WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Català
Čeština
Deutsch
Español
Esperanto
فارسی
Français
한국어
Italiano
עברית
Nederlands
日本語
Polski
Português
Русский
Türkçe
Українська
اردو
Tiếng Việt
中文
Edit links

Discrete logarithm

From Wikipedia, the free encyclopedia

This article includes a list of references, related reading or external links, **but its sources remain unclear because it lacks inline citations**. Please improve this article by introducing more precise citations. *(October 2013)*

In mathematics, a **discrete logarithm** is an integer k solving the equation $b^k = g$, where b and g are elements of a finite group. Discrete logarithms are thus the finite-group-theoretic analogue of ordinary logarithms, which solve the same equation for real numbers b and g, where b is the base of the logarithm and g is the value whose logarithm is being taken.

Computing discrete logarithms is believed to be difficult. No efficient general method for computing discrete logarithms on conventional computers is known, and several important algorithms in public-key cryptography base their security on the assumption that the discrete logarithm problem has no efficient solution.

Contents [hide]

Example [edit]

Discrete logarithms are perhaps simplest to understand in the group $(\mathbf{Z}_p)^\times$. This is the group of multiplication modulo the prime p. Its elements are congruence classes modulo p, and the group product of two elements may be obtained by ordinary integer multiplication of the elements followed by reduction modulo p.

The kth power of one of the numbers in this group may be computed by finding its kth power as an integer and then finding the remainder after division by p. When the numbers involved are large, it is more efficient to reduce modulo p multiple times during the computation. Regardless of the specific algorithm used, this operation is called modular exponentiation. For example, consider $(\mathbf{Z}_{17})^\times$. To compute 3^4 in this group, compute $3^4 = 81$, and then divide 81 by 17, obtaining a remainder of 13. Thus $3^4 = 13$ in the group $(\mathbf{Z}_{17})^\times$.

The discrete logarithm is just the inverse operation. For example, consider the equation $3^k \equiv 13 \pmod{17}$ for k. From the example above, one solution is $k = 4$, but it is not the only solution. Since $3^{16} \equiv 1 \pmod{17}$—as follows from Fermat's little theorem—it also follows that if n is an integer then $3^{4+16n} \equiv 3^4 \times (3^{16})^n \equiv 13 \times 1^n \equiv 13 \pmod{17}$. Hence the equation has infinitely many solutions of the form $4 + 16n$. Moreover, since 16 is the smallest positive integer m satisfying $3^m \equiv 1 \pmod{17}$, i.e. 16 is the order of 3 in $(\mathbf{Z}_{17})^\times$, these are the only solutions. Equivalently, the set of all possible solutions can be expressed by the constraint that $k \equiv 4 \pmod{16}$.

Definition [edit]

In general, let G be any group, with its group operation denoted by multiplication. Let b and g be any elements of G. Then any integer k that solves $b^k = g$ is termed a **discrete logarithm** (or simply **logarithm**, in this context) of g to the base b. We write $k = \log_b g$. Depending on b and g, it is possible that no discrete logarithm exists, or that more than one discrete logarithm exists. Let H be the subgroup of G generated by b. Then H is a cyclic group, and integral $\log_b g$ exists for all g in H. If H is infinite, then $\log_b g$ is also unique, and the discrete logarithm amounts to a group isomorphism

$$\log_b \colon H \to \mathbf{Z}.$$

On the other hand, if H is finite of size n, then $\log_b g$ is unique only up to congruence modulo n, and the discrete logarithm amounts to a group isomorphism

$$\log_b \colon H \to \mathbf{Z}_n,$$

where $\mathbf{Z}_n$ denotes the ring of integers modulo n. The familiar base change formula for ordinary logarithms remains valid: If c is another generator of H, then

$$\log_c(g) = \log_c(b) \cdot \log_b(g).$$

Algorithms [edit]

See also: Discrete logarithm records

List of unsolved problems in computer science

Can the discrete logarithm be computed in polynomial time on a classical computer?

No efficient classical algorithm for computing general discrete logarithms $\log_b g$ is known. The naive algorithm is to raise b to higher and higher powers k until the desired g is found; this is sometimes called *trial multiplication*. This algorithm requires running time linear in the size of the group G and thus exponential in the number of digits in the size of the group. There exists an efficient quantum algorithm due to Peter Shor.[1]

More sophisticated algorithms exist, usually inspired by similar algorithms for integer factorization. These algorithms run faster than the naive algorithm, some of them linear in the *square root* of the size of the group, and thus exponential in half the number of digits in the size of the group. However none of them runs in polynomial time (in the number of digits in the size of the group).

- Baby-step giant-step
- Function field sieve
- Index calculus algorithm
- Number field sieve
- Pohlig–Hellman algorithm
- Pollard's rho algorithm for logarithms
- Pollard's kangaroo algorithm (aka Pollard's lambda algorithm)

Comparison with integer factorization [edit]

While computing discrete logarithms and factoring integers are distinct problems, they share some properties:

- both problems are difficult (no efficient algorithms are known for non-quantum computers),
- for both problems efficient algorithms on quantum computers are known,
- algorithms from one problem are often adapted to the other, and
- the difficulty of both problems has been used to construct various cryptographic systems.

Cryptography [edit]

There exist groups for which computing discrete logarithms is apparently difficult. In some cases (e.g. large prime order subgroups of groups $(\mathbf{Z}_p)^\times$) there is not only no efficient algorithm known for the worst case, but the average-case complexity can be shown to be about as hard as the worst case using random self-reducibility.

At the same time, the inverse problem of discrete exponentiation is not difficult (it can be computed efficiently using exponentiation by squaring, for example). This asymmetry is analogous to the one between integer factorization and integer multiplication. Both asymmetries have been exploited in the construction of cryptographic systems.

Popular choices for the group G in discrete logarithm cryptography are the cyclic groups $(\mathbf{Z}_p)^\times$ (e.g. ElGamal encryption, Diffie–Hellman key exchange, and the Digital Signature Algorithm) and cyclic subgroups of elliptic curves over finite fields (*see* elliptic curve cryptography).

References [edit]

1. ^ Shor, Peter (1997). "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". *SIAM Journal on Computing* **26** (5): 1484–1509. arXiv:quant-ph/9508027. doi:10.1137/s0097539795293172.

- Richard Crandall; Carl Pomerance. Chapter 5, *Prime Numbers: A computational perspective*, 2nd ed., Springer.
- Stinson, Douglas Robert (2006), *Cryptography: Theory and Practice* (3rd ed.), London: CRC Press, ISBN 978-1-58488-508-5

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	

v · t · e **Public-key cryptography** [show]

Categories: Modular arithmetic | Group theory | Cryptography | Logarithms | Finite fields | Binary operations | Computational hardness assumptions | Unsolved problems in computer science