# Secure Systems

Goal: Safety net approach
Protection as a negative goal

Design principles

- Economy of mechanism: simplicity
- Fail-safe defaults:
  permission, not exclusion
- Complete mediation: check everything
- Open design
- Explicitness: assumptions apparent
- Least privilege: "need-to-know"
- Least common mechanism:
  minimize shared mechanisms to
  reduce potential information paths
- Psychological acceptability: ease of use
- Feedback and interaction in process

# Confidentiality in shared systems

Virtual memory protection

- Distinct information paging:
  all references go through page map, authority checks memory location for each access.
- Distinct address space:
  all memory references through page map address register.
- Permission: user and kernel mode bits for processes.
- Protection of permission bit

## Confidentiality via cryptography

Sealing:

1. Symmetric: shared key K

   Alice: $C \leftarrow seal(M, K)$

   Alice: Send ciphertext C to Bob

   Bob: $M \leftarrow unseal(C, K)$

2. Asymmetric: public-key crypto

   public key – sealing

   private key – unsealing

   Alice: public $K_A$, private $K^{-1}_A$

   Bob: public $K_B$, private $K^{-1}_B$

   Alice: $C \leftarrow seal(M, K_B)$

   Alice: Send C to Bob

   Bob: $M \leftarrow unseal(C, K^{-1}_B)$

# Threat model: types of attacks

1. Ciphertext-only attack
   Eve sees $C_1 \ldots C_n$

2. Known-plaintext attack
   Eve sees $\{M_1, C_1\} \ldots \{M_n, C_n\}$

3. Chosen-plaintext attack
   Lucifer chooses $M_1 \ldots M_n$
   Lucifer sees $\{M_1, C_1\} \ldots \{M_n, C_n\}$

4. Adaptive chosen-plaintext
   Lucifer chooses $M_1$
   Lucifer sees $\{M_1, C_1\}$
   …
   Lucifer chooses $M_n$
   Lucifer sees $\{M_n, C_n\}$

5. Chosen-ciphertext
   Lucifer chooses $C_1 \ldots C_n$
   Lucifer sees $\{M_1, C_1\} \ldots \{M_n, C_n\}$

**6.** Adaptive chosen ciphertext

# Sealing algorithms: examples

## 1. One-Time Pad (XOR)

$$C = M \oplus K$$

- Perfect secrecy
- Key random string as long as message
- Key used only once

$$(M_1 \oplus K) \oplus (M_2 \oplus K) = M_1 \oplus M_2$$

## 2. DES: Data Encryption Standard

- Symmetric key cipher
- Cipher Block Chaining (CBC) mode:

$$C_o = IV$$
$$C_i = E(M_i \oplus C_{i-1}, K)$$

Cascading change propagation

Random IV yields different ciphertexts of same message

## 3. RSA

Public-key cryptosystem:

Generate primes p, q
Public modulus $n = p \times q$
Key Generation:
$e\, d \equiv 1 \pmod{(p-1)(q-1)}$

Public key (e, n), private d

Seal: $C \leftarrow M^e \bmod n$
Unseal: $M \leftarrow C^d \bmod n$

Finding d from (e,n) is equivalent to factoring!

Assumption: factoring is hard!

Need for longer keys as computation power increases

# Authentication

1. Message Authentication Codes (MAC)

    - symmetric key primitives

2. Digital signatures

    - public-key primitives:
      public key – verifying
      private key – signing

      Alice: $\sigma \leftarrow \text{sign}(M, K^{-1}_A)$
      Bob: $\{0,1\} \leftarrow \text{verify}(\sigma, K_A)$

    - non-repudiation

# Confidentiality vs. Authentication

Confidentiality only:
    Alice seals her message

Authentication only:
    Alice appends MAC or signature

Both confidentiality and authentication
    Alice:
        1. appends authentication tag
        2. seals (plaintext message, tag)
    Alice:
        1. seals (plaintext message)
        2. appends authentication tag