Security Threats

• We want to protect the services and data offered by distributed systems against security threats
• Four types of security threats
  – **Interception**: An unauthorized party attempts to gain access to a service or data (e.g., breaking into a system and copying data, sniffing passwords being sent over the network)
  – **Interruption**: Services or data become unavailable, unusable, or destroyed (e.g., deletion of data, denial-of-service attacks)
  – **Modification**: Unauthorized changing of data or tampering with a service so that it no longer adheres to its specifications (e.g., intercepting and changing transmitted data, changing a program so that it secretly logs the activities of its user)
  – **Fabrication**: Generation of additional data or activity that would normally not exist (e.g., adding an entry to a password database, replaying previously sent messages)

Security Policies & Mechanisms

• A **security policy** describes which actions the entities in a system are allowed to take and which ones are prohibited
• Security policies are enforced by **security mechanisms**
• Important security mechanisms
  – **Encryption**: transforms data into something an attacker cannot understand
  – **Authentication**: verifies the claimed identity of an entity
  – **Authorization**: allows entities to only access those resources that they are entitled to access
  – **Auditing**: logs which clients accessed what, and which way
Cryptography

- **Cryptography** provides the basis for most computer security mechanisms. It is the art of encoding information in a format that only the intended recipients can decode.

![Diagram of encryption and decryption process]

Types of Cryptosystems

- **Symmetric cryptosystems**: a single key is used for both encryption and decryption, i.e., $E_k = D_k$
  - Algorithms are usually fast and suitable for bulk encryption tasks
  - A secure channel is needed so that two communicating parties can establish a shared secret key
  - Examples: DES, IDEA

- **Asymmetric cryptosystems** (or public-key systems): $E_k \neq D_k$ and computing one key from the other is computationally infeasible
  - Each entity has a public/private key pair, where $E_k$ is the public key and $D_k$ is the private key
  - Algorithms are too slow to encrypt large volumes of data
  - Examples: RSA, ElGamal
Data Encryption Standard (DES)

- DES was approved as a US national standard in 1976, and superseded by the Advanced Encryption Standard (AES) in 2002
- The same key is used to encrypt and decrypt a message
- Encryption operates on 64-bit blocks of data with a 56-bit key
  - An input block is transformed into an encrypted output block in 16 rounds, plus an initial permutation and a final permutation
  - Each round uses a different 48-bit subkey derived from the 56-bit key
- Decryption is the same as encryption except that the subkeys are applied in the reverse order when decrypting

(a) The principle of DES. (b) Outline of one encryption round.
Replacements to DES

- **Triple DES or 3DES (1999)**
  - Apply the DES algorithm three times to each data block, 3 keys are used
    
    ciphertext = $E_{k_3}(D_{k_2}(E_{k_1}(plaintext)))$
    
    plaintext = $D_{k_1}(E_{k_2}(D_{k_3}(ciphertext)))$
  
  - Fixed block size: 128 bits
  - Variable key size: 128, 192, or 256 bits
  - A machine that could crack 56-bit DES in one second would take 149 trillion years to crack a 128-bit AES key!