Integer Factorization Algorithms

Ali El Kaafarani

[1]Mathematical Institute
[2] PQShield Ltd.

Outline

1. Factorization algorithms

Integer factorization

- Given a composite number N, compute its (unique) factorization $N = \prod p_i^{e_i}$ where p_i are prime numbers
- Equivalently: compute one non-trivial factor p_i
- We will assume $N = pq$, where p and q are primes

Trial Division

- How it works: try every prime number up to $\sqrt{N}$. Running time is, at worst, $O(\sqrt{N})$.

Pollard's rho

- It can be used to factor any arbitrary integer $N = pq$.
- Idea: find a **good** pair (x, y) such that $[x = y \bmod p]$ but $[x \neq y \bmod N]$.
- This implies that $\gcd(x - y, N) = p$ and therefore a non-trivial factor of N is obtained by computing this $\gcd$.
- Define some "pseudorandom" iteration function f (a standard choice would be $f(x) = x^2 + 1 \mod N$).
- Compute iterates x_i, x_{2i} and compute $\gcd(x_i - x_{2i}, N)$.
- By birthday's paradox, a pair $(x_i = x_{2i})$ s.t. $[x_i = x_{2i} \bmod p]$ is expected to be found after $O(p^{1/2})$ trials on average.

Pollard's Rho

Algorithm

Given: Integer N, a product of two n-bit primes.

$a := b \leftarrow \mathbb{Z}_N^*$

for $i = 1$ *to* $2^{n/2}$*:*

$a := f(a)$

$b := f(f(b))$

$p := \gcd(a - b, N)$

if $p \notin \{1, N\}$ ***return*** p.

Pollard's $p-1$ and Elliptic curve factorization methods

- Pollard's $p-1$ is an effective method if $p-1$ has only "small" prime factors.
- Elliptic curve factorization method generalizes previous method when neither $p-1$ nor $q-1$ are smooth.
- The group order $\#E(\mathbb{F}_p)$ of an elliptic curve can be smooth even when $p-1$ is not!
- Choosing *strong primes* for RSA, i.e. $p-1$ and $q-1$ both have large prime factors, can help against Pollard's $p-1$, but not against Elliptic curve factorization method or Number Field Sieve.

Quadratic Sieve Algorithm

- It runs in sub-exponential time, good choice for numbers up to about 300 bits long.
- Try to factor 8051.

Quadratic Sieve Algorithm

- It runs in sub-exponential time, good choice for numbers up to about 300 bits long.
- Try to factor 8051. $8051 = 90^2 - 7^2$.

Quadratic Sieve Algorithm

- It runs in sub-exponential time, good choice for numbers up to about 300 bits long.
- Try to factor 8051. $8051 = 90^2 - 7^2$. Difference of squares, $8051 = 83 \times 97$.
- Idea: find a, b for which $[a^2 = b^2 \mod N]$ and $[a \neq \pm b \mod N]$. $\gcd(a - b, N)$ gives one non trivial factor of N.

Quadratic Sieve Algorithm

- Fix some bound B, and let $F = \{p_1, \ldots, p_k\}$ the set of primes less than or equal to B.
- Search for integers $q_i = [x_i^2 \mod N]$, for $x = \left\lceil \sqrt{N} \right\rceil, \left\lceil \sqrt{N} \right\rceil + 1, \ldots$ that are B-smooth and factor them.
- Find a subset of $\{q_i\}_i$ whose product is a square, i.e.

$$S \subset \{q_i\}_i, \quad \prod_{j \in S} q_j = \prod_{i=1}^{k} p_i^{\sum_{j \in S} e_{j,i}}$$

- This product is a square iff the exponent of each prime p_i is even.

Quadratic Sieve Algorithm

- Define the matrix of exponents as follows:

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

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

Quadratic Sieve Algorithm

- Take $N = 377753$. We can compute the following;

$$620^2 \bmod N = 17^2 \cdot 23$$
$$621^2 \bmod N = 2^4 \cdot 17 \cdot 29$$
$$645^2 \bmod N = 2^7 \cdot 13 \cdot 23$$
$$655^2 \bmod N = 2^3 \cdot 13 \cdot 17 \cdot 29$$

$$[620 \cdot 621 \cdot 645 \cdot 655 \bmod N]^2 = [2^7 \cdot 13 \cdot 17^2 \cdot 23 \cdot 29 \bmod N]^2 \mod N$$
$$\Rightarrow 127194^2 = 45335^2 \bmod N$$
where $127194 \neq \pm 45335 \bmod N$,
compute $\gcd(127194 - 45335, 377753) = 751,$ a non trivial factor of N

Complexity Analysis of Quadratic Sieve

- Exercise. Hint: look at the complexity analysis of the index calculus in previous slides.

Bonus slide: Bilinear Maps (Pairings)

A *bilinear map* can be defined as a function that maps any pair of elements from two given groups (e.g. groups of points on an elliptic curve) to an element in another group (subgroup of a multiplicative group of a finite field, which is the case for the Tate Pairing).

Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be three groups of the same prime order p, a *pairing* is an efficiently computable function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, satisfying that:

1. $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$, for all $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$ and all $a, b \in \mathbb{Z}_p$.
2. *Non-degeneracy*, which is, if g_1 is a generator of $\mathbb{G}_1$, g_2 is a generator of $\mathbb{G}_2$ then $e(g_1, g_2)$ is a generator of $\mathbb{G}_T$.

Further Reading (1)

- 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 (2)

- 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.