Integer Factorization

Christophe Clavier

University of Limoges

Master 2 Cryptis

Factorization versus Primality Testing

Proving or disproving the primality of a large integer is somewhat easy . . .

- $M_p = 2^p - 1$ has been proven prime for $p = 57\,885\,161$ (17 425 170 decimal digits).
- The largest general (with no particular mathematical property) number proven prime has 26 643 digits.

. . . but this is a difficult problem to find the prime factors of a composite number:

- One can factorize 5-digit to 7-digit numbers by hand.
- General purpose softwares (Mathematica, PARI-GP, . . .) tend to have difficulty above 60-digit numbers.
- A large number of computers using an efficient implementation of the best mathematical method known, computing during several month, found in December 2009 the record factorization of a 232-digits general integer.

Which numbers to factorize ?

- RSA modules
  - These are $n = p \cdot q$ numbers, product of two primes of equal size.
  - These numbers are the most difficult ones to factorize.
- Numbers with particular mathematical properties:
  - Fermat numbers: $F_n = 2^{2^n} + 1$ (Generalization: $F_{a,b} = 2^{2^a} + 2^{2^b}$)
    (http://www.prothsearch.net/fermat.html)
  - Mersenne numbers: $M_n = 2^n - 1$
    (http://www.mersenne.org)
  - Cunningham numbers: $a^b \pm 1$, with $a \leqslant 12$, and b small
    (http://homes.cerias.purdue.edu/%7Essw/cun/index.html)
  - Partition numbers: $p(n)$ the number of ways to express n as a sum of integers.
    (http://www.asahi-net.or.jp/%7EKC2H-MSM/mathland/part)
  - Successive terms of aliquote sequences.
    (http://christophe.clavier.free.fr/Aliquot/site/Aliquot.html)
  - ...

Why factorizing ?

- For breaking the RSA key of your enemy's bank account ...
- ...or at least for evaluating the difficulty of factorizing RSA modules of different sizes.
  - A challenge was proposed by *RSA Laboratories* (rewarded by cash prizes).
    (http://www.rsasecurity.com/rsalabs/node.asp?id=2094)
- For helping to prove or disprove some famous mathematical conjectures:
  - There exists no odd perfect number (proven up to 10^{300}).
    (http://www.oddperfect.org)
  - All aliquote sequences terminate on a prime or on a cycle of sociable numbers.
    (Catalan's conjecture, 1888)
- Because it's fun ! ...

Trial Divisions

One wish to find the prime factors p_i of the integer n:

$$n = \prod_{i=1}^{k} p_i^{e_i}$$

It is sufficient to search for n prime factors only in $[1, \sqrt{n}]$.

- One tries to divide n by all successive primes $p \leqslant \sqrt{n}$.

This method allows only to find factors up to 10^8 or so.

Random Mappings

- Let S a finite set of cardinal n.
- Let $\mathrm{f} : S \to S$ be a ramdomly chosen mapping.
- Consider the iterative sequence $z_{i+1} = \mathrm{f}(z_i)$, z_0 arbitrary.
- This sequence will eventualy enter in a cycle. When ?
- One defines the tail length (τ) and the cycle length (γ) of the sequence to be the least integers verifying

$$z_\tau = z_{\tau+\gamma}$$

- A main result in random mapping theory states that the average lengths of the tail and cycle of this sequence are given by:

$$\mathrm{E}(\tau) = \mathrm{E}(\gamma) = \sqrt{\frac{\pi n}{8}} \in \mathcal{O}(\sqrt{n})$$

Pollard's ρ Method

- This method allows to find relatively small factors ($\simeq$ 15 digits) p of any arbitrarily large integer n.
- One considers the iterative sequence $(x_i)_i$ modulo p :

$$\begin{cases} x_0 \equiv 2 \pmod{p} \\ x_{i+1} \equiv f(x_i) \equiv x_i^2 + 1 \pmod{p} \end{cases}$$

- The Pollard's ρ method will succeed by detecting a cycle in the sequence $x_0, x_1, x_2, \ldots$
- The problem is that p is unknown, so we can not compute modulo p. One can only compute the sequence $(y_j)_j$ defined modulo n :

$$\begin{cases} y_0 \equiv 2 \pmod{n} \\ y_{j+1} \equiv f(y_j) \equiv y_j^2 + 1 \pmod{n} \end{cases}$$

- Note that reduced modulo p, the sequences $(y_j)_j$ and $(x_i)_i$ are equal.

Pollard's ρ Method

- How to detect a collision (modulo p) on the sequence $(x_i)_i$?

  That is, how to notice that $x_{i_1} \equiv x_{i_2} \pmod{p}$ for some i_1 and i_2 ?
- Since $(x_i)_i$ values are not computable, this collision detection can not be achieved by comparison.
- The equality $x_{i_1} \equiv x_{i_2} \pmod{p}$ will be revealed by a GCD computation between $(y_{i_2} - y_{i_1})$ and n :

$$\begin{aligned} x_{i_1} = x_{i_2} &\Rightarrow x_{i_2} - x_{i_1} \equiv 0 \pmod{p} \\ &\Rightarrow y_{i_2} - y_{i_1} \equiv 0 \pmod{p} \\ &\Rightarrow p \mid (y_{i_2} - y_{i_1}) \\ &\Rightarrow p \mid \gcd(y_{i_2} - y_{i_1}, n) \end{aligned}$$

- The search for this collision may use Floyd's algorithm for cycle detection and requires $\mathcal{O}(\sqrt{p})$ steps.

Pollard's $p-1$ Method

Definition

An integer n is said to be B-smooth with respect to some bound $B > 0$ when all its prime factors p_i are smaller than B.

- The Pollard's $p-1$ method will succeed in finding some prime factor p of n provided that $p-1$ is B-smooth. (typically $B \approx 10^6$ to 10^8)
- Let λ be the LCM of all powers of primes $q < B$ which are smaller than n:

$$\lambda = \prod_{q \leq B} q^{\lfloor \ln n / \ln q \rfloor}$$

Example: if $n = 7\,663$, and $B = 30$, then

$$\begin{aligned} \lambda &= 2^{12} \cdot 3^8 \cdot 5^5 \cdot 7^4 \cdot 11^3 \cdot 13^3 \cdot 17^3 \cdot 19^3 \cdot 23^2 \cdot 29^2 \\ &= 883974047274131592034257281280000 \ . \end{aligned}$$

Pollard's $p-1$ Method

- Crucial remark:

If $p-1$ is B-smooth, then $p-1 \mid \lambda$.

Theorem (Fermat's little theorem)

If p is prime, and $\gcd(a,p) = 1$, then $a^{p-1} \equiv 1 \pmod{p}$.

- So, if $p-1$ is B-smooth and $\gcd(a,p) = 1$, then we have:

$$\begin{aligned} p-1 \mid \lambda \quad &\Rightarrow \quad a^{\lambda} \equiv 1 \pmod{p} \\ &\Rightarrow \quad p \mid a^{\lambda} - 1 \\ &\Rightarrow \quad p \mid \gcd(a^{\lambda} - 1, n) \end{aligned}$$

Pollard's $p-1$ Complexity

Note that defining λ as $\prod_{q \leq B} q^{\lfloor \ln B / \ln q \rfloor}$ instead of $\prod_{q \leq B} q^{\lfloor \ln n / \ln q \rfloor}$ allows to greatly improve the effienciency with only a minor penalty on the success probability.

Complexity

- Given B, size of $p \nearrow \implies$ proba($p-1$ is B-smooth) $\searrow$
- Given size of p, $B \nearrow \implies$ proba($p-1$ is B-smooth) $\nearrow$
- $B \nearrow \implies$ test complexity $\nearrow$

Conclusion

Finding large factors requires either a important computational effort, or luck !

Pollard's $p-1$ Records

- gmp-ecm (from Paul Zimmermann) is probably the best available implementation of Pollard's $p-1$ factorization method.
  (http://www.komite.net/laurent/soft/ecm/ecm-6.0.1.html)
- The largest factor ever found by $p-1$ method is a 66-digits factor of $960^{119}-1$ (T. Nohara, 29 June 2006)
  (http://www.loria.fr/%7Ezimmerma/records/Pminus1.html)

Elliptic Curve Method

- The Elliptic Curve Method (ECM) was discovered by H. W. Lenstra in 1985.
- It may be viewed as a kind of randomized variation of the $p-1$ method.

Definition

An elliptic curve $\mathcal{E} = \mathcal{S} \cup \mathcal{O}$ defined over $\mathbb{F}_p$ is the set $\mathcal{S}$ of points (x, y) verifying

$$y^2 \equiv x^3 + ax + b \pmod p$$

(where a and b are two constants, with $4a^3 + 27b^2 \not\equiv 0$)

together with a special point $\mathcal{O}$ called 'point at infinity' (may be viewed as $(0, \infty)$).

- A addition law $(+)$ on points of the elliptic curve $\mathcal{E}$ is defined, giving to $(\mathcal{E}, +)$ the structure of a group with neutral element $\mathcal{O}$.

Université de Limoges

Elliptic Curve Method

- For any point $P \in \mathcal{E}$, the scalar multiplication by any positive integer k is defined to be

$$k \cdot P = \underbrace{P + P + \ldots + P}_{k \text{times}}$$

- We will have $k \cdot P = \mathcal{O}$ as soon as k is a multiple of $\text{ord}(P)$.
  In particular, when k is a multiple of the curve order $\#\mathcal{E}$.
- As with a^λ in the $p-1$ method, a 'power' $\lambda \cdot P_0$ of some initial point $P_0 \in \mathcal{E}$ is computed modulo n.
- If $\#\mathcal{E} \mid \lambda$, then computations modulo n will produce a numerical exception when evaluating $\mathcal{O}$. This will reveal p by GCD.
- A prime factor p of n is found if the order $\#\mathcal{E}$ of the curve defined over $\mathbb{F}_p$ is B-smooth.

Université de Limoges

Elliptic Curve Method

The very interesting point with ECM is that the curve order $\#\mathcal{E}$ depends on the parameters (a, b) of the curve.

Theorem (Hasse)

The order of an elliptic curve defined over $\mathbb{F}_p$ is:

$$\#\mathcal{E} = p + 1 - t \ , \quad \text{with } |t| < 2\sqrt{p}$$

- If the order of some curve is not B-smooth, change its parameters, and try again ...
- Contrary to $p - 1$ method, there is no more 'good' or 'bad' numbers to be factored: only 'good' or 'bad' curves.
- A winning curve will eventually be chosen and p be revealed.
- The expected number of trials increases with the size of p, and decreases with B.

ECM Records

- The average complexity of the ECM method is sub-exponential.
  - It depends on the size of the factor p, and nearly not on the size of the number n.
  - It is impossible today to factor a 200-digit number product of two 100-digit primes by ECM.
  - It is easy for ECM to find a 30-digit factor of a 2 000-digits number.
- `gmp-ecm` is the best available ECM implementation.
  (https://gforge.inria.fr/projects/ecm/)
- Essentially, ECM method is highly parallelizable.
  A client-server application, ECMNet, use this property to distributed the factorization of hard to factor numbers over many computers.
  (http://home.wi.rr.com/mrodenkirch)
- The largest factor ever found by ECM is a 83-digits factor of the Cunningham number $7^{337} + 1$. (7 September 2013)
  (http://www.loria.fr/%7Ezimmerma/records/top50.html)

Quadratic Sieve

- When one obtains a relation of the form

$$x^2 \equiv y^2 \pmod{n}\text{ , with } x \not\equiv \pm y \pmod{n}$$

it is possible to derive a non trivial factor of n.

- For obtaining such relations, one builds many relations of the form

$$r^2 \equiv p_1^{e_1} \cdot p_2^{e_2} \cdots p_t^{e_t} \pmod{n}$$

where $(p_i)_{i \leqslant t}$ forms a base of small primes.

- When more than t such relations are obtained, one must find a subset J of them whose product have only even exponents:

$$\begin{array}{lcll} \prod_{j \in J} r_j^2 & \equiv & \prod_{j \in J} p_{1,j}^{e_{1,j}} \cdot p_{2,j}^{e_{2,j}} \cdots p_{t,j}^{e_{t,j}} & \\ & \equiv & p_1^{f_1} \cdot p_2^{f_2} \cdots p_t^{f_t} & \text{with } f_i = \sum_{j \in J} e_{i,j} \text{ even} \\ \left(\prod_{j \in J} r_j\right)^2 & \equiv & \left(p_1^{\frac{f_1}{2}} \cdot p_2^{\frac{f_2}{2}} \cdots p_t^{\frac{f_t}{2}}\right)^2 & \pmod{n} \end{array}$$

Quadratic Sieve

- The Quadratic Sieve (QS) has three phases:
  - Sieve (to collect the relations, slow)
  - Linear algebra (for the exponents, memory consuming)
  - Square root (several trials, fast)
- The average complexity of the QS method is sub-exponential.
  - It does not depend on the size of the factors, but only on the size of the number n.
  - The largest composite factored by QS is a 129-digit RSA challenge module.
  - For 100-digit or larger number QS is slower than GNFS.
- `msieve` is a good implementation of the Quadratic Sieve.
  (http://www.boo.net/%7Ejasonp/qs.html)

Number Field Sieve

- The Number Field Sieve (NFS) is a (much more complicated) generalization of the Quadratic Sieve.
- The goal is also to collect relations, but the way to obtain them is more efficient and makes use of more advanced theoretical concepts.
- Contrary to Quadratic Sieve, one does not work in the ring of integers modulo n, but rather in number fields which are fields containing the rationals and some polynomial roots.
- There are two versions of NFS:
  - The Special Number Field Sieve (SNFS) applies very efficiently to integers of the form $n = r^e - s$, for small r and $|s|$.
  - The General Number Field Sieve (GNFS) has been invented later, is less efficient than SNFS, but applies to any number.

Number Field Sieve

- The average complexity of both GNFS and SNFS methods is sub-exponential.
  - It does not depend on the size of the factors, but only on the size of the number n.
  - The largest composite factored by GNFS is a 232-digit RSA module.
  - GNFS is a reference method to analyse the security of factorization based cryptography (RSA).
- `ggnfs` is one of the rare available implementation of NFS.
  (http://www.math.ttu.edu/%7Ecmonico/software/ggnfs)