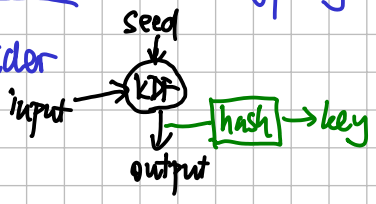
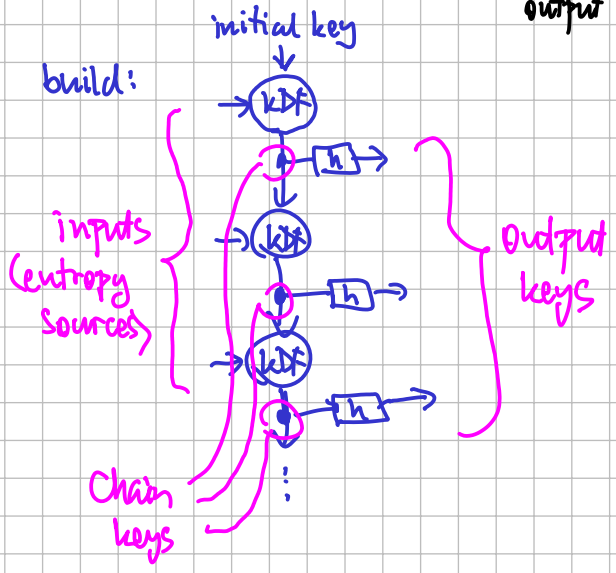
Double Ratchet Protocol for encrypting two-party messaging [e.g. Signal, Matrix]

① KDF chains: consider



build:



Properties:

- output keys appear random to those who don't know chain keys
- fwd secrecy: leaking a chain key doesn't compromise output keys in the past
- recovery from break-in if inputs provide sufficient entropy

② symmetric-key ratchet:

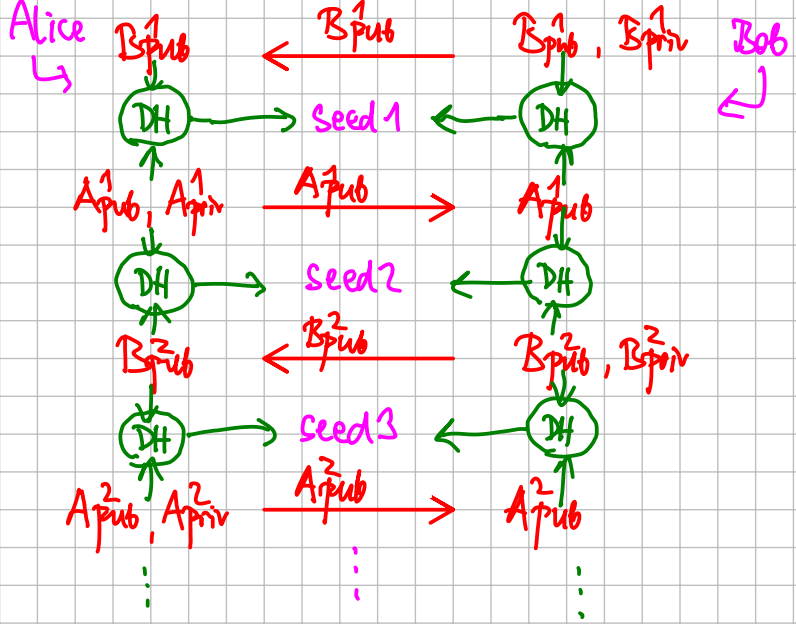
① with constant inputs

↓

one chain for each direction, output keys used to encrypt/sign messages

③ DH ratchet: use DH to supply shared randomness

- each party generates (pub, priv) DH pair
- pub part sent with a message
- my priv + received pub → new random string
- then re-generate my pair & change direction



④ Combine ② with ③

↓

Double Ratchet

provides break-in recovery

- root chain seeded by the DH ratchet

provides fwd secrecy

- whenever root chain makes 1 step, it provides an initial key for a new symmetric chain
- odd chains used for A→B, even chains for B→A
- messages carry chain ID & seq. no on their chain

↓

can cope with reordering

- old chain keys can be deleted