ECDH Key exchange

Generate Ks_A, Kp_A

Kp_A

1. Generate Ks_B, Kp_B
2. Derive $k = kex(Ks_B, Kp_A)$

Kp_B, $enc(k, m)$

1. Derive $k = kex(Ks_A, Kp_B)$
2. Decrypt $m = dec(k, enc(k, m))$

Diffie-Hellman-Merkle provides a way to generate a shared key from two asymmetric key pairs

$$kex(Ks_A, Kp_B) = kex(Ks_B, Kp_A) = k$$

- ✓ Mutual contribution to the key generation
- ✓ No need to send the encrypted shared key

A widely used key exchange protocol

ECDH is in many protocols

- SSH
- TLS (used by HTTPS)
- Signal (used by most messaging apps like Whatsapp)
- and so on …

✓ It is fast and requires two exchanges only

◉ But how to make sure Alice is talking to Bob and vice-versa? Diffie-Hellman-Merkle alone **does not ensure authentication**