-
- Wikiwand W
-
-
- *EN*
-
-
- Get Wikiwand

Baby-step giant-step

Connected to:

Algorithm Discrete logarithm Daniel Shanks

From Wikipedia, the free encyclopedia

In group theory, a branch of mathematics, the **baby-step giant-step** is a meet-in-the-middle algorithm for computing the discrete logarithm or order of an element in a finite abelian group by Daniel Shanks.[1] The discrete log problem is of fundamental importance to the area of public key cryptography.

Many of the most commonly used cryptography systems are based on the assumption that the discrete log is extremely difficult to compute; the more difficult it is, the more security it provides a data transfer. One way to increase the difficulty of the discrete log problem is to base the cryptosystem on a larger group.

Theory

The algorithm is based on a space–time tradeoff. It is a fairly simple modification of trial multiplication, the naive method of finding discrete logarithms.

Given a cyclic group G of order n, a generator α of the group and a group element β, the problem is to find an integer x such that

$$\alpha^x = \beta\,.$$

The baby-step giant-step algorithm is based on rewriting x:

$$x = im + j$$
$$m = \lceil \sqrt{n} \rceil$$
$$0 \leq i < m$$
$$0 \leq j < m$$

Therefore, we have:

$$\alpha^x = \beta$$
$$\alpha^{im+j} = \beta$$

$$\alpha^j = \beta\left(\alpha^{-m}\right)^i$$

The algorithm precomputes α^j for several values of j. Then it fixes an m and tries values of i in the right-hand side of the congruence above, in the manner of trial multiplication. It tests to see if the congruence is satisfied for any value of j, using the precomputed values of α^j.

The algorithm

Input: A cyclic group G of order n, having a generator α and an element β.

Output: A value x satisfying $\alpha^x = \beta$.

1. $m \leftarrow$ Ceiling($\sqrt{n}$)
2. For all j where $0 \le j < m$:
    1. Compute α^j and store the pair (j, α^j) in a table. (See § In practice)
3. Compute α^{-m}.
4. $\gamma \leftarrow \beta$. (set $\gamma = \beta$)
5. For all i where $0 \le i < m$:
    1. Check to see if γ is the second component (α^j) of any pair in the table.
    2. If so, return $im + j$.
    3. If not, $\gamma \leftarrow \gamma \bullet \alpha^{-m}$.

In practice

The best way to speed up the baby-step giant-step algorithm is to use an efficient table lookup scheme. The best in this case is a hash table. The hashing is done on the second component, and to perform the check in step 1 of the main loop, γ is hashed and the resulting memory address checked. Since hash tables can retrieve and add elements in $O(1)$ time (constant time), this does not slow down the overall baby-step giant-step algorithm.

The running time of the algorithm and the space complexity is $O(\sqrt{n})$, much better than the $O(n)$ running time of the naive brute force calculation.

The Baby-step giant-step algorithm is often used to solve for the shared key in the Diffie Hellman key exchange, when the modulus is a prime number. If the modulus is not prime, the Pohlig–Hellman algorithm has a smaller algorithmic complexity, and solves the same problem.

Notes

- The baby-step giant-step algorithm is a generic algorithm. It works for every finite cyclic group.
- It is not necessary to know the order of the group G in advance. The algorithm still works if n is merely an upper bound on the group order.
- Usually the baby-step giant-step algorithm is used for groups whose order is prime. If the order of the group is composite then the Pohlig–Hellman algorithm is more efficient.
- The algorithm requires O(m) memory. It is possible to use less memory by choosing a smaller m in the first step of the algorithm. Doing so increases the running time, which then is O(n/m). Alternatively one can use Pollard's rho algorithm for logarithms, which has about the same running time as the baby-step giant-step algorithm, but only a small memory requirement.

- While this algorithm is credited to Daniel Shanks, who published the 1971 paper in which it first appears, a 1994 paper by Nechaev[2] states that it was known to Gelfond in 1962.
- There exist optimized versions of the original algorithm, such as using the collision-free truncated lookup tables of [3] or negation maps and Montgomery's simultaneous modular inversion as proposed in.[4]

Further reading

- H. Cohen, A course in computational algebraic number theory, Springer, 1996.
- D. Shanks, Class number, a theory of factorization and genera. In Proc. Symp. Pure Math. 20, pages 415—440. AMS, Providence, R.I., 1971.
- A. Stein and E. Teske, Optimized baby step-giant step methods, Journal of the Ramanujan Mathematical Society 20 (2005), no. 1, 1–32.
- A. V. Sutherland, Order computations in generic groups, PhD thesis, M.I.T., 2007.
- D. C. Terr, A modification of Shanks' baby-step giant-step algorithm, Mathematics of Computation 69 (2000), 767–773. doi:10.1090/S0025-5718-99-01141-2

References

1. ^ Daniel Shanks (1971), "Class number, a theory of factorization and genera", *In Proc. Symp. Pure Math.*, Providence, R.I.: American Mathematical Society, **20**, pp. 415–440
2. ^ V. I. Nechaev, Complexity of a determinate algorithm for the discrete logarithm, Mathematical Notes, vol. 55, no. 2 1994 (165-172)
3. ^ Panagiotis Chatzigiannis, Konstantinos Chalkias and Valeria Nikolaenko (2021-06-30). *Homomorphic decryption in blockchains via compressed discrete-log lookup tables*. CBT workshop 2021 (ESORICS). Retrieved 2021-09-07.
4. ^ Steven D. Galbraith, Ping Wang and Fangguo Zhang (2016-02-10). *Computing Elliptic Curve Discrete Logarithms with Improved Baby-step Giant-step Algorithm*. Advances in Mathematics of Communications. Retrieved 2021-09-07.

External links

- Baby step-Giant step – example C source code

Number-theoretic algorithms

Number-theoretic algorithms

Primality tests	AKS APR Baillie–PSW Elliptic curve Pocklington Fermat Lucas *Lucas–Lehmer* *Lucas–Lehmer–Riesel* *Proth's theorem* *Pépin's* Quadratic Frobenius 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's square forms
- Trial division
- Shor's

Multiplication

- Ancient Egyptian
- Long
- Karatsuba
- Toom–Cook
- Schönhage–Strassen
- Fürer's

Euclidean division

- Binary
- Chunking
- Fourier
- Goldschmidt
- Newton-Raphson
- Long
- Short
- SRT

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]
- [Berlekamp]

Other algorithms

- Chakravala
- Cornacchia
- Exponentiation by squaring
- Integer square root
- Integer relation (LLL)
- Modular exponentiation
- Montgomery reduction
- Schoof

- *Italics* indicate that algorithm is for numbers of special forms

Categories
Categories:

- Group theory
- Number theoretic algorithms

- Home
- About Us
- Press
- Site Map
- Terms Of Service
- Privacy Policy

Baby-step giant-step

Listen to this article