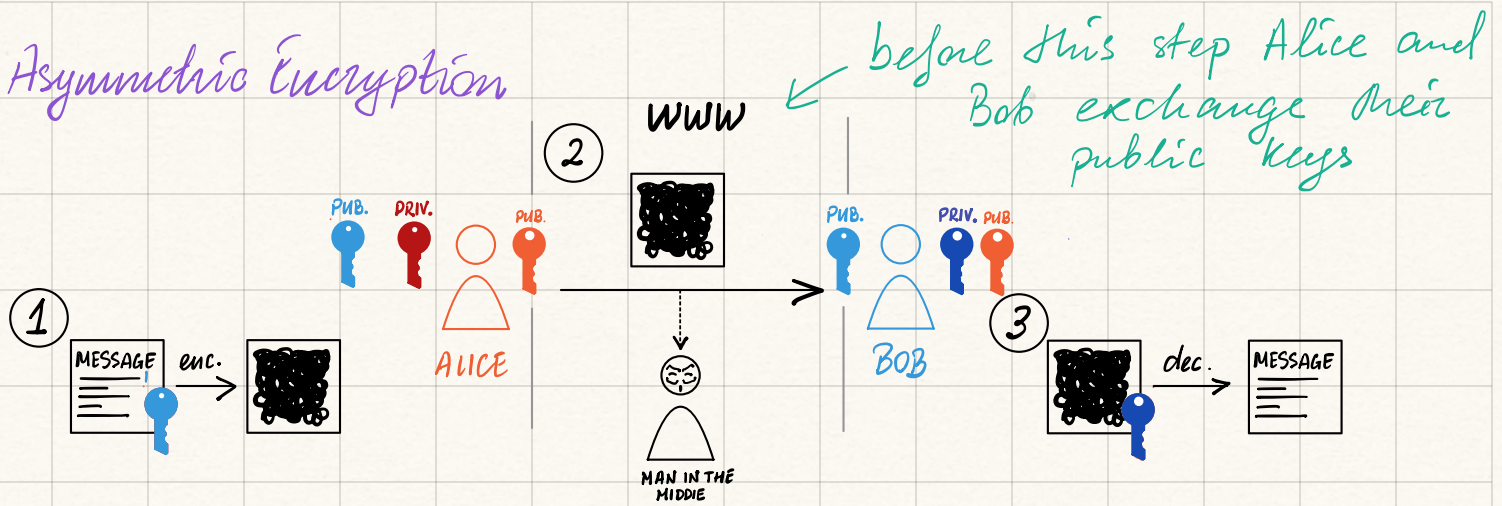



## Asymmetric Encryption



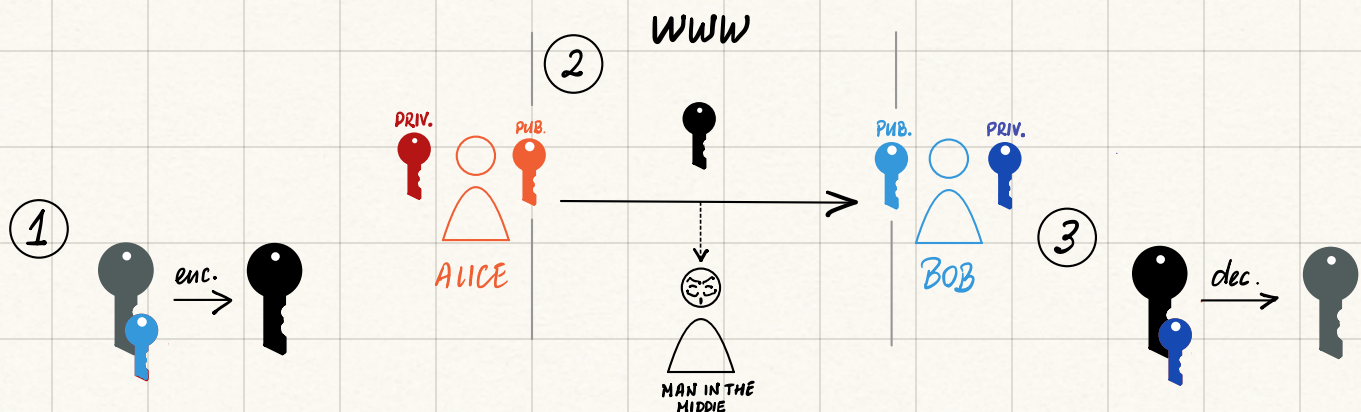
## 3 Pillars of Secure Communication

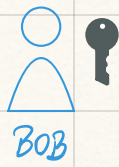
**Confidentiality:** The message is encrypted in-transit

**Authenticity:** Bob knows Alice is the one who sent the message (otherwise  would not have decrypted the message)

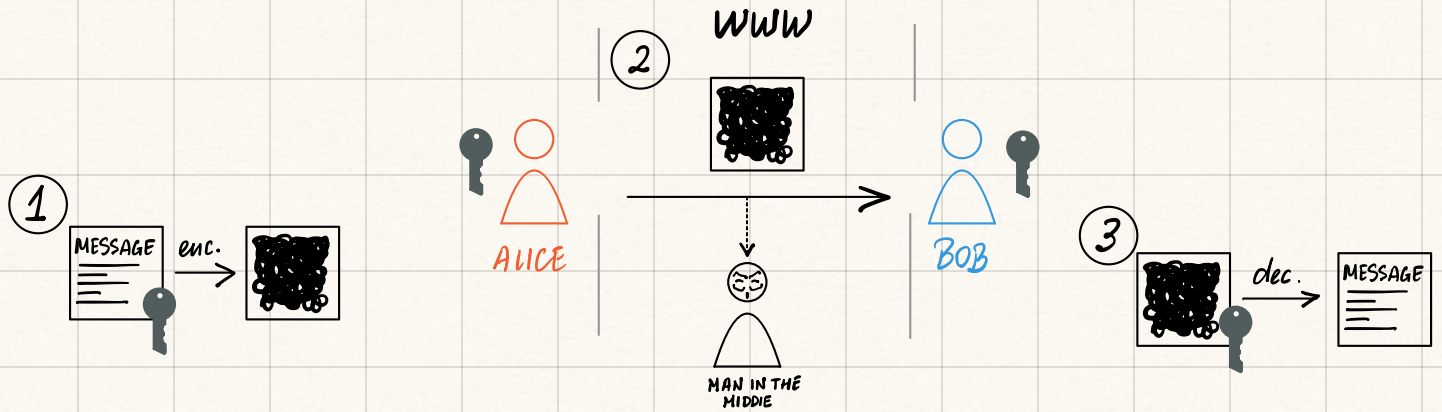
**Integrity:** Bob knows Alice's message has not been altered by the man in the middle (for the same reason)

## Symmetric Encryption





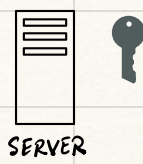
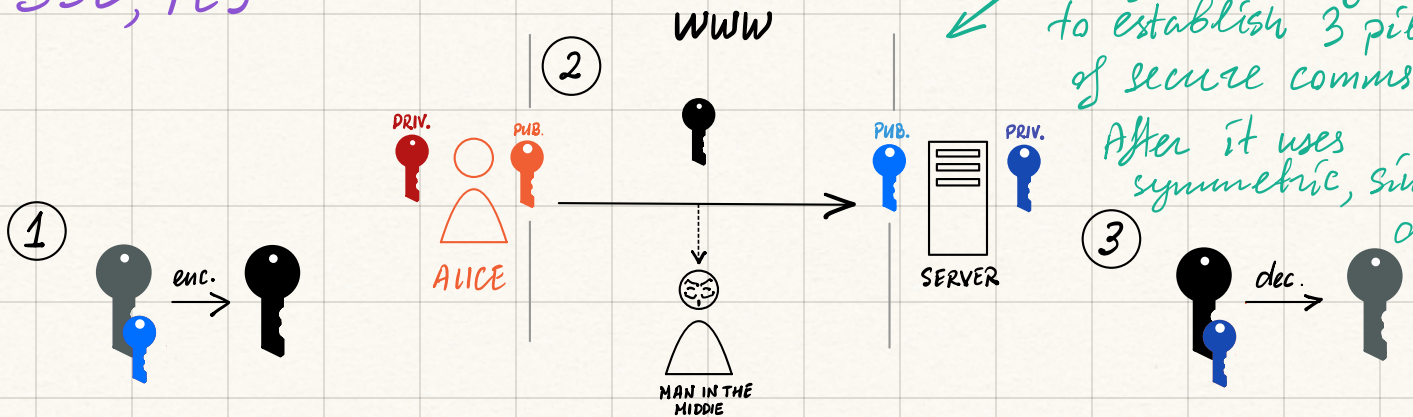
Now Alice and Bob are the only ones that have the symmetric key



SSL, TLS

SSL uses Asymmetric Encryption only once to establish 3 pillars of secure comms.

After it uses symmetric, since that's cheaper.



Now Alice and the Server are the only ones that have the symmetric key.

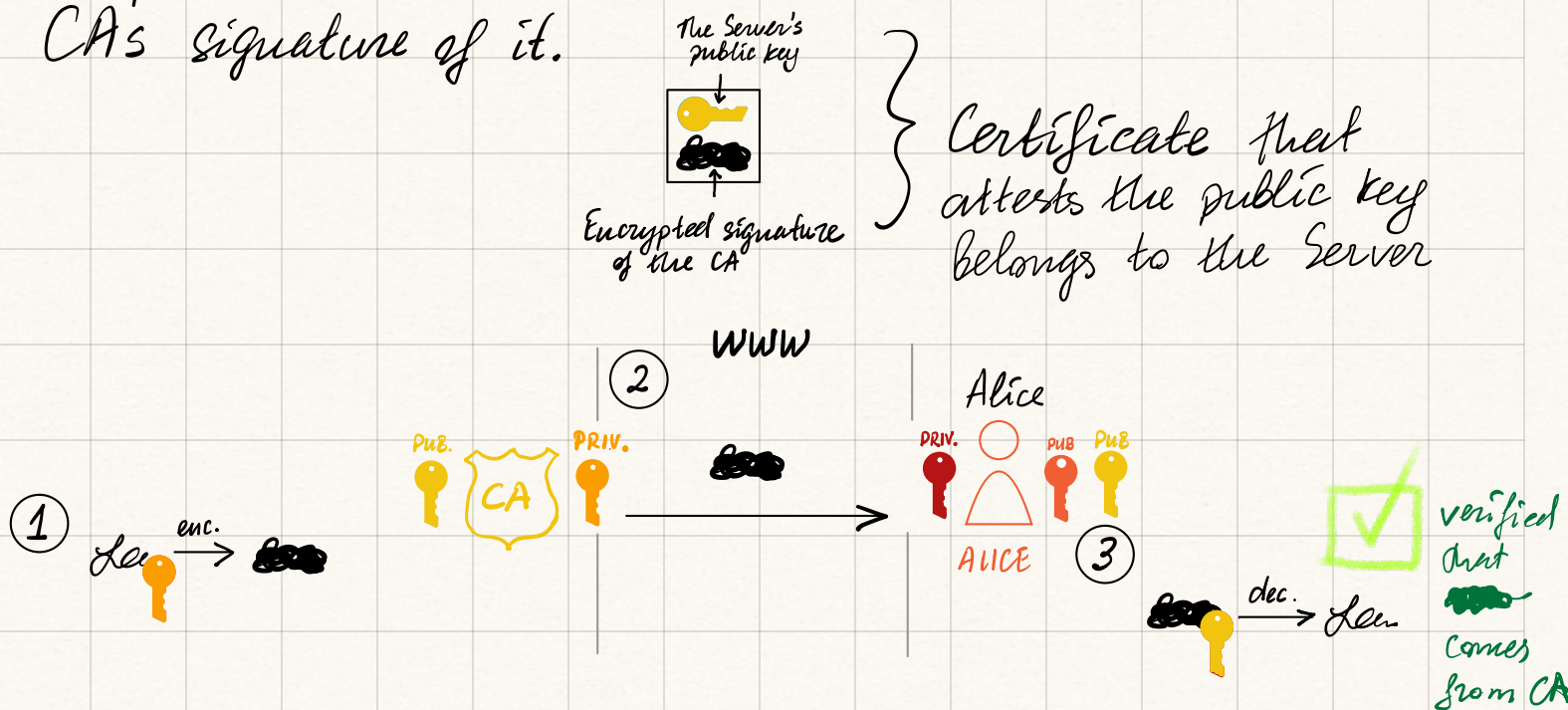
Now there is confidentiality & integrity established between Alice and the Server.

But, unlike in the example with Bob, Alice doesn't know the Server. As anyone can generate a set



of private and public keys, there's no way of knowing if the Server's public key really does belong to the Server. So, when there's **no prior established trust** there is no **authenticity** guarantee.

**Certificate Authority** issues a certificate to the Server consisting of the Server's public key and the CA's signature of it.



Alice trusts the CA so by verifying its signature **authentication** is established.



Now Alice knows that the server is who they say they are and the SSL flow between Alice

and the Server can continue.

To sum up, Asymmetric Encryption is used twice in the SSL flow. Once to establish a Symmetric key and another time to verify with the Certificate Authority.