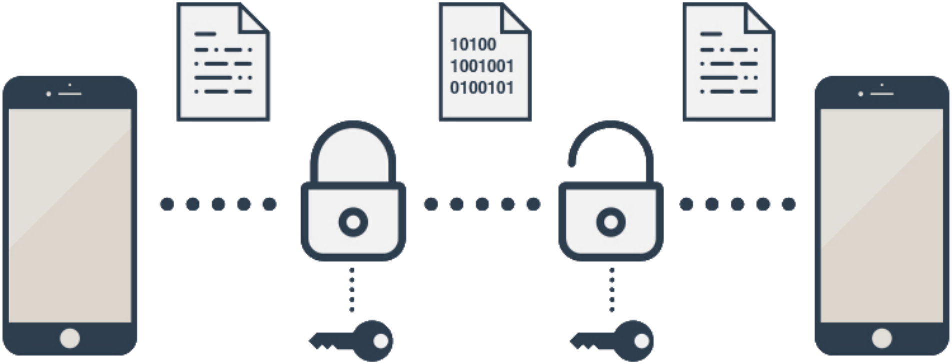
What is...Diffie–Hellman key exchange?

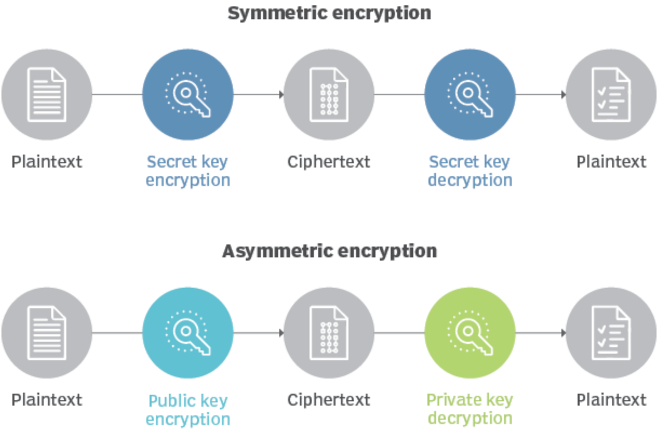
Or: How not to transfer the encryption key

The problems in end-to-end encryption (E2EE)



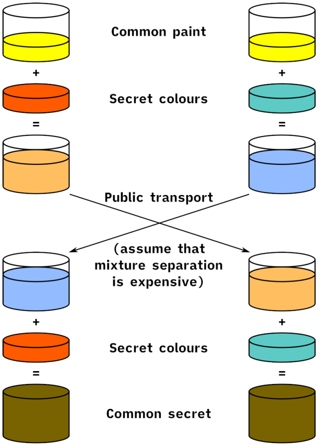
- E2EE Only the two communicating parties should decrypt the message
- Problem How to transfer the encryption key?
- Diffie–Hellman (DH) Addresses this problem

Asymmetry rocks!



- **Symmetric** Both parties us the same secret key
- **Problem (still)** How to transfer the encryption key?
- **Asymmetric** Both parties have a public and a private key, no sharing needed

DH in action



- DH Two secrets a, b, public g, send mix ag or gb and get agb
- Catch Relies on the mixtures to be hard ot decompose
- BTW Using colors is not very practical ;-)

Enter, the theorem/idea

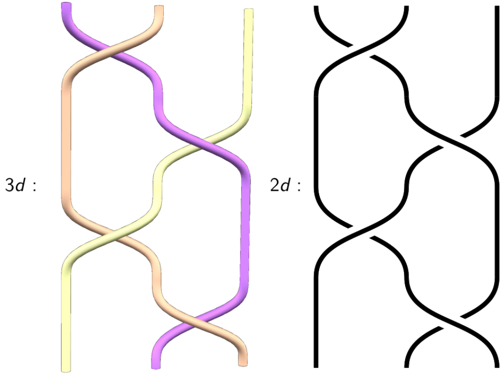
The original DH key exchange:

- Fix $\mathbb{Z}/p\mathbb{Z}$ and $g \in (\mathbb{Z}/p\mathbb{Z})^*$ Public
- Party A fixes $a \in \mathbb{Z}$, party B fixes $b \in \mathbb{Z}$ Private
- Party A sends g^a mod p, party B sends g^b mod p Public
- Party A computes $(g^b \bmod p)^a \bmod p$, party B computes $(g^a \bmod p)^b \bmod p$
  A does not know b and B does not know a
- Common secret $(g^b \bmod p)^a \bmod p = g^{ab} \bmod p = (g^a \bmod p)^b \bmod p$

Theorem/idea

Party C knows only p, g, g^a mod p and g^b mod p, and needs to find g^{ab} mod p
Finding g^{ab} mod p is the discrete logarithm problem which does not appear to have an efficient algorithm (but there are efficient quantum algorithms)

Variation of DH: conjugacy search problem (CSP)



- Group-bases $g^x = xgx^{-1}$ for x in some group G
- Same game, different names $g \in G$ public, $a, b \in G$ private
- Any group G works, but the conjugacy problem should be hard in G
- Proposed candidates include braid groups (albeit these are not optimal)

Thank you for your attention!

I hope that was of some help.