Pollard's $p - 1$ algorithm

From Wikipedia, the free encyclopedia

Pollard's $p - 1$ algorithm is a number theoretic integer factorization algorithm, invented by John Pollard in 1974. It is a special-purpose algorithm, meaning that it is only suitable for integers with specific types of factors; it is the simplest example of an algebraic-group factorisation algorithm.

The factors it finds are ones for which the number preceding the factor, $p - 1$, is powersmooth; the essential observation is that, by working in the multiplicative group modulo a composite number N, we are also working in the multiplicative groups modulo all of N's factors.

The existence of this algorithm leads to the concept of safe primes, being primes for which $p - 1$ is two times a Sophie Germain prime q and thus minimally smooth. These primes are sometimes construed as "safe for cryptographic purposes", but they might be *unsafe* — in current recommendations for cryptographic strong primes (*e.g.* ANSI X9.31), it is necessary but not sufficient that $p - 1$ has at least one large prime factor. Most sufficiently large primes are strong; if a prime used for cryptographic purposes turns out to be non-strong, it is much more likely to be through malice than through an accident of random number generation. This terminology is considered obsolescent by the cryptography industry. [1]

Contents [hide]

Base concepts [edit]

Let n be a composite integer with prime factor p. By Fermat's little theorem, we know that for all integers a coprime to p and for all positive integers K:

$$a^{K(p-1)} \equiv 1 \pmod{p}$$

If a number x is congruent to 1 modulo a factor of n, then the gcd$(x - 1, n)$ will be divisible by that factor.

The idea is to make the exponent a large multiple of $p - 1$ by making it a number with very many prime factors; generally, we take the product of all prime powers less than some limit B. Start with a random x, and repeatedly replace it by $x^w \mod n$ as w runs through those prime powers. Check at each stage, or once at the end if you prefer, whether gcd$(x - 1, n)$ is not equal to 1.

Multiple factors [edit]

It is possible that for all the prime factors p of n, $p - 1$ is divisible by small primes, at which point the Pollard $p - 1$ algorithm gives you n again.

Algorithm and running time [edit]

The basic algorithm can be written as follows:

Inputs: n: a composite number
Output: a nontrivial factor of n or failure

1. select a smoothness bound B
2. define $M = \prod_{\text{primes } q \le B} q^{\lfloor \log_q B \rfloor}$ (note: explicitly evaluating M may not be necessary)
3. randomly pick a coprime to n (note: we can actually fix a, random selection here is not imperative)

4. compute $g = \gcd(a^M - 1, n)$ (note: exponentiation can be done modulo n)
5. if $1 < g < n$ then return g
6. if $g = 1$ then select a larger B and go to step 2 or return failure
7. if $g = n$ then select a smaller B and go to step 2 or return failure

If $g = 1$ in step 6, this indicates there are no prime factors p for which $p-1$ is B-powersmooth. If $g = n$ in step 7, this usually indicates that all factors were B-powersmooth, but in rare cases it could indicate that a had a small order modulo n.

The running time of this algorithm is $O(B \times \log B \times \log^2 n)$; larger values of B make it run slower, but are more likely to produce a factor.

How to choose B? [edit]

Since the algorithm is incremental, it can just keep running with the bound constantly increasing.

Assume that $p - 1$, where p is the smallest prime factor of n, can be modelled as a random number of size less than $\sqrt{n}$. By Dixon's theorem, the probability that the largest factor of such a number is less than $(p - 1)^{\varepsilon}$ is roughly $\varepsilon^{-\varepsilon}$; so there is a probability of about $3^{-3} = 1/27$ that a B value of $n^{1/6}$ will yield a factorisation.

In practice, the elliptic curve method is faster than the Pollard $p - 1$ method once the factors are at all large; running the $p - 1$ method up to $B = 10^6$ will find a quarter of all twelve-digit factors and 1/27 of all eighteen-digit factors, before proceeding to another method.

Two-stage variant [edit]

A variant of the basic algorithm is sometimes used; instead of requiring that $p - 1$ has all its factors less than B, we require it to have all but one of its factors less than some B_1, and the remaining factor less than some $B_2 \gg B_1$. After completing the first stage, which is the same as the basic algorithm, instead of computing a new

$$M' = \prod_{\text{primes } p \le B_2} q^{\lfloor \log_q B_2 \rfloor}$$

for B_2 and checking $\gcd(a^{M'} - 1, n)$, we compute

$$Q = \prod_{\text{primes } q \in (B_1, B_2]} (H^q - 1)$$

where $H = a^M$ and check if $\gcd(Q, n)$ produces a nontrivial factor of n. As before, exponentiations can be done modulo n.

Let $\{q_1, q_2, \ldots\}$ be successive prime numbers in the interval $(B_1, B_2]$ and $d_n = q_n - q_{n-1}$ the difference between consecutive prime numbers. Since typically $B_1 > 2$, d_n are even numbers. The distribution of prime numbers is such that the d_n will all be relatively small. It is suggested that $d_n \le \ln^2 B_2$. Hence, the values of H^2, H^4, H^6, … (mod n) can be stored in a table, and H^{q_n} be computed from $H^{q_{n-1}} \cdot H^{d_n}$, saving the need for exponentiations.

Implementations [edit]

- The GMP-ECM package includes an efficient implementation of the $p - 1$ method.
- Prime95 and MPrime, the official clients of the Great Internet Mersenne Prime Search, use p - 1 to eliminate potential candidates.

See also [edit]

- Williams' p + 1 algorithm

References [edit]

- Pollard, J. M. (1974). "Theorems of factorization and primality testing". *Proceedings of the Cambridge Philosophical Society* **76** (3): 521–528. doi:10.1017/S0305004100049252.
- Montgomery, P. L.; Silverman, R. D. (1990). "An FFT extension to the $P - 1$ factoring algorithm". *Mathematics of Computation* **54** (190): 839–854. doi:10.1090/S0025-5718-1990-1011444-3.

External links [edit]

- Pollard's $p - 1$ Method
- Pollard's $p - 1$ Algorithm source code

v · t · e	Number-theoretic algorithms [hide]
Primality tests	AKS TEST · APR TEST · Baillie–PSW · ECPP TEST · Elliptic curve · Pocklington · Fermat · Lucas · *LUCAS–LEHMER* · *LUCAS–LEHMER–RIESEL* · *PROTH'S THEOREM* · *PÉPIN'S* · Quadratic Frobenius test · Solovay–Strassen · Miller–Rabin
Prime-generating	Sieve of Atkin · Sieve of Eratosthenes · Sieve of Sundaram · Wheel factorization
Integer factorization	Continued fraction (CFRAC) · Dixon's · Lenstra elliptic curve (ECM) · Euler's · Pollard's rho · ***p* − 1** · *p* + 1 · Quadratic sieve (QS) · General number field sieve (GNFS) · *Special number field sieve (SNFS)* · Rational sieve · Fermat's · Shanks' square forms · Trial division · Shor's
Multiplication	Ancient Egyptian · Long · Karatsuba · Toom–Cook · Schönhage–Strassen · Fürer's
Discrete logarithm	BABY-STEP GIANT-STEP · Pollard rho · Pollard kangaroo · POHLIG–HELLMAN · Index calculus · Function field sieve
Greatest common divisor	Binary · Euclidean · Extended Euclidean · Lehmer's
Modular square root	Cipolla · Pocklington's · Tonelli–Shanks
Other algorithms	Chakravala · Cornacchia · Integer relation · Integer square root · Modular exponentiation · Schoof's
	Italics indicate that algorithm is for numbers of special forms · SMALLCAPS indicate a deterministic algorithm

Categories: Integer factorization algorithms