



Asymmetric encryption for integrity

Alice encrypts a message  $m$  with her private key  $K_{s_A}$

➔ Everybody can decrypt  $m$  using Alice's public key  $K_{p_A}$

✓ Authentication with non-repudiation (a.k.a Digital Signature)







KsA, KpA

KpA



KpA

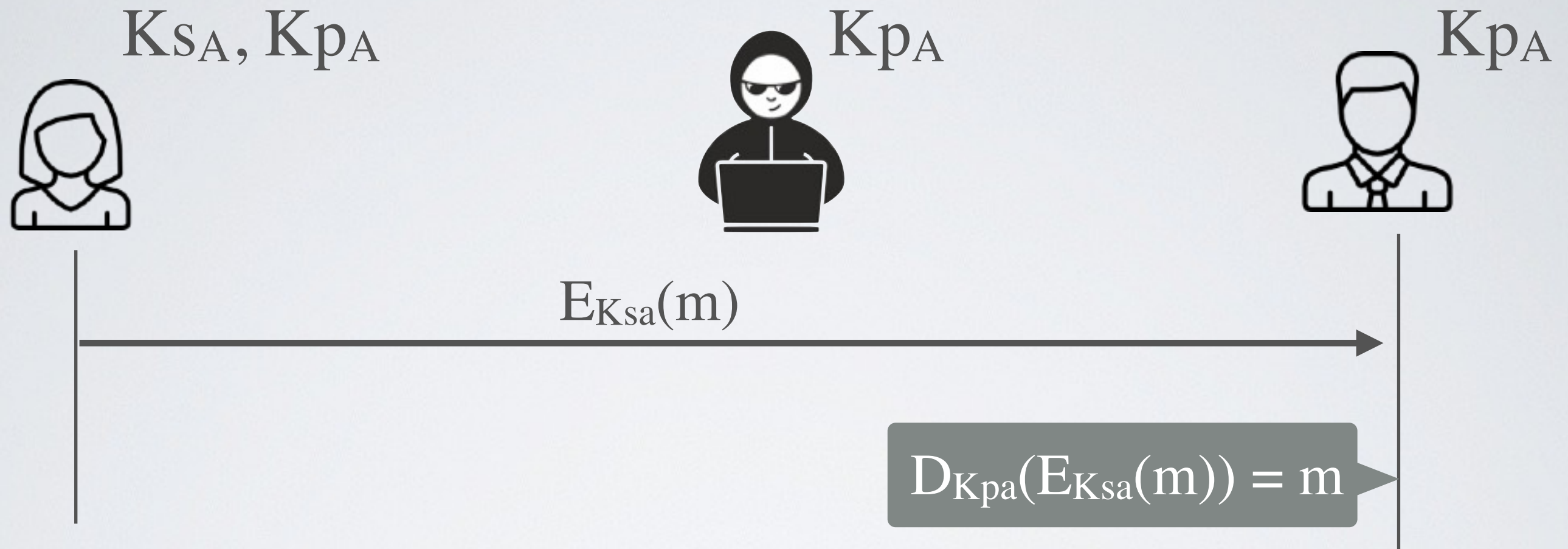





$$E_{Ksa}(n)$$

$$D_{Kpa}(E_{Ksa}(m)) = m$$

# Asymmetric encryption for **integrity**



Alice encrypts a message  $m$  with her private key  $K_{SA}$

➔ Everybody can decrypt  $m$  using Alice's public key  $K_{PA}$

✓ Authentication with non-repudiation (a.k.a Digital Signature)

# Functional Requirements

$D_{K_s}(E_{K_p}(m)) = m$  and  $D_{K_p}(E_{K_s}(m)) = m$  for every pair  $(K_p, K_s)$

- ✓ Generating a pair  $(K_p, K_s)$  is easy to compute (polynomial)
- ✓ Encryption is easy to compute (either polynomial or linear)
- ✓ Decryption is easy to compute (either polynomial or linear)
- Finding a matching key  $K_s$  for a given  $K_p$  is hard (exponential)
- Decryption without knowing the corresponding key is hard (exponential)