The Diffie-Hellman-Merkel key exchange protocol

A, p, g

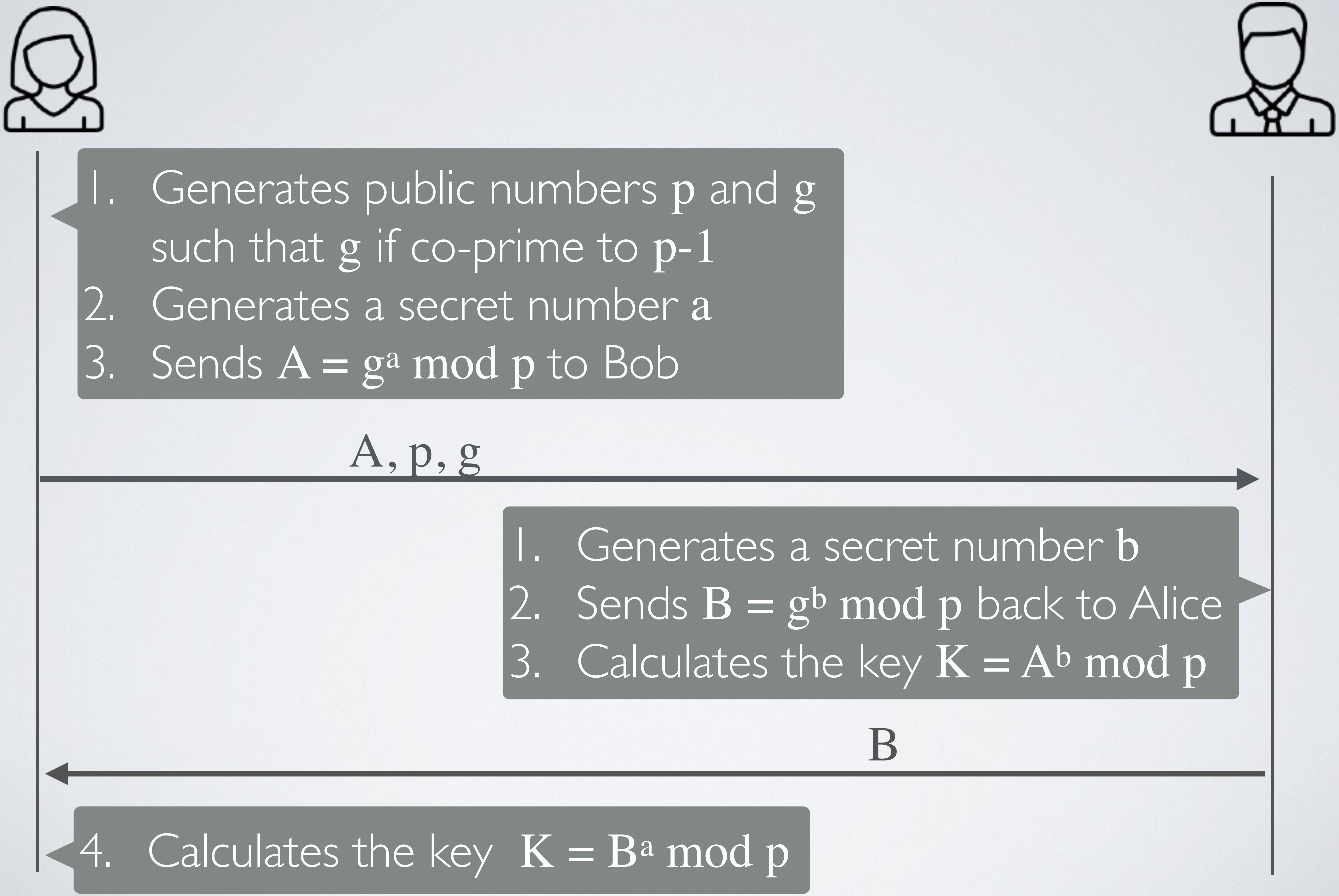
B

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

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

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

The Diffie-Hellman-Merkel key exchange protocol



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

A, p, g

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

B

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

Diffie-Hellman-Merkle in practice

- $\mathbf{g}$ is small (either 3, 5 or 7 and fixed in practice)
- $\mathbf{p}$ is at least 2048 bits (and fixed in practice)
- private keys $\mathbf{a}$ and $\mathbf{b}$ are 2048 bits as well

➡ So the public values $\mathbf{A}$ and $\mathbf{B}$
and the master key $\mathbf{k}$ are 2048 bits

➡ Use $\mathbf{k}$ to derive an AES key using a Key Derivation Function
(usually HKDF - the HMAC-based Extract-and-Expand key derivation function)