Cryptography Protocols

Thierry Sans

Security goals vs attacker's model

Interception
Modification
Fabrication
Interruption

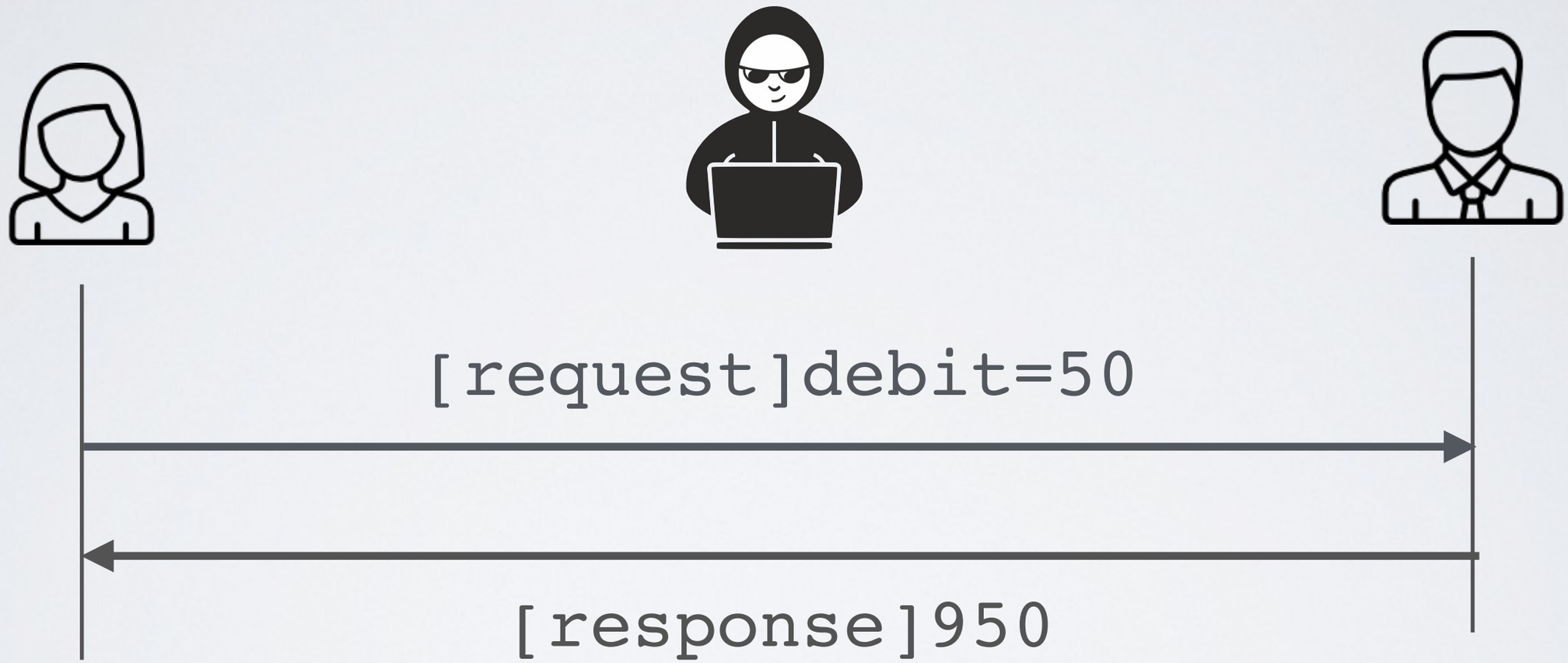
Let us consider **confidentiality, integrity ~~and availability~~**

Design of a cryptography protocol

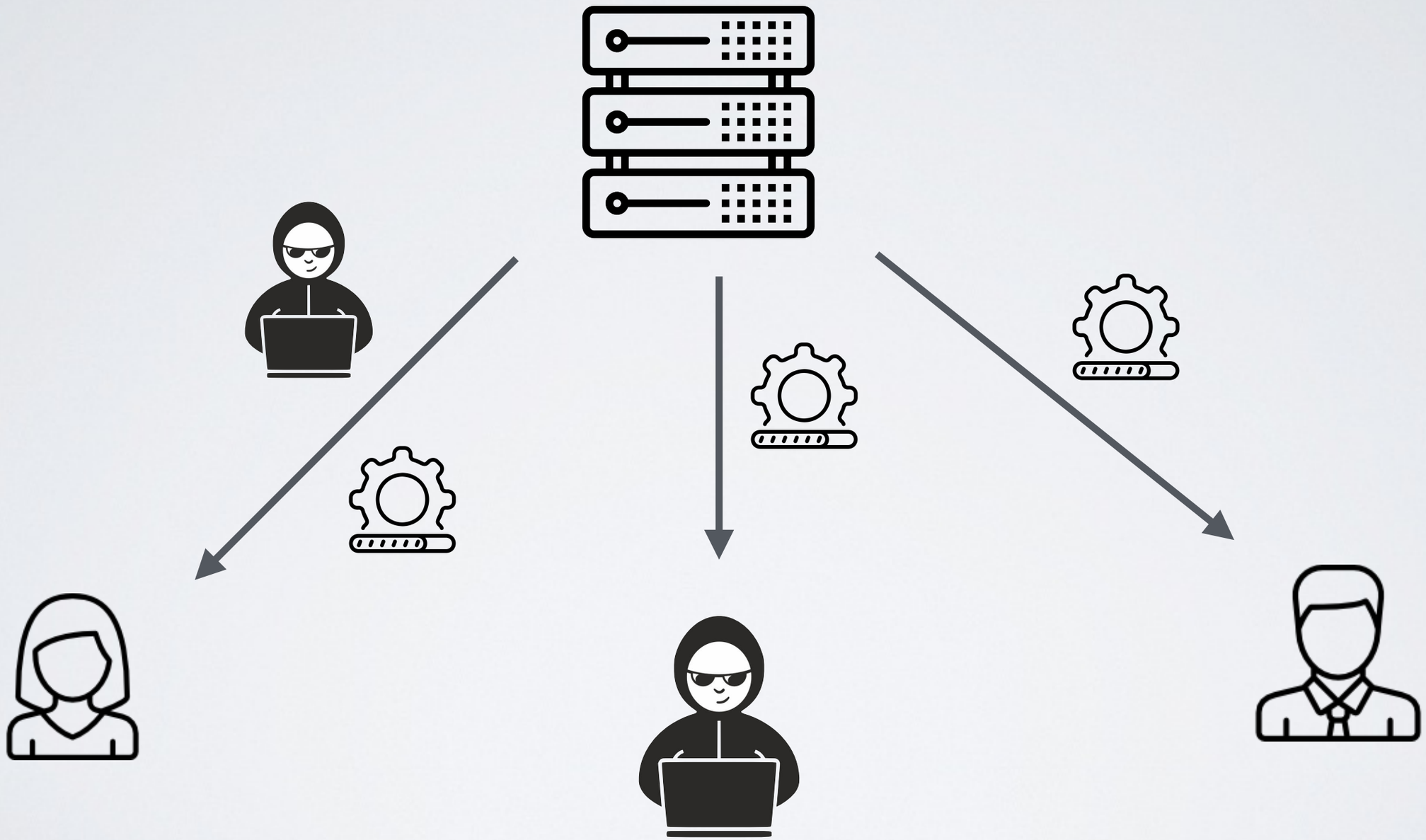
The hypothesis on the system

- **What is the network model?**
  bandwidth, latency, reliability, message ordering, synchronous vs asynchronous
- **What trusted setup is assumed?**
  pre-shared keys, key generation
- **How powerful are the parties vs. attacker?**
  computing power, source of randomness
- **Which adversary model is considered?**
  outsider vs insider, passive vs active, man-in-the-middle, man-at-the-end, corruption
- **What kinds of failures are tolerated?**
  crash faults, byzantine faults
- **What exact security properties are being claimed?**
  confidentiality, integrity, authentication, non-repudiation, forward secrecy

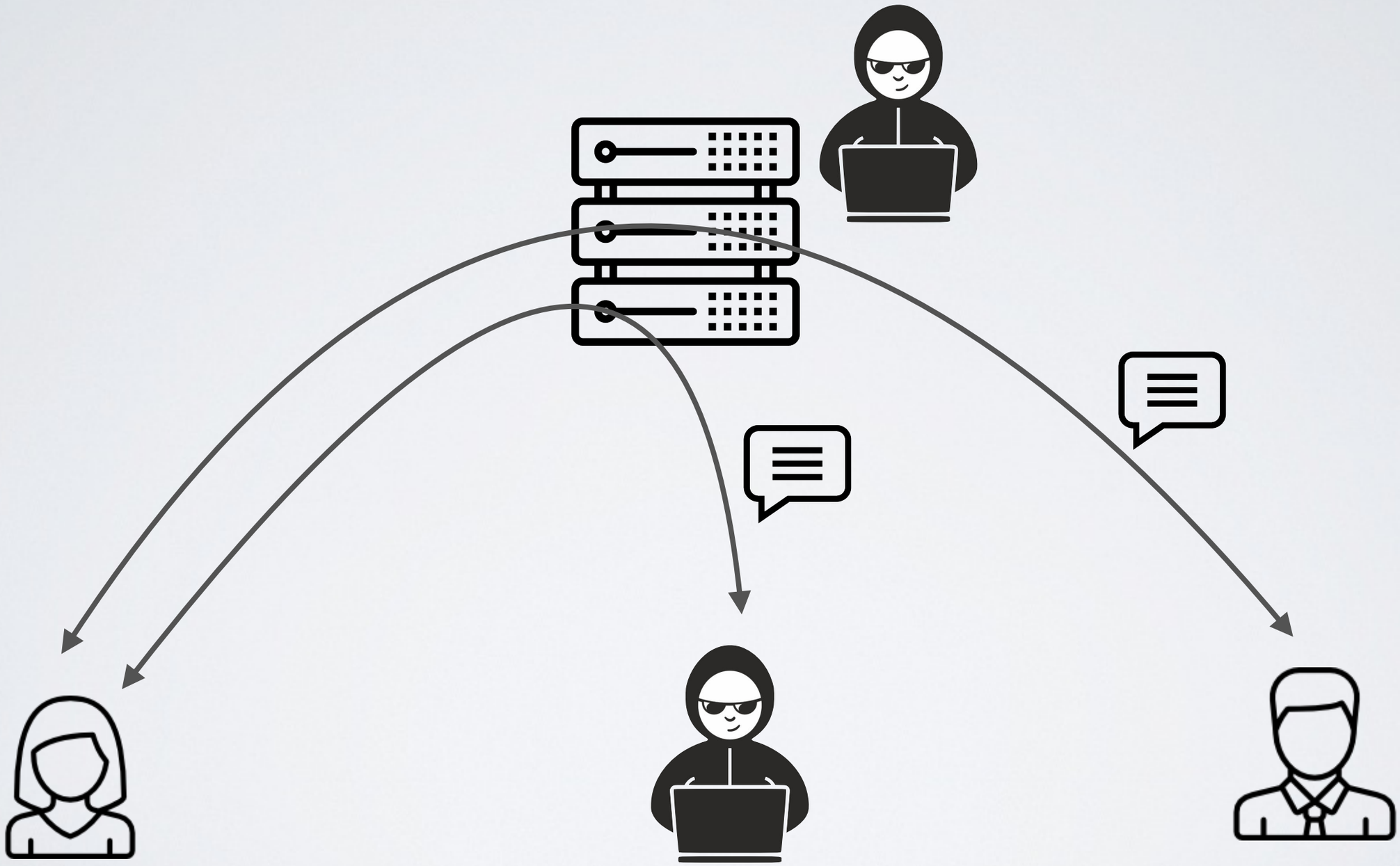
Example 1 - **Interactive Protocol**



Example 2 - **Distribution Centre**

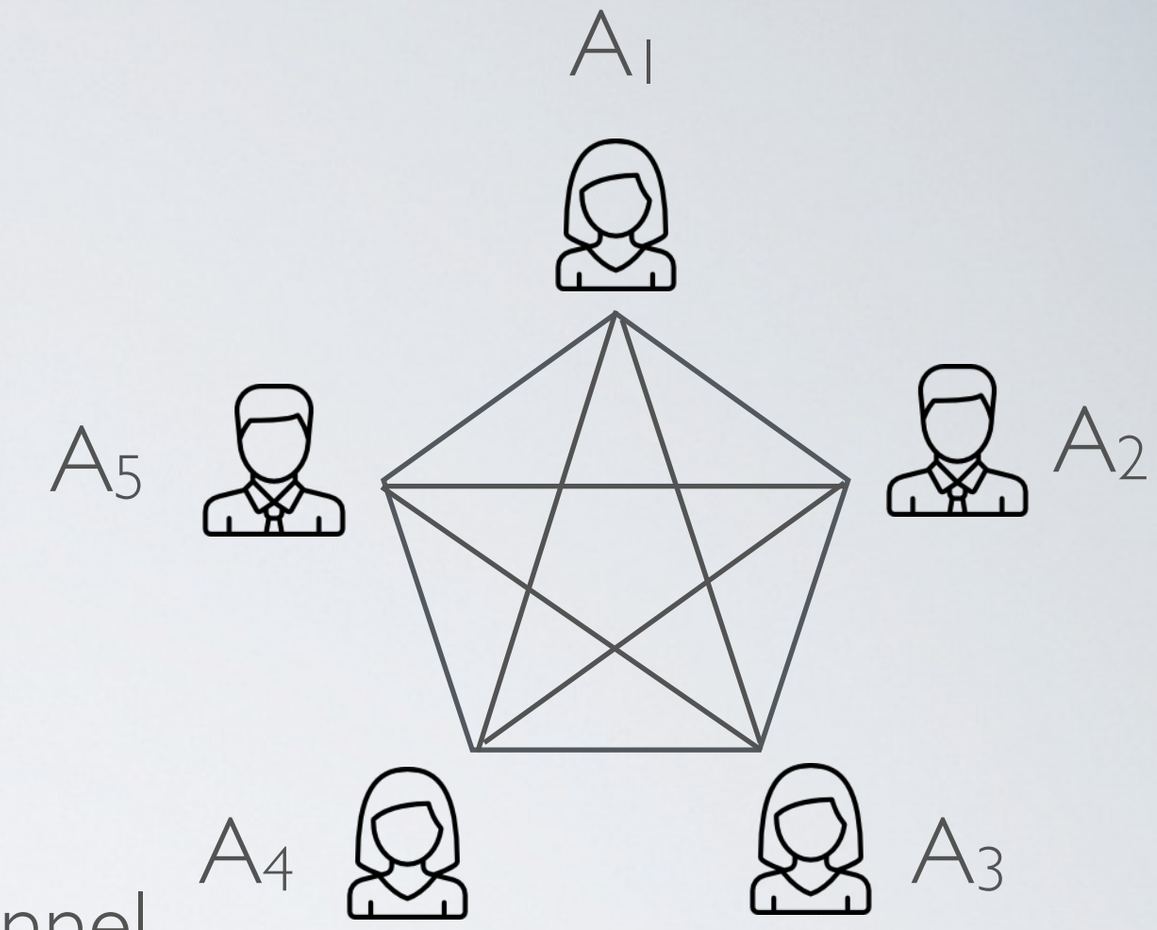


Example 3 - **Asynchronous Messaging**



Replay attacks

Interactive Protocol



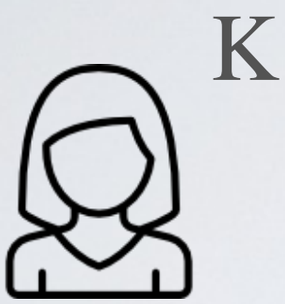
System Hypothesis

- Synchronous communication channel
- Each participant share a unique symmetric key with each other
- Mallory can read, modify and forge message send over the network

Goals

- When two participant exchange a message, the system should protect the confidentiality and integrity of the message

Using Authenticated Encryption



K

K

A, $E_k(m)$

$E_k(m')$

Problem : replay attack

A, $E_k(m)$

$E_k(m')$

A, $E_k(m)$

$E_k(m'')$

Counter replay attacks

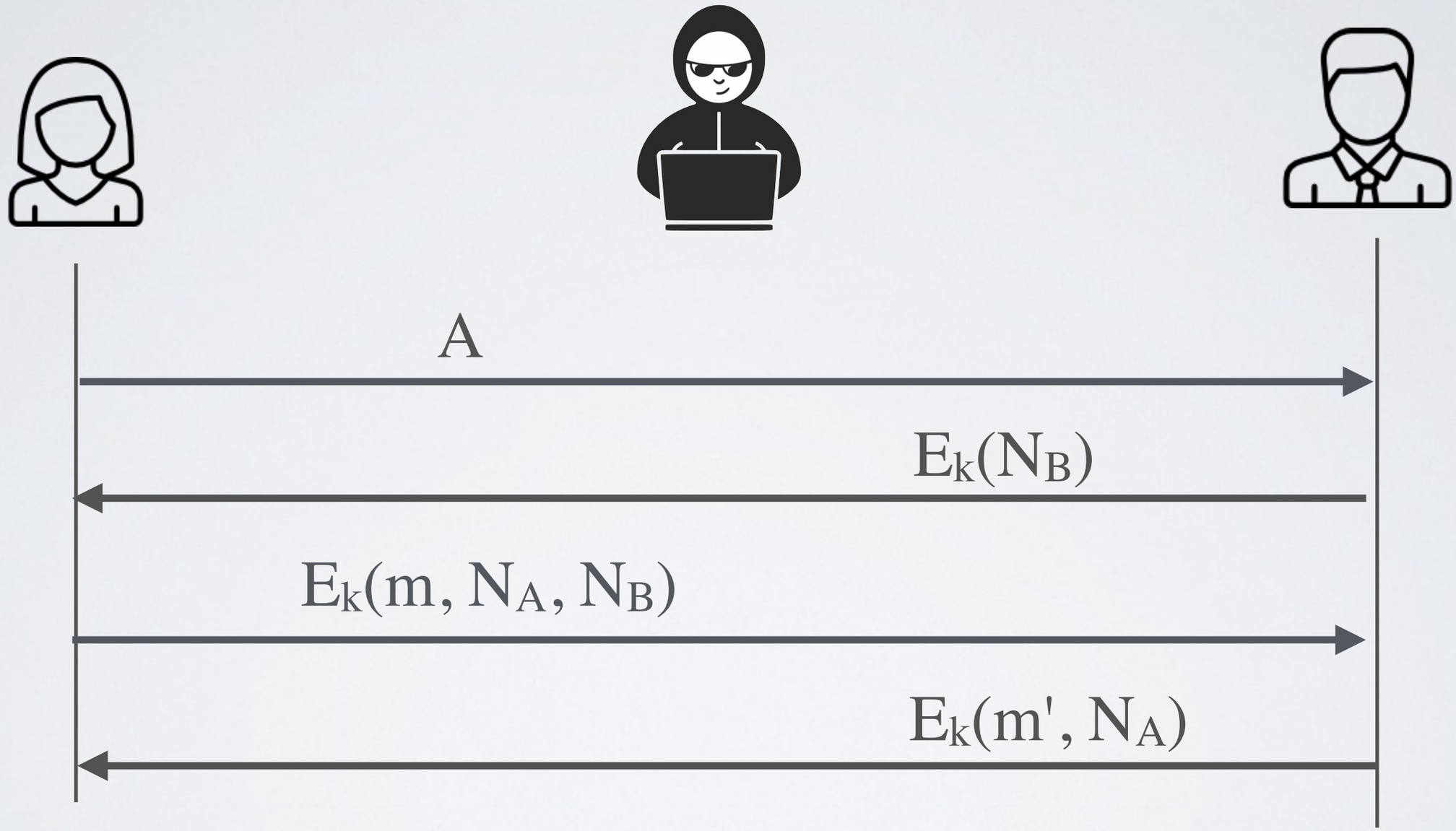
✓ **Storage-based solution**
Store the message entirely (log), or ID or encryption nonce and check whether the same message has been replayed

◉ Problem: this solution can be expensive

✓ **Protocol-based solution**
Add a nonce in the interaction and verify that the nonce is sent back

➡ The nonce should be random enough that is does not repeat itself over time

Double Nonce Protocol



Are we secure yet?

Two major issues:

1. **Key distribution**
   If A1, A2 … A5 want to talk, then (n x (n-1) / 2) keys must be exchanged physically using a secure channel

2. Does not ensure **Perfect-Forward Secrecy**
   If somehow Mallory is able to compromise one of the participant at some point in time, she can decrypt all previous communications between Alice and Bob

Session Keys

Interactive Protocol

System Hypothesis

- Synchronous communication channel
- **Each participant has a public key pair and everybody knows everyone's public keys**
- Mallory can read, modify and forge message send over the network

Goals

- When two participant exchange a message, the system should protect the confidentiality, integrity **and perfect forward secrecy** of the messages

[broken] Key Wrapping

pk_B

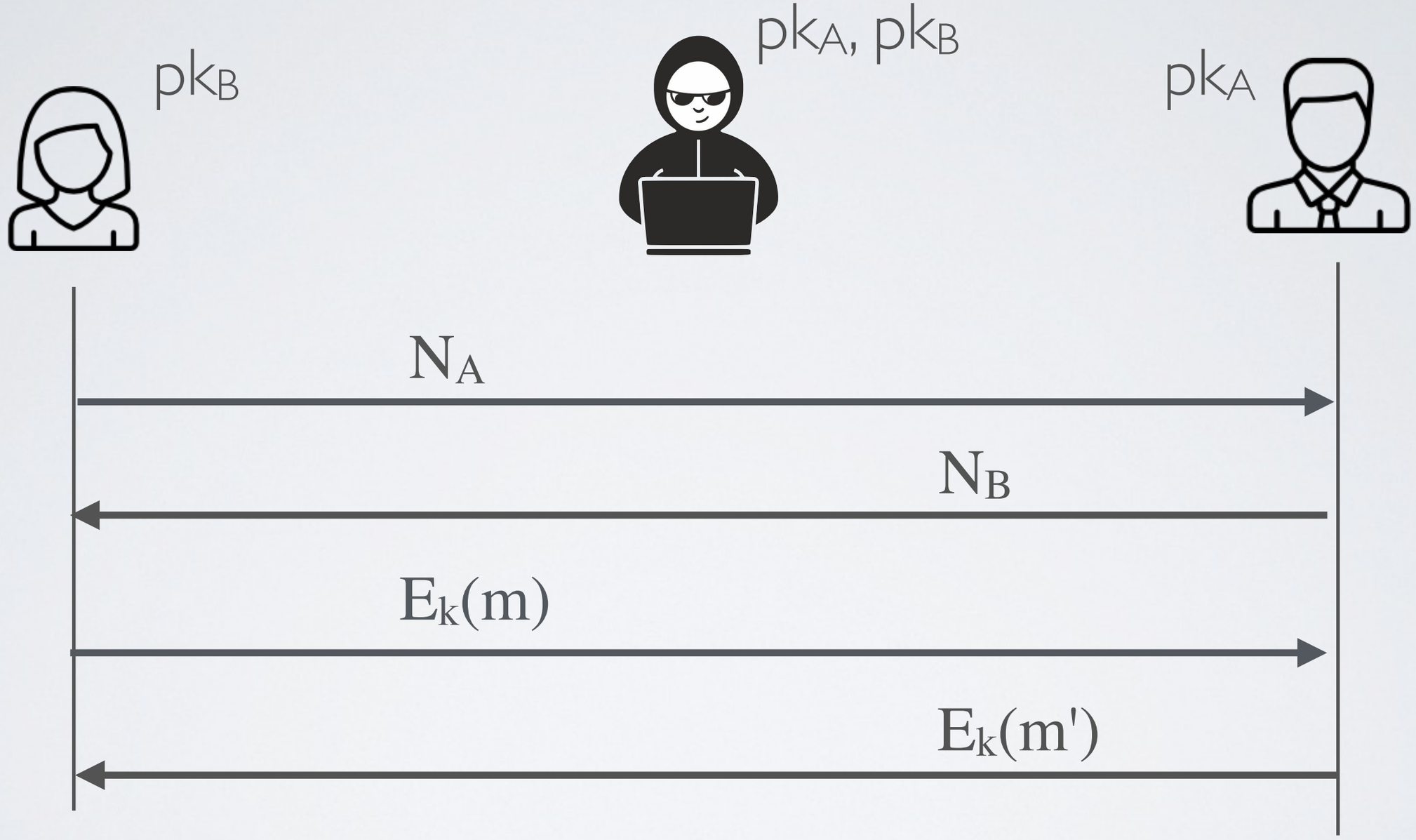
pk_A, pk_B

pk_A

$E_{pkB}(k), E_k(m)$

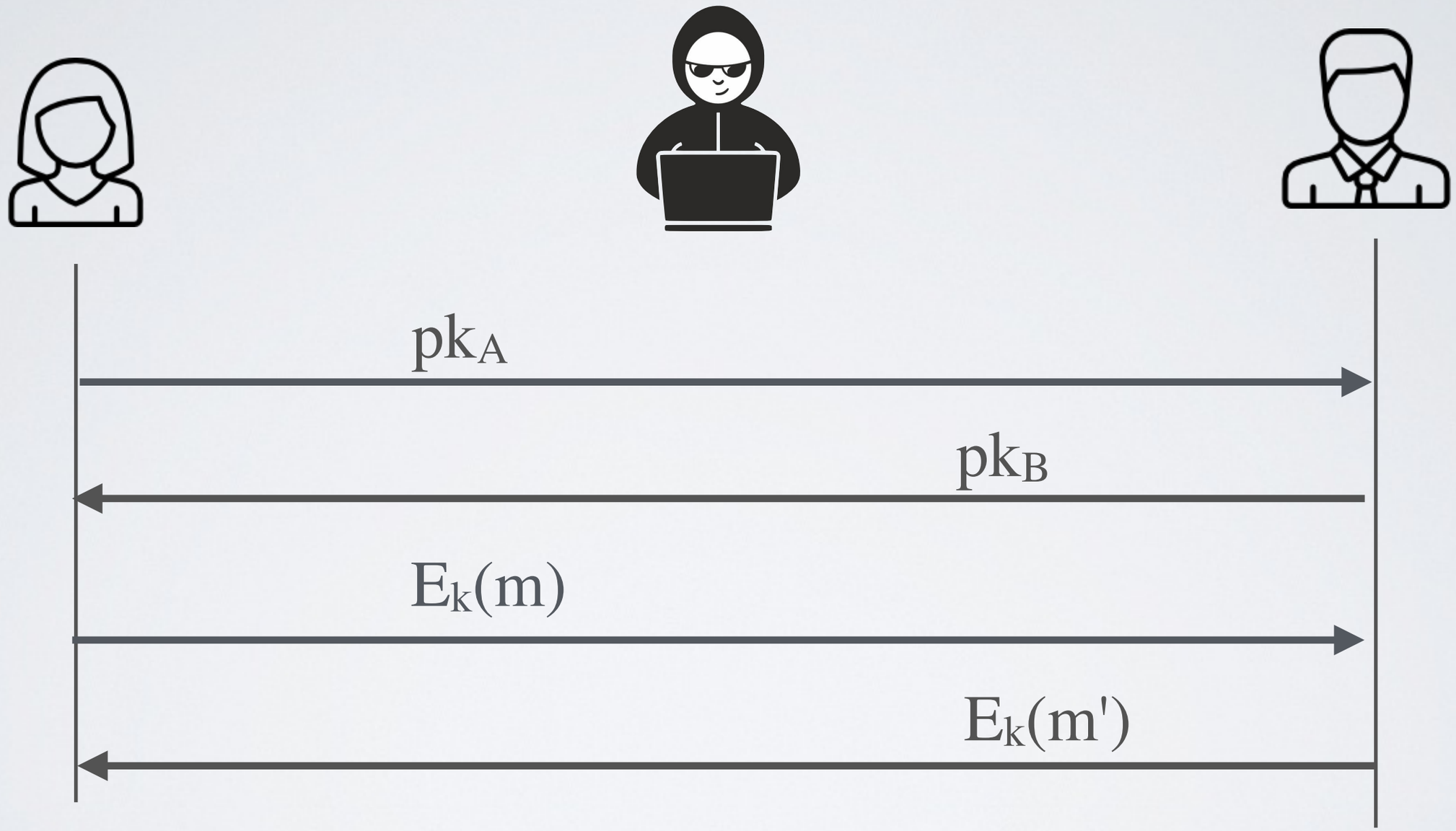
$E_{pkA}(k'), E_k(m')$

[broken] Key Derivation using Long-Term Keys



$$k = ECDH(sk_A, pk_B, N_A, N_B) = ECDH(sk_B, pk_A, N_A, N_B)$$

[broken] Key Derivation using Short-Term Keys



$$k = ECDH(sk_A, pk_B) = ECDH(sk_B, pk_A)$$