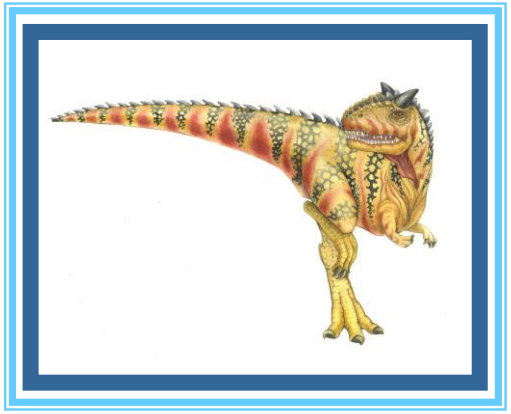
Xi'an Jiaotong-Liverpool University
西交利物浦大學

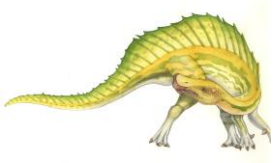
CPT104 - Operating Systems Concepts

Protection & Security in OS



Protection & Security

- Protection
  - Principles of Protection
  - Access Matrix
- Security
  - Cryptography as a Security Tool
  - User Authentication
  - Implementing Security Defenses



PROTECTION

How do we provide controlled access to programs and data stored in a computer system?

Goals of Protection

Protection refers to a mechanism which controls the access of programs, processes, or users to the resources defined by a computer system.

Goals of Protection:

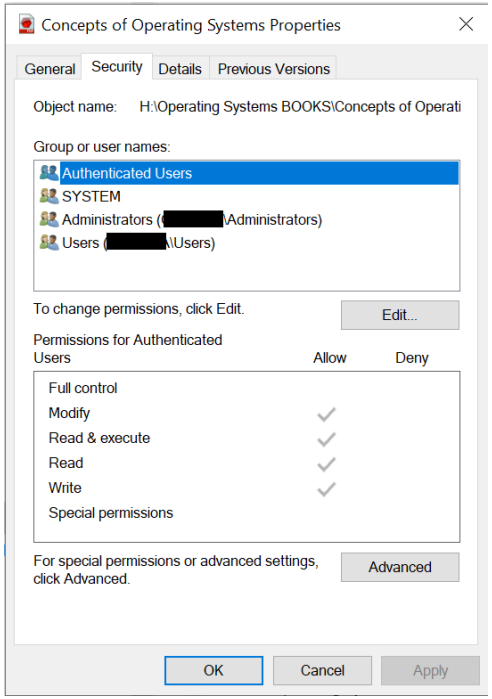
- To prevent the access of unauthorized users,
- To ensure that each active programs or processes in the system uses resources only as the stated policy,
- To ensure that errant programs cause the minimal amount of damage possible,
- To improve reliability by detecting latent errors.

Principles of Protection

- **Principle of least privilege**
- The principle of least privilege dictates that programs, users, and systems be given just enough privileges to perform their tasks.
  - *the user account* limited privileges
  - *the root account*
- Can be **static** (during life of system, during life of process)
- Or **dynamic** (changed by process as needed) – *domain switching, privilege escalation*

Access Matrix

- View protection as a matrix (**access matrix**)
- Rows represent **domains** (a domain is a set of object and right pairs)
- Columns represent **objects** (resources)
- **Access(i, j)** is the **set of operations that a process executing in Domain$_i$ can invoke on Object$_j$**

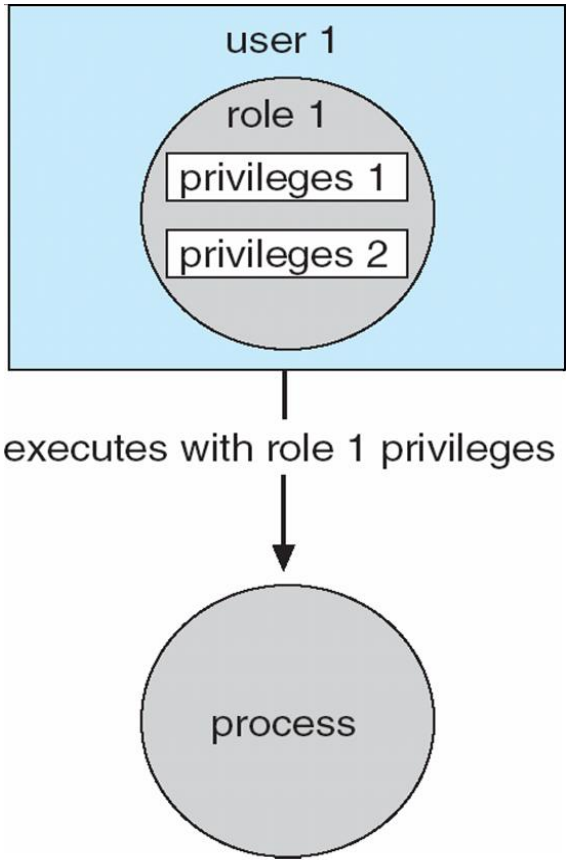


permission settings on Windows

domain \ object	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

Access Control Policy

Role-based access control (RBAC) is a security feature for controlling user access to tasks that would normally be restricted to the root user.



Role-Based Access Control, RBAC, assigns *first* the roles and *then* all the permissions are assigned.

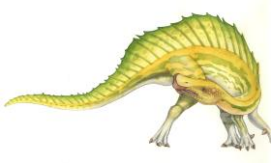
- A user can be assigned multiple roles.
- Multiple users can be assigned the same role.
- A role can have multiple access rights.

SECURITY

Security is the practice of the confidentiality, integrity, and availability of data.

Security Violation Categories

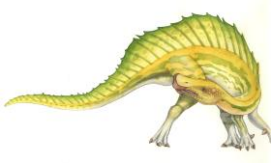
- **Breach of confidentiality**
  - Unauthorized reading of data
- **Breach of integrity**
  - Unauthorized modification of data
- **Breach of availability**
  - Unauthorized destruction of data
- **Theft of service**
  - Unauthorized use of resources
- **Denial of service (DOS)**
  - Prevention of illegitimate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



The Security Problem

Concepts used in security:

- **Intruders** (**crackers**) attempt to breach security
- **Threat** is potential security violation
- **Attack** attempts to breach security
  - Attack can be accidental or malicious
  - Easier to protect against accidental than malicious misuse



Security Violation Methods

- **Masquerading** (breach **authentication**)
  - attacker pretends to be an authorized user to escalate privileges
- **Replay attack**
  - attacker delays, replays, or repeats data transmission between the user and the site.
- **Man-in-the-middle attack**
  - intruder sits in data flow, masquerading as sender to receiver and vice versa
- **Hijacking**
  - type of network security attack in which the attacker takes control of computer systems, software programs etc.

Program Threats

- **Trojan Horse** - is able to carry out any action that a legitimate user could perform, such as exporting files, modifying data, deleting files or otherwise altering the contents of the device.
- **Trap Door** - deliberately inserts a security hole that they can use later to access the system.
- **Logic Bomb** - is a piece of code intentionally inserted into a software system that will set off a malicious function when specified conditions are met.
- **Stack** and **Buffer Overflow** focus on buffers/memory.
  - *Stack overflow* - when the execution stack grows beyond the memory that is reserved for it.
  - *Buffer overflow* refers to any case in which a program writes beyond the end of the memory allocated for any buffer.
- **Viruses** - is a malicious executable code attached to another executable file that can be harmless or can modify or delete data.

System and Network Threats

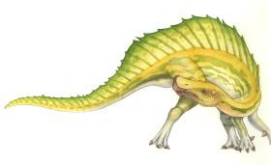
- **Worm** - is a type of malware or malicious software that can replicate rapidly and spread across devices within a network. Worms consume system resources, often blocking out other, legitimate processes.
- **Port Scanning** is technically not an attack, but rather a search for vulnerabilities to attack.
- **Denial of Service (DOS)** attacks that attempt to lock down systems so much that they can no longer be used for any useful activity. DOS attacks can also involve social engineering.

Security Measure Levels

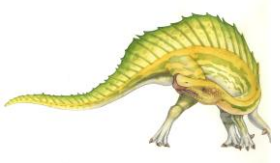
Security measures at four levels:

- **Physical**
  - Data centers, servers, connected terminals
- **Human**
  - Avoid *social engineering, phishing* (involves sending an innocent-looking e-mail), *dumpster diving* (searching the trash or other locations for passwords), *password cracking,* etc.
- **Operating System**
  - System must protect itself from accidental or purposeful security breaches: *runaway processes* (DOS denial of service), *memory-access violations, stack overflow violations, launching of programs with excessive privileges*, etc.
- **Network**
  - protecting the network itself from attack and protecting the local system from attacks coming in through the network (intercepted communications, interruption, DOS, etc.).

CRYPTOGRAPHY AS A SECURITY TOOL



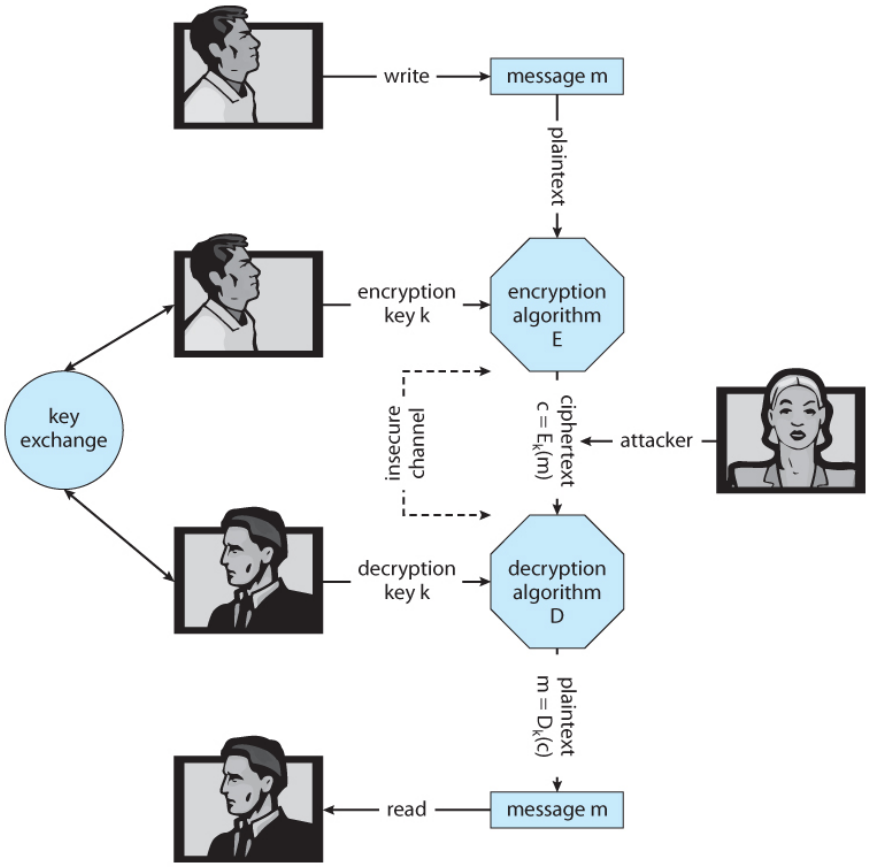
- Encryption
- Symmetric Encryption
- Asymmetric Encryption
- Authentication
- Key Distribution



Encryption

Cryptography is a technique to hide the message using *encryption*.

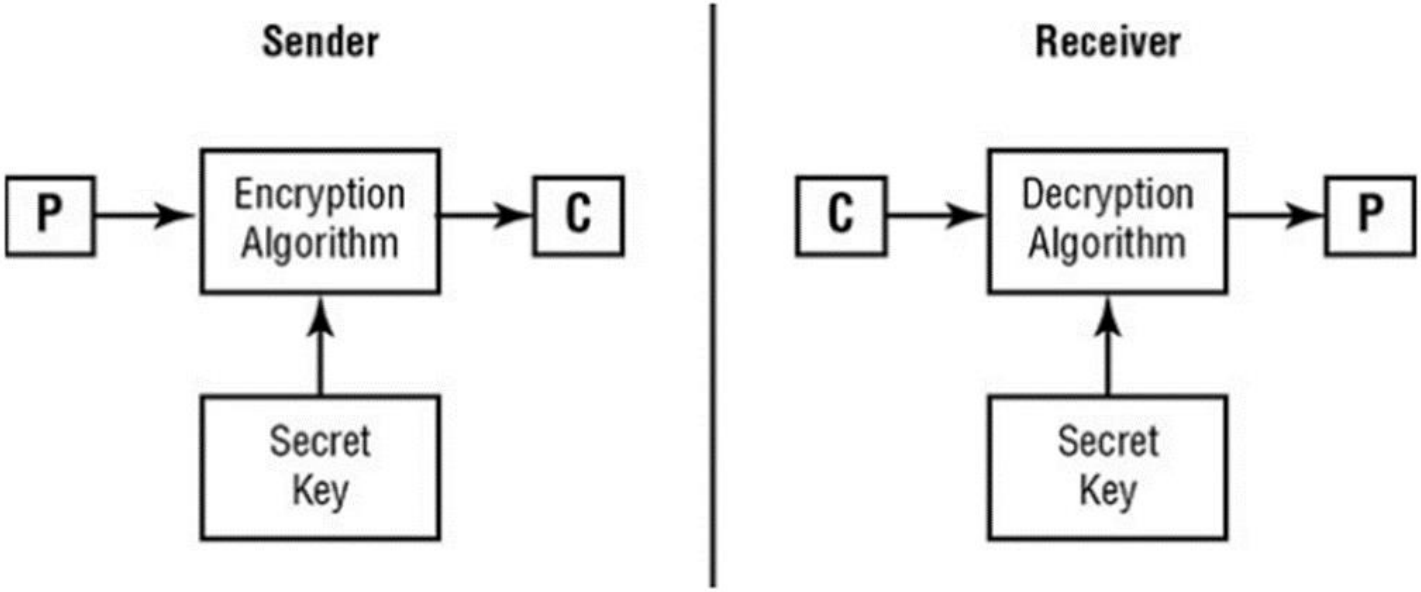
Encryption is a process of encoding a message so that its meaning cannot be easily understood by unauthorized people.



- Symmetric Encryption
- Asymmetric Encryption

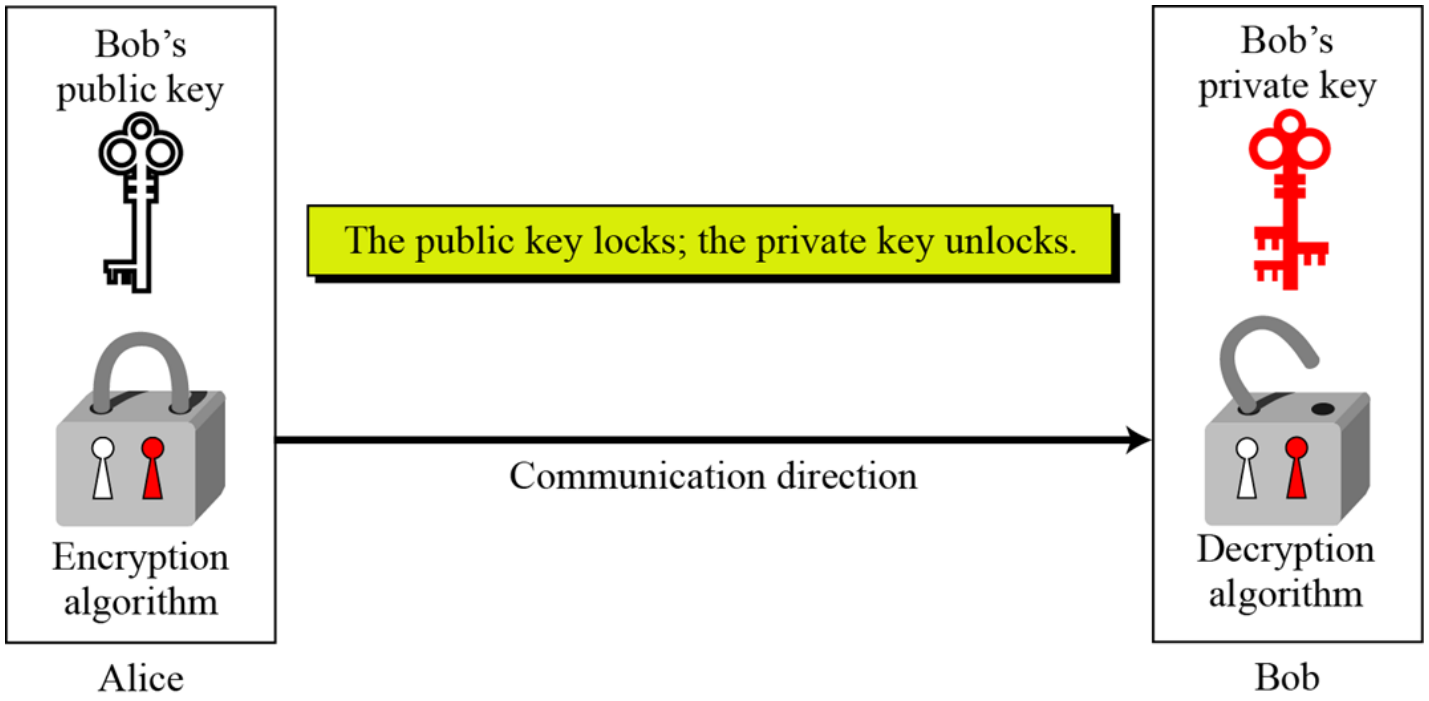
Symmetric Encryption

- **Same key used to encrypt and decrypt**
- **Data Encryption Standard (DES)** was most commonly used symmetric block-encryption algorithm (created by US Govt)
- **Triple-DES** considered more secure
- **Advanced Encryption Standard** (**AES**)
- **Rivest Cipher RC4** is most common symmetric stream cipher, but known to have vulnerabilities



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
- Most common is **RSA** (RSA = Ron Rivest, Adi Shamir and Leonard Adleman) based on prime numbers



Digital Certificates

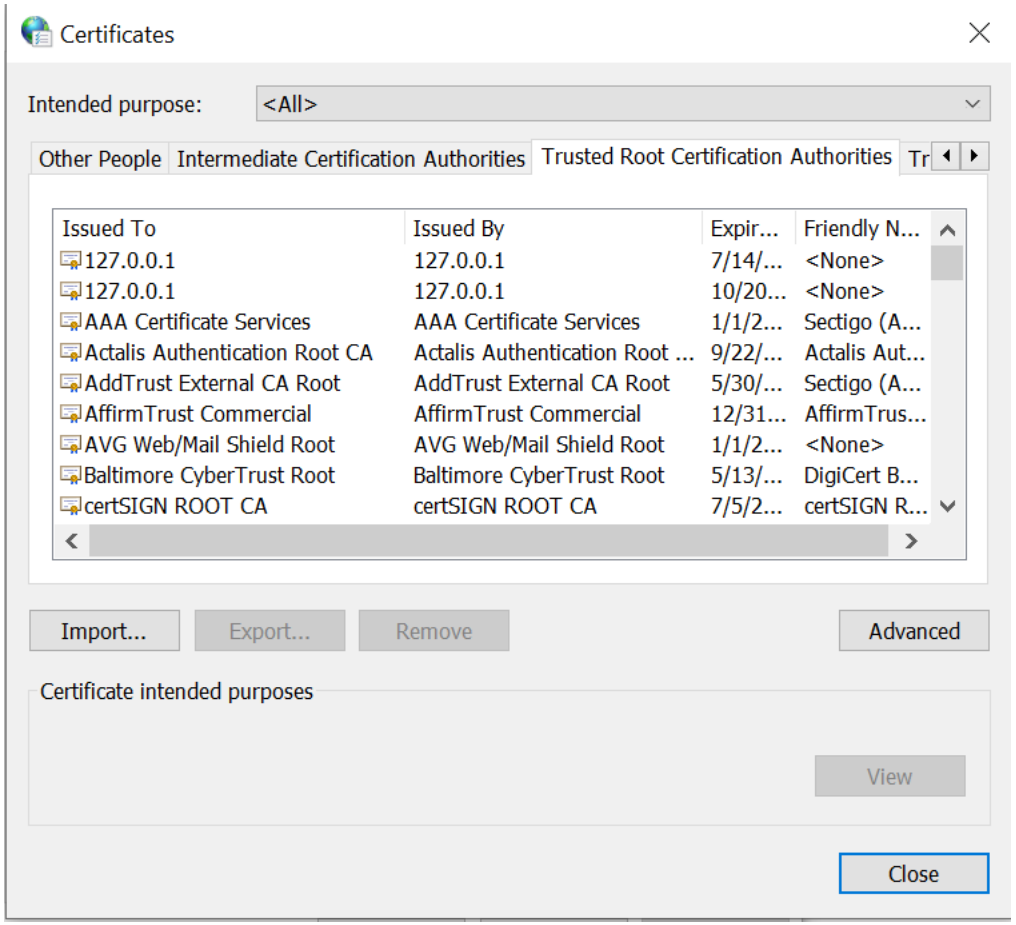
Is the public key safe?

One solution : ***Digital Certificates***

A digital certificate is a mechanism that allows users to verify the authenticity of a key / document.

- Proof of **who** or **what** owns a public key
- **Public key** digitally signed by 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

In *Windows* you can view these certificates by selecting Control Panel → Internet Options → Content → Certificates



Key Distribution Management

- Keys in **Symmetric encryption** is a **major problem**
  - One option is to send them **Out-of-band**, say via **paper** or a **confidential conversation or One-time pad**
- Keys in **Asymmetric encryption** - the public keys **are not confidential.**
  - the **key-ring** can be easily stored and managed (**key-ring** is simply a file with keys in it.).

Even asymmetric key distribution needs care – *man-in-the-middle attack*

USER AUTHENTICATION

Only certain users were allowed to perform certain tasks. But how does one verify that identity to begin with?

Authentication

When a user logs into a computer, the OS needs to determine the identity of the user.

The user authentication has two steps:

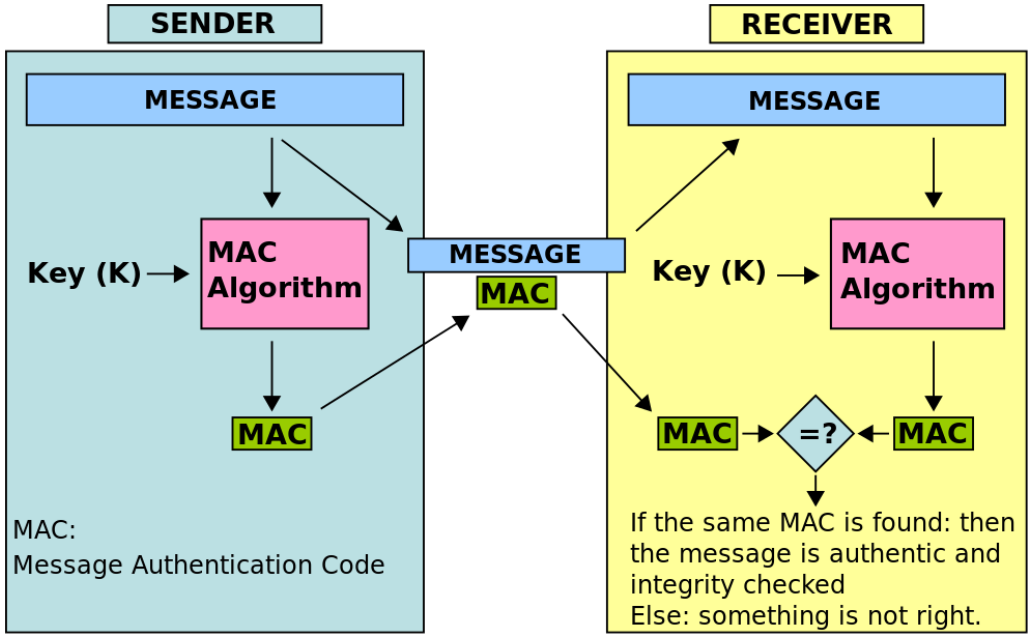
- **IDENTIFICATION** - a unique identifier is specified to the user to authenticate.
  - a **Signing function** - produces an **authenticator**: a value to be used to authenticate a user.
- **VERIFICATION** of a user - performed against the unique identifier, that is, it confirms the binding between the user and the identifier.
  - a **Verification function** - produces a value of "**true**" if the authenticator was created from the user, and "**false**" otherwise.

Authentication

Two main authentication algorithms :

1. Message Authentication Code (MAC) - uses symmetric encryption.

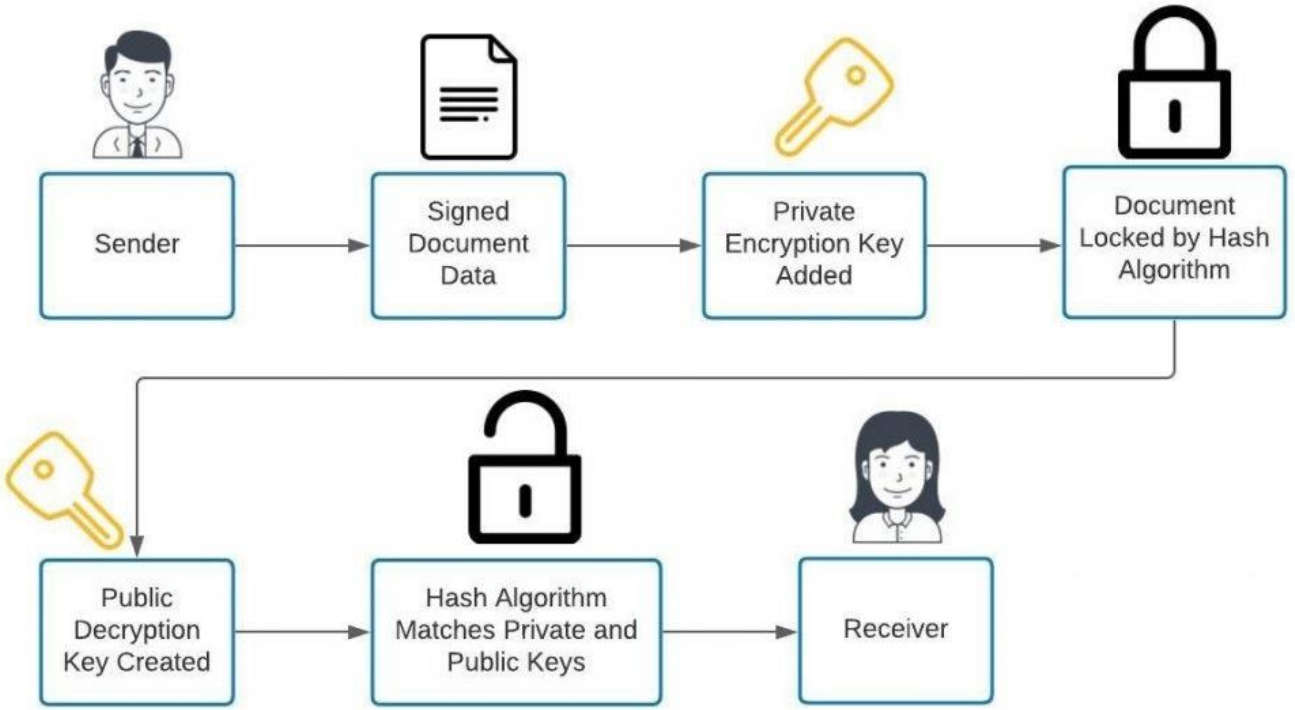
- used for authenticating and integrity-checking a message.
- to confirm that the message came from the stated sender (its authenticity) and has not been changed (its integrity).



Authentication

2. Digital-signature algorithm - uses asymmetric encryption.

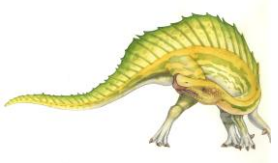
- to authenticate and verify documents and data.



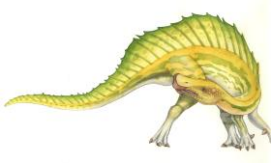
Common forms of user authentication

- **Passwords**
  - **Password Vulnerabilities !!**
  - **Securing Passwords** - modern systems do not store passwords in clear-text form.
  - **One-time passwords** resist shoulder surfing and other attacks .
- **Biometrics** involve a physical characteristic of the user.

Multifactor authentication is better.



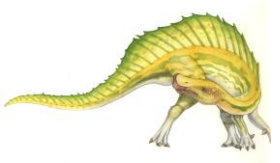
IMPLEMENTING SECURITY DEFENSES



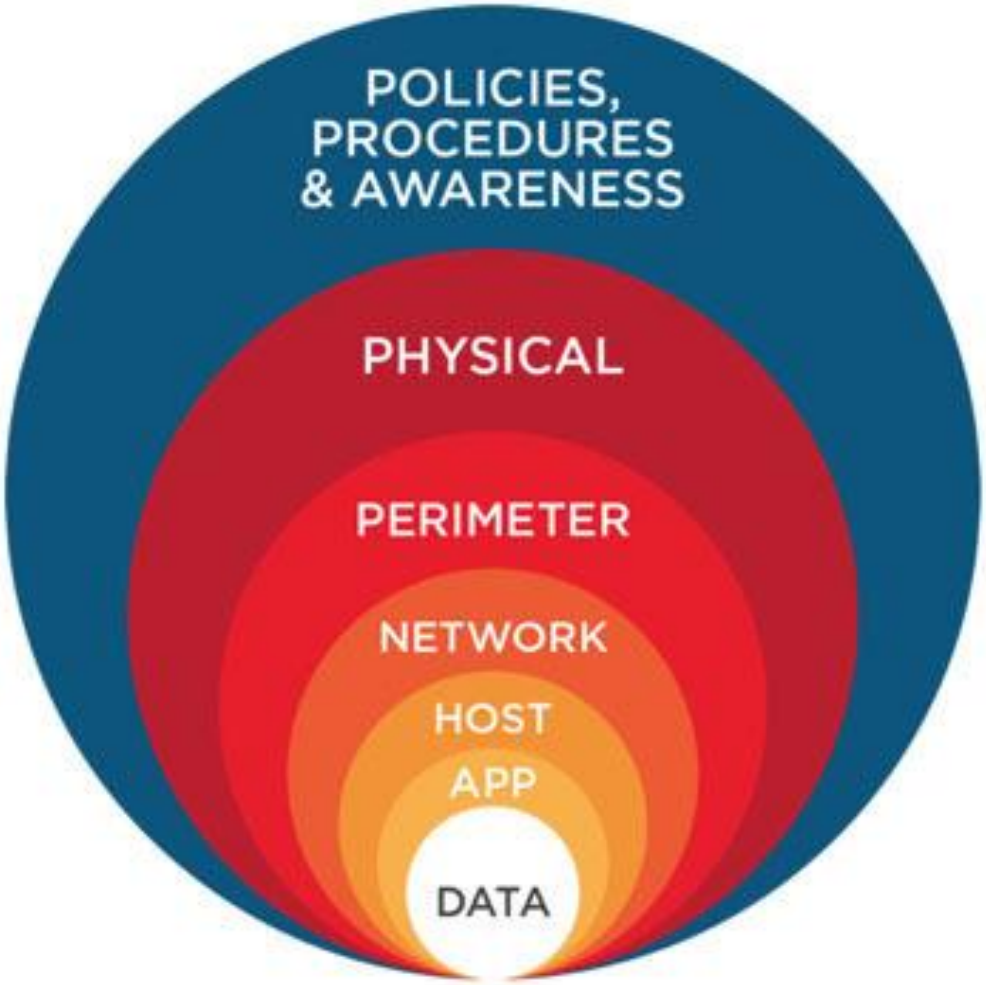
Implementing Security Defenses

The major methods, tools, and techniques that can be used to improve security:

- Security Policy
- Vulnerability Assessment
- Intrusion Detection
- Virus Protection
- Auditing, Accounting, and Logging



Defense in depth is most common security theory – **multiple layers of security**

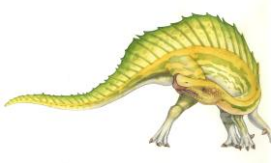


Vulnerability Assessment

How can we determine whether a security policy has been correctly implemented?

Periodically examine the system to detect vulnerabilities.

- Port scanning.
- Check for bad passwords.
- Unauthorized programs in system directories.
- Incorrect permission bits set.
- Program checksums / digital signatures which have changed.
- Unexpected or hidden network daemons.
- New entries in startup scripts, shutdown scripts or other system scripts or configuration files.
- New unauthorized accounts.



End of Lecture

- **Summary**

- Protection
  - Principles of Protection
  - Access Matrix
- Security
  - Cryptography as a Security Tool
  - User Authentication
  - Implementing Security Defenses

- **Reading**

  - Textbook 9th edition, **ch. 14 + ch. 15 of the module textbook**