0 TODO: add link to comp systems summary <15-10-20, yourname> 0

[[toc]]

Worst case assumptions

- interfaces are exposed: e.g. socket interface is open to the public
- networks are insecure: messages can be looked at, falsified, copied
- limit the lifetime and scope of each secret
- algorithms and program code are available to attackers:
  - the larger the secret the greater the risk of its disclosure
  - open source provides benefit in finding security problems, and scrutiny of methods
- attackers may have access to large resources
  - note Moore's law: resources of attackers are likely to grow exponentially over the lifetime of the product
- minimise the trusted base

Cryptography

Standard participants

- Alice, Bob, Carol, Dave: general participants
- Eve: eavesdropper
- Mallory: malicious
- Sara: server

Notation

- k_B Bob's secret key
- k_{AB} shared secret key between Alice and Bob
- $k_{A\,\text{priv}}$ Alice's **private** key: known only to Alice
- $k_{A\text{pub}}$ Alice's **public** key: shared freely
- $\{M\}_k$ message M **encrypted** with key k
- $[M]_k$ message M **signed** with key k
- $\{M\}_k = E(M,k), M = D(\{M\}_k, k)$
  - E encryption algorithm
  - D a decryption algorithm

Scenario 1: Ensuring Secrecy

Alice and Bob share a secret key k_{AB} encryption/decryption algorithm. If the decrypted message makes sense or contains an agreed upon-value (checksum etc).

Bob can be confident: - the message came from Alice - the message hasn't been tampered with

Issues: - how to securely **send the shared key**? - how can Bob know any message is not a **replay**?

Alice needs to send something with the message so that Bob can verify it isn't a replay

Scenario 2: Authentication

Alice wants to access Bob's resource. Sara is a securely managed authentication server. Sara issues passwords to all users, and knows k_A, k_B, as they are derived from the passwords.

- Alice sends a plaintext message to Sara stating identity and requesting a ticket for access to Bob
- Sara sends a ticket to Alice encrypted with k_A containing ticket encrypted by k_B, and a new secret key k_{AB}

0 TODO: <15-10-20, yourname> 0

Issues: - how to trust server? - how to enrol in the system?

Scenario 3: Challenge-response

- common use: avoid sending passwords in the clear

Scenario 4: Authenticated communication with public keys

- Alice accesses key distribution service Sara to obtain a **pub-key certificate** `Cert` giving Bob's public key
  -

Digital Signature

- **digital signature**: binds an identity to a message
  - for public/private key exchange, the *identity* is the key pair itself
- **digest**: maps an arbitrary message to a fixed length message

Certificates