

2.4 Faktorisierungsalgorithmen

Wie schnell kann man große Zahlen faktorisieren?

- Es gibt „schnelle“ Faktorisierungsverfahren für Zahlen der Gestalt $a^b \pm c$ mit „kleinen“ Werten a und c , z. B. für die MERSENNE- und FERMAT-Zahlen $2^b \pm 1$. Die Wahrscheinlichkeit, bei der Erzeugung von RSA-Schlüsseln aus zufällig gewählten Primzahlen einen solchen Modul zu konstruieren, ist verschwindend gering und wird gewöhnlich vernachlässigt.
- Die FERMAT-Faktorisierung von n : Man sucht eine Zahl $a \geq \sqrt{n}$, so dass $a^2 - n$ eine Quadratzahl $= b^2$ ist. Dann ist

$$n = a^2 - b^2 = (a + b)(a - b)$$

faktorisiert. [Beispiel: $n = 97343$, $\sqrt{n} \approx 311.998$, $312^2 - n = 1$, $n = 313 \cdot 311$.] Diese Methode ist effizient, wenn a nahe bei \sqrt{n} gefunden wird, also $a^2 \approx n$ ist, also im Fall $n = pq$, wenn die Differenz $|p - q|$ der Faktoren klein ist.

- Die schnellsten allgemein anwendbaren Faktorisierungsverfahren –
 - Zahlkörpersieb (SILVERMAN 1987, POMERANCE 1988, A. K. LENSTRA/ H. W. LENSTRA/ MANASSE/ POLLARD 1990),
 - Elliptische-Kurven-Faktorisierung (H. W. LENSTRA 1987, ATKIN/ MORAIN 1993), –

haben einen Zeitaufwand der Größenordnung

$$L_n := e^{\sqrt[3]{\ln n \cdot (\ln \ln n)^2}},$$

sind also „subexponentiell“, aber noch „superpolynomiell“. *Insbesondere ist die Faktorisierung als Angriff auf das RSA-Verfahren wesentlich effizienter als die vollständige Suche.*

Daraus ergeben sich folgende Schätzungen:

Zahl	Bits	Dezimalstellen	Aufwand (MIPS-Jahre)	Status
rsa120	399	120	100	auf PC < 1 Woche
rsa154	512	154	100 000	TE RIELE 1999
rsa200	665	200	(*)	FRANKE 2005
	1024	308	10^{11}	nicht mehr sicher
	2048	616	10^{15}	kurzfristig sicher

(*) mit 80 Rechnern à 2.2 GHz in 4.5 Monaten

Bei der Extrapolation der Aufwandsschätzungen ist zu beachten:

- Sie sind mit großer Unsicherheit behaftet,
- sie gelten nur, solange keine wesentlich schnelleren Faktorisierungsalgorithmen gefunden werden.

Insbesondere ist *bisher nicht bewiesen*, dass es nicht vielleicht doch einen polynomiellen Faktorisierungsalgorithmus gibt.

Neuere Entwicklungen (in der obigen Tabelle schon berücksichtigt) sind:

- Ein Artikel von A. K. LENSTRA/ E. VERHEUL, *Selecting cryptographic key sizes*, der den Stand der Technik im Jahr 2000 zusammenfasst und extrapoliert. Die Abschätzungen wurden als zu pessimistisch bewertet, da der Speicherbedarf der Verfahren nicht berücksichtigt wurde.
- Ein Vorschlag von BERNSTEIN, *Circuits for integer factorization*, der die Stellenzahl (!) verdreifacht, die bei festem Aufwand mit dem schnellsten Faktorisierungsverfahren erreichbar ist.
- Spezielle Hardware-Designs von SHAMIR und Mitarbeitern:
 - TWINKLE (The WEIZMANN Institute Key Locating Machine) – die Hardware-Umsetzung einer Idee von LEHMER aus den dreißiger Jahren, die das Faktorisieren um den Faktor 100 – 1000 beschleunigt (1999),
 - TWIRL (The WEIZMANN Institute Relation Locator) – der das Faktorisieren um einen weiteren Faktor 1000 – 10000 beschleunigt (2003) unter Berücksichtigung des Vorschlags von BERNSTEIN,also insgesamt ein Faktor etwa 10^6 (oder 2^{20}) für das Zahlkörpersieb, ohne allerdings die Größenordnung L_n der Komplexität zu ändern.

Insbesondere sieht die Abschätzung von LENSTRA/ VERHEUL jetzt eher zu optimistisch aus. *1024-Bit-Schlüssel sollten schnellstens aus dem Verkehr gezogen werden*. 2048-Bit-Schlüssel sind gerade noch für ein paar Jahre als sicher zu betrachten.

Empfehlung: Die Primzahlen p und q , aus denen der RSA-Modul $n = pq$ konstruiert wird, sollen mit mindestens 1024 Bit Länge gewählt werden, und zwar so, dass auch ihre Differenz $|p - q|$ eine Länge in der Größenordnung 1024 Bit hat.