CSC 4103 - Operating Systems
Spring 2008

LECTURE - XX

PROTECTION AND SECURITY - I

Tevfik Koşar

Louisiana State University

April 15th, 2008

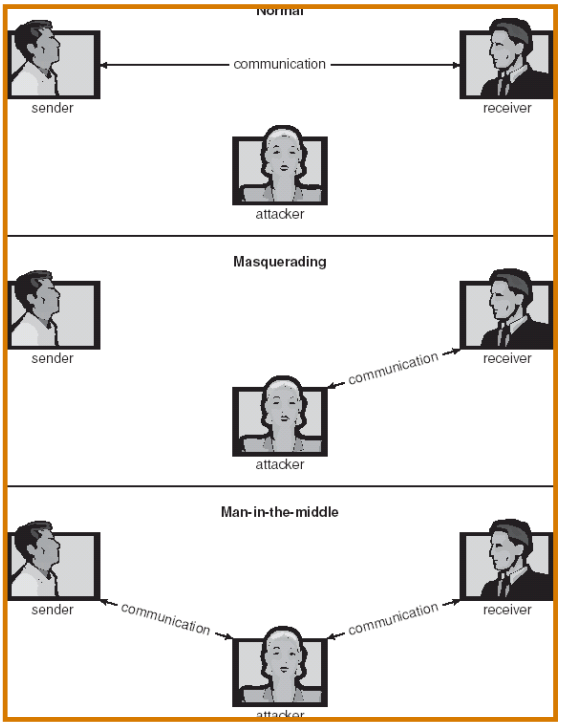
The Security Problem

- Security must consider external environment of the system, and protect the system resources
- Intruders (crackers) attempt to breach security
- **Threat** is potential security violation
- **Attack** is attempt to breach security
- Attack can be accidental or malicious
- Easier to protect against accidental than malicious misuse

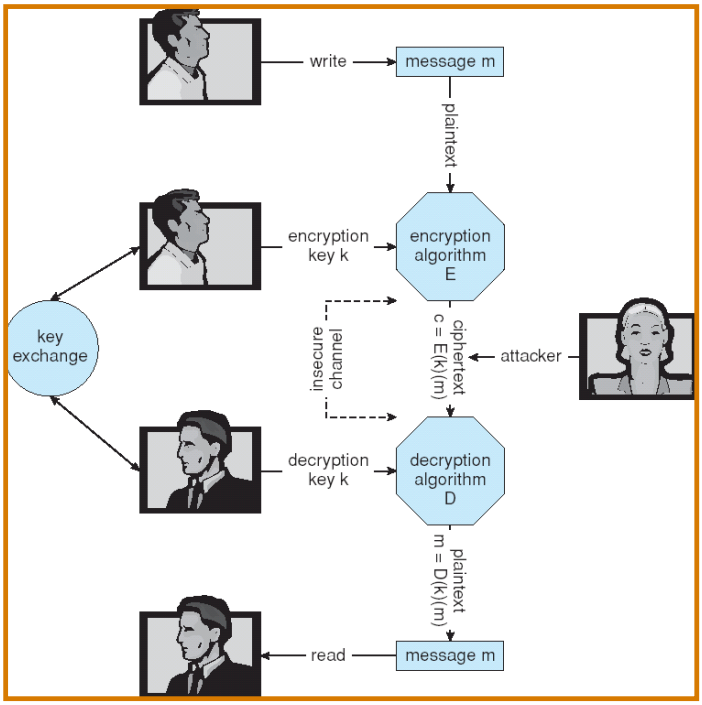
Security Violations

- Categories
  - **Breach of confidentiality** (information theft, identity theft)
  - **Breach of integrity** (unauthorized modification of data)
  - **Breach of availability** (unauthorized destruction of data)
  - **Theft of service** (unauthorized use of resources)
  - **Denial of service** (crashing web servers)
- Methods
  - **Masquerading (breach authentication)**
    - **Pretending to be somebody else**
  - **Replay attack (message modification)**
    - **Repeating a valid data transmission (eg. Money transfer)**
    - **May include message modification**
  - **Session hijacking**
    - **The act of intercepting an active communication session**
  - **Man-in-the-middle attack**
    - **Masquerading both sender and receiver by intercepting messages**

Standard Security Attacks



Secure Communication over Insecure Medium



Encryption

- Encryption algorithm consists of
  - Set of K keys
  - Set of M Messages
  - Set of C ciphertexts (encrypted messages)
  - A function $E : K \rightarrow (M \rightarrow C)$. That is, for each $k \in K$, $E(k)$ is a function for generating ciphertexts from messages.
  - A function $D : K \rightarrow (C \rightarrow M)$. That is, for each $k \in K$, $D(k)$ is a function for generating messages from ciphertexts.
  -

Encryption

- An encryption algorithm must provide this essential property: Given a ciphertext $c \in C$, a computer can compute m such that $E(k)(m) = c$ only if it possesses $D(k)$.
  - Thus, a computer holding $D(k)$ can decrypt ciphertexts to the plaintexts used to produce them, but a computer not holding $D(k)$ cannot decrypt ciphertexts.
  - Since ciphertexts are generally exposed (for example, sent on the network), it is important that it be infeasible to derive $D(k)$ from the ciphertexts

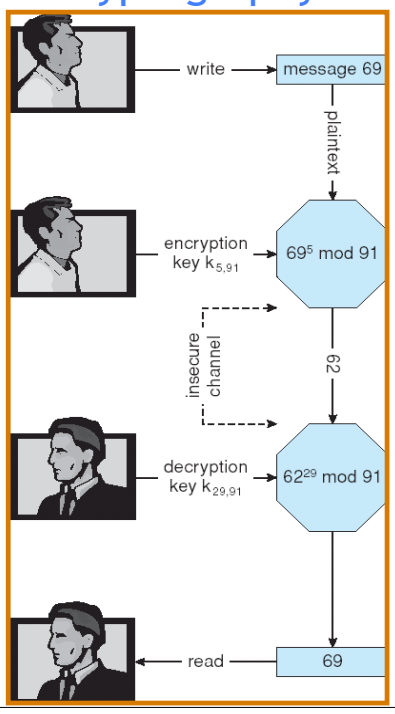
Symmetric Encryption

- Same key used to encrypt and decrypt
  - $E(k)$ can be derived from $D(k)$, and vice versa
- **DES** is most commonly used symmetric block-encryption algorithm (created by US Govt)
  - Encrypts a block of data at a time (64 bit messages, with 56 bit key)
- **Triple-DES** considered more secure (repeat DES three times with three different keys)
- Advanced Encryption Standard (**AES**) replaces DES
  - Key length upto 256 bits, working on 128 bit blocks
- **RC4** is most common symmetric stream cipher (works on bits, not blocks), but known to have vulnerabilities
  - Encrypts/decrypts a stream of bytes (i.e wireless transmission, web browsers)
  - Key is a input to psuedo-random-bit generator
    - Generates an infinite **keystream**

Asymmetric Encryption

- Encryption and decryption keys are different
- Public-key encryption based on each user having two keys:
  - public key – published key used to encrypt data
  - private key – key known only to individual user used to decrypt data
- Must be an encryption scheme that can be made public without making it easy to figure out the decryption scheme
  - Most common is RSA *(Rivest, Shamir, Adleman)* block cipher

Encryption and Decryption using RSA Asymmetric Cryptography



Asymmetric Encryption (Cont.)

- Formally, it is computationally infeasible to derive $D(k_d , N)$ from $E(k_e , N)$, and so $E(k_e , N)$ need not be kept secret and can be widely disseminated
  - $E(k_e , N)$ (or just k_e) is the **public key**
  - $D(k_d , N)$ (or just k_d) is the **private key**
  - N is the product of two large, randomly chosen prime numbers p and q (for example, p and q are 512 bits each)
  - Select k_e and k_d, where k_e satisfies $k_e k_d \bmod (p-1)(q-1) = 1$
  - Encryption algorithm is $E(k_e , N)(m) = m^{k_e} \bmod N$,
  - Decryption algorithm is then $D(k_d , N)(c) = c^{k_d} \bmod N$

Asymmetric Encryption Example

- For example. choose $p = 7$ and $q = 13$
- We then calculate $N = 7*13 = 91$ and $(p-1)(q-1) = 72$
- We next select k_e relatively prime to 72 and< 72, yielding 5
- Finally,we calculate k_d such that $k_e k_d \bmod 72 = 1$, yielding 29
- We how have our keys
  - Public key, k_e, $N = 5, 91$
  - Private key, k_d , $N = 29, 91$
- Encrypting the message 69 with the public key results in the cyphertext 62 ($E=69^5 \bmod 91$)
- Cyphertext can be decoded with the private key
  - Public key can be distributed in cleartext to anyone who wants to communicate with holder of public key

Cryptography (Cont.)

- Note symmetric cryptography based on transformations, asymmetric based on mathematical functions
  - Asymmetric much more compute intensive
  - Typically not used for bulk data encryption
  - Used for authentication, confidentiality, key distribution