gerät immer wieder in die Diskussion, z. B. im Zusammenhang mit der Veröffentlichung aktueller Faktorisierungserfolge. Ungeachtet dessen ist das RSA-Verfahren seit seiner Veröffentlichung vor mehr als 20 Jahren unangefochtener De-facto-Standard. Der Beitrag gibt eine aktuelle Analyse der Sicherheit, Vorzüge und Grenzen des Verfahrens.[1]

Dr. Frank Bourseau

dvg Hannover GmbH
Arbeitsschwerpunkt: Strategie IT-Sicherheit

E-Mail: frank.bourseau@dvg.de

Dipl.-Inform. Dirk Fox

Security Consultant und Geschäftsführer der Secorvo Security Consulting GmbH.
Arbeitsschwerpunkt: Public Key Infrastrukturen, Digitale Signaturen, Sicherheit in Netzen.
E-Mail: fox@secorvo.de

Dr. Christoph Thiel

SIZ – Informatikzentrum der Sparkassenorganisation GmbH
Arbeitsschwerpunkt: Biometrie, Kryptografie
E-Mail: christoph.thiel@siz.de

[1] Der Beitrag basiert auf den Ergebnissen einer für das Informatikzentrum der Sparkassenorganisation (SIZ) durchgeführten Studie.

Einleitung

Ronald Rivest, Adi Shamir und Leonard Adleman gelang im Jahre 1977 am MIT die Konstruktion des ersten (praktisch einsetzbaren) asymmetrischen kryptographischen Verfahrens [RiSA_78], das den von Whitfield Diffie und Martin Hellman in ihrem wegweisenden Veröffentlichung „New Directions in Cryptography“ geforderten Eigenschaften genügte [DiHe_76]. Dieses nach den Anfangsbuchstaben der Namen der Autoren RSA genannte Verfahren ist inzwischen zum de-facto-Standard in der asymmetrischen Kryptographie geworden.

Dass RSA heute das bekannteste und verbreitetste asymmetrische kryptographische Verfahren ist, liegt zum einen an der vergleichsweise einfachen, leicht verständlichen Struktur, zum anderen wohl auch an der von der Firma RSA Security, Inc., geförderten Integration von RSA in nationale und internationale Standards und Sicherheitslösungen.

So sind das RSA-Verschlüsselungsverfahren und das RSA-Signatursystem heute Teil der folgenden Standards und Spezifikationen:

- 1991: *Digital signature scheme giving message recovery*, Internationaler Standard ISO/IEC 9796, Anhang A [ISO_91]
- 1993: *Privacy Enhanced Mail* (PEM), RFC 1421-1424, vorgeschlagener Internet-Standard [Linn_93, Kent_93, Bale_93, Kali_93]
- 1996: *Pretty Good Privacy* (PGP), RFC 1991/2015, Internet-Spezifikationen [AtSZ_96, Elki_96]
- 1997: *Secure Electronic Transaction* (SET), Spezifikation von Visa und MasterCard [SET_97]
- 1997: *Domain Name System Security Extensions* (DNSSEC), RFC 2065, vorgeschlagener Internet-Standard [EaKa_97]
- 1997: *Electronic Data Interchange for administration, commerce and transport (EDIFACT)*, Internationaler Standard ISO 9735-5 [ISO_97]
- 1998: *OpenPGP*, RFC 2440, vorgeschlagener Internet-Standard [CDFT_98]
- 1998: *Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)*, X9.31, ANSI-Standard [ANSI_98]
- 1999: *Digital Signatures with Appendix – Part 3: Certificate-based mechanisms*, Internationaler Standard ISO/IEC 14888-3 [ISO_99]
- 1999: *Secure Sockets Layer* (SSL/TLS), RFC 2246, de-facto-Standard-Protokoll [DiAl_99]
- 1999: *S/MIME Version 3*, RFC 2632-2633, vorgeschlagener Internet-Standard [Rams_99, Rams2_99]
- 2000: *Digital Signature Standard* (DSS), FIPS 186-2, NIST-Standard [NIST_00]
- 2000: Public-Key and Attribute Certificate Frameworks, X.509, ITU-T Recommendation [ITU_00]
- 2000: *Homebanking-Computer-Interface* (HBCI), Version 2.2, Schnittstellenspezifikation des ZKA [ZKA_00]
- 2001: *XML-Signature Syntax and Processing*, RFC 3075, vorgeschlagener Internet-Standard [EaRS_01]
- 2001: *Empfehlung geeigneter Kryptoalgorithmen in Erfüllung der Anforderungen nach § 17 (1) SigG* (22.05.2001) in Verbindung mit § 17 (2) SigV (22.10.1997) des BSI [BSI_01]

Die Zahl der Spezifikationen, die die Verwendung von RSA als asymmetrisches Kryptoverfahren empfehlen oder sogar erfordern, hat seit dem Auslaufen des amerikanischen RSA-Patents No. 4.405.829 aus dem Jahr 1983 im September 2000, das von der Firma RSA Security, Inc. gehalten wurde, erkennbar zugenommen: Eine Zurückhaltung, das RSA-Verfahren in Standards zu integrieren, lässt sich seit Ende der 90er Jahre nicht mehr beobachten.

1 Sicherheit

Die Sicherheit des RSA-Verfahrens basiert – wie die aller kryptographischen Verfahren – auf den folgenden vier zentralen „Säulen“:

- der Komplexität des dem Verfahren zugrunde liegenden zahlentheoretischen Problems (hier: der Faktorisierung großer Zahlen),
- der Wahl geeigneter Sicherheitsparameter (hier: Länge des Moduls),
- der geeigneten Anwendung des Verfahrens sowie der Schlüsselerzeugung zum Schutz vor bekannten Angriffen und
- der korrekten Implementierung des Algorithmus.

Die beiden erstgenannten Punkte sind dabei grundsätzliche Eigenschaften des Verfahrens und unabhängig von einer konkreten Implementierung oder Anwendung. Sie sind entscheidend für die prinzipielle Eignung des RSA-Verfahrens und werden im folgenden einer aktuellen Analyse unterzogen.[2]

1.1 Komplexität

Die Vertraulichkeit RSA-verschlüsselter Daten und die Fälschungssicherheit von RSA-Signaturen beruhen auf der Schwierigkeit, zu einem Schlüsseltext c den korrespondierenden Klartext m mit

$$c = m^{ek} \bmod n$$

bzw. einen Wert *Sig* zu finden, so dass für ein bestimmtes m gilt:

$$Sig^{pk} \bmod n = m.$$

Dabei kann die Kenntnis des Moduls n und des jeweiligen öffentlichen Schlüssels (*ek* respektive *pk*) vorausgesetzt werden.

Mathematisch ausgedrückt erfordert ein erfolgreiches Entschlüsseln oder eine Signaturfälschung ohne Kenntnis des geheimen Schlüssels also, die *ek*-te bzw. *pk*-te Wurzel aus c bzw. m modulo n zu ziehen.

Da bekanntermaßen die zugehörigen geheimen Schlüssel, nämlich die multiplikativen Inversen zu *ek* und *pk* mit Kenntnis von $\phi(n)$ bestimmt werden können[3], gilt: Sind die Primfaktoren p und q von n bekannt, dann lässt sich der Klartext m bzw. eine Signatur *Sig* leicht berechnen. Könnte ein Angreifer also jedes Modul n hinreichend schnell faktorisieren, dann wäre das RSA-Verfahren vollständig gebrochen.

Daher kann das Brechen des RSA-Verfahrens nicht schwieriger sein als die Faktorisierung des Moduls n. Die Lösungskomplexität des Faktorisierungsproblems liegt mit dem besten heute bekannten Algorithmus, einer Weiterentwicklung des ursprünglich für Zahlen mit bestimmter Darstellung (z.B. Fermatzahlen) entwickelten *general number field sieve* (GNFS) von Pollard [Poll_93], asymptotisch bei[4] [LeLe_93, Silv_00]

$$o(l) = e^{(c+o(1))\cdot(l\cdot\ln 2)^{1/3}(\ln(l\cdot\ln 2))^{2/3}}$$

Die Formel zeigt, dass das Faktorisierungsproblem damit zur Komplexitätsklasse der Probleme mit subexponentieller Berechnungskomplexität zählt.

als sicher, dass das „RSA-Problem“ und das Faktorisierungsproblem komplexitätstheoretisch äquivalent sind [MeOV_96]. Die Einordnung in die Komplexitätsklasse für Probleme mit subexponentieller Berechnungskomplexität bedeutet jedoch nicht, dass das Faktorisierungsproblem nicht möglicherweise auch polynomiell gelöst werden kann. Zwar ist bis heute kein Lösungsalgorithmus mit (asymptotisch) polynomiellem Aufwand bekannt; dass kein solcher Algorithmus existiert, ist jedoch eine unbewiesene Annahme.

Jahr	Zahl	Stellen	Bits	Durch	Methode	MIPS-Jahre
1970	$2^{128}+1$	39	128	Brillhardt/Morrison	CFRAC	
1978	$2^{150}-1$	45	149	Wunderlich	CFRAC	0,001
1981	$3^{225}-1$	47	156	Gerver[1]	QS	
1982	$5^{91}-1$	51	169	Wagstaff	CFRAC	
1983	$11^{93}+1$	63	209	Davis/Holdridge	QS	
1984	$10^{71}-1$	71	236	Davis/Holdridge/ Simmons	QS	0,1
1986	$5^{128}+1$	87	289	Silverman	MPQS	
1987	$5^{160}+1$	90	299	Silverman	MPQS	
1988	$11^{104}+1$	100	332	Internet	MPQS	
1988		106	350	Lenstra/Manasse	QS	140
1990	$2^{484}+1$	111	369	Lenstra/Manasse	MPQS	
1991	RSA-100	100	332		QS	7
1991	$10^{142}+1$	116	384	Lenstra/Manasse	MPQS	
1992	RSA-110	110	365		QS	75
1993	RSA-120	120	399	Denny/Dodson/ Lenstra/Manasse	QS	825
1994	RSA-129	129	426	Atkins	MPQS	5.000
1995		119	395	Dodson/Lenstra	GNFS	250
1996	RSA-130	130	432	Lenstra	GNFS	750
1998	RSA-140	140	465	Montgomery	GNFS	2.000
1999	RSA-512	155	512	Montgomery	GNFS	8.000

Tabelle 1: Erfolgreiche Faktorisierungen großer Moduln

Die umgekehrte Aussage, dass das RSA-Verfahren ausschließlich durch eine Faktorisierung des Moduls n gebrochen werden kann, ist bis heute nicht bewiesen. Zwar kann gezeigt werden, dass sowohl das Finden einer passenden multiplikativen Inversen zum öffentlichen Schlüssel als auch die Bestimmung von $\phi(n)$ zugleich das Faktorisieren von n erlauben, also beide äquivalent zum Faktorisierungsproblem sind, nicht aber, dass es keinen anderen Weg zum Fälschen von RSA-Signaturen gibt. Allerdings gilt es unter Zahlentheoretikern

Die Sicherheit des RSA-Verfahrens beruht also auf den beiden unbewiesenen Annahmen, dass

- die Entschlüsselung bzw. die Bestimmung einer gültigen Signatur *Sig*, d.h. der *ek*-ten bzw. *pk*-ten Wurzel aus c bzw. m modulo n ohne Kenntnis von *dk* bzw. *sk* oder der Primfaktoren von n, äquivalent zum Faktorisierungsproblem ist, und
- das Faktorisierungsproblem eine hinreichend hohe Berechnungskomplexität besitzt.

1.2 Sicherheitsparameter

Die Komplexität des einem konkreten RSA-Schlüssels zugrundeliegenden Faktorisie-

[2] Zu technischen Anforderungen an die Implementierung des RSA-Signaturverfahrens siehe z.B. [Fox_97].

[3] Eulersche Phi-Funktion: $\phi(n) = (p-1)(q-1)$ mit $n = p \cdot q$.

[4] $o(l)$ gibt die Zahl der durchschnittlich erforderlichen Prozessoroperationen abhängig von der Länge l der zu faktorisierenden zusammengesetzten Zahl n an. Für den besten allgemeinen Faktorisierungsalgorithmus ist $c = (64/9)^{1/3} \approx 1{,}923$ [LeLe_93].

rungsaufwands wird im wesentlichen von der Länge *l* des Moduls *n* bestimmt. Daher gilt *l* als der Sicherheitsparameter des RSA-Verfahrens. Das Sicherheitsniveau eines Schlüssels kann daher anhand dieses Parameters festgelegt bzw. abgeschätzt werden.

1977 veröffentlichten Ronald Rivest, Adi Shamir und Leonard Adleman einen mit dem RSA-Algorithmus bezüglich einem 129-stelligen Modul (l = 426 bit) verschlüsselten Text und setzten ein Preisgeld von 100 US-$ auf die Entschlüsselung aus [Gard_77]. Mit den besten damals bekannten Algorithmen und einer Maschine mit einer – auch heute unrealistischen – Rechenleistung von 3 Mio. MIPS[5] schätzte Ronald Rivest die Faktorisierungszeit auf bis zu 4 Billiarden Jahren, das entspricht einem Berechnungsaufwand von $12 \cdot 10^{21}$ MIPS-Jahren. Dennoch empfahlen Rivest, Shamir und Adleman bereits 1978 eine Modullänge von mindestens 200 Dezimalstellen (l = 665 bit) für *n* [RiSA_78].

In den vergangenen zwanzig Jahren wurden Faktorisierungsalgorithmen jedoch erheblich verbessert (siehe Tabelle 1). Am 2. April 1994 gelang mit einer verteilten Implementierung des von Pomerance 1982 entwickelten *quadratic sieve*-Algorithmus (QS) [Pome_84] unter Nutzung von 1600 Workstations im Internet nach knapp acht Monaten die spektakuläre Faktorisierung des 1977 veröffentlichten 129stelligen Moduls. Die Rechenleistung summierte sich dabei auf etwa 5.000 MIPS-Jahre [AG LL_94]. Fünf Jahre später, am 22. August 1999, wurde unter Verwendung des asymptotisch ab ca. 116 Dezimalstellen effizienteren und von Buhler, Lenstra und Pomerance entwickelten *general number field sieve*-Algorithmus' (GNFS) [BuLP_93] auch ein 512 bit langer (154stelliger) Modul innerhalb von 5 Monaten faktorisiert – eine Modullänge, die noch wenige Jahre zuvor als sicher galt und in vielen Produkten eingesetzt wurde.

Diese Faktorisierung einer 512-Bit-Zahl nach wenig mehr als fünf Monaten parallelen Rechnens von 160 Sun Workstations und 120 Pentium-PCs zeigte, dass eine Modullänge von 512 keinen Schutz mehr vor Angreifern bietet, die über vergleichsweise große Rechenressourcen verfügen.

Konsequenterweise werden daher in der gemäß der deutschen Signaturverordnung (SigV) im Bundesanzeiger veröffentlichten Empfehlung geeigneter Algorithmen des Bundesamtes für Sicherheit in der Informationstechnik (BSI) vom 05.07.2001 bis Ende 2005 Modullängen von mindestens 1024 bit gefordert; anschließend werden 2048 bit als Länge des Moduls empfohlen [BSI_01].

Abschätzungen über die zukünftige Entwicklung der Sicherheit unterschiedlicher RSA-Modullängen sind allerdings immer von verschiedenen Annahmen abhängig und daher sehr häufig ungenau.

So zum Beispiel eine etwas ältere, sehr häufig zitierte Aufwandsabschätzung des GNFS aus dem Jahr 1995 für die Faktorisierung großer Modullängen von Odlyzko [Odly_95]: Dort findet sich u.a. eine Aufwandsschätzung für die Faktorisierung von

Modullänge	GNFS [MY]	SNFS [MY]
512 bit	$3 \cdot 10^4$	
768 bit	$2 \cdot 10^8$	$1 \cdot 10^5$
1024 bit	$3 \cdot 10^{11}$	$3 \cdot 10^7$
1280 bit	$1 \cdot 10^{14}$	$3 \cdot 10^9$
1536 bit	$3 \cdot 10^{16}$	$3 \cdot 10^{11}$
2048 bit	$3 \cdot 10^{20}$	$4 \cdot 10^{14}$

Tabelle 2: Geschätzter absoluter Aufwand der Faktorisierung verbreiteter Modullängen [Odly_95]

Modullänge	MIPS-Jahre	RAM Client[1]	RAM Server[1]
512 bit	$4{,}0 \cdot 10^5$	$1{,}3 \cdot 10^8$	$5{,}0 \cdot 10^{10}$
1024 bit	$3 \cdot 10^{12}$	$3 \cdot 10^{11}$	$2 \cdot 10^{14}$
2048 bit	$3 \cdot 10^{21}$	$8 \cdot 10^{15}$	$7 \cdot 10^{18}$
4096 bit	$2 \cdot 10^{33}$	$8 \cdot 10^{21}$	$7 \cdot 10^{24}$

Tabelle 3: Geschätzter Aufwand der Faktorisierung verbreiteter Modullängen nach [ANSI_98]

Zahlen mit spezieller Darstellung[6], wie z.B. Fermat-Zahlen, bei denen das erheblich effizientere *special number field sieve* (SNFS) angewendet werden kann (Tabelle 2). Gemäß dieser Abschätzung wäre nach der Faktorisierung der Fermatzahl $F_7 = 2^{128}+1$ im Jahr 1970, $F_8 = 2^{256}+1$ im Jahr 1980 und $F_9 = 2^{512}+1$ im Jahr 1990 eine Faktorisierung von $F_{10} = 2^{1024}+1$ im Jahr 2000 zu erwarten gewesen. Tatsächlich wurde im April 1999 die Zahl 10^{211}-1 (eine Zahl der Länge 698 bit) mit einer Rechenleistung von etwa $2 \cdot 10^3$ MIPS-Jahren faktorisiert.

Im ANSI-Standard X9.31 findet sich die in Tabelle 3 zusammengefasste Abschätzung des Rechen- und Speicheraufwands einer Faktorisierung verbreiteter Modullängen aus dem Jahr 1998, bei der der Rechenaufwand etwa um den Faktor fünf höher liegt als der tatsächlich im Jahr 1999 für die Faktorisierung von RSA-512 benötigte Aufwand [ANSI_98].

Zu etwas niedrigeren Schätzwerten kommen Lenstra und Verheul in ihrer Anfang 2000 veröffentichten Abschätzung (siehe Tabelle 4) [LeVe_00].

Eine der wesentlichen, einer solchen Abschätzung zugrundeliegenden Annahmen

Modullänge	MIPS-Jahre
488 bit	$2{,}46 \cdot 10^6$
777 bit	$5 \cdot 10^8$
1028 bit	$2{,}06 \cdot 10^{10}$
1562 bit	$1{,}21 \cdot 10^{13}$
2054 bit	$1{,}45 \cdot 10^{15}$

Tabelle 4: Abschätzung des Faktorisierungsaufwands nach [LeVe_00]

betrifft die für einen Angriff realistischerweise verfügbare Rechenleistung. Tabelle 5 gibt die von Odlyzko zugrundegelegte

Jahr	Leistung eines PC	Geheim. Angriff	Verteilter Angriff
2004	10^3 MIPS	10^8 MY	$2 \cdot 10^9$ MY
2014	10^4-10^5 MIPS	10^{10}-10^{11} MY	10^{11}-10^{13} MY

Tabelle 5: Abschätzung der verfügbaren Ressourcen für Faktorisierungsangriffe [Odly_95]

Schätzung der Entwicklung der für Angriffe auf einzelne Moduln einsetzbaren Rechenkapazität (in MIPS-Jahren, MY) im Jahr 1995 und – unter Annahme der Gültigkeit des Gesetzes von Moore: Verdoppelung der Rechenleistung alle 18 Monate – der Entwicklung der Rechenleistung wieder [Odly_95].

Für einen verteilten Angriff nahm Odlyzko dabei an, dass ca. 0,1% der weltweit verfügbaren ungenutzten und über das Internet erreichbaren Ressourcen für einen

[5] Zum Vergleich: Ein mit 200 MHz getakteter Pentium-Pro-Prozessor erreicht etwa 440 MIPS.

[6] Das SNFS erlaubt eine effiziente Faktorisierung von Zahlen, die sich in der Form r^e–s für kleine Werte r und s darstellen lassen. Fermat-Zahlen haben die spezielle Gestalt 2^e+1. Für das SNFS liegt die Konstante der Aufwandsformel bei $c = (32/9)^{1/3} \approx 1{,}526$.

Angriff gewonnen werden könnten.[7] Für spezielle Organisationen wie die NSA, die organisierte Kriminalität oder auch große Unternehmen schätzte er die für einen konzentrierten, geheimgehaltenen Angriff auf einen Modul verfügbare Rechenleistung auf etwa 1-10% der Ressourcen, die über das Internet konzentriert werden könnten.

Der 1999 tatsächlich für die Faktorisierung eines RSA-Moduls von 512 bit benötigte Aufwand lag um den Faktor vier unter dem 1995 von Odlyzko geschätzten Aufwand. Allerdings decken sich seine Abschätzungen mit jüngeren Zahlen, in denen das Verhältnis des Faktorisierungsaufwands größerer Schlüssellängen zur Faktorisierung des 512-bit-Moduls angegeben wird. So wird der Aufwand für die Faktorisierung heute empfohlener Schlüssellängen relativ zum Aufwand für die Faktorisierung einer 512 bit langen Zahl von Experten wie in Tabelle 6 angegeben geschätzt [Cont_99, Silv_00].

Modul-länge	Faktor Aufwand	Faktor RAM
512 bit	1	1
768 bit	ca. $6 \cdot 10^3$	ca. $8 \cdot 10$
1024 bit	ca. $7 \cdot 10^6$	$2{,}65 \cdot 10^3$
2048 bit	ca. $9 \cdot 10^{15}$	$9 \cdot 10^7$

Tabelle 6: Geschätzter Aufwand der Faktorisierung heutiger Modullängen relativ zu RSA-512

Schwierigkeiten bereitet bei der Faktorisierung solcher großen Zahlen vor allem die Lösungsmatrix, die sehr viel Arbeitsspeicher benötigt. Der Speicherbedarf wächst etwa quadratisch in der Modullänge *l*. Die Faktorisierung von 2048 bit langen Zahlen erfordert eine Lösungsmatrix, die den Adressraum heutiger 64-bit-Prozessoren übersteigt. Daher dürfte es für die heute bekannten Faktorisierungsalgorithmen innerhalb der kommenden 5-10 Jahre schwierig sein, überhaupt genügend Rechner zu finden, die für einen Faktorisierungsangriff auf 1024 bit lange Zahlen geeignet sind.

Unter der Annahme des Moore'schen Gesetzes würde mit derselben Rechneranzahl, mit der RSA-512 faktorisiert wurde, erst in etwa 34 Jahren die Zerlegung von RSA-1024 möglich sein. Zusätzlich müssen allerdings Verbesserungen in den Faktorisierungsalgorithmen und eine Zunahme der Gesamtrechnerkapazität berücksichtigt werden. Robert Silverman erwartet daher eine erfolgreiche Faktorisierung einer 768 bit langen Zahl frühestens in ca. 10 Jahren [Silv_00].

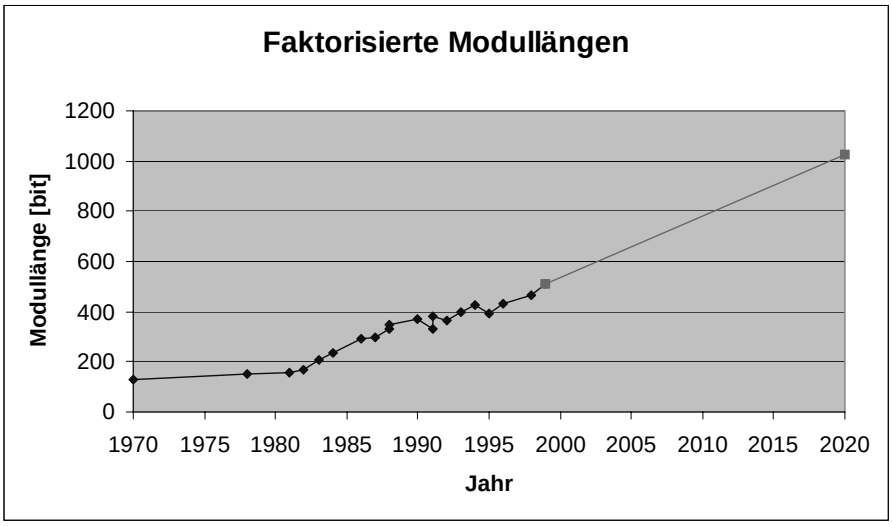

Bild 1: Prognostizierte Entwicklung der Faktorisierungserfolge 1970-2020 (Fox)

Prognostiziert man die Entwicklung der Faktorisierungserfolge ausgehend von der Entwicklung der vergangenen 20 Jahre, ergibt sich die in Bild 1 dargestellte Erwartung: 1024 bit lange RSA-Schlüssel ließen sich danach frühestens im Jahr 2020 mit einer erheblichen Konzentration verteilter Rechenleistung über einen längeren, der RSA-512-Faktorisierung vergleichbaren Zeitraum realisieren. Bei dieser Prognose wird implizit angenommen, dass sowohl die Entwicklung der Rechenleistungen als auch die Fortschritte bei der Entwicklung neuer Faktorisierungsalgorithmen in etwa der selben Entwicklungsgeschwindigkeit verläuft wie in den vergangenen 20 Jahren. Diese Prognose erscheint vor dem Hintergrund der bisherigen überraschend linearen Entwicklung sehr realistisch.

2 Vorzüge

Zweifellos besitzt das RSA-Verfahren eine große Zahl von Vorzügen, die wesentlich zu der heutigen Bedeutung und weiten Verbreitung des Verfahrens beigetragen haben. Die wichtigsten dieser Vorzüge sind die folgenden:

- **Eignung für asymmetrische Verschlüsselung und digitale Signaturen**: Der RSA-Algorithmus selbst kann sowohl zum Verschlüsseln als auch für die Erzeugung und Prüfung digitaler Signaturen eingesetzt werden. Es genügt daher, ein einziges Verfahren zu implementieren und zu testen, um einer Anwendung beide Funktionen zur Verfügung zu stellen.
- **Integration in technische Standards**: Fast alle wichtigen Standards und Spezifikationen für Protokolle und Sicherheitsmechanismen, die asymmetrische kryptographische Verfahren nutzen, enthalten heute das RSA-Verfahren. Diese Spezifikationen umfassen insbesondere Formate und Austauschformate für Schlüssel, verschlüsselte Daten sowie OIDs. Das sorgt für eine sehr große Interoperabilität kryptographischer Funktionen auf der Basis des RSA-Verfahrens.
- **Große Verbreitung**:
  Mehr als 20 Jahre nach seiner Entwicklung ist RSA de facto zum Standard-Verfahren der asymmetrischen Kryptographie geworden. In nahezu allen Lösungen und Produkten, die asymmetrische Verfahren verwenden, ist RSA implementiert. Inzwischen sind auch eine große Zahl von Hardware-Beschleunigern für RSA verfügbar sowie Smartcards, die RSA-Operationen mit Schlüssellängen bis 1024 bit durchführen können.
- **Patentsituation**:
  Nach Ablauf des amerikanischen Patents auf den RSA-Algorithmus im September 2000 werden von der Firma RSA Security, Inc. (USA) für die Integration des RSA-Algorithmus in auf dem amerikanischen Markt vertriebene Produkte keine Lizenzgebühren mehr erhoben. Das wird

[7] Die Faktorisierung von des 512-bit-RSA-Moduls konnte ca. 0,03% der im Internet verfügbaren Ressourcen erreichen.

die Verbreitung des Verfahrens auch weiterhin befördern.

- **Einsatz mit *message recovery* möglich**: Das RSA-Signatursystem kann bei kurzen Nachrichten (kürzer als Modul *n*) nicht nur als Signatursystem *with appendix* (Signatur als Nachrichtenanhang), sondern auch mit *message recovery* eingesetzt werden: Die Verifikationsfunktion liefert als Nebenergebnis die originale Nachricht – sofern diese nicht zuvor durch eine Einweg-Hashfunktion auf einen Hashwert hash(*m*) abgebildet wurde. Auf diese Weise kann die Verlängerung einer Nachricht durch die Signatur auf wenige Bits beschränkt werden, eine Eigenschaft, die vor allem für Bitströme von besonderer Bedeutung ist.
- **Implementierung**:
  Das RSA-Verfahren lässt sich auf der Basis einer Langzahlarithmetik sehr einfach implementieren. Ein wichtiger Vorteil ist, dass die Verschlüsselungs- und die Signieroperation keine Zufallszahlen benötigen; lediglich bei der Schlüsselgenerierung muss eine zufällige Wahl der Schlüsselparameter erfolgen.
- **Berechnungseffizienz**:
  Beim RSA-Verfahren können insbesondere die Operationen mit öffentlichen Schlüsseln stark beschleunigt werden, indem kurze Schlüssel mit wenigen 1-Bits (z.B. die vierte Fermat-Zahl $2^{16}+1$) verwendet werden. Dies ist besonders bei der Verwendung von RSA als digitales Signatursystem von Bedeutung, da Signaturen unter einem Dokument oder einer Nachricht in der Regel erheblich häufiger geprüft als erzeugt werden. Auch für die Nutzung von RSA in *Personal Digital Assistents* (PDA) mit geringen Rechen- und Speicherressourcen, wie Smartcards oder Mobilfunkendgeräte, sind effiziente Operationen mit öffentlichen Schlüsseln sehr wichtig.
- **Sicherheit**:
  Bis heute ist das Verfahren trotz der erheblichen Fortschritte bei Faktorisierungsalgorithmen nicht gefährdet. Selbst unter der Annahme, dass die Faktorisierungsalgorithmen eine ähnliche Entwicklung wie in den vergangenen 20 Jahren nehmen, erscheint eine Schlüssellänge von 1024 bit für die kommenden 20 Jahre hinreichend. Obwohl bislang kein Äquivalenzbeweis für das Brechen von RSA und die Faktorisierung des Moduls geführt werden konnte, wird die Äquivalenz des „RSA-Problems“ mit dem Faktorisierungsproblem von Zahlentheoretikern angenommen. Bis heute ist kein effizientes Verfahren für einen systematischen Angriff auf das RSA-Verfahren bekannt.

3 Grenzen

Allerdings unterliegt das RSA-Verfahren auch einer Reihe von Einschränkungen. Die wichtigsten dieser Begrenzungen sind die folgenden:

- **Berechnungseffizienz**:
  Optimierungsmöglichkeiten für die Berechnung einer modularen Exponentiation, des hinsichtlich des Berechnungsaufwands entscheidenden Teils des RSA-Algorithmus, sind ausgereizt. Durch die Verwendung kürzerer Exponenten können zwar die Verschlüsselung und die Prüfung einer RSA-Signatur beschleunigt werden; es gibt aber keinen Beweis, dass dadurch die Sicherheit des Verfahrens nicht beeinträchtigt wird. Zwar ist das Verfahren für viele Anwendungen Client-seitig hinreichend effizient; Server-basierte Anwendungen wie beispielsweise E-Commerce-Applikationen oder auch Sicherheitsprotokolle wie DNSSEC stellen jedoch sehr hohe Anforderungen an die Bandbreite der RSA-Berechnungen, die sich häufig nur unter Einsatz von Spezialhardware und oft nur aufwändig realisieren lassen. Der Rechenaufwand für die Berechnung der RSA-Operationen steigt etwa quadratisch in der Schlüssellänge; d.h. die Verwendung größerer Moduln verringert die Bandbreite der RSA-Berechnung spürbar.
- **Länge von RSA-Signaturen und Verschlüsselungsblöcken**:
  RSA-Signaturen und RSA-verschlüsselte Datenblöcke sind ebenso lang wie der verwendete Modul. Mit einer Schlüssellänge von 1024 bit benötigt jede digitale RSA-Signatur und jeder RSA-verschlüsselte Nachrichtenschlüssel 128 Byte. Dabei wäre für digitale Signaturen (Schutz vor Kollisionsangriffen nach dem „Geburtstags-Paradoxon“[8]) derzeit eine Signaturlänge von 20 Byte ausreichend. Beim Einsatz von RSA zur Verschlüsselung von Nachrichtenschlüsseln wie z.B. eines sieben Byte langen DES- oder eines 16 Byte langen AES-Schlüssels entsteht eine Redundanz von 85% bis 95%.
- **Sicherheit**:
  Zwar ist bis heute kein effizienter Angriff auf RSA bekannt; auch gilt es unter Experten als sehr unwahrscheinlich, dass ein solcher Angriff gefunden werden kann. Allerdings kann er mathematisch nicht ausgeschlossen werden, da die Äquivalenz von Faktorisierung und dem Brechen des RSA-Verfahrens nicht nachgewiesen ist. Auch die Entwicklung der Faktorisierungsalgorithmen könnte einen qualitativen Sprung machen: Gelänge es, eine erheblich effizienteren Lösungsalgorithmus zu finden, würde das die Sicherheit des RSA-Verfahrens erheblich beeinträchtigen.

[8] Siehe Fox, Gateway, DuD 11/2001, S. 682.

Fazit

Das RSA-Verfahren ist heute der De-Facto-Standard für asymmetrische Kryptoverfahren. Daran wird sich auch in den kommenden zehn Jahren voraussichtlich wenig ändern. Auch wenn es bis heute keine Indizien für eine Unsicherheit des Verfahrens gibt, ist allerdings allein die Tatsache, dass es zum RSA-Verfahren keine etablierte Alternative gibt, beunruhigend. Eine Kompromittierung des RSA-Verfahrens würde schlagartig nahezu alle Lösungen gefährden, deren Sicherheit auf der Verwendung asymmetrischer Kryptographie beruht. Zur Beseitigung dieses „Single Point of Failure“ ist daher die Suche nach und die Weiterentwicklung von alternativen kryptographischen Ansätzen unverzichtbar und dringlich.

Literatur

AGLL_94 Atkins, Derek; Graff, Michael; Lenstra, Arjen K.; Leyland, Paul C.: *The magic words are squeamish ossifrage*. In: Pieprzyk, J.; Safavi-Naini, R. (Hrsg.): Proceedings of Asiacrypt ‘94, LNCS 917, Springer, Berlin 1995, S. 263-277.

ANSI_98 American National Standards Institute (ANSI): *Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA).* ANSI X9.31, 1998.

AtSZ_96 Atkins, D.; Stallings, W.; Zimmermann, P.: *PGP Message Exchange Formats*, Request for Comments (RFC) 1991, August 1996.

Bale_93 Balenson, D.: *Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes and Identifiers.* Request

for Comments (RFC) 1423, Februar 1993.

BSI_01 Bundesamt für Sicherheit in der Informationstechnik (BSI): *Geeignete Kryptoalgorithmen in Erfüllung der Anforderungen nach § 17 (1) SigG vom 22. Mai 2001 in Verbindung mit § 17 (2) SigV vom 22. Oktober 1997*, vom 05.07.2001.

BuLP_93 Buhler, J.P.; Lenstra, H.W.; Pomerance, C.: *Factoring integers with the number field sieve*. In: Lenstra, A.K.; Lenstra, H.W. (Hrsg.): The Development of the Number Field Sieve. Lecture Notes in Mathematics, Vol. 1554, Springer, Heidelberg 1993, S. 50-94.

CDFT_98 Callas, J.; Donnerhacke, Lutz; Finney, H.; Thayer, R.: *OpenPGP Message Format*. Request for Comments (RFC) 2440, Standards Track, November 1998.

Cont_99 Contini, Scott: *The Factorisation of RSA-140*. RSA Laboratories' Bulletin, No. 10, 08.03.1999, S. 1-2.

DiAl_99 Dierks, T.; Allen, C.: *The TLS Protocol, Version 1.0*. Request for Comments (RFC) 2246, Standards Track, January 1999.

DiHe_76 Diffie, Whitfield; Hellman, Martin E.: *New Directions in Cryptography*. IEEE Transactions on Information Theory, Vol. IT-22, No. 6, 1976, S. 644-654.

EaKa_97 Eastlake, D.; Kaufman, C.: *Domain Name System Security Extensions*. Request for Comments (RFC) 2065, Standards Track, January 1997.

EaRS_01 Eastlake, D.; Reagle, J.; Solo, D.: *XML-Signature Syntax and Processing*, Request for Comments (RFC) 3075, Standards Track, March 2001.

Elki_96 Elkins, M.: *MIME Security with Pretty Good Privacy (PGP)*, Request for Comments (RFC) 2015, Oktober 1996.

Fox_97 Fox, Dirk: *Fälschungssicherheit digitaler Signaturen*. Datenschutz und Datensicherheit (DuD), 2/1997, S. 69-74.

Gard_77 Gardner, Martin: *Mathematical games. A new kind of cipher that would take millions of years to break*. Scientific American, August 1977, S. 120-124.

ISO_91 International Organization for Standardization (ISO): *Information technology – Security techniques – Digital signature scheme giving message recovery*. International Standard ISO/IEC 9796, Genf 1991.

ISO_97 International Organization for Standardization (ISO): *Electronic Data Interchange for administration, commerce and transport (EDIFACT) – Application level syntax rules. Part 5: Security Rules for Batch EDI*. ISO 9735-5, Genf 1997.

ISO_99 International Organization for Standardization (ISO): *Information technology – Security techniques – Digital Signatures with Appendix – Part 3: Certificate-based mechanisms.* International Standard ISO/IEC 14888-3, Genf 1999.

ITU_00 International Telecommunication Union: *Information Technology – Open Systems Interconnection – The Directory: Public-Key and Attribute Certificate Frameworks*, ITU-T Recommendation X.509 (2000)

Kali_93 Kaliski, Burt: *Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services*. Request for Comments (RFC) 1424, Februar 1993.

Kent_93 Kent, Stephen T.: *Privacy Enhancement for Internet Electronic Mail. Part II: Certificate-Based Key Management*. Request for Comments (RFC) 1422, Februar 1993.

LeLe_93 Lenstra, A.; Lenstra, H.: *The Development of the Number Field Sieve*. Lecture Notes in Mathematics 1554, Springer, New York 1993.

LeVe_00 Lenstra, Arjen K.; Verheul, Eric: *Selecting Cryptographic Key Sizes*. Datenschutz und Datensicherheit (DuD), 3/2000, S. 166.

Linn_93 Linn, John: *Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures*. Request for Comments (RFC) 1421, Februar 1993.

MeOV_96 Menezes, A.J.; van Oorschot, P.C.; Vanstone, S.A.: *Handbook of Applied Cryptography*. CRC Press, 1996.

NIST_00 National Institute of Standards and Technology (NIST): *Digital Signature Standard (DSS)*. Federal Information Processing Standards Publication 186-2 (FIPS-PUB), 27.01.2000.

Odly_95 Odlyzko, Andrew M.: *The Future of Integer Factorisation*. Cryptobytes, Summer 1995, Vol. 1, No. 2, S. 5-12.

Poll_93 Pollard, J. M.: *The Development of the Number Field Siev*e. In: Lenstra, Arjen K.; Lenstra, H.W. (Hrsg.): Factoring with cubic integers. Lecture Notes in Mathematics, 1554, Springer-Verlag, 1993, S. 4-10.

Pome_84 Pomerance, C.: *The quadratic sieve factoring algorithm*. In: Blakley, G.R.; Chaum, D. (Hrsg.): Proceedings of Crypto '84, LNCS 196, Springer, Berlin 1995, S. 169-182.

Rams_99 Ramsdell, Blake: *S/MIME Version 3 Certificate Handling*. Request for Comments (RFC) 2632, June 1999.

Rams2_99 Ramsdell, Blake: S*/MIME Version 3 Message Specification*. Request for Comments (RFC) 2633, June 1999.

RiSA_78 Rivest, Ronald L.; Shamir, Adi; Adleman, Leonard: *A Method for obtaining Digital Signatures and Public Key Cryptosystems*. Communications of the ACM, Vol. 21, No. 2, 1978, S. 120-126.

SET_97 *SET Secure Electronic Transaction Specification*. Specification by Visa and MasterCard. Book 1-3, Version 1.0, 31.05.1997.

Silv_00 Silverman, Robert D.: *A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths*. In: RSA Laboratories Bulletin, No. 13, April 2000, S. 1-22.

ZKA_00 Zentraler Kreditausschuß (ZKA): *Homebanking-Computer-Interface (HBCI)*, Schnittstellenspezifikation, Version 2.2, 10.05.2000.