CS343 - Operating Systems

Module-8C

Security Mechanisms in Operating Systems

Dr. John Jose

Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology Guwahati, Assam.

http://www.iitg.ac.in/johnjose/

Overview

- Cryptography
- User Authentication
- Implementing Security Defenses
- Firewalling to Protect Systems and Networks

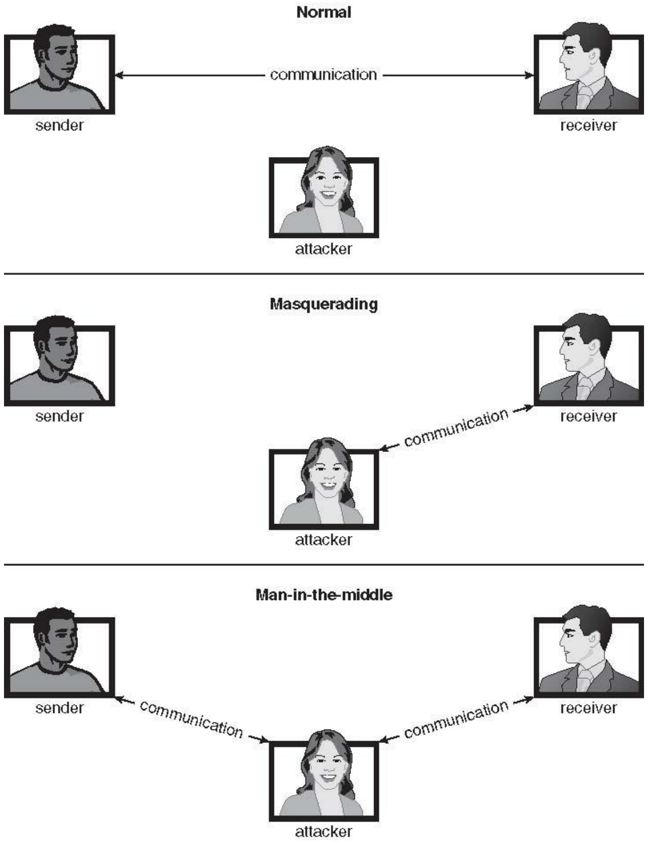
Objectives

- ❖ To explain the fundamentals of encryption, authentication, and hashing
- ❖ To examine the uses of cryptography in computing
- ❖ To describe the various countermeasures to security attacks

Security Violation Categories

- **Breach of confidentiality**
  - Unauthorized reading of data
- **Breach of integrity**
  - Unauthorized modification of data
- **Breach of availability**
  - Unauthorized destruction of data
- **Theft of service**
  - Unauthorized use of resources
- **Denial of service (DOS)**
  - Prevention of legitimate use

Standard Security Attacks



Cryptography as a Security Tool

- Broadest security tool - Art of secret writing
- Source and destination of messages can be known and protected
    - OS creates, manages, protects process IDs, communication ports
- Source and destination of messages on network cannot be trusted without cryptography
    - Local network – IP address?
        - Consider unauthorized host added
    - WAN / Internet – how to establish authenticity
        - Not via IP address

Cryptography

- Means to constrain potential senders (sources) and / or receivers (destinations) of messages
  - Based on secrets (**keys**)
  - Confirmation of source
  - Receipt only by certain destination
  - Trust relationship between sender and receiver
- Symmetric cryptography based on transformations
- Asymmetric cryptography based on mathematical functions
  - Asymmetric much more compute intensive
  - Typically not used for bulk data encryption

Encryption

- Constrains the set of possible receivers of a message
- **Encryption** algorithm consists of
  - Set K of keys
  - Set M of Messages
  - Set C of 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
    - E_k for any k should be efficiently computable functions
  - 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
    - D_k for any k should be efficiently computable functions

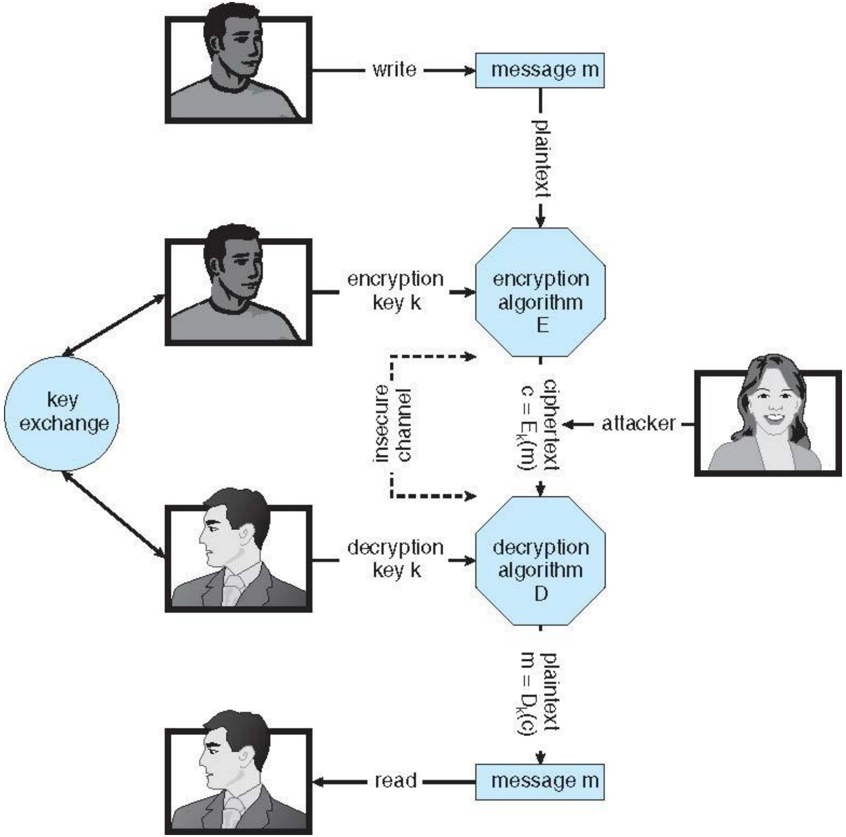
Encryption

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

Symmetric Encryption

- Same key used to encrypt and decrypt
  - Therefore *k* must be kept secret
- DES was most commonly used symmetric block-encryption algorithm
- Triple-DES considered more secure $c = E_{k3}(D_{k2}(E_{k1}(m)))$
- Block cipher - Advanced Encryption Standard (**AES**)
  - Keys of 128, 192, or 256 bits, works on 128 bit blocks

Secure Communication over Insecure Medium



Asymmetric Encryption

- **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** block cipher

Asymmetric Encryption

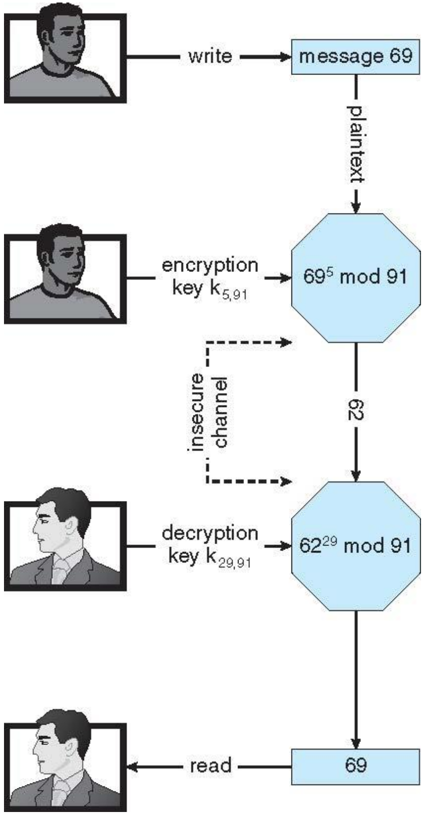
- Formally, it is computationally infeasible to derive $k_{d,N}$ from $k_{e,N}$, and so k_e need not be kept secret and can be widely disseminated
  - k_e is the **public key**
  - k_d is the **private key**
  - N is the product of two large, randomly chosen prime numbers p and q (for example, p and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

Asymmetric Encryption Example

- Assume, $p = 7$ 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
  - 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
- 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

Encryption using RSA Asymmetric Cryptography

- ❖ Public key, $k_{e,N}$ = 5, 91
- ❖ Private key, $k_{d,N}$ = 29, 91
- ❖ Encryption algorithm: $E_{ke,N}(m) = m^{ke} \bmod N$.
- ❖ Decryption algorithm: $D_{kd,N}(c) = c^{kd} \bmod N$



Authentication

- Constraining set of potential senders of a message
  - Complementary to encryption
  - Also can prove message unmodified
- A set K of keys, set M of messages, A set A of authenticators
  - A function $S : K \rightarrow (M \rightarrow A)$
    - That is, for each $k \in K$, S_k is a function for generating authenticators from messages
    - Both S and S_k for any k should be efficiently computable functions
  - A function $V : K \rightarrow (M \times A \rightarrow \{true, false\})$. That is, for each $k \in K$, V_k is a function for verifying authenticators on messages
    - Both V and V_k for any k should be efficiently computable functions

Authentication

- For a message m, a computer can generate an authenticator $a \in A$ such that $V_k(m, a)$ = true only if it possesses k
- Thus, computer holding k can generate authenticators on messages so that any other computer possessing k can verify them
- Computer not holding k cannot generate authenticators on messages that can be verified using V_k
- Since authenticators are generally exposed (for example, they are sent on the network with the messages themselves), it must not be feasible to derive k from the authenticators
- Practically, if $V_k(m,a)$ = **true** then we know m has not been modified and that send of message has k
    - If we share k with only one entity, know where the message originated

Authentication – Hash Functions

- Basis of authentication
- Creates small, fixed-size block of data **message digest** (**hash value)** from m
- Hash Function H must be collision resistant on m
  - Must be infeasible to find an $m' \neq m$ such that $H(m) = H(m')$
- If $H(m) = H(m')$, then $m = m'$
  - The message has not been modified

Authentication – Hash Functions

- Common message-digest functions include **MD5**, which produces a 128-bit hash, and **SHA-1**, which outputs a 160-bit hash
- Not useful as authenticators
  - For example H(m) can be sent with a message
    - But if H is known someone could modify m to m’ and recompute H(m’) and modification not detected
    - So must authenticate H(m)

Authentication - MAC

- Symmetric encryption used in **message-authentication code** (**MAC**) authentication algorithm
- Cryptographic checksum generated from message using secret key
  - Can securely authenticate short values
- If used to authenticate H(m) for an H that is collision resistant, then obtain a way to securely authenticate long message by hashing them first
- Note that k is needed to compute both S_k and V_k, so anyone able to compute one can compute the other

Authentication – Digital Signature

- **Digital signatures -** based on asymmetric keys to verify authenticity of m.
- Similar to the RSA encryption algorithm, but the key use is reversed
- k_v is the public key and k_s is the private key
- Computationally infeasible to derive k_s from k_v
- RSA digital-signature algorithm
  - Digital signature of message $S_{ks}(m) = H(m)^{ks} \bmod N$
  - The key k_s again is a pair (d, N), where N is the product of two large, randomly chosen prime numbers p and q
  - Verification algorithm is $V_{kv}(m, a)$ $(a^{kv} \bmod N = H(m))$
    - Where k_v satisfies $k_v k_s \bmod (p - 1)(q - 1) = 1$

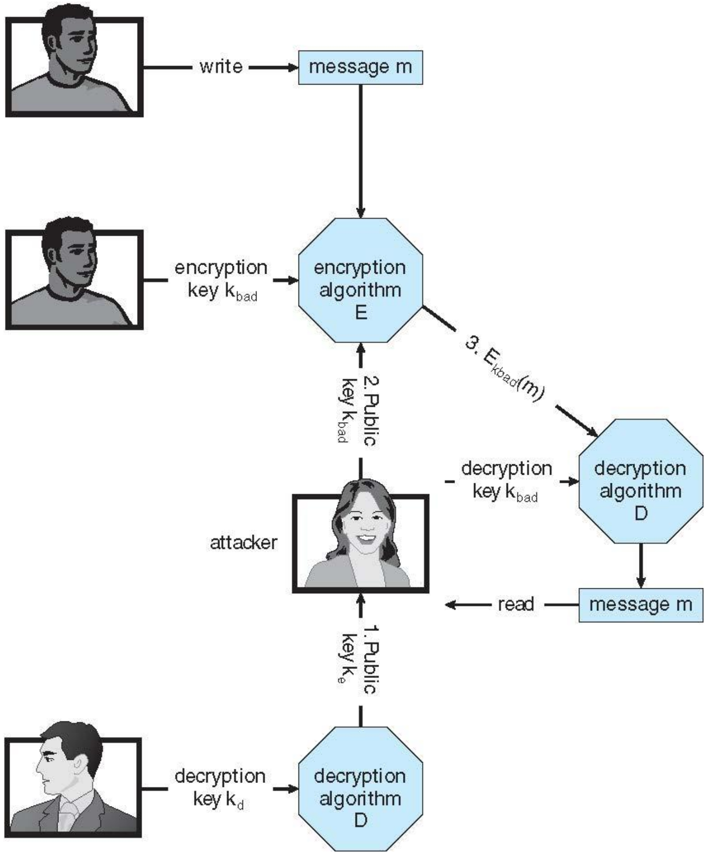
Key Distribution

- Delivery of symmetric key is huge challenge
  - Sometimes done **out-of-band**
- Asymmetric keys can proliferate – stored on **key ring**
  - Even asymmetric key distribution needs care – man-in-the-middle attack

Digital Certificates

- Proof of who or what owns a public key
- Public key digitally signed a trusted party
- Trusted party receives proof of identification from entity and certifies that public key belongs to entity
- **Certificate authority** are trusted party – their public keys included with web browser distributions
    - They vouch for other authorities via digitally signing their keys, and so on

Man-in-the-middle Attack - Asymmetric Cryptography



Implementation of Cryptography

- Can be done at various **layers** of ISO Reference Model
  - SSL at the Transport layer
  - Network layer is typically **IPSec**
    - **IKE** for key exchange
    - Basis of **Virtual Private Networks (VPNs)**

OSI model

7. Application Layer

NNTP · SIP · SSI · DNS · FTP · Gopher · HTTP · NFS · NTP · SMPP · SMTP · SNMP · Telnet · Netconf · (more)

6. Presentation Layer

MIME · XDR · TLS · SSL

5. Session Layer

Named Pipes · NetBIOS · SAP · L2TP · PPTP · SPDY

4. Transport Layer

TCP · UDP · SCTP · DCCP · SPX

3. Network Layer

IP (IPv4, IPv6) · ICMP · IPsec · IGMP · IPX · AppleTalk

2. Data Link Layer

ATM · SDLC · HDLC · ARP · CSLIP · SLIP · GFP · PLIP · IEEE 802.3 · Frame Relay · ITU-T G.hn DLL · PPP · X.25 · Network Switch · DHCP

1. Physical Layer

EIA/TIA-232 · EIA/TIA-449 · ITU-T V-Series · I.430 · I.431 · POTS · PDH · SONET/SDH · PON · OTN · DSL · IEEE 802.3 · IEEE 802.11 · IEEE 802.15 · IEEE 802.16 · IEEE 1394 · ITU-T G.hn PHY · USB · Bluetooth · Hubs

This box: view · talk · edit

OSI Model

	Data unit	Layer	Function
Host layers	Data	7. Application	Network process to application
		6. Presentation	Data representation, encryption and decryption, convert machine dependent data to machine independent data
		5. Session	Interhost communication
	Segments	4. Transport	End-to-end connections and reliability, flow control
Media layers	Packet/Datagram	3. Network	Path determination and logical addressing
	Frame	2. Data Link	Physical addressing
	Bit	1. Physical	Media, signal and binary transmission

Encryption Example - SSL

- Insertion of cryptography at one layer of network model (transport layer)
- SSL – Secure Socket Layer (also called TLS)
- Cryptographic protocol that limits two computers to only exchange messages with each other
- Used between web servers and browsers for secure communication (credit card numbers)
- The server is verified with a **certificate** assuring client is talking to correct server
- Asymmetric cryptography used to establish a secure **session key** (symmetric encryption) for bulk of communication during session
- Communication between each computer then uses symmetric key cryptography

User Authentication

- Crucial to identify user correctly, as protection systems depend on user ID
- User identity most often established through **passwords**, can be considered a special case of either keys or capabilities
- Passwords must be kept secret
  - Frequent change of passwords
  - History to avoid repeats
  - Use of "non-guessable" passwords
  - Log all invalid access attempts (but not the passwords themselves)
  - Unauthorized transfer
- Passwords may also either be encrypted or allowed to be used only once

Passwords

- Encrypt to avoid having to keep secret
  - But keep secret anyway
  - Use algorithm easy to compute but difficult to invert
  - Only encrypted password stored, never decrypted
- One-time passwords
  - Use a function based on a seed to compute a password, both user and computer
- Biometrics
  - Some physical attribute (fingerprint, hand scan)
- Multi-factor authentication

Implementing Security Defenses

- **Defense in depth** – multiple layers of security
- **Security policy** describes what is being secured
- Vulnerability assessment compares real state of system / network compared to security policy
- Intrusion detection endeavors to detect attempted or successful intrusions
  - **Signature-based** detection spots known bad patterns
  - **Anomaly detection** spots differences from normal behavior
  - **False-positives** and **false-negatives** a problem

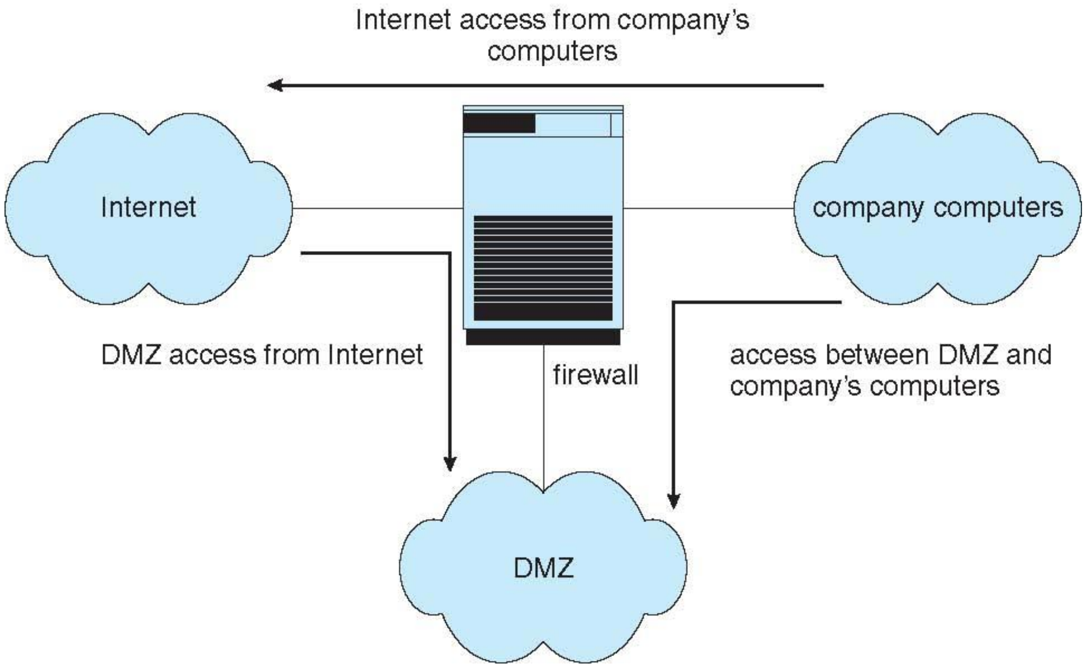
Implementing Security Defenses

- Virus protection
  - Searching all programs or programs at execution for known virus patterns
- Auditing, accounting, and logging of all or specific system or network activities
- Practice **safe computing** – avoid sources of infection, download from only good sites, etc

Firewalling to Protect Systems and Networks

- A network **firewall** is placed between trusted and untrusted hosts
- The firewall limits network access between these two **security domains**
- Firewall rules typically based on host name or IP address which can be spoofed
- **Personal firewall** is software layer on given host
  - Can monitor / limit traffic to and from the host
- **Application proxy firewall** understands application protocol and can control them (i.e., SMTP)
- **System-call firewall** monitors all important system calls and apply rules to them (i.e., this program can execute that system call)

Network Security Through Firewall



Thank you

johnjose@iitg.ac.in
http://www.iitg.ac.in/johnjose/