Diffie-Hellman Key Exchange

The 1st published public-key algorithm was invented by Whitfield Diffie and Martin Hellman in 1976 and is generally referred to as Diffie-Hellman key exchange. The purpose of the algorithm is to enable two users to exchange a key securely that can then be used for subsequent encryption of messages. The algorithm itself is limited to exchange of the keys.

The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms. Briefly, we can define the discrete logarithm as follows. First, we define a primitive root of a prime number p as one whose powers generate all the integers from 1 to p-1. That is, if a is a primitive root of the pumber p, then the numbers

$a \bmod p, a^2 \bmod p, .., a^{p-1} \bmod p$

Diffie-Hellman Key Exchange (Cont 1)

are distinct and consist of the integers from 1 through p-1 in some permutation. For any integer b and a primitive root a of prime number p, we can find a unique exponent i such that

$$b \equiv a^i \bmod p, 0 \le i < p \,.$$

The exponent i is referred to as the discrete logarithm, or index of b for the base a, mod p. This value is denoted as $ind_{a,p}(b)$. Diffie-Hellman key exchange is summarized in Figure 10.7:

Global Public Elements

q	prime number
α	$\alpha < q$ and α a primitive root of q

User A Key Generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$

Generation of Secret Key by User A

$K = (Y_B)^{X_A} \bmod q$

Generation of Secret Key by User B

$K = (Y_A)^{X_B} \bmod q$

Figure 10.7 The Diffie-Hellman Key Exchange Algorithm

Diffie-Hellman Key Exchange (Cont 2)

Because X_A and X_B are private, the opponent is forced to take a discrete logarithm to determine the key. For example, attacking the secret key of user B, the opponent must compute

$$X_B = ind_{\alpha,q}(Y_B)$$

The opponent then can calculate the key K in the same manner as user B calculates it. For large primes, such an attack is considered infeasible.

Let's consider example. Key exchange is based on the use of the prime number q=353 and a primitive root of 353, in this case α=3. A and B select secret keys X_A=97 and X_B=233, respectively.
Each computes its public key:
A computes $Y_A=3^{97} \bmod 353 = 40$,
B computes $Y_B=3^{233} \bmod 353 = 248$.
After they exchange public keys, each can compute the common secret key:
A computes K=$(Y_B)^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160$,
B computes K=$(Y_A)^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160$.
We assume an attacker would have available the following information:
q=353, α=3, Y_A= 40, Y_B= 248.
In this simple example, it would be possible by brute force attack to determine the secret key 160. In particular, the attacker E can determine the common key by discovering a solution to the equation $3^a \bmod 353 = 40$ or the equation $3^a \bmod 353 = 248$. The brute-force attack is to calculate powers of 3 modulo 353, stopping when result equals either 40 or 248. The desired answer is reached with the exponent value of 97, which provides

$$3^{97} \bmod 353 = 40$$

With larger numbers, problem becomes impractical.