



The Diffie-Hellman-Merkle key exchange protocol









A, p, g







1. Generates public numbers  $p$  and  $g$  such that  $g$  is co-prime to  $p-1$
2. Generates a secret number  $a$
3. Sends  $A = g^a \bmod p$  to Bob

1. Generates a secret number  $b$
2. Sends  $B = g^b \bmod p$  back to Alice
3. Calculates the key  $K = A^b \bmod p$

4. Calculates the key  $K = B^a \bmod p$

# The Diffie-Hellman-Merkel key exchange protocol



1. Generates public numbers  $p$  and  $g$  such that  $g$  is co-prime to  $p-1$
2. Generates a secret number  $a$
3. Sends  $A = g^a \bmod p$  to Bob

$A, p, g$

1. Generates a secret number  $b$
2. Sends  $B = g^b \bmod p$  back to Alice
3. Calculates the key  $K = A^b \bmod p$

$B$

4. Calculates the key  $K = B^a \bmod p$

# Diffie-Hellman-Merkle in practice

- $g$  is small (either 3, 5 or 7 and fixed in practice)
  - $p$  is at least 2048 bits (and fixed in practice)
  - private keys  $a$  and  $b$  are 2048 bits as well
- ➔ So the public values  $A$  and  $B$   
and the master key  $k$  are 2048 bits
- ➔ Use  $k$  to derive an AES key using a Key Derivation Function  
(usually HKDF - the HMAC-based Extract-and-Expand key derivation function)