Pollard's kangaroo algorithm

From Wikipedia, the free encyclopedia

In computational number theory and computational algebra, **Pollard's kangaroo algorithm** (aka **Pollard's lambda algorithm**, see Naming below) is an algorithm for solving the discrete logarithm problem. The algorithm was introduced in 1978 by the number theorist J. M. Pollard, in the same paper [1] as his better-known ρ algorithm for solving the same problem. Although Pollard described the application of his algorithm to the discrete logarithm problem in the multiplicative group of units modulo a prime *p*, it is in fact a generic discrete logarithm algorithm—it will work in any finite cyclic group.

Contents [hide]

The algorithm [edit]

Suppose G is a finite cyclic group of order n which is generated by the element α, and we seek to find the discrete logarithm x of the element β to the base α. In other words, we seek $x \in Z_n$ such that $\alpha^x = \beta$. The lambda algorithm allows us to search for x in some subset $\{a, \ldots, b\} \subset Z_n$. We may search the entire range of possible logarithms by setting $a = 0$ and $b = n - 1$, although in this case Pollard's rho algorithm is more efficient. We proceed as follows:

1. Choose a set S of integers and define a pseudorandom map $f : G \rightarrow S$.

2. Choose an integer N and compute a sequence of group elements $\{x_0, x_1, \ldots, x_N\}$ according to:

- $x_0 = \alpha^b$
- $x_{i+1} = x_i \alpha^{f(x_i)} \text{ for } i = 0, 1, \ldots, N-1$

3. Compute

$$d = \sum_{i=0}^{N-1} f(x_i).$$

Observe that:

$$x_N = x_0 \alpha^d = \alpha^{b+d}.$$

4. Begin computing a second sequence of group elements $\{y_0, y_1, \ldots\}$ according to:

- $y_0 = \beta$
- $y_{i+1} = y_i \alpha^{f(y_i)} \text{ for } i = 0, 1, \ldots, N-1$

and a corresponding sequence of integers $\{d_0, d_1, \ldots\}$ according to:

$$d_n = \sum_{i=0}^{n-1} f(y_i).$$

Observe that:

$$y_i = y_0 \alpha^{d_i} = \beta \alpha^{d_i} \text{ for } i = 0, 1, \ldots, N-1$$

5. Stop computing terms of $\{y_i\}$ and $\{d_i\}$ when either of the following conditions are met:

A) $y_j = x_N$ for some j. If the sequences $\{x_i\}$ and $\{y_j\}$ "collide" in this manner, then we have:

$$x_N = y_j \Rightarrow \alpha^{b+d} = \beta \alpha^{d_j} \Rightarrow \beta = \alpha^{b+d-d_j} \pmod{n} \Rightarrow x \equiv b+d-d_j \pmod{n}$$

and so we are done.

B) $d_i > b - a + d$. If this occurs, then the algorithm has failed to find x. Subsequent attempts can be made by changing the choice of S and/or f.

Complexity [edit]

Pollard gives the time complexity of the algorithm as $O(\sqrt{b-a})$, based on a probabilistic argument which follows from the assumption that *f* acts pseudorandomly. Note that when the size of the set {*a*, …, *b*} to be searched is measured in bits, as is normal in complexity theory, the set has size log(*b* − *a*), and so the algorithm's complexity is $O(\sqrt{b-a}) = O(2^{\frac{1}{2}\log(b-a)})$, which is exponential in the problem size. For this reason, Pollard's lambda algorithm is considered an exponential time algorithm. For an example of a subexponential time discrete logarithm algorithm, see the index calculus algorithm.

Naming [edit]

The algorithm is well known by two names.

The first is "Pollard's kangaroo algorithm". This name is a reference to an analogy used in the paper presenting the algorithm, where the algorithm is explained in terms of using a *tame* kangaroo to trap a *wild* kangaroo. Pollard has explained[2] that this analogy was inspired by a "fascinating" article published in the same issue of *Scientific American* as an exposition of the RSA public key cryptosystem. The article[3] described an experiment in which a kangaroo's "energetic cost of locomotion, measured in terms of oxygen consumption at various speeds, was determined by placing kangaroos on a treadmill".

The second is "Pollard's lambda algorithm". Much like the name of another of Pollard's discrete logarithm algorithms, Pollard's rho algorithm, this name refers to the similarity between a visualisation of the algorithm and the Greek letter lambda (λ). The shorter stroke of the letter lambda corresponds to the sequence $\{x_i\}$, since it starts from the position b to the right of x. Accordingly, the longer stroke corresponds to the sequence $\{y_i\}$, which "collides with" the first sequence (just like the strokes of a lambda intersect) and then follows it subsequently.

Pollard has expressed a preference for the name "kangaroo algorithm",[4] as this avoids confusion with some parallel versions of his rho algorithm, which have also been called "lambda algorithms".

See also [edit]

- Rainbow table

References [edit]

1. ^ J. Pollard, *Monte Carlo methods for index computation mod p*, Mathematics of Computation, Volume 32, 1978
2. ^ J. M. Pollard, *Kangaroos, Monopoly and Discrete Logarithms*, Journal of Cryptology, Volume 13, pp 437-447, 2000
3. ^ T. J. Dawson, *Kangaroos*, Scientific American, August 1977, pp. 78-89
4. ^ http://sites.google.com/site/jmptidcott2/

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: Number theoretic algorithms | Computer algebra | Logarithms

This page was last modified on 6 June 2014, at 10:02.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Developers Mobile view