Diffie Hellman Key Exchange

Diffie Hellman Key Exchange

- First published public key algorithm.

- Enables two users to securely exchange a secret key that can be used for subsequent encryption of messages.

- Suppose Alice & Bob wish to exchange a key.

- Both of them agree upon a prime number, ***p*** & an integer, ***g***, which is a primitive root of ***p***.

- g is a generator of order p-1 in the group $< Z_p^* , x >$

- The group and the generator need not be confidential.

- Alice selects a large random integer, x such that $0 \leq x \leq p-1$, computes, $R_1 = g^x \bmod p$ & sends R_1 to Bob.

- Bob selects another large random integer, y such that $0 \leq y \leq p-1$, computes, $R_2 = g^y \bmod p$ & sends R_2 to Alice.

- Both Alice and Bob keeps the x and y values private.

- The values R_1 and R_2 are publicly available to the other side.

- Alice computes the key, ***K*** = $\boldsymbol{R_2}^{\boldsymbol{x}}$ mod ***p***.

- Bob computes the key, ***K*** = $\boldsymbol{R_1}^{\boldsymbol{y}}$ mod ***p***.

- K= $(g^x \bmod p)^y \bmod p = (g^y \bmod p)^x \bmod p = g^{xy} \bmod p$

- Both of these calculations produce identical results.

- Thus both of them have successfully exchanged a secret key value.

- The security of Diffie Hellman lies on the discrete logarithmic problem.

- Let p be a very large prime number, g is a primitive root in group G = < Zp* ,× > and x is an integer, then it’s easy to compute $R_1 = g^x \bmod p$

- Given R_1 ,g, p it’s computationally infeasible to calculate $x = \log_g R_1 \bmod p$.

D-H Key Exchange Example

- Suppose A & B agree to use p = 23 & g = 7.
- A & B select secret keys, x = 3 & y = 6.
- Each of them computes its public key as,

  $R_1 = 7^3 \bmod 23 = 21.$

  $R_2 = 7^6 \bmod 23 = 4 .$
- After exchanging the public keys, the secret key is calculated as,

  By Alice, $K = 4^3 \bmod 23 = 18.$

  By Bob, $K = 21^6 \bmod 23 = 18.$

Discrete Logarithmic Attack

- To make the scheme safe from discrete logarithmic attack,the following are recommended.

- The prime p must be very large.

- Bob and Alice should destroy the values of x and y after they have calculated the secret key.

Man-in-the-Middle Attack

- Suppose Alice & Bob wish to exchange keys & Darth is an attacker. The attack proceeds as follows.

  - Alice chooses x,calculates $R_1 = g^x \bmod p$ and sends R_1 to bob.
  - Darth intercepts R_1 .
  - Darth chooses z, calculates $R_2 = g^z \bmod p$ and sends R_2 to both Alice and Bob.
  - Bob chooses y, calculates, $R_3 = g^y \bmod p$ and sends R_3 to Alice.
  - R_3 is intercepted by Darth and never reaches Alice.

- Alice and Darth calculates, $K_1 = g^{xz} \bmod p$ and it's the shared secret key between Alice and Darth.

- Bob and Darth calculates, $K_2 = g^{zy} \bmod p$ and it's the shared secret key between them.

- Alice and Bob never knows about it.

- Alice & Bob think that they share a secret key.
- Actually, Alice & Darth share secret key K_1 & Alice & Bob share secret key K_2.
- All future communication is compromised .

- This problem occurs because this algorithm does not authenticate the participants.

- This can be overcome with the use of digital signatures.