Pohlig–Hellman algorithm

From Wikipedia, the free encyclopedia

In number theory, the **Pohlig–Hellman algorithm** sometimes credited as the **Silver–Pohlig–Hellman algorithm**[1] is a special-purpose algorithm for computing discrete logarithms in a multiplicative group whose order is a smooth integer.

The algorithm was discovered by Roland Silver, but first published by Stephen Pohlig and Martin Hellman (independent of Silver).

We will explain the algorithm as it applies to the group $\mathbf{Z}^*_p$ consisting of all the elements of $\mathbf{Z}_p$ which are coprime to *p*, and leave it to the advanced reader to extend the algorithm to other groups by using Lagrange's theorem.

Input Integers *p*, *g*, *e*.

Output An Integer *x*, such that $e \equiv g^x \pmod p$ (if one exists).

1. Determine the prime factorization of the order of the group :
$$\varphi(p) = p_1 \cdot p_2 \cdots p_n$$
(All the p_i are considered small since the group order is smooth.)
2. From the Chinese remainder theorem it will be sufficient to determine the values of *x* modulo each prime power dividing the group order. Suppose for illustration that p_1 divides this order but p_1^2 does not. Then we need to determine *x* mod p_1, that is, we need to know the ending coefficient b_1 in the base-p_1 expansion of *x*, i.e. in the expansion $x = a_1 p_1 + b_1$. We can find the value of b_1 by examining all the possible values between 0 and p_1-1. (We may also use a faster algorithm such as baby-step giant-step when the order of the group is prime.[2]) The key behind the examination is that:
$$\begin{aligned} e^{\varphi(p)/p_1} &\equiv (g^x)^{\varphi(p)/p_1} \pmod p \\ &\equiv (g^{\varphi(p)})^{a_1} g^{b_1 \varphi(p)/p_1} \pmod p \\ &\equiv (g^{\varphi(p)/p_1})^{b_1} \pmod p \end{aligned}$$
(using Euler's theorem). With everything else now known, we may try each value of b_1 to see which makes the equation be true. If $g^{\varphi(p)/p_1} \not\equiv 1 \pmod p$, then there is exactly one b_1, and that b_1 is the value of *x* modulo p_1. (An exception arises if $g^{\varphi(p)/p_1} \equiv 1 \pmod p$ since then the order of *g* is less than φ(*p*). The conclusion in this case depends on the value of $e^{\varphi(p)/p_1} \mod p$ on the left: if this quantity is not 1, then no solution *x* exists; if instead this quantity is also equal to 1, there will be more than one solution for *x* less than φ(*p*), but since we are attempting to return only one solution *x*, we may use b_1=0.)
3. The same operation is now performed for p_2 through p_n.
A minor modification is needed where a prime number is repeated. Suppose we are seeing p_i for the (*k* + 1)st time. Then we already know c_i in the equation $x = a_i p_i^{k+1} + b_i p_i^k + c_i$, and we find either b_i or c_i the same way as before, depending on whether $g^{\varphi(p)/p_i} \equiv 1 \pmod p$.
4. With all the b_i known, we have enough simultaneous congruences to determine *x* using the Chinese remainder theorem.

Complexity [edit]

The worst-case time complexity of the Pohlig–Hellman algorithm is $O(\sqrt{n})$ for a group of order *n*, but it is more efficient if the order is smooth. Specifically, if $\prod_i p_i^{e_i}$ is the prime factorization of *n*, then the complexity can be stated as $O\left(\sum_i e_i(\log n + \sqrt{p_i})\right)$.[3]

Notes [edit]

1. ^ Mollin 2006, pg. 344
2. ^ Menezes, et. al 1997, pg. 109
3. ^ Menezes, et. al 1997, pg. 108

References [edit]

- Mollin, Richard (2006-09-18). *An Introduction To Cryptography* (2nd ed.). Chapman and Hall/CRC. p. 344. ISBN 978-1-58488-618-1.
- S. Pohlig and M. Hellman (1978). "An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significance" (PDF). *IEEE Transactions on Information Theory* (24): 106–110.
- Menezes, Alfred J.; van Oorschot, Paul C.; Vanstone, Scott A. (1997). "Number-Theoretic Reference Problems" (PDF). *Handbook of Applied Cryptography*. CRC Press. pp. 107–109. ISBN 0-8493-8523-7.

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