Prime Factorization

Anna Friesen

Ladislaus von Bortkiewicz Chair of Statistics
Humboldt–Universität zu Berlin
http://lvb.wiwi.hu-berlin.de

Motivation

- RSA cryptosystem
- published in 1977 at the MIT
- by Ron **R**ivest, Adi **S**hamir and Leonard **A**delman
- algorithm is based on two pairs of keys
  - first pair (N, E) to encrypt data
  - second pair (N, D) to decrypt the data
- N is product of two large primes p, q $(\log(i) > 232, i = p, q)$
- N cannot be factored by now (computationally intractable for classical (non-quantum) computers)
- using quantum computers, Shor's algorithm could factor N
- RSA ist still used to encrypt (still secure)

The RSA algorithm:

- let OI be the original integer to be encrypted
- or transfer your text into an integer (e. g. number in alphabet)

1. choose two random primes p, q
   - $p \cdot q = N > OI$ and $p \neq q$ and $| \, p - q \, |$ not too small
2. choose E (E prime suffices all conditions)
   - $E \in \{n \in \mathbb{N} \mid 2 \nmid n\}$ and $E \nmid N$ and $E > (p-1) \cdot (q-1)$
3. choose D
   - $(E \cdot D) mod((p-1) \cdot (q-1)) = 1$

How to use the pairs of keys:

- $OI \xrightarrow[\text{encrypt}]{} CI : CI = OI^E mod(N)$
- $CI \xrightarrow[\text{decrypt}]{} OI : OI = CI^D mod(N)$

Trial Division

- the simplest (and most simple-minded) prime factorization algorithm
- assume n is not prime, test for $2 \leq i \leq \lfloor \sqrt{n} \rfloor$ whether $i \mid n$
- works quite well, because most coposite numbers have small prime factors
- inefficient if n has large prime factors
  - **efficient** algorithm has **polynomial** running time $f(\log_2(n))$, i.e. $f(\log_2(n)) \in O((\log_2(n))^k)$ mit $k \geq 0$
- modifications:
  - choose $i \in \{k \in \mathbb{N} \mid 2 \nmid k\}$
  - choose fixed bound B: find $i \leq B$ (often $B = 10^6$)
  - test primes $i \leq \lfloor \sqrt{r} \rfloor$ (sieve of Eratosthenes)

Sieve of Eratosthenes

- named by the greek mathematician Eratosthenes of Cyrene (3rd century BC)
- invented the name *sieve* for known algorithm
- deterministic algorithm
- creates list of primes that are smaller than or eqal to a given integer $r \geq 2$
- running time $\log_2(r-1) \cdot (r-1)$ is *exponentional*

Modern factorization algorithms

- devided into two groups
  - special purpose algorithms
    - efficiency depends on factors of number being factored
    - e.g. Pollard (p-1) factorization method
    - not dangerous to RSA (RSA uses large primes)
  - general purpose algorithms
    - efficiency depends only on number being factored
    - e.g. general number field sieve

Pollard (p-1) factorization method

- based on Fermat´s little theorem

Theorem (Fermat´s little theorem)

Let p be prime ($p \in \mathbb{P}$) and $a \in \mathbb{Z}$ with $a < p$. Then it holds:

$$p \in \mathbb{P} \;\Rightarrow\; a^p \;\equiv\; a \,(mod\, p)$$

If additionally a and p have no common divisor, i.e. $\gcd(p, a) = 1$, *then it even holds:*

$$p \in \mathbb{P} \;\Rightarrow\; a^{p-1} \;\equiv\; 1 \,(mod\, p)$$

- cannot be used as a primality test
- $\Leftarrow$ does not hold

- invented by John Pollard in 1974
- is a derivative of the Pollard rho method
- suitable for n that is B-smooth for $p-1$
  - A smooth (or friable) number is an integer which factors completely into small prime numbers. For example, a 7-smooth number is a number whose prime factors are all at most 7.
- uses that for any a and $\forall p \in \mathbb{P} : a^{p-1} \equiv 1 \pmod{n}$, i.e. $a^{p-1} - 1 \equiv 0 \pmod{n}$

algorithm of Pollard´s rho method:

1. pick two random numbers: $x \pmod{n}$ and $y \pmod{n}$
2. If $x - y = 0 \pmod{n}$ we found a factor $gcd(x-y, n)$, else go to step 1

Algorithm of the Pollard (p-1) factorization algorithm

1. Input: $n \geq 2$
2. choose a with $1 \leq a \leq n - 1$ and arbitrary $B \in \mathbb{N}$
3. $\forall q \in \mathbb{P}, q \leq B$:
   - $a := a^q \pmod{n}$
   - $p := gcd(a - 1, n)$
   - if $p \mid n$ break
   - else select new a and go to step 3
4. return p

RSA-challenge

- in 2009 RSA-768 was factored over the span of two years
- it is 768 bits and 232 decimal digits of size
- it is the largest solved RSA-challenge so far
- calculated with the general number field sieve

RSA-768 =
33478071698956898786044169848212690817704794983
71376856891243138898288379387800228761471165253174
3087737814467999489
x
36746043666799590428244633799627952632279158164
34308764267603228381573966651127923337341714339681
02700927987363089I7

For Further Reading

- Lasse Rempe, Rebecca Waldecker
  Primzahltests für Einsteiger
  1. Auflage, Vieweg+Teubner Verlag, Wiesbaden

- Martin Dietzfelbinger
  Primality Testing in Polynomial Time - From Randomized Algorithms to "‘PRIMES is in P"’
  1.Auflage, Springer Verlag, Berlin Heidelberg

- Richard P. Brent
  Recent Progress and Prospects for Integer Factorisation Algorithms

For Further Reading

Thorsten Kleinjung and Kazumaro Aoki and Jens Franke and Arjen Lenstra and Emmanuel Thomé and Joppe Bos and Pierrick Gaudry and Alexander Kruppa and Peter Montgomery and Dag Arne Osvik and Herman te Riele and Andrey Timofeev and Paul Zimmermann
Factorization of a 768-bit RSA modulus
Cryptology ePrint Archive, Report 2010/006
available on , 2018