2.4 Factoring Algorithms

A crucial question for the security of RSA is: *How fast can we factorize large integers?*

- There are "fast" factoring algorithms for integers of the form $a^b \pm c$ with "small" values a and c, the most prominent examples are the Mersenne and Fermat primes $2^b \pm 1$. The probability that the generation of RSA keys from random primes yields such a module is extremely low and usually neglected.

- Fermat factoring of n: Find an integer $a \geq \sqrt{n}$ such that $a^2 - n$ is a square $= b^2$. This yields a decomposition

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

  Example: $n = 97343$, $\sqrt{n} \approx 311.998$, $312^2 - n = 1$, $n = 313 \cdot 311$. This method is efficient provided that we find an a close to $\sqrt{n}$, or $a^2 \approx n$. In the case $n = pq$ of two factors this means a small difference $|p - q|$. (Un-) fortunately finding a seems to be hard.

- The fastest general purpose factoring algorithms

  - number field sieve (Silverman 1987, Pomerance 1988, A. K. Lenstra/ H. W. Lenstra/ Manasse/ Pollard 1990),
  - elliptic curve factoring (H. W. Lenstra 1987, Atkin/ Morain 1993),

  need time of size

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

  hence are "subexponential", but also "superpolynomial". Anyway they show that *factoring is a significantly more efficient attack on RSA than exhaustion ("brute force").*

This results in the following estimates for factoring times:

integer	bits	decimal places	expense (MIPS years)	status
rsa120	399	120	100	<1 weak on a PC
rsa154	512	154	100 000	te Riele 1999
rsa200	665	200	$(*)$	Franke 2005
	1024	308	10^{11}	insecure
	2048	616	10^{15}	for short-term security

$(*)$ 80 CPUs à 2.2 GHz in 4.5 months

When we extrapolate these estimates we should note:

- they are rough approximations only,
- they hold only as long as nobody finds significantly faster factoring algorithms.

Remember that the existence of a polynomial factoring algorithm is an *open problem*.

Some recent developments are already incorporated into the table:

- A paper by A. K. LENSTRA/ E. VERHEUL, *Selecting cryptographic key sizes* summarizes the state of the art in the year 2000 and extrapolates it.
- A proposal by BERNSTEIN, *Circuits for integer factorization* triples (!) the length of integers that can be factorized with a given expense, using the fastest factoring algorithms.
- Special-purpose hardware designs by SHAMIR and his collaborators:
  - TWINKLE (The WEIZMANN Institute Key Locating Machine) (1999) is the realization in hardware of an idea by LEHMER from the 1930s that accelerates factoring 100–1000 times,
  - TWIRL (The WEIZMANN Institute Relation Locator) (2003) accelerates factoring another 1000–10000 times following BERNSTEIN's idea.

  Taken together these approaches make factoring 10^6 (or 2^{20}) times faster using the number field sieve. However the order of magnitude L_n of the complexity is unaffected.

This progress lets the LENSTRA/ VERHEUL estimates look overly optimistic. 1024-*bit keys should no longer be used.* 2048-bit keys might be secure enough to protect information for a few years.

Recommendation: Construct your RSA module $n = pq$ from primes p and q that have bit lengths of at least 2048 bits, and choose them such that also their difference $|p - q|$ has a bit length of about 2048 bits.