Title: Diffie-Hellman (D-H) key exchange calculations

Technology: Security: IPSec VPN

Related Courses: IINS, SNRS, SNAF and SNAA

Submitted by Pat Lao (R&S CCIE/CCSP/CISSP/CCSI) from Learning@Cisco

DH is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher. Learn more about DH by reading this short paper on it!

The Diffie-Hellman (DH) algorithm is the basis of most modern automatic key exchange methods. The Internet Key Exchange (IKE) protocol in IP Security (IPsec) Virtual Private Networks (VPNs) uses DH algorithms extensively to provide a reliable and trusted method for key exchange over untrusted channels.

Whitfield Diffie and Martin Hellman invented the DH algorithm in 1976. Its security stems from the difficulty of calculating the discrete logarithms of very large numbers. The DH algorithm provides secure key exchange over insecure channels and is frequently used in modern key management to provide keying material for other symmetric algorithms, such as DES, 3DES or AES

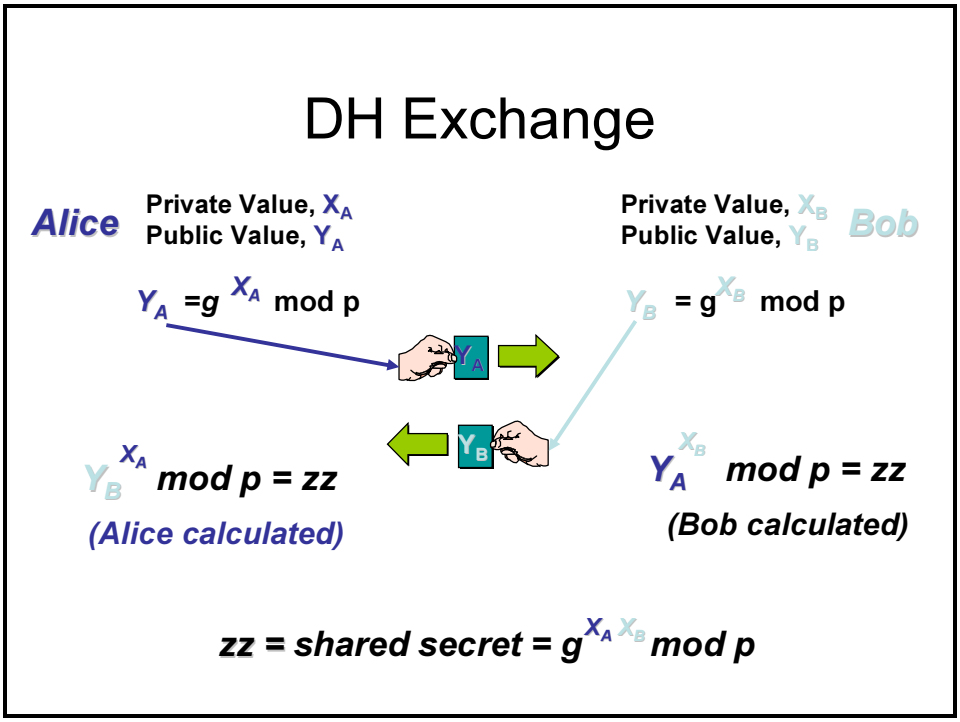
In order to start a DH exchange, the two parties must agree on two nonsecret numbers. The first number is *g,* the generator, and the second number is *p*, the modulus. These numbers can be made public and are usually chosen from a table of known values. *g* is usually a very small number, such as 2, 3, and 4, and *p* is a very large prime number. Next, every party generates its own secret value. Then, based on *g*, *p*, and the secret value of each party, each party calculates its public value. The public value is computed according to the following formula:

$$Y=g^x \bmod p$$

In this formula *x* is the secret value of the entity, and *Y* is the public value of the entity.

After computing the public values, the two parties exchange their public values. Each party then exponentiates the received public value with its secret value to compute a common shared-secret value. When the algorithm completes, both parties have the same shared secret, which they have computed from their secret value and the public value of the other party.

No one listening on the channel can compute the secret value, because only *g*, *p*, Y_A and Y_B are known; at least one secret value is needed to calculate the shared secret. Unless the attacker can compute the discrete algorithm of the above equation to recover X_A or X_B, they cannot obtain the shared secret.



The following steps describe a DH exchange:

Step 1 Alice and Bob agree on generator g and modulus p.

Step 2 Alice chooses a random large integer X_A and sends Bob its public value, Y_A where $Y_A=g^{x(A)} \bmod p$.

Step 3 Bob chooses a random large integer X_B and sends Alice his public value, Y_B, where $Y_B=g^{x(B)} \bmod p$.

Step 4 Alice computes $k=Y_B^{x(A)} \bmod p$.

Step 5 Bob computes $k'=Y_A^{x(B)} \bmod p$.

Step 6 Both k and k' are the equal to $g^{x(A)x(B)} \bmod p$.

Alice and Bob now have a shared secret ($k=k'$) and even if someone has listened on the untrusted channel, there is no way they could compute the secret from the captured information, assuming that computing a discrete logarithm of Y_A or Y_B is practically unfeasible.

Note	More details about the values of g and p can be found in RFCs 2409 and 3526.