Introduction to Cryptology

13.1 - Factorisation Algorithms

Federico Pintore

Mathematical Institute, University of Oxford (UK)

UNIVERSITY OF OXFORD

Michaelmas term 2020

Integer factorization

Problem: Given the product N of two n-bit primes, compute (one of) its factors.

Trial Divison: try every prime number up to $\sqrt{N}$. Worst-case complexity is $O(\sqrt{N} \cdot \text{polylog}(N))$.

Integer factorization

Problem: Given the product N of two n-bit primes, compute (one of) its factors.

Trial Divison: try every prime number up to $\sqrt{N}$. Worst-case complexity is $O(\sqrt{N} \cdot \mathrm{polylog}(N))$.

Anything better?

Pollard's Rho Algorithm

It is a general purpose algorithm.

It aims at obtaining a pair (x, y) s.t. $x = y \pmod{p}$ but $x \neq y \pmod{N}$, since $\gcd(x - y, N) = p$.

Pollard's Rho Algorithm

It is a general purpose algorithm.

It aims at obtaining a pair (x, y) s.t. $x = y \pmod{p}$ but $x \neq y \pmod{N}$, since $\gcd(x - y, N) = p$.

- Define some "pseudorandom" iteration function f such that, if $x = x' \pmod{p}$, then $f(x) = f(x') \pmod{p}$.
- A standard choice would be $f(x) = x^2 + 1 \pmod{N}$.
- Approximately $\sqrt{p}$ congruence classes are obtained computing iteratively f.

Pollard's Rho Algorithm

Input: integer N (a product of two n-bit primes)

$x \leftarrow \mathbb{Z}_N^*$

$x' := x$

for $i = 2, \cdots 2^{n/2}$ do

  $x := f(x)$

  $x' := f(f(x'))$

  $p := \gcd(x - x', N)$

  if $p \notin \{1, N\}$

    return p

Pollard's $p-1$ and the EC factorisation method

Pollard's $p-1$ is an effective method if $p-1$ is *smooth*, e.i. it has only "small" prime factors.

The Elliptic-curve factorisation method generalises it when neither $p-1$ nor $q-1$ are smooth.

The order of the group $\#E(\mathbb{Z}_p)$, where E is an elliptic curve, can be smooth even when $p-1$ is not.

Quadratic Sieve Algorithm

It runs in time sub-exponential in $\| N \|$ $(2^{o(\|N\|)})$.

It is a good choice for numbers up to about 300 bits.

It aims at finding a, b s.t. $a^2 = b^2 \pmod N$ but $a \neq \pm b \pmod N$, since $\gcd(a-b, N)$ gives a non trivial factor of N.

<u>Example</u>: $8051 = 90^2 - 7^2 = (90-7)(90+7) = 83 \times 97$.

Quadratic Sieve Algorithm

- Fix some bound $B \in \mathbb{N}$, and let $\mathcal{F} = \{p_1, \ldots, p_k\}$ be the set of primes less than or equal to B.
- Among $x_1 = \left\lceil \sqrt{N} \right\rceil$, $x_2 = \left\lceil \sqrt{N} \right\rceil + 1, \ldots$, select those integers x_i s.t. $q_i := x_i^2 \pmod N$ is B-smooth[1], and factor them.
- Find a subset S of $\{q_i\}_i$ such that the product of its elements is a square, i.e.

$$\prod_{j \in S} q_j = \prod_{\ell=1}^{k} p_\ell^{\sum_{j \in S} e_{j,\ell}} \quad \text{s.t.} \quad \sum_{j \in S} e_{j,\ell} = 0 \pmod 2 \quad \forall \ell \in \{1, \ldots, k\}$$

- S can be found using linear algebra.

[1] An integer is B-smooth if all its prime factors are less than or equal to B.

Quadratic Sieve Algorithm

In order to find S, define the matrix of exponents (modulo 2) as follows:

$$\begin{pmatrix} e_{1,1} \pmod 2 & e_{1,2} \pmod 2 & \dots & e_{1,k} \pmod 2 \\ \vdots & \vdots & \ddots & \vdots \\ e_{m,1} \pmod 2 & e_{m,2} \pmod 2 & \dots & e_{m,k} \pmod 2 \end{pmatrix}$$

If $m = k + 1$, then there exists a nonempty subset S of rows that sum to the zero vector modulo 2.

Quadratic Sieve Algorithm

Example (Katz-Lindell book)

Take $N = 377753$ and $B = 30$.

- Testing $x_1 = \left\lceil \sqrt{N} \right\rceil$, $x_2 = \left\lceil \sqrt{N} \right\rceil + 1, \ldots$, we obtain:

$$\begin{aligned}
620^2 &= 17^2 \cdot 23 \pmod{N} \\
621^2 &= 2^4 \cdot 17 \cdot 29 \pmod{N} \\
645^2 &= 2^7 \cdot 13 \cdot 23 \pmod{N} \\
655^2 &= 2^3 \cdot 13 \cdot 17 \cdot 29 \pmod{N}
\end{aligned}$$

- $(620 \cdot 621 \cdot 645 \cdot 655)^2 = (2^7 \cdot 13 \cdot 17^2 \cdot 23 \cdot 29)^2 \pmod{N}$
$\Rightarrow 127194^2 = 45335^2 \pmod{N}$.

- Since $127194 \neq \pm 45335 \pmod{N}$, $\gcd(127194 - 45335, N)$ gives a non trivial factor of N, i.e 751.

Further Reading I

Andrew Granville.
Smooth numbers: computational number theory and beyond.
Algorithmic number theory: lattices, number fields, curves and cryptography, 44:267–323, 2008.

Antoine Joux, Andrew Odlyzko, and Cécile Pierrot.
The past, evolving present, and future of the discrete logarithm.
In Open Problems in Mathematics and Computational Science, pages 5–36. Springer, 2014.

Hendrik W Lenstra Jr.
Factoring integers with elliptic curves.
Annals of mathematics, pages 649–673, 1987.

Further Reading II

Carl Pomerance.
Smooth numbers and the quadratic sieve.
Algorithmic Number Theory, Cambridge, MSRI publication, 44:69–82, 2008.

Carl Pomerance.
A tale of two sieves.
Biscuits of Number Theory, 85, 2008.

Victor Shoup.
Lower bounds for discrete logarithms and related problems.
In Advances in Cryptology—EUROCRYPT'97, pages 256–266. Springer, 1997.