Alice: give me your PK → Bob

PK_b

M ← Enc(PK_b, m) → m

Attacker MiTM

PK_adv ← PK_b

Enc(PK_adv, m) ← Enc(PK_b, M)

Trusted directory:

random
drawn
cylic
d by Alice
Alice
PK_TD

random
nonce sent by PK Bob
PK_b, signed (PK_b) (Bob's
nonce)
MITM

name
PK
Bob
0x52.... 0x2e

MITM

sign(PK_TD) is MITM

Updating a key

Assume update happens securely

Replay attack:

Attacker replays old information
(old sig with old PK)
Alice embeds nonce in her request checks sig from TA to contain nonce & to verify with PK_TA & contains Bob's name ⇒ knows PK of Bob is latest

Drawbacks of TA

- Scalability (store & serve all PKs)
- TA is a central point of attack/trust
- difficult to recover from TA compromise
- updating key requires trust
- TA has to be always available - central point of failure
Approach 2: Digital Certificates

- association between name & PK by a CA (certificate authority)
  eg. Vensign

Certificate: $\text{sign}(SK_{CA}, \text{Bob's PK is } 0x54... \rangle$ with expiry date $\Rightarrow$ $\text{cert}_{\text{Bob}}$

Assume browsers have $\text{PK}_{\text{CA}}$ hardcoded

[+ anyone can serve $\text{PK}_{\text{Bob}}, \text{cert}_{\text{Bob}}$

Alice checks:
- $\text{cert}_{\text{Bob}}$ verifies with $\text{PK}_{\text{CA}}$
  is not expired, is for Bob

Alice no longer contacts TA to fetch $\text{PK}_{\text{Bob}}$, but can contact local server, e.g. Bob's Server