Operating Systems: Protection and Security

Neerja Mhaskar

Department of Computing and Software, McMaster University, Canada

Acknowledgements: Material based on the textbook Operating Systems Concepts (Chapter 16 and 17)

Protection

Protection Goal of OS

- Prevent malicious misuse of the system.
- Ensure that each shared resource is used only in accordance with system policies
- Ensure that errant programs cause the minimal amount of damage possible

Protection – how is it achieved?

Protection is achieved by **controlling the access** of programs, processes, or users to the resources defined by a computer system.

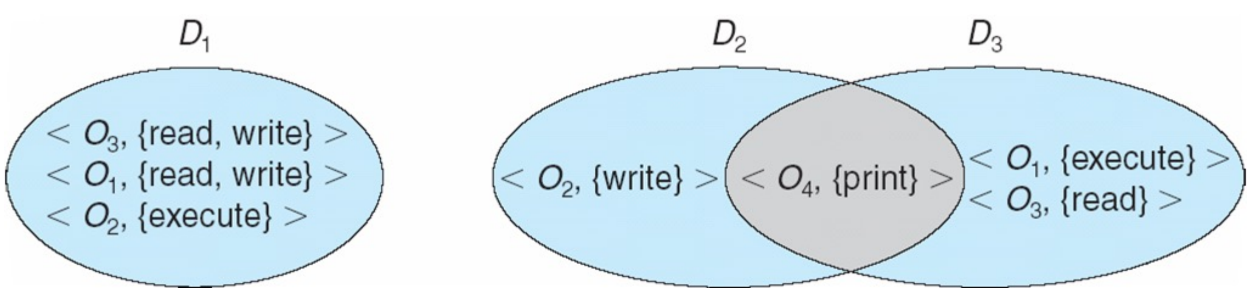
- Guiding principle – **principle of least privilege**
  - Programs, users and systems should be given just enough **privileges** to perform their tasks.

Domain of Protection

- A computer system is a collection of processes and objects:
  - Hardware objects (e.g.: CPU, memory, printers, disks), and
  - Software objects (such as files).
- Processes or users operate within a **(protection) domain**, which specifies the resources (objects) a process may access.

Domain of Protection

- Each domain is defined by a **set of objects and the types of operations** that may be invoked on each object, and
- A domain is represented as a set of pairs of ***<object-name*, access *rights-set>***, where *access rights-set* is a subset of all valid operations that can be performed on the object.
  - E.g.: <file *F*, {read, write}>



Domain of Protection continued

- Domains may be realized in different ways
  - As users (**UNIX associates domains with users**), or
  - As processes/procedures

Security

- Protection is providing controlled access to programs and data stored in a computer system.
  - Protection deals with internal threats
- **Security**, on the other hand, requires not only an adequate protection system but also consideration of the external environment within which the system operates.
  - Security deals with external threats/intruders

Security

- **Intruders** attempt to breach security

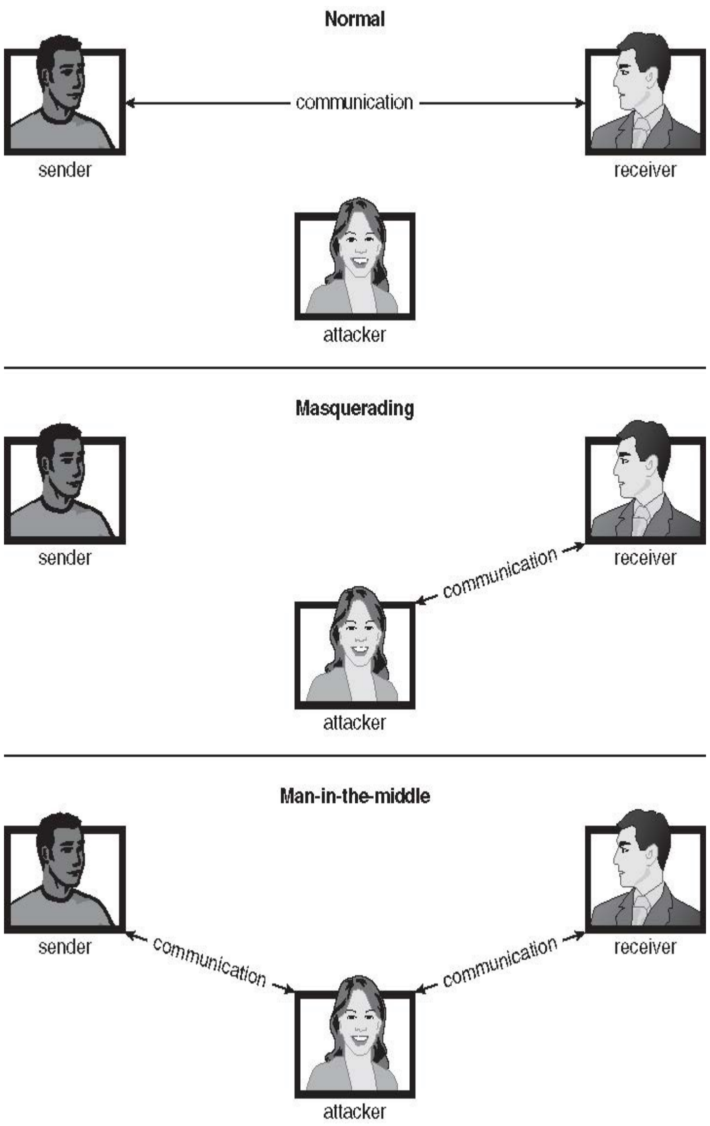
- **Threat** is potential security violation

- **Attack** is an attempt to breach security

  - Attack can be accidental (easier to protect from) or malicious.

Standard Security Attacks

- **Masquerading**
  - Pretending to be an authorized user to escalate privileges
- **Man-in-the-middle attack**
  - Intruder sits in data flow, masquerading as sender to receiver and vice versa
- **Session hijacking**
  - In a network communication, a man-in-the-middle attack may be preceded by a session hijacking, in which an active communication session is intercepted.



Threats

Many variations, many names. Broadly classified as program, system and network threats.

- Trojan Horse
- Spyware
- Trap Door
- Logic Bomb
- Stack and Buffer Overflow
- Viruses
- Worm
- Port scanning
- Denial of Service

Cryptography

- Cryptography means to constrain potential senders (*sources*) and/or receivers (*destinations*) of *messages.*
  - Based on secrets called **keys** used to process messages.
- Cryptography helps with the following two major scenarios:
  - **Encryption** - Enables a sender to send a message to the intended receiver.
    - This is achieved by encoding the message, such that it can only be understood (decrypted) by the receiver.
  - **Authentication** - Enables a recipient of a message <u>to verify</u> sender.
    - It is also used to check if a message has been modified.

Encryption

- **Encryption** is the process of encoding messages (called **ciphertexts**) using keys.

- **Decryption** is the process of decoding messages using keys.

- An algorithm used for encryption must provide the following essential property:

  - Given a ciphertext, a computer can compute the message only if it possesses the key

  - Given a ciphertext, it is impossible to derive the key from it.

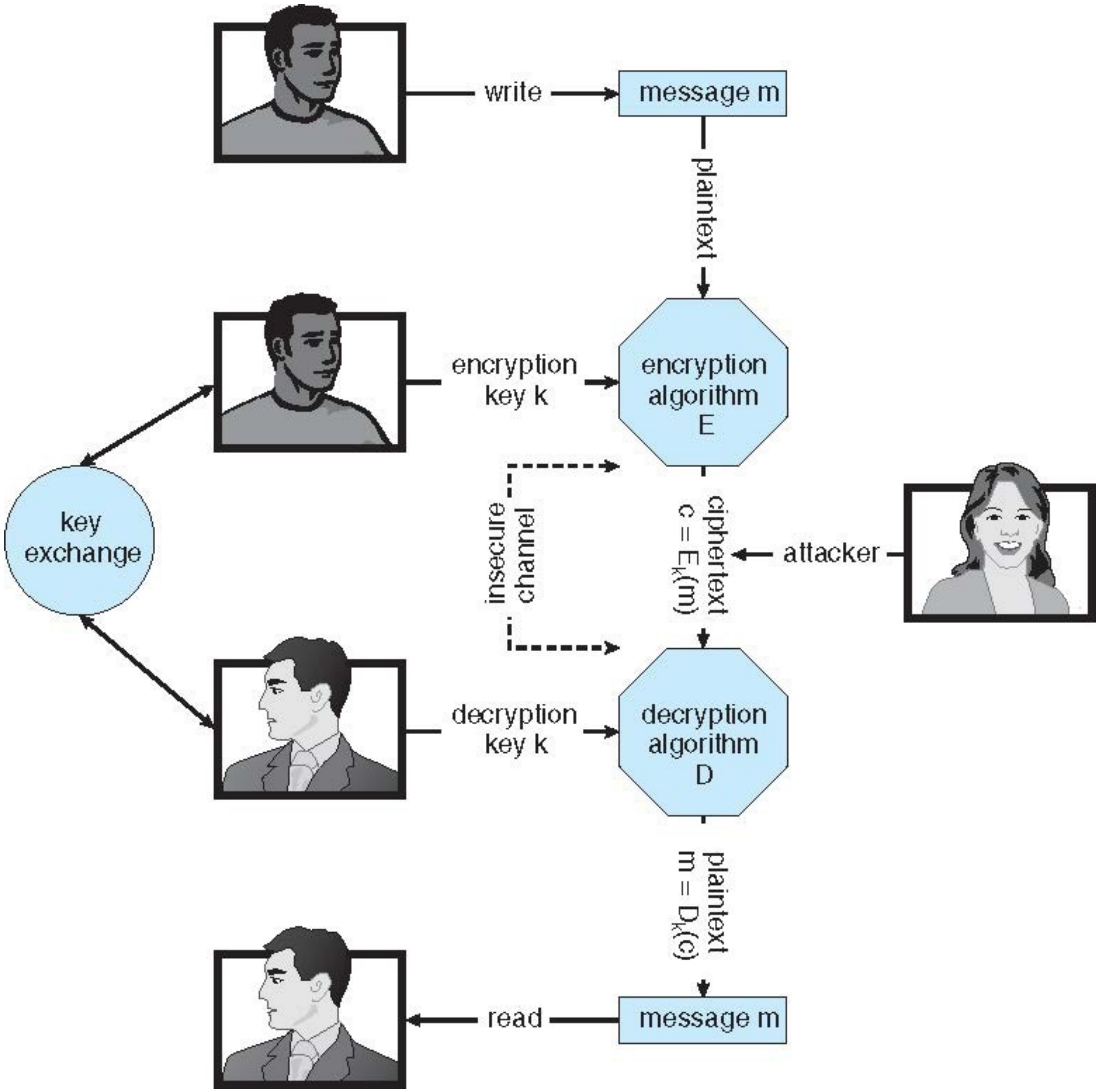
Encryption Algorithms

- There are two main types of encryption algorithms:
  - Symmetric
  - Asymmetric

Symmetric Encryption

- **Same key** used to encrypt and decrypt messages
  - Therefore, *key* must be kept secret and safely guarded.
- Examples of symmetric encryption algorithms are:
  - Data-Encryption Standard (DES)
  - Triple-DES
  - Advanced Encryption Standard (AES)

Secure Communication over Insecure Medium



Asymmetric Encryption

- **Asymmetric encryption** is based on having two different keys to encrypt and decrypt messages.
  - **public key** – is used to **encrypt data** and is published.
  - **private key** – is used to **decrypt data** and is private; that is, key known only to individual decrypting message
- **RSA Algorithm** is one of the most widely used asymmetric encryption algorithms.
- However, RSA is computationally intensive.
  - Therefore, used primarily to encrypt and decrypt small sized data. For example, keys.

RSA Algorithm

Formally, it is computationally infeasible to derive $k_{d,N}$ from $k_{e,N}$, and so k_e need not be kept secret and can be widely distributed

- $K_{e,N} = (k_e, N)$ is the **public key**
- $K_{d,N} = (k_d, N)$ is the **private key**
- $N = p * q$, *where p, q are* two large, randomly chosen prime (for example 512 bits long)
- K_e satisfies the condition that it is relatively prime to $(p–1)(q–1)$ and $< (p–1)(q–1)$
  - *Relatively prime numbers don't share any factors > 1*

RSA Algorithm

Encryption algorithm is $E_{ke,N}(m) = m^{k_e}$ mod N, where m is the message.

Decryption algorithm is then $D_{kd,N}(c) = c^{k_d}$ mod N, *where C is the ciphertext (encrypted message).*

RSA Algorithm Example

- For example, let $p = 7$ 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 (m) 69 with the public key results in the ciphertext (C) = $E_{ke,N}(m) = 69^5 \bmod 91 = 62$
- The ciphertext C is decrypted using the decryption algorithm = $D_{kd,N}(C) = 62^{29} \bmod 91 = 69 = m$

Encryption using RSA Asymmetric Cryptography