Diffie-Hellman算法，简称DH算法，由W.Diffie和M.E.Hellman在1976年公布的一种密钥一致性算法，该算法是一种建立密钥的方法，并非加密方法，但其产生的密钥可用于加密、密钥管理或任何其它的加密方式，这种密钥交换技术的目的在于使两个用户间能安全地交换密钥(KEY)以便用于今后的报文加密。该算法需要公开两个参数：质数 n 和其原根 g，同时通信双方 A 和 B 随机选择自己的私钥 x 和 y，通过交换 mod n 和 mod n 后，它们就可以生成两者之间的会话密钥了。DH算法对公开密钥密码编码学产生了深远的影响。DH算法是一种确保共享KEY安全穿越网络的方法。

算法描述

离散对数：定义素数p的原始根是能生成1-(p-1)之间所有数的一个数，设a为p的原始根，则：a mod p， mod p，…， mod p是各不相同的整数，且以某种排列方式组成了从1到p-1的所有整数。对于任意数b及素数p的原始根a，可以找到一个唯一的指数i，满足：b= mod p，其中0≤i≤p-1，那么指数i 称为b的以a为基数的模p的离散对数。

Diffie-Hellman算法的有效性依赖于计算离散对数的难度，其含义是：当已知大素数p和它的一个原根a后，对于给定的b，要计算出i 被认为是很困难的，而给定i 计算b却相对容易。

假设网络上有两个用户A和B，彼此之间协商共同的密码，算法过程如图1所示。

图1 Diffie-Hellman算法过程示意图

图1 Diffie-Hellman算法过程示意图

假设交换密钥的值为k。

(1)A和B事先约好大素数p和它的原始根a；

(2)A随机产生一个数x，计算X= mod p，然后把X发给B；

(3)B随机产生一个数y，计算Y= mod p，然后把Y发给A；

(4)A计算k= mod p；

(5)B计算 = mod p。

因为：k= mod p= mod p= mod p= mod p； = mod p；所以k= 。

不安全网络上的窃听者只能得到a、p、X、Y，除非能计算离散对数x和y，否则将无法得到密钥k，但对于大素数p，计算离散对数是十分困难的，因此k为用户A和B独立计算出的密钥。

幂模运算特性：

Cryptographic explanation[edit]

The simplest and the original implementation[2] of the protocol uses the multiplicative group of integers modulo p, where p is prime, and g is a primitive root modulo p. These two values are chosen in this way to ensure that the resulting shared secret can take on any value from 1 to p. Here is an example of the protocol, with non-secret values in blue, and secret values in red.

Alice and Bob publicly agree to use a modulus p = 23 and base g = 5 (which is a primitive root modulo 23).

Alice chooses a secret integer a = 4, then sends Bob A = ga mod p

A = 54 mod 23 = 4

Bob chooses a secret integer b = 3, then sends Alice B = gb mod p

B = 53 mod 23 = 10

Alice computes s = Ba mod p

s = 104 mod 23 = 18

Bob computes s = Ab mod p

s = 43 mod 23 = 18

Alice and Bob now share a secret (the number 18).

Both Alice and Bob have arrived at the same value s, because, under mod p,