1

INTRODUCTION

Unit Structure

1.0 OBJECTIVES

The objective of this module is to learn the concept of Information Security. Why security is required in the first place? After that we will discuss the key principles of security. These principles help us to identify the various areas, which are crucial while determining the security threats and possible solutions to tackle them. This is followed by a discussion of the services and attacks. Finally we will discuss the functional requirements of security and current trends in security.

1.1 INTRODUCTION TO INFORMATION SECURITY

Information is a valuable asset like any other asset. So, information needs to be secured from attacks. Now the question is what is security?

Security is

- Freedom from risk or danger; safety.
- Freedom from doubt, anxiety, or fear;
- confidence

Why do we need security?

- Protect vital information while still allowing access to those who need it. Eg. Medical records, Trade secrets etc
- Provide authentication and access control for resources
- Guarantee availability of resources

In short to secure information, three security goals must be achieved

Confidentiality: Means information needs to be hidden from unauthorized access.

Integrity: Means information protected from unauthorized change.

Availability: Means information available to an authorized entity when it is needed.

In early days, the information collected by an organization was stored on physical files. The confidentiality of the files was achieved by restricting the access to only trusted people in the organization. And also, only a few authorized people were allowed to change the contents of the files. Availability was achieved by designating at least one person who would have access to the files at all times.

With the invention of computers, information storage become electronic, means it was stored in computers. The three security requirements, however, did not change. The files stored in computers also require confidentiality, integrity and availability (CIA). The implementation of these requirements, however is different and more challenging

During the last two decades, computer networks created a revolution in the use of information. Information is now distributed. Authorized people can send and retrieve information from a distance using computer networks. Although the three above mentioned requirements confidentiality, integrity and availability have not changed, they now have some new dimensions.

1.2 PRINCIPLES OF SECURITY

Now we classify the principles related to security, which help us understand the attacks better and also help us to tackle the attacks. By taking one example will understand these concepts.

Let us assume that a person A wants to send a check worth 10000 Rs. to another person B. So, A will write the check for 10000 Rs, put it inside an envelope and send it to B.

- A will like to ensure that no one except B gets the envelope and even if someone else gets it, she does not come to know about the details of the check. This is the principle of **confidentiality**
- A and B will further like to make sure that no one can tamper with the contents of the check (such as its amount, date, signature, name of the payee, etc.). This is the principle of **integrity**
- B would like to be assured that the check has indeed come from A and not from someone else posing as A (as it could be a fake check in that case). This is the principle of **authentication.**

- What will happen tomorrow if B deposits the check in his account, the money is transferred from A's account to B's account and then A refuses having written/sent the check? The court of law will use A's signature to disallow A to refuse this claim and settle the dispute. This is the principle of **non-repudiation.**

These are the four chief principles of security. There are two more, **access control** and **availability**, which are not related to a particular message, but are linked to the overall system as a whole.

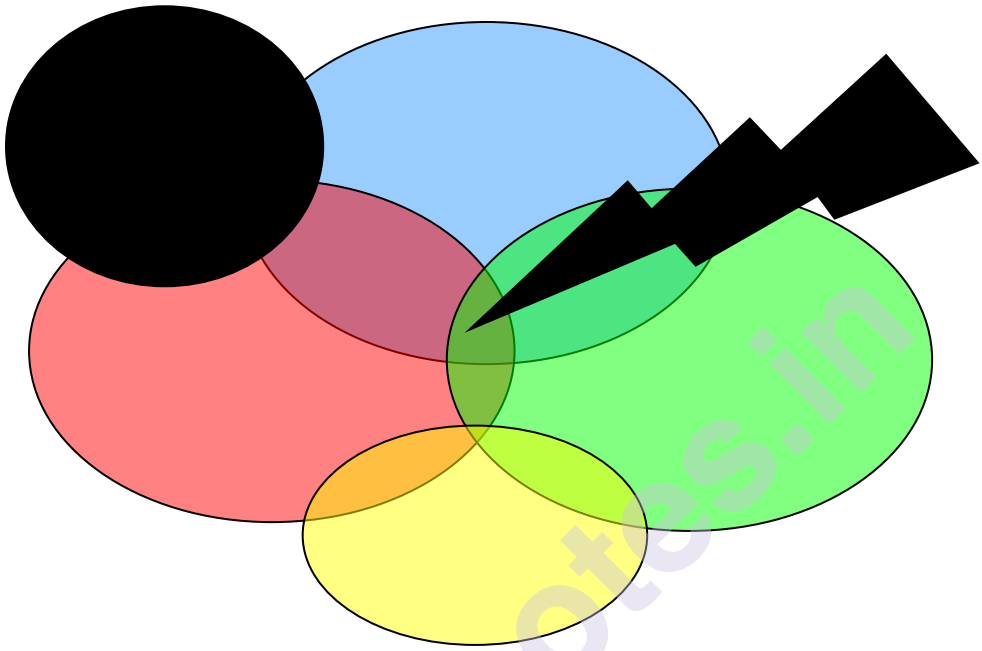


Fig. 1.2 Principles of Security

Examples of Security Requirements

- Confidentiality – student grades
- Integrity – patient information
- Availability – authentication service
- Authenticity – admission ticket
- Non-Repudiation – stock sell order

1.3 ATTACKS

Security attack is an action that compromises the security of information owned by an organization. Network security attacks are unauthorized actions against private, corporate or governmental IT assets in order to destroy, modify or steal sensitive data.

There is a flow of information from source to destination. This is normal flow, as shown in Fig 1.3.1

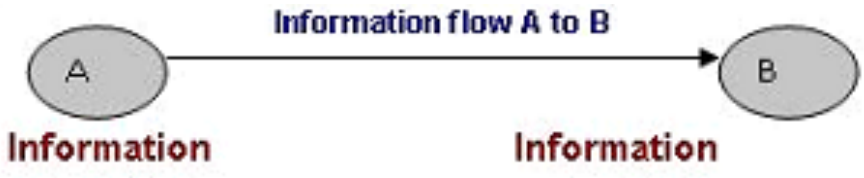


Fig. 1.3.1 Normal flow

Security attacks are classified into four general categories:

Interruption

This is an attack on availability. An asset of the system is destroyed or becomes unavailable. **Ex:** cutting of a communication line, disabling of the file management system, destruction of piece of hardware.

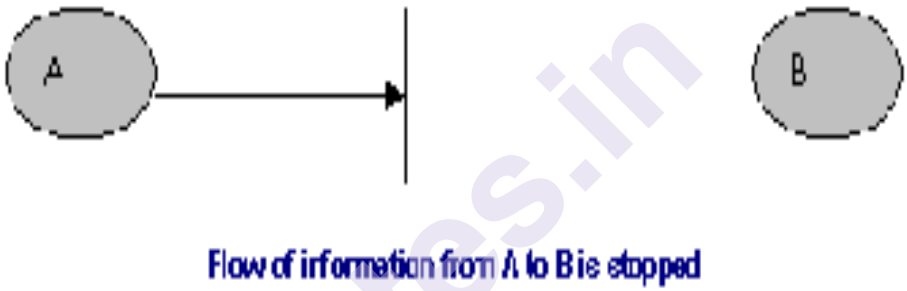


Fig. 1.3.2 Interruption

Interception

This is an attack on confidentiality. An unauthorized party gain access to an asset. Ex: wire tapping to capture data in a network, illicit copying of files.

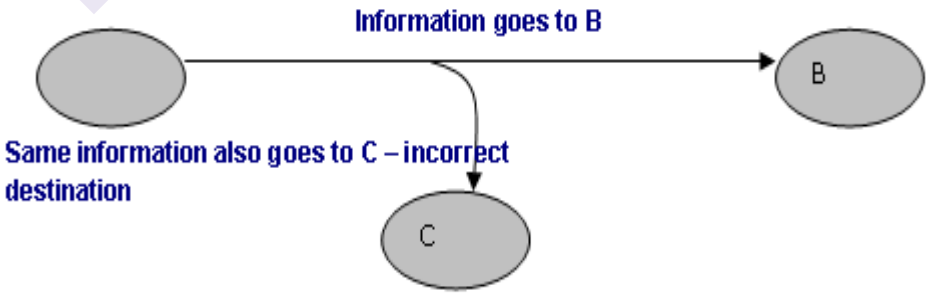


Fig. 1.3.3 Interception

Modification

This is an attack on integrity. An unauthorized party not only gains access but tampers with an asset. Ex: changing values in data file, altering a program, modifying the contents of messages being transmitted in a network.

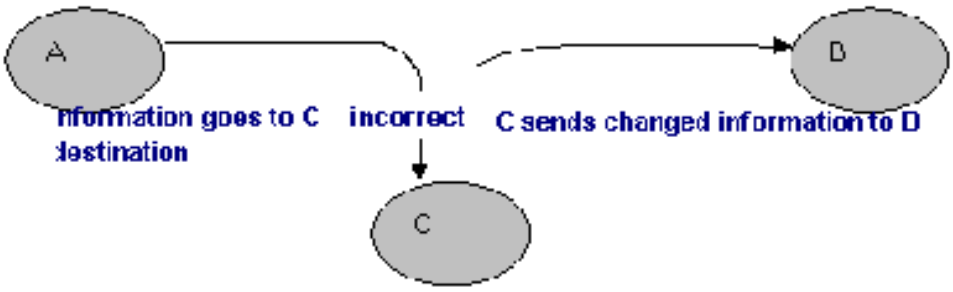


Fig. 1.3.4 Modification

Fabrication:

This is an attack on authenticity. An unauthorized party inserts counterfeit objects into the system. Ex: the insertion of spurious messages in a network or the addition of records to a file

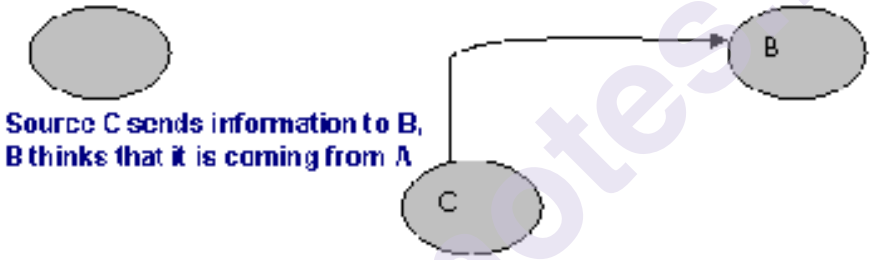


Fig. 1.3.5 Fabrication

Security attacks are classified into two:

a. Passive attacks
b. Passive attacks

a. Passive attacks

A passive attack attempts to learn or make use of information from the system but does not affect system resources. In a passive attack, the attacker's goal is just to obtain information. The attack does not modify data or harm the system, and the system continues with its normal operation. Example of attack threatening to confidentiality is Release of message contents and traffic analysis

Passive attacks are of two types:

1. **Release of message contents:** A telephonic conversation, an email message and a transferred file may contain sensitive or confidential information; we would like to prevent an opponent from learning the content of these transmissions. As shown in Figure 1.3.6, Darth will try to read the contents of message sent from Bob to Alice.

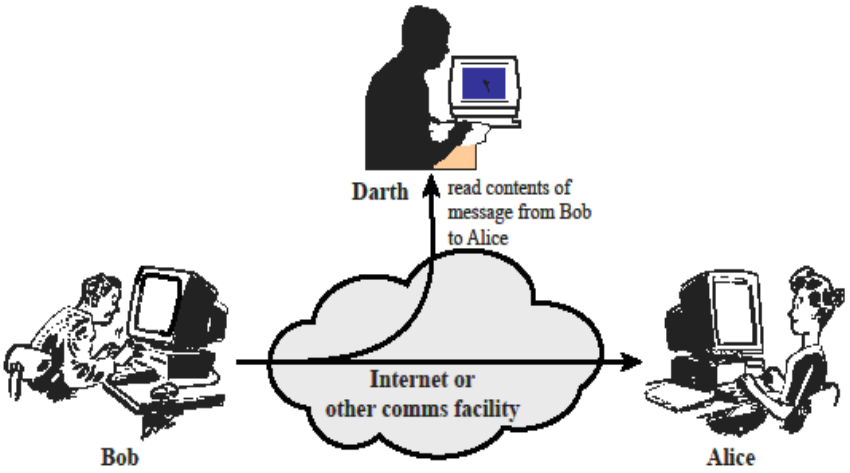


Fig. 1.3.6 Release of message contents

2. **Traffic analysis**: In traffic analysis attack opponents will observe the pattern of messages from sender to receiver. As shown in Figure 1.3.7, Darth will try to observe the traffic or message pattern from Bob to Alice.

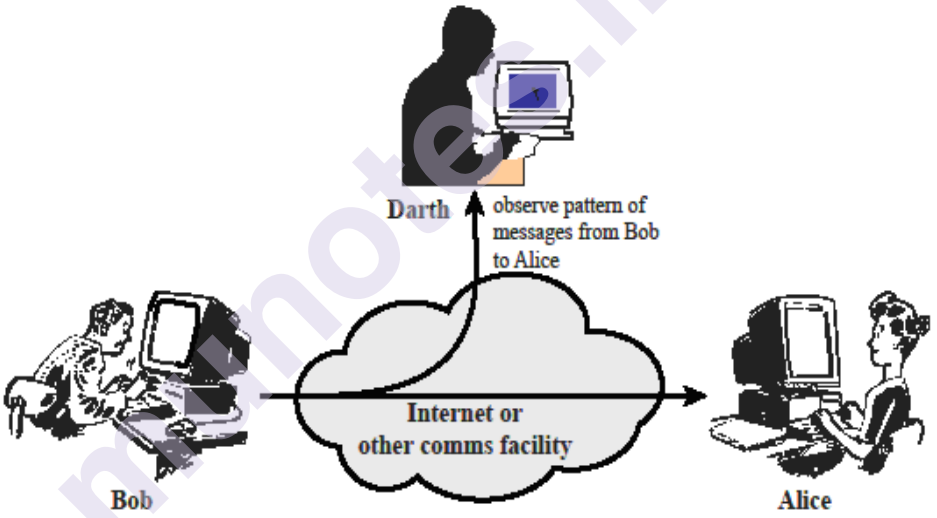


Fig. 1.3.7 Traffic analysis

Passive attacks are very difficult to detect because they do not involve any alteration of data. However, it is feasible to prevent the success of these attacks.

b. Active attacks

An active attack attempts to alter system resources or affect their operation. An active attack may change the data or harm the system. Example of attack threatening to integrity is masquerading, replaying, modification and Repudiation. Example of attack threatening to availability is Denial of Service (DoS).

Active attacks can be subdivided into five types:

1. Masquerade: It takes place when one entity claims to be a different entity. One of the other forms of active attack is a masquerade attack. Figure 1.3.8 depicts masquerade attack where Darth sends message to Alice pretending to be Bob and Alice thinks that message is from Bob. Alice is unaware that message is actually send by Darth.

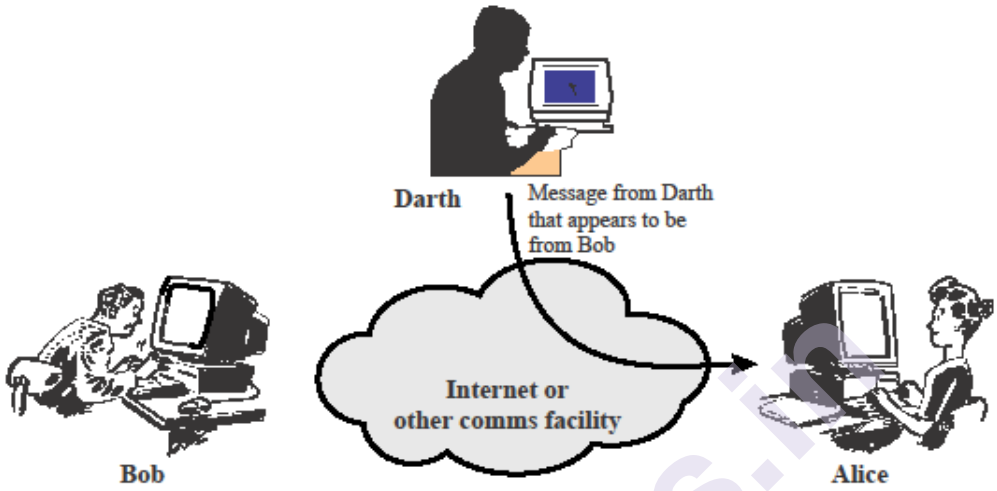


Fig. 1.3.8 Masquerade

2. Replay: It involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

For example consider Figure 1.3.9 with scenario that Bob sends message to Alice to add amount of Rs. 500/- to Darth's account. Darth captures the message and replays after few days. Alice assumes that this is new message from Bob and he adds amount to Darth's account.

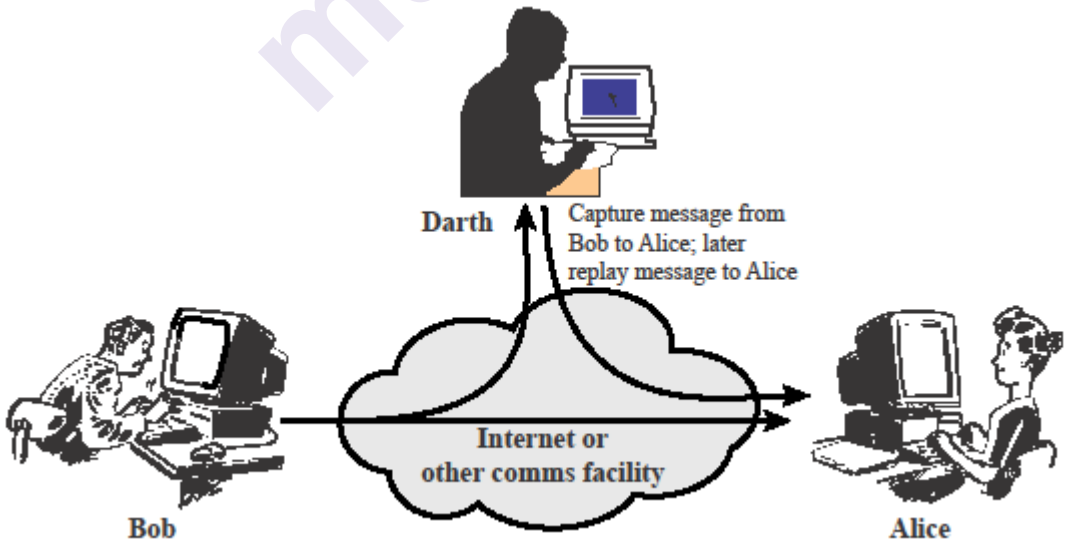


Fig. 1.3.9 Replay

3. Modification of message: It involves some portions of message is altered or the messages are delayed or recorded, to produce an unauthorized effect.

For example, Bob sends message to Alice as “Allow John to access confidential file X". In transmission the Darth intercept the message and change it for its own benefit as “Allow Darth to access confidential X file."

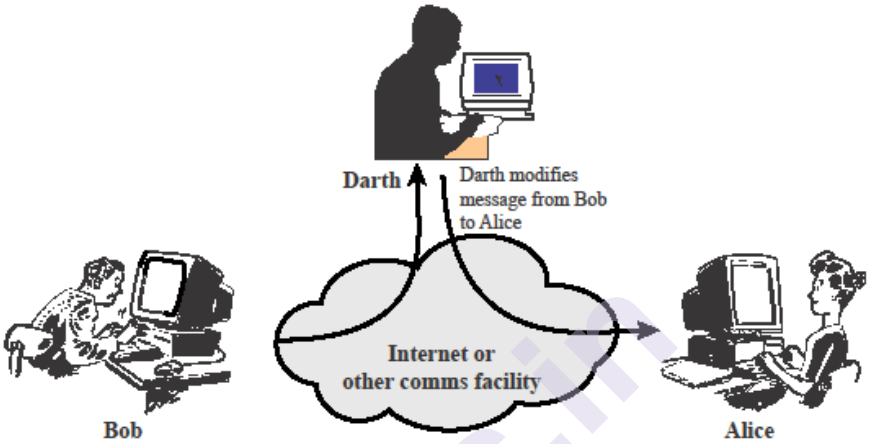


Fig. 1.3.10 Modification of messages

4. Repudiation

Sender or receiver performs this attack. The sender or recipient might subsequently deny sending or receiving a communication. The client, for instance, asks his bank "To transfer the sum to someone" and, subsequently, refuses the sender (customer) to make the request. This is disapproval. Figure 1.3.11 represents Repudiation where Darth denies previously sent message to Alice

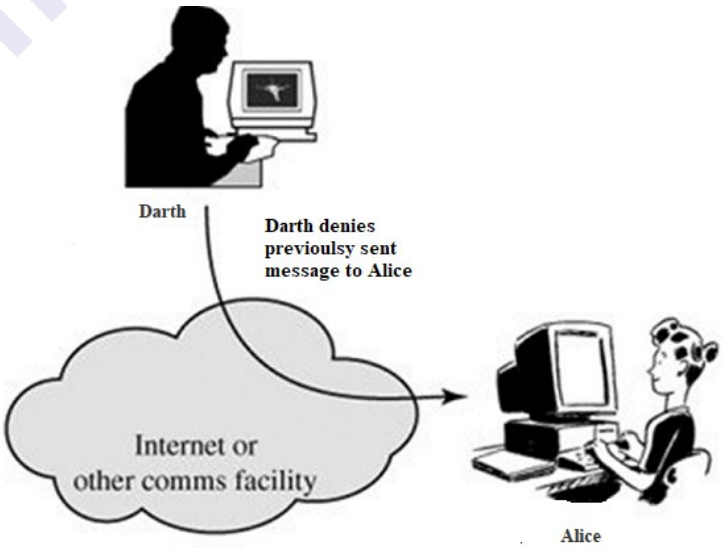


Fig. 1.3.11 Repudiation

5. Denial of service (DoS): Fabrication causes denial of service attacks. Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

As in Figure 1.3.12, Darth saturates a server with an overwhelming number of packets, resulting in denial-of-service attack and the server is unable to service the request of genuine user (Bob). In order for most DoS flood attacks to be successful, the malicious Darth must have more available bandwidth than the target

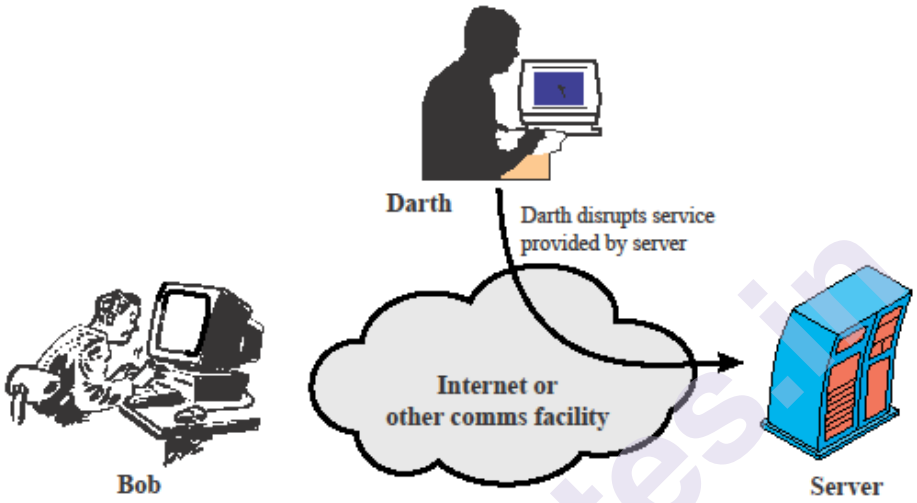


Fig. 1.3.12 Denial of service

The usual use of communication systems can be prevented. This attack might have a specific goal. For instance, the entity can remove all communications to a certain location. The interruption of the whole network by deactivating the network and/or overloading it by messages to deteriorate performance is also a type of service denial.

It is quite difficult to prevent active attacks absolutely, because to do so would require physical protection of all communication facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them.

Attacks	*Passive/Active*	*Threatening*
Snooping Traffic analysis	Passive	Confidentiality
Modification Masquerading Replaying Repudiation	Active	Integrity
Denial of service	Active	Availability

Comparative Points of Passive and Active Attack:

ACTIVE ATTACK	PASSIVE ATTACK
Attacker needs to have physical control of the media or network.	Attacker merely needs to observe the communication in the media or network.
It can be easily detected.	It cannot be easily detected.
It affects the system.	It does not affect the system.
It involves a modification of data.	It involves the monitoring of data.
Types of active attacks are Masquerade, session replay, denial of service, distributed denial of service.	Types of passive attacks are the Release of a message, traffic analysis.
It does not check for loopholes or vulnerabilities.	It scans the ports and network in the search of loopholes and vulnerabilities.
It is difficult to prevent network from active attack.	Passive attacks can be prevented.

1.4 SECURITY SERVICES

The International Telecommunication Union-Telecommunication Standardization Section (ITU-T) provides some security services and some mechanisms to implement those services

Security service means a processing or communication service that is provided by a system to give a specific kind of protection to system resources.

ITU-T (X.800)divides Security services as shown in the following Figure 1.1.4.1

Fig 1.4.1 Security Services

1. **DATA CONFIDENTIALITY:**

It's designed to protect data from disclosure attacks. There are four types of confidentiality services defined by ITU-T standard.

- **Connection Confidentiality:** All user data on a connection link is protected.
- **Connectionless Confidentiality:** All user data protected data in a single data block
- **Selective-Field Confidentiality:** Selected fields within the user data on a connection or in a single data block are kept confidential.
- **Traffic-flow Confidentiality:** The protection of the information that might be obtained from traffic flow observation.

2. **DATA INTEGRITY:**

It is designed to protect data from modification, insertion, deletion and replay. There are five types of integrity services defined by ITU-T standard.

- **Connection Integrity with Recovery:** Maintains the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence with recovery attempted.
- **Connection Integrity without Recovery:** Maintains the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence without recovery attempted.
- **Selective-Field Connection Integrity:** Determines whether selected fields within the user data of a data block transmitted over a connection link have been modified, inserted, deleted, or replayed.
- **Connectionless Integrity:** Provides integrity for a single connectionless data block which can provide data change detection.
- **Selective-Field Connectionless Integrity:** Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.

3. AUTHENTICATION :

The confirmation that the communicating party is who it says it is. There are two types of authentication services defined by ITU-T standard.

- **Peer Entity Authentication:** Used in association with a logical connection, it provides assurance that the people connected are who they say they are.
- **Data-origin Authentication:** In a connectionless transfer, provides assurance that the source of received data is as claimed.

4. NONREPUDIATION:

Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication. There are two types of nonrepudiation services defined by ITU-T standard.

- **Nonrepudiation, Origin :** Proof that the message was sent by the specified party.
- **Nonrepudiation, Destination :** Proof that the message was received by the specified party.

5. ACCESS CONTROL:

Access control provides prevention of unauthorized access of a resource (includes hardware, software, firmware, information/ data, and telecommunications). Access control is the ability to limit and control the access to host systems and applications via communications links. This service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do.

1.5 FUNCTIONAL REQUIREMENTS OF SECURITY

Functional requirements define the basic system behavior. Essentially, they are what the system does or must not do, and can be thought of in terms of how the system responds to inputs. Functional requirements usually define if/then behaviors and include calculations, data input, and business processes.

Functional requirements are features that allow the system to function as it was intended. Put another way, if the functional requirements are not met, the system will not work. Functional requirements are product features and focus on user requirements.

The functional requirements of security should be

- Able to uniquely identify individual system users
- It must include two-factor authentication for system access
- It must include notification and user acknowledgment at the login
- It must manage passwords and password processing securely
- It should encrypt authentication and authorization mechanisms
- It must grant the minimum, sufficient access or privileges to the user according to roles of users or their job duties
- Allow access to sensitive data only as necessary for users job duties
- User should log out or lock unattended workstations
- Users access should be revoked upon termination of appointments
- Owner should review accounts at least annually

- It should designate owners to manage privileged and shared accounts
- It should meet related regulatory and/or contractual obligations

Functional requirements describe what a system has to do. So functional security requirements describe functional behavior that enforces security. Functional requirements can be directly tested and observed. Requirements related to access control, data integrity, authentication, and wrong password lockouts fall under functional requirements.

In short, we can say that the functional requirement of the system should have the ability to protect data, resources, and services from unauthorized access and other threats or attacks that could potentially result in harm to the information system. Ultimately, the data should be accurate, secured, and consistent across the entire system

1.6 CURRENT TRENDS IN SECURITY

The computing environment is now a highly interconnected network of smaller systems. Now a day's anyone can access a computer from almost anywhere on the planet. So it complicates the job of the security professional. Day by day the attackers are increasingly stealing and committing fraud and other crimes for their benefit. Computer criminals are increased because the rewards are getting more than robbing the local store.

Below are three current trends in the information security field are

1. Wireless Security
2. Bluetooth Technology and Security
3. Mobile Security

1. Wireless Security

Wireless communications carry the information through the radio frequency carriers. Wireless communications can occur over a wireless network when a wireless device connects to a wired network, or between two wireless devices. Laptops, mobile phones, Personal computers, and tablets communicate with each other through a wireless network. Modulators, demodulators, transmitters, receivers, and wireless networks ensure that the messages sent from a correct source are received at the intended destination. Ease of use including the mobility of access points and no need for extensive cabling, comparatively lower cost of installation, and ability to work on multiple types of networks like 2G/3G, etc. Wireless access points connect you to the internet or a wired network. The systems and the devices which connect to the wireless access point are known as stations. Wi-Fi adapters are required by stations to communicate with the wireless access point. 3G or 4G USB-based data cards can be used for Wireless access.

Though it is cost-effective, easy to use, and easy implementation it has certain information security issues which we have to consider. Plain text communication through wireless is vulnerable to sniffing, eavesdropping, and man-in-the-middle attack. The solution to information security issues is wireless encryption. The two Wi-Fi encryption standards are Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA). WPA is more effective but WEP can be easily cracked.

Service Set Identifier (SSID) is used to identify each Wi-Fi network. It is recommended that this SSID be changed to a name that is neutral in representation. Wi-Fi authentication can happen using a centralized authentication server, open system authentication, or shared key authentication by the access point. To strengthen the authentication process MAC filtering may be enabled. Various discovery and analysis tools are available for a price or for free. These enable the analysis of the wireless frames and man-in-the-middle attacks, including denial of service attacks, by sending fake packets or sending fake Clear to Send (CTS) signals in the name of a fake client making other clients wait indefinitely for their turn. The tools like NetSurveyor, NetStumbler, WiFi Hopper, and Meraki WiFi Stumbler allow for easy network discovery. Powerful jamming signals can be used to jam the entire Wi-Fi network of an organization this leads to denial of service attack.

Best practices to be followed to avoid or reduce wireless attacks:

- Default SSID should be reset in the beginning into a neutral name.
- MAC filtering should be enabled on the access point or the router.
- Firewalls should be enabled between the access point and the internal network for corporations.
- Encryption should be enabled.
- Pass-phrase should be periodically changed.
- A strong access password should be set on the router.
- The strength of the wireless signal should not be high so that it can be accessed from the outside of the organization.
- Better encryption mechanisms should be used like WPA.
- Use a centralized server for authentication.
- Access points should not be accessed by others.
- Drivers should be up-to-date.
- If possible use Wireless Intrusion Prevention Systems.
- Periodic audits should be carried out for wireless systems and if any issues are found fix them.
- Penetration testing should be carried out at least once a year on wireless network systems and if any issues found are fixed appropriately.

2. Bluetooth Technology and Security

For short-range communication, Bluetooth is used. Bluetooth technology is used for transferring files from one mobile device to another other. Though the range of communication is short the possibility of hacking cannot be denied. The two devices that want to connect need to be paired. The Bluetooth devices communicate with each other using wireless networks so there is the possibility of attacks, such as denial of service, and copying of the files from the connected devices. So, Bluetooth is turned on only when required. If Bluetooth is enabled and the configuration is in pairable mode, then other devices can connect and copy the files or information from the connected device.

Bluetooth attackers may use others' mobile phones to send messages to others.

Similarly, Bluetooth attackers may insert malicious code on mobile phones using Bluetooth technology and then the attacker can have complete control over the phone of the victim. Contact details can be stolen and misused. Other mobile phones can be used for making phone calls or for connecting to the internet.

Best practices to be followed to avoid or reduce attacks on Bluetooth communication:

- Have the devices securely with you or store them securely when not used by you.
- Pre-shared key authentication and encryption should be used for Bluetooth communication.
- Bluetooth should be enabled only when required to communicate.
- Bluetooth should be set to non-discoverable and non-pairable mode.
- Set discoverable and pairable mode only when you need to connect to other devices or vice versa.
- Always remove unwanted devices from the list of paired devices.
- Carry out the pairing of the devices only in a secure area. Have the paired device as near as possible to the other device with which it is pairing.
- Anti-virus systems should be in place.
- The device firewall should be active all time.
- Regular patching of the Bluetooth devices should be done.

3. Mobile Security

Now a day's usage of mobile phones, smartphones, and tablets is increasing. Different operating systems are used in these devices. Mobile devices are being used for sending e-mails, instant messaging, gaming, and various official / personal purposes. Many mobile apps are available free or at a very low cost. Due to mobile devices lives of users become easy and more active; however, at the same time, they have created a number of security issues.

If unauthorized users will be able to access the mobile device then there is a possibility of leaking personal information such as login ids, credit card details, and passwords. The security of the various apps or games we download is also questionable. The apps or games might have been created with malicious intent and can infect mobile devices. The attackers might misuse the intercepted information.

Operating systems used by mobile devices may have multiple security issues or vulnerabilities. These security issues may be misused by the attackers.

Generally, users do not install anti-virus software on mobile phones. It is advised to install good anti-virus software such as Avast, Norton, or McAfee on mobile phones to protect from malicious attacks. Mobile phone settings should be reviewed regularly and set appropriately. Always disable unnecessary settings/features.

Best practices to be followed to avoid or reduce attacks on Mobile Communication:

- On mobile devices download apps from only authorized stores only.
- Download unknown apps or even known apps from trusted websites only.
- Regular updates of the mobile operating system and the apps should be done.
- Always mobile phones should be locked with a strong PIN.
- On mobile phones do not open links from unknown sources.
- Strong encryption mechanism should be used while using Wi-Fi.
- Confidential data and other sensitive information from your mobile phone should be transferred to other security devices or secondary storage devices under your control.
- Keep limited data on your mobile phone.
- Secure websites should not be accessed using unsecured Wi-Fi connections. There is the possibility of credentials being captured or sniffed by others.
- Mobile phones should be locked after a certain amount of idle time.
- Unwanted settings should be enabled only when required.

1.7 SUMMARY

In this chapter we studied what is security, the need of information security, the security principles, and types of attack and security services. Here we also discussed the current trends of security and functional security requirements.

1.8 MULTIPLE CHOICE QUSETION ANSWERS

i. The _____attack is related to authentication.

A. Interception
B. Fabrication
C. Modification
D. Interruption

Ans : B

ii. The _____attack is related to integrity.

A. Interception
B. Fabrication
C. Modification
D. Interruption

Ans : C

iii. Interruption attacks are also called _____attacks.

A. Denial of Service (DoS)
B. Fabrication
C. Masquerade
D. Replay

Ans : A

iv. Which following considered is not considered violation confidentiality?

A. Stealing passwords
B. Eavesdropping
C. Social engineering
D. Hardware destruction

Ans : D

v. In _____ attack the message contents are modified.

A. passive
B. active.
C. Both of the above
D. None of the above.

Ans : B

1.9 TRUE OR FALSE

i. **DOS attacks are caused by Fabrication.**
---False

ii. **Violations of confidentiality are limited to direct intentional attacks.**
---False

iii. In passive attack the message contents are modified**.**
---False

iv. If a computer system is not accessible, the principal of availability is violated.
----True

1.10 SAMPLE QUESTIONS

1. What are the key principles of security?
2. What is the difference between passive and active security attack?
3. Discuss the different security services.
4. List the functional requirements of security.
5. Explain in detail current trends in security.

1.11 LIST OF REFERENCES

1. https://www.opensecurityarchitecture.org/cms/definitions/it_security_requirements
2. https://link.springer.com/content/pdf/10.1007%2F978-1-4302-6383-8_16.pdf
3. Atul Kahate, “Cryptography and Network Security”, McGraw Hill
4. Cryptography and Network Security: Principles and Practice, William Stallings
5. Cryptography and Network Security, Behrouz A Forouzan

❄❄❄❄❄❄❄

2

CRYPTOGRAPHY

Unit Structure

2.0 OBJECTIVE

- Understand the concepts of cryptography and its type and its applications.
- Discuss the requirement of information security, private and public key algorithms and to examine the mathematics of cryptography.
- Learn different ways of securing data by using different algorithm.
- To review integer arithmetic, concentrating on divisibility and finding the greatest common divisor using the Euclidean algorithm.
- To emphasize the importance of modular arithmetic and the modulo operator, because they are extensively used in cryptography.

2.1 INTRODUCTION

Cryptography is the study and practice of strategies for at ease communication in the presence of 1/3 parties called adversaries. It deals with the developing and analyzing protocols which prevents malicious 1/3 parties from retrieving statistics being shared between two entities thereby following the various elements of data security.

The idea of verbal exchange refers to the scenario wherein the message or facts shared among two events can't be accessed by adversary. In cryptography, an adversary is a malicious entity, which aims to retrieved treasured data or records thereby undermining the standards of records safety.

Information Confidentiality, Statistics, Integrity, Authentication and Non-repudiation are core ideas of modern-day-day-cryptography.

2.2 CRYPTOGRAPHY

Cryptography is the look at a securing communication from out of doors observers. Encryption algorithms take the unique message or plain text and convert it into ciphertext, which is not converts it into ciphertext, which is not understandable. The important thing permits the consumer to decrypt the message, for that reason making sure on they could read the message. The strength of the randomness of an encryption is also studies, which makes it tougher for every person to wager the important thing or input of the algorithm. Cryptography is how we will attain greater at ease and robust connections to elevate our privacy. Advancements in cryptography makes it harder to break encryptions in order that encrypted documents, folders, or community connections are most effective handy to legal customers.

Definition

It is the science of using mathematics to encrypt and decrypt data.

In another language you can say that cryptography is the way of securing data. Where cryptanalysis is the way of securing and breaking the communication.

History

It is the art of writing. As civilizations evolved, human beings got organized in tribes, groups, and kingdoms. This led to the emergence of ideas such as power, battles, supremacy, politics. These ideas further filled the natural need of people to communication secretly with selective recipient which in turn ensured the continuous evolution of cryptography as well.

The cryptography was found in Roman and Egyptian civilizations.

A message or text is plaintext or called cleartext. This process of converting a message or text in such a way as to hide its essence encryption. An encrypted message is ciphertext. The process of again ciphertext back into plaintext is decryption.

Caser Shift Cipher

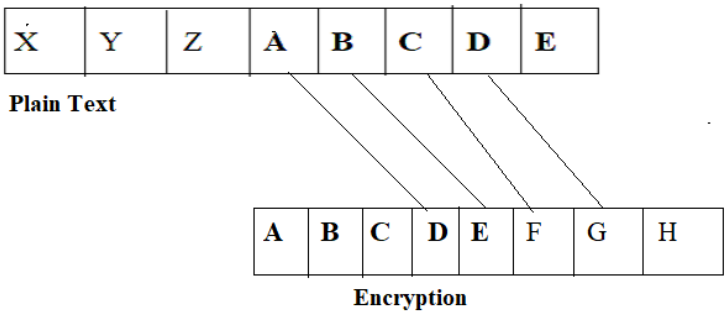
Caser Shift Cipher have logic to shift the letters of a message by number (three were common choice) the recipient of this message would then shift the letters back by the same number and obtain the original message. It is the simple and easy method of encryption. It's a type of substitution cipher that is each letter of a given text is replaced by a letter some fixed number of positions down the alphabet.

Encryption is the process of converting normal message (plaintext) into meaningless message (Ciphertext)

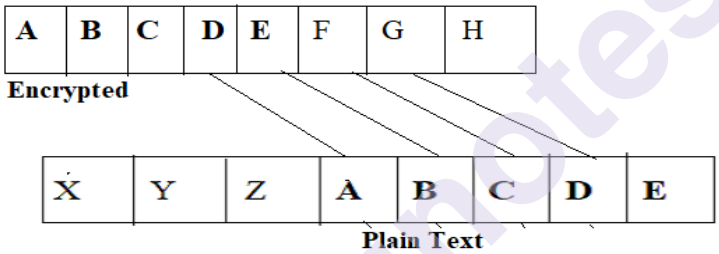
Decryption is the process of converting meaningless message(ciphertext)into its original form(plaintext)

Examples:

Suppose we have plaintext ABCDE and convert into Caesar Shift Cipher.so we must made it into encrypted text and decrypted text for original plain text.



To get original text that is plaintext we have decrypt text from encrypted text.



In other way means when you make it mathematical way you can give the numbers to the alphabets that is

A B C D E F G H I J K L M N

0 1 2 3 4 5 6 7 8 9 10 11 12 13

O P Q R S T U V W X Y Z

14 15 16 17 18 19 20 21 22 23 24 25

Suppose “text” message which is converted into 1942319

That is,

T->19

e->4

x->23

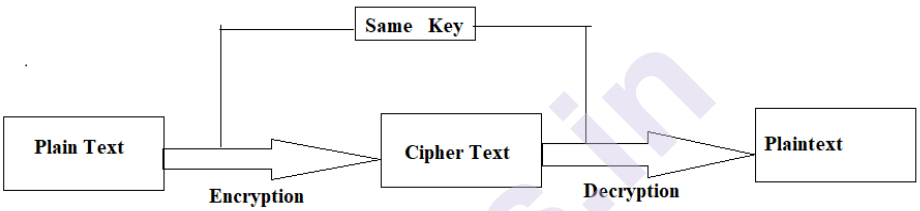
t->19

2.2.1 Types of Cryptography

1) Symmetric Key Cryptography
2) Asymmetric Key Cryptography
3) Hash Function

1) **Symmetric Key Cryptography:** It is also called Conventional cryptography or secret Key Cryptography. In this while doing the encryption process the sender and receiver shares common single key for encryption and decryption of text or message. The key for encrypting and decrypting had known to sender as well as receiver. Otherwise, the message could not be decrypted by conventional means.

The algorithm use is also known as a symmetric algorithm or sometimes called a secret key algorithm.



Advantages of symmetric cryptography

1) It is simple and fast.
2) The sender and receiver exchange the key in secure way.

Disadvantage of symmetric cryptography

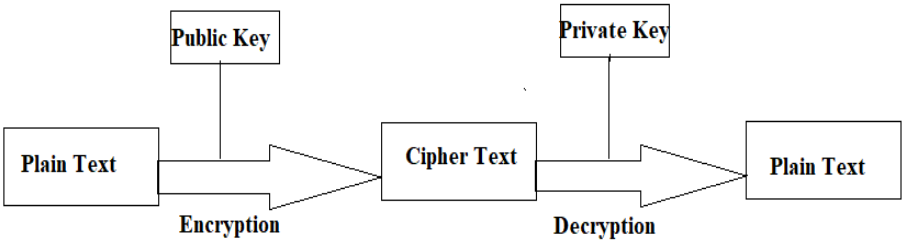
1) If the key is leaked, the message can be easily changed that considered risk.

Symmetric Key Cryptography examples

a) Data Encryption Standard (DES)
b) Triple Encryption
c) Advanced Encryption Standard (AES)

2) Asymmetric Key Cryptography:

It is also known as public-key cryptography, refers to a cryptography algorithm which requires two separate keys, one of private and another is public. The first key which is used to encrypt the message that is public and another key which is used to decrypt the message that is private.



Asymmetric Key Cryptography Examples

a) Digital Signature Standard (DSS)
b) Algorithm-RSA
c) EI Gamal

Difference between Symmetric and Asymmetric Encryption

Sr. No	Symmetric Encryption	Asymmetric Encryption
1	It is used single key for encryption and decryption.	It uses two different keys one for encryption and another for decryption.
2	It is used to transmit large amount of data.	It is used to transmit for small amount of data.
3	It uses low resource consumption.	It uses high resource consumptions.
4	It is Fast Technique.	It is slow technique.
5	Length of the keys are 128- or 256-bit size	Length of the key are (RSA) 2048 bit
7	It is less secured as uses single key.	Much secure as uses two different keys.
8	It is an old technique.	It is modern encryption technique.
9	Ex.RC4, AES, DES,3DES and QUAD	EX.RSA, Diffe-Hellman, ECC Algorithm

3) Hash Function:

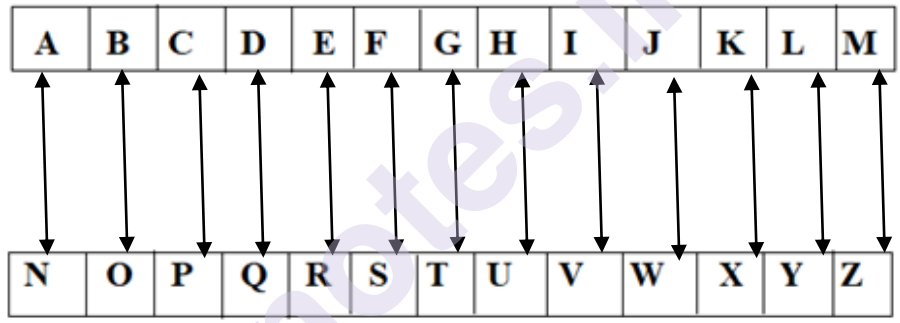
It is a hash function that takes random size input and yields a fixed size output. It is a easy to calculate but difficult to access an original data. It is strong and difficult to duplicate the same hash with unique inputs as hash function .it is one-way function so once you started you cannot go back. It is same as Digest, message digest, checksum etc.

By using hash function

1) Check Digital Signature
2) Message assurance
3) Source integrity service using MAC (Media Access Control) address.
4) Key establishments algorithm
5) To generate random numbers

2.2.2 Mathematics of Cryptography:

Cryptography is the mathematics behind encrypting data. Whenever user wants to transform plaintext data into ciphertext information. There are operations or a set of operations that is used. For example, ROT13 is a type of symmetric encryption algorithm that use some rotational operational on a plaintext data. hence this operation is symmetric and reversible. Thus, the ciphertext data can turn back into a plaintext data.so consider all alphabets with 13 alphabets.



Ex. HELLO - Plain text

ROT13

URYYB- Cipher text

Use of mathematics in cryptography. Whenever we must encode text plaintext to into encrypted form that time, we have to use mathematical technique to encode plain text with hash and perform crypto analysis from encrypted keys to identify the original text.

The maths plays important role in cryptography to find out how keys behave at sender and receiver side.

There are other types of algorithms that use cryptography mathematics like

- Big ‘O’ Notation: It is Asymptotic Notation which is used to describe the running time of an algorithm.

- Prime Factorization: This is commonly used technique uses in multiplication of two large prime numbers.
- Pseudo Random Number Generation: To generate random number sequence
- The Birthday Problem: In group number of people have birthday on same day. It is useful for checking probability
- The RSA Algorithm: It is asymmetric algorithm.
- The Diffie-Hellman Algorithm: This algorithm is used for data security. It is also called DH Algorithm.

2.2.3 Modular Arithmetic Additive Inverse:

In this, we are interested in only one of the outputs, the remainder r. modular arithmetic we use in daily life. Example we use a clock to measure time. Our clock system uses modulo 12 arithmetic. However, instead of a 0 we use the number12.

Consider we are divided two integers and have equations like

$$\frac{A}{B} = Q$$

Then we can say from these equations.

R-remainder
A-divided
B-Divisor
Q-quotient

When we want to work on remainder that time A/B for theses we use operator modulo operator called mod.

Example:(10 mod 3) =3 remainder 1

When we are working in modular arithmetic, we often need to find the inverse of a number relative to an operation. We are normally looking for an additive inverse (relative to an addition operation) or a multiplicative inverse (relative to a multiplication operation).

In this, each integer has an additive inverse. The sum of an integer and its additive inverse is congruent to modulo n.

Congruence Modulo

$$A \equiv B \pmod{C}$$

You can say A is congruent to B modulo C

In Zn, two numbers a and b are additive inverses of each other if

$$a + b \equiv 0 \pmod{n}$$

Where To show that two integers are congruent, we use the congruence operator (≡) example

3 ≡ 8 (mod 5) and 15≡25 (mod 10).

Ex.1) Find all additive inverse pairs in Z10.

Sol=> We have m=10

By using modular arithmetic, the sum of an integer and its additive inverse is congruent to 0 modulo n.

So, we must find all pairs that is equal to 10

The six pairs of additive inverses are (0, 0), (1, 9), (2, 8), (3, 7), (4, 6), and (5, 5).

2.2.4 Multiplicative Inverse:

In modular arithmetic, an integer may or may not have a multiplicative inverse. When it does the product of the integer, and its multiplicative inverse is congruent to 1 modulo n.

In Zn, two numbers a and b are the multiplicative inverse of each other if

$$a \times b \equiv 1 \pmod{n}$$

suppose a = 3, m = 11

sol: 4

Since (4*3) mod 11 = 1, 4 is modulo inverse of 3(under 11).

One might think, 15 also as a valid output as "(15*3) mod 11"

is also 1, but 15 is not in ring {1, 2, ... 10}, so not

valid.

suppose a = 10, m = 17

sol: 12

Since (10*12) mod 17 = 1, 12 is modulo inverse of 10(under 17).

Ex.1) Find all multiplicative inverses in Z10.

Sol=> We have n=10

By using Multiplicative Inverse,

$$a \times b \equiv 1 \pmod{n}$$

so, there are only three pairs (1, 1), (3, 7) and (9, 9). The numbers 0, 2, 4, 5, 6, and 8 do not have a multiplicative inverse.

Ex.2) Find the multiplicative inverse of 8 in Z10.

Sol=> We have n=10

By using Multiplicative Inverse,

a x b ≡ 1 (mod n)

so, there is no multiplicative inverse because gcd (10, 8) = 2 ≠ 1. In other words, we cannot find any number between 0 and 9 such that when multiplied by 8, the result is congruent to 1.

Ex.3) Find all multiplicative inverse pairs in Z11.

Sol=> We have n=10

By using Multiplicative Inverse,

`a x b ≡ 1 (mod n)`

so, we have seven pairs (1, 1), (2, 6), (3, 4), (5, 9), (7, 8), (9, 9), and (10, 10)

2.2.5 Euclidean Algorithm

GCD (Greatest Common Divisor) of the two numbers is the largest that divides both. Suppose two numbers A and B of two integers and we must find the Greatest Common Divisor (GCD).

This algorithm has technique to find the GCD of two integers quickly.

Definition:

A method of finding the greatest common divisor of two numbers by dividing the larger by the smaller, the smaller by the reminder, the first remainder by the second remainder you have to find until exact division is found once the greatest common divisor is the exact divisor.

Fact 1: gcd (a, 0) = a

Fact 2: gcd (a, b) = gcd (b, r), where r is

the remainder of dividing a by b

The Euclidean Algorithm for finding GCD (A, B) is as follows:

- If A = 0 then GCD (A, B) =B, since the GCD (0, B) =B, and we can stop.
- If B = 0 then GCD (A, B) =A, since the GCD(A,0) =A, and we can stop.
- Write A in quotient remainder form (A = B·Q + R)
- Find GCD (B, R) using the Euclidean Algorithm since GCD (A, B) = GCD (B, R)

Ex1). First let me show the computations for a=210 and b=45.

Sol=> Divide 210 by 45, and get the result 4 with remainder 30, so 210=4·45+30.

Divide 45 by 30, and get the result 1 with remainder 15, so 45=1·30+15.

Divide 30 by 15, and get the result 2 with remainder 0, so 30=2·15+0.

15 is the greatest common divisor of 210 and 45.

Ex.2) Find the GCD of 270 and 192

Sol=> A=270, B=192

A ≠0

B ≠0

By using division to find that 270/192 = 1 with a remainder as 78. We can say like this

270 = 192 * 1 +78

Ex.3) Find GCD (192,78), since GCD (270,192) =GCD (192,78)

Sol=> A=192, B=78

A ≠0

B ≠0

By using division to find that 192/78 = 2 with a remainder as 36. We can say like this

192 = 78 * 2 + 36

Ex.4) Find GCD (78,36), since GCD (192,78) =GCD (78,36)

Sol=> A=78, B=36

A ≠0

B ≠0

By using division to find that 78/36 = 2 with a remainder as 6. We can say like this

78 = 36 * 2 + 6

Ex.5) Find GCD (36,6), since GCD (78,36) =GCD (36,6)

Sol=> A=36, B=6

A ≠0

B ≠0

By using division to find that 36/6 = 6 with a remainder of 0. We can say like this

36 = 6 * 6 + 0

Ex.6) Find GCD (6,0), since GCD (36,6) =GCD (6,0)

Sol=> A=6, B=0

A ≠0

B =0, GCD (6,0) =6

So, we have shown:

GCD (270,192) = GCD (192,78) = GCD (7 8,36) = GCD (36,6) = GCD(6,0) = 6

So GCD (270,192) = 6

Ex.3) Find the GCD (1424, 3084) using Euclidean Algorithm.

Sol=> By using Euclidean Algorithm method perform successive division, first of the smaller of the two numbers into the larger, followed by the resulting remainder divided into the divisor of each division until we get the remainder is equal to zero. At that point, look at the remainder of the previous division that will be the greatest common divisor.

1)

```
          2
      ____________
1424 | 3084
     - 2848
     ------
       236   Reminder
```

2)

```
          6
      ____________
236  | 1424
     - 1416
     ------
        8    Reminder
```

3)

```
          29
      ____________
8    | 236
     - 16
     ------
       76
      -72
      ----
        4    Reminder
```

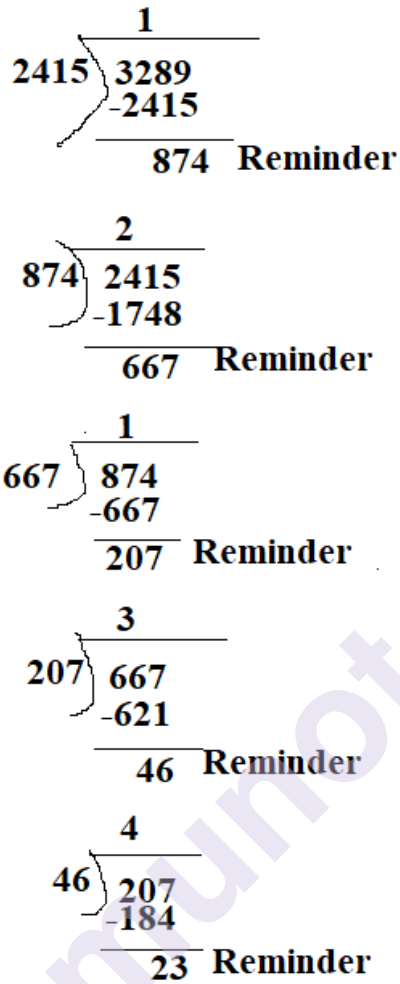
4)

```
          2
      ____________
4    | 8
     -8
     ---
     0    Reminder
```

So, GCD (1424, 3084) is 4.

Ex. GCD (2415, 3289)

Sol=> By using Euclidean Algorithm method perform successive division, first of the smaller of the two numbers into the larger, followed by the resulting remainder divided into the divisor of each division until we get the remainder is equal to zero. At that point, look at the remainder of the previous division that will be the greatest common divisor.



2.2.6 Extended Euclidean Algorithm

While the Euclidean Algorithm calculate sates only the greatest common divisor (GCD) of the two integers and the extended version also finds a way to represent GCD in terms of and i.e., coefficients and for which

a.x+b. y=gcd (a.b)

In other language you can say the Extended Euclidean Algorithm finds a linear combination of a and b equal to (a, b)

This algorithm builds on top of the basic Euclidean Algorithm and helps us in solving certain linear Diophantine equations as well as finding the modular multiplicative inverse, in addition to calculating the greatest common divisor.

Algorithm

We will denote the GCD of a and b with g

The changes to the original algorithm are very simple. If we recall the algorithm, we can see that the algorithm ends with b=0 and a=g. For these parameters we can easily find coefficients, namely g.1+0.0=g

Starting from these coefficients (x, y) = (1,0), we can go backwards up the recursive calls. All we need to do is to figure out how the coefficients x and y change during the transition from (a, b) to (b, a mod b)

Let us assume we found the coefficients (x, y) for (b, a mod b):

b.x1+ (a mod b). y1=g

and we want to find the pair (x, y) for (a, b):

a.x +b. y=g

We can represent a mod b as:

a mod b=a-[a/b]. b

Substituting this expression in the coefficient equation of gives

g=b.x1+ (a mod b). y1=b.x1+(a-[a/b]. b. y1

and after rearranging the terms

g=a. y1+b. (x1-y1. [a/b])

we found the values of x and y

{x=y1

{y=x1-y1. [a/b]

The extended Euclidean algorithm finds the multiplicative inverses of b in Zn
when n and b are given and gcd (n, b) = 1. The multiplicative inverse of b is the
value of t after being mapped to Zn

Linear Diophantine equation:

A linear Diophantine equation of two variables is

a x + by = c

Solution:

x0 = (c/d) s and y0 = (c/d) t

General solutions:

$x = x0 + k\ (b/d)$ and $y = y0 - k(a/d)$ where k is an integer.

2.3 SUMMARY

It is the science of using mathematics to encrypt and decrypt data. In another language you can say that cryptography is the way of securing data. Where cryptanalysis is the way of securing and breaking the communication. Encryption is the process of converting normal message (plaintext) into meaningless message (Ciphertext)Decryption is the process of converting meaningless message(ciphertext)into its original form(plaintext)types of ssymmetric Key Cryptographyand asymmetric Key Cryptography and hash function.

In Mathematics of Cryptography, where we added concepts of math. Modular Arithmetic Additive Inverse and Multiplicative Inverse. To find GCD Euclidean Algorithm and Extended Euclidean Algorithm.

2.4 REFERENCE FOR FURTHER READING.

1) https://www.itu.int/en/ITU-D/Cybersecurity/Documents/01-Introduction%20to%20Cryptography.pdf

2) https://www.cs.umd.edu/~waa/414-F11/IntroToCrypto.pdf

3) https://nayakuch.files.wordpress.com/2015/08/cryptography-network-security-atul-kahate.pdf

4) https://thisismyclassnotes.blogspot.com/2017/05/cryptography-euclidean-algorithm.html

5) https://www.cs.siue.edu/~tgamage/archieved/S15/CS490/L/CR01.pdf

6) https://www.rit.edu/academicsuccesscenter/sites/rit.edu.academicsuccesscenter/files/documents/math-handouts/DM6_EuclideanAlgorithm_BP_9_22_14.pdf

7) https://cp-algorithms.com/algebra/extended-euclid-algorithm.html

8) Cryptography And Information Security, V. K. Pachghare

9) Atul Kahate, “Cryptography and Network Security”, McGraw Hill

Self-learning topics:

Variations of DES – 2DES and 3DES, Symmetric and Asymmetric Key Cryptography together

QUESTIONS

1) What is mean by cryptography? Explain its type.
2) Difference between Symmetric and Asymmetric Encryption.
3) Explain Mathematical Cryptography.
4) What is mean Modular Arithmetic Additive Inverse?
5) What is mean Multiplicative Inverse?
6) Explain Euclidean Algorithm?
7) Explain Extended Euclidean Algorithm?

❄❄❄❄❄❄❄

3

STREAM CIPHER AND BLOCK CIPHER

Unit Structure

3.0 OBJECTIVES

- Understand the concepts of cipher and its type.
- Concept of stream cipher and block cipher.
- Discuss the difference between the stream cipher and block cipher.
- You learn the concept of Confusion and Diffusion.
- You learn the difference mode of operation of block cipher.
- Discuss the algorithm of RSA

3.1 INTRODUCTION

Cipher: The text which is comes from output of encryption algorithm applied to plain text. When a person or device lacking the cipher is unable to read it when data is said to be encrypted. Need to convert into the plain text so that it is in readable form.

Definition: It is an algorithm which is applied to plain text or message to get ciphertext. Cipher text is the unreadable output of an encryption algorithm. The term "cipher" is sometimes used as an alternative term for ciphertext. Ciphertext is not understandable until it has been converted into plain text using a key.

3.2 CIPHER

3.2.1 Types of cipher

1) Caesar Cipher
2) Mono Alphabetic Cipher
3) Homophonic Substitution Cipher
4) Polygram Substitution Cipher
5) Vigenere Cipher

Stream Cipher: It is the type of a symmetric key (secret key) cipher that operates on small units of data (as small as a single bit) at a time. It generates a key stream or sequence of bits using the secret key as a seed.

Stream cipher is that they encrypt data one bit, or byte, at a time. As this cipher uses only one key for encryption as well as decryption so it has fast implementations with low resource consumption. This is useful for encrypting wireless signals, which more naturally fit a streaming model than transmitting data in larger, fixed-size chunks.

This is used one time pad the key typically used with a stream cipher one-time pad is unbreakable because it's always at least the exact same size as the message it is encrypting. Cryptographers also refer to the symmetric key used in a stream cipher as a keystream.

Example of stream cipher is simple substitution.

3.2.2 Types of stream cipher

1) Synchronous stream ciphers: In which the keystream is generated independently of the plaintext and of the ciphertext. The keystream is usually produced by a pseudorandom generator, parameterized by a key, which is the secret key of the whole scheme.

2) Self-synchronizing stream ciphers: the key stream depends on the secret key of the scheme, but also of a fixed number, say *t*, of cipher text digits.

Advantages:

1) Speed-Stream cipher is father transformation.
2) Ease of use-as single key used.
3) Low Consumption-Low resources required.

Disadvantage:

1) Low diffusion: plaintext symbol is contained in a single cipher text symbol.
2) Susceptibility to insertions/ modifications: As it works on bits.

Block Cipher-

Encrypt a group of plaintext symbols as one block. The key is applied to blocks mostly 64 bits size at a time. Size of block 128, 192, or 256 bits. (128 bits used by AES and 64 bits used by DES)

Consider a 128-bit block cipher it required 128 bits of plaintext and encrypts it into 128 bits of ciphertext. Where the amount of plaintext is less than 128 bits.

Advantages:

1) Block Ciphers provides us integrity protection like MAC.
2) It provides ease of implementation and less restrictive required.

Disadvantage:

1) Block Cipher are slow and less memory efficient.
2) In this transmission errors are more caused.

We can say that on block cipher is the process in blocks or multiple bits where stream cipher is the process them to bit or bytes.

Difference between Stream Cipher and Block Cipher.

Sr. No	**Stream Cipher**	**Block Cipher**
1	It encrypted one bit at a time. One time pad)	It encrypted block at a time.
2	It uses only confusion.	It uses both confusion and diffusion.
3	It uses substitution technique.	It uses transportation technique.
4	It is faster than block cipher.	It is slower than stream cipher.
5	It requires less code.	It requires more code.
6	It is more complex.	It is simple.

3.2.3 Confusion:

It means that the key does not relate in a simple way to the ciphertext. Each character of the ciphertext should depend on several parts of the key.

As Stream cipher used one time pad so uses confusion. Nonlinear functions are responsible for confusion.

Confusion = Substitution

a —>b

Caesar cipher

Each bit of the ciphertext block has highly nonlinear relations with the plaintext block bits and the key bits.

X plaintext E $_k$(x) k-key Y ciphertext

Each bit of the ciphertext block has highly nonlinear relations with the plaintext block bits and the key bits. Example: Suppose that x, y and k all have 8 bits.

y1 = x1 + x2 + x3 + x4 + k1 + k2 + k3 + k4

y2 = x2 + x3 + x4 + x5 + k2 + k3 + k4 + k5

y3 = x3 + x4 + x5 + x6 + k3 + k4 + k5 + k6

y4 = x4 + x5 + x6 + x7 + k4 + k5 + k6 + k7

y5 = x5 + x6 + x7 + x8 + k5 + k6 + k7 + k8

y6 = x6 + x7 + x8 + x1 + k6 + k7 + k8 + k1

y7 = x7 + x8 + x1 + x2 + k7 + k8 + k1 + k2

y8 = x8 + x1 + x2 + x3 + k8 + k1 + k2 + k3

As the linear relations bad confusion.

3.2.4 Diffusion

It means that if we change a character of the plaintext, then several characters of the ciphertext should change, and similarly, if we change a character of the ciphertext, then several characters of the plaintext should change.

Block cipher uses confusion and diffusion. Linear functions are responsible for diffusion.

Diffusion = Transposition or Permutation

abcd —>dacb

DES

We can say confusion means relationship between plaintext and ciphertext. where diffusion means spreads the plaintext statistics through the cipher text.

Each plaintext block bit or key bit affects many bits of the ciphertext block.

X plaintext E $_k$(x) k-key Y ciphertext

x, y and k all have 8 bits. If

y1 = x1 + x2 + x3 + x4 + k1 + k2 + k3 + k4

y2 = x2 + x3 + x4 + x5 + k2 + k3 + k4 + k5

y3 = x3 + x4 + x5 + x6 + k3 + k4 + k5 + k6

y4 = x4 + x5 + x6 + x7 + k4 + k5 + k6 + k7

$y5 = x5 + x6 + x7 + x8 + k5 + k6 + k7 + k8$
$y6 = x6 + x7 + x8 + x1 + k6 + k7 + k8 + k1$
$y7 = x7 + x8 + x1 + x2 + k7 + k8 + k1 + k2$
$y8 = x8 + x1 + x2 + x3 + k8 + k1 + k2 + k3$

then it has very good diffusion, because each plaintext bit or key bit affects half of the bits in the output block y.

Difference between confusion and diffusion.

Sr. No	Confusion	Diffusion
1	It obscures the relationship between the plaintext and ciphertext.	It spread the plaintext statistics through the cipher.
2	It is possible through substitution algorithm.	It is possible through transportation algorithm.
3	It is used stream cipher.	It is used only block cipher.
4	Vagueness is increased in resultant.	redundancy is increased in resultant.
5	If a single bit in the key is changed, all the bits in the ciphertext will also have to be changed.	In case a symbol in the plaintext is changed, several or all symbols in the cipher text will also have to be changed.
6	The relation between the cipher text and the key is masked by confusion.	While The relation between the cipher text and the plain text is masked by diffusion.

3.2 5 Modes of Operation of Block Cipher:

The mode of operation of block cipher are configuration methods are allowed to work with large data streams with providing security. The block cipher process into the data block that is fixed block consider n bits means n-bits input of plaintext produce output of n-bits cipher text. If the output is longer than n-bits, then we must divide in sequential block.

Suppose consider the size of a message is larger than the block size in the modes so the long message is divided into a series of sequential message blocks which divides it into blocks, and the cipher operates on these blocks one at a time in the cryptography.

Block cipher are procedural rules of generic block. with the use of different modes of operations.

Criteria of evaluation

1) Identical messages checks that ciphertext of two identical messages are the same.

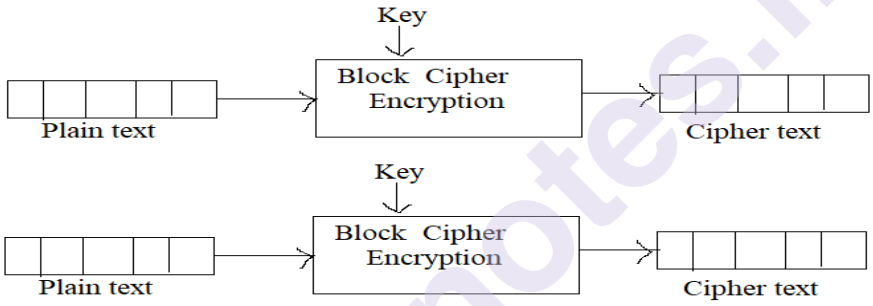
2) Chaining dependencies check adjacent plaintext blocks affect encryption of a plaintext block
3) Error propagation that are resistance to channel noise.
4) Efficiency pre-processing and parallelization random access.

Types of Mode of Operations

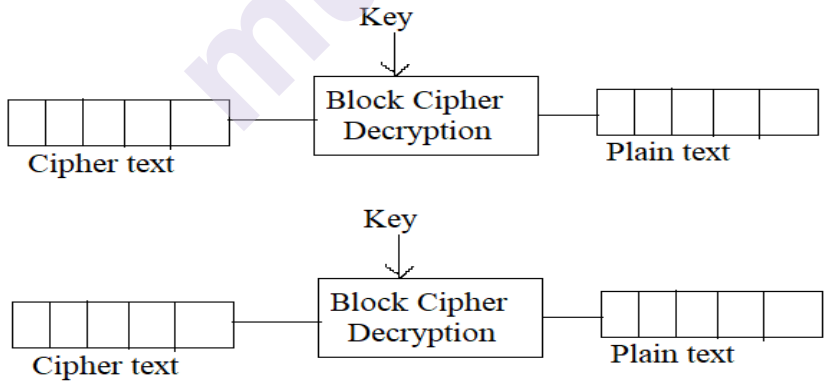
1) Electronic Code Book (ECB)
2) Cipher Block Chaining (CBC)
3) Cipher feedback (CFB)
4) Output Feedback (OFB)

1) **Electronic Code Book (ECB):**

This mode of operation is applied for encryption and decryption. Each block is encrypted independently. It is simple as direct encryption of each block of input plaintext and output is in form of blocks of encrypted ciphertext. Suppose a message is larger than n bits in size, it can be broken down into a bunch of blocks and the procedure is repeated.



Electronic Code Book Encryption



Electronic Code Book Decryption

Advantages

1) It is simple.
2) It is faster because at a time of blocks of bits encrypted.

Disadvantages

1) Not suitable for long message.

2) Cipher Block Chaining (CBC):

In this, data is encrypted in specific blocks, and each block is dependent on the blocks before it for decryption. The process uses something called an initialization vector to help tie these blocks of encrypted data together. the first block of the plaintext is exclusive-OR'd (XOR'd), which is a binary function or operation that compares two bits and alters the output with a third bit, with an initialization vector (IV) prior to the application of the encryption key.

If the first block has index 1, CBC encryption is

$C_i = E_k(p_i \oplus C_{i-1})$,

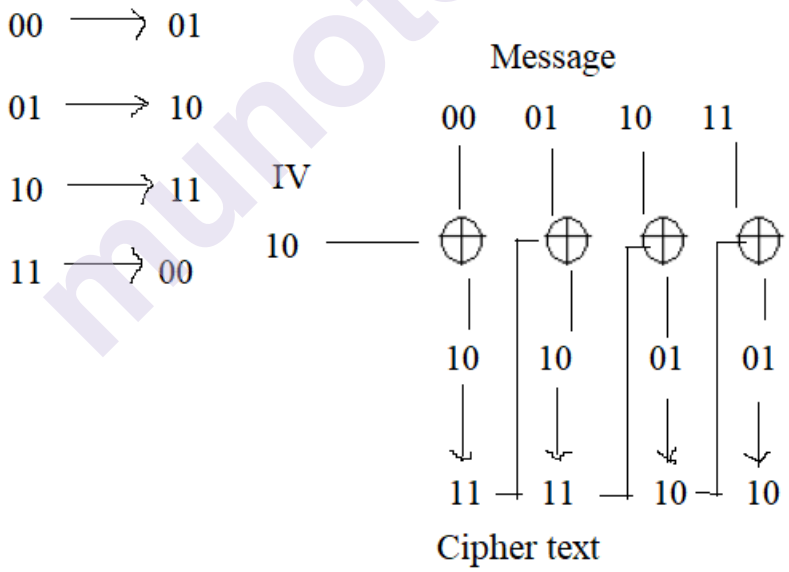
$C_0 = IV$

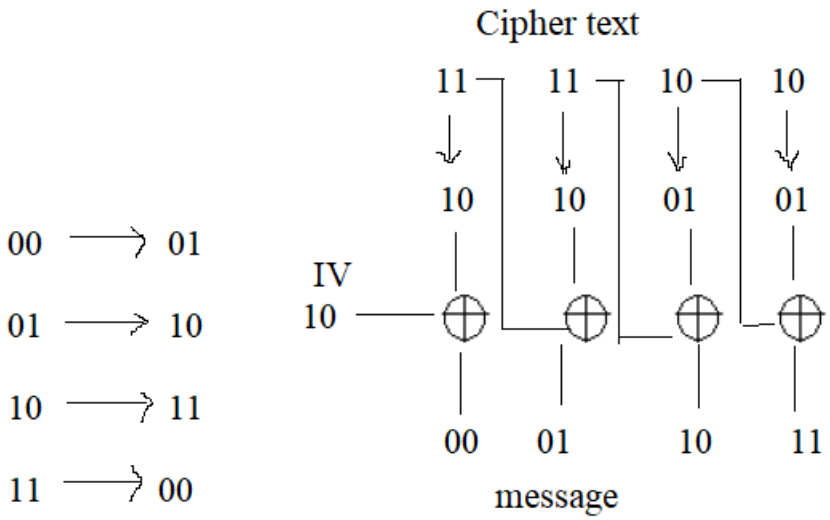
CBC decryption is

$P_i = D_k(C_i) \oplus C_{i-1}$,

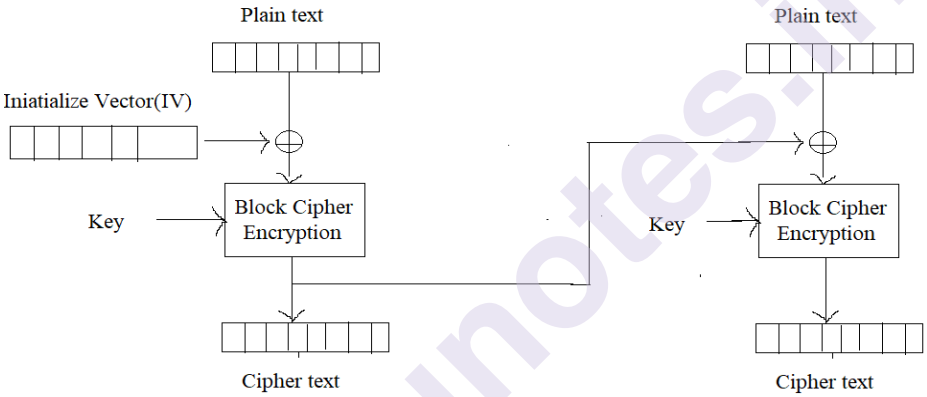
$C_0 = IV$

Ex.

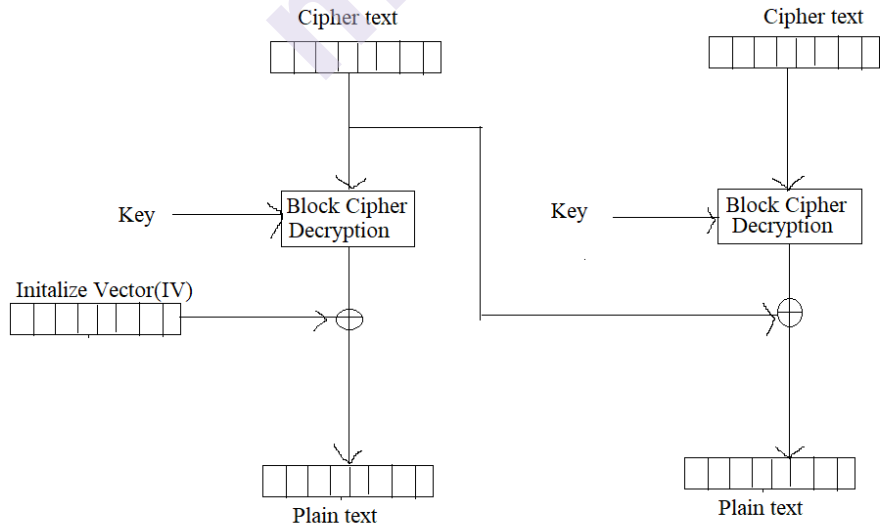




Cipher Block Chaining Encryption



Cipher Block Chaining decryption



Advantages

1) It is more secure rather than electronic code book.
2) It has good authentication mechanism.
3) It also works those have input greater than n bits.

Disadvantages

1) Parallel cipher block chaining is not possible.

3) Output Feedback (OFB) mode: -

Output feedback mode have output from encryption function that is feedback to shift register in cryptography. OFB modes operates on full blocks of plaintext, ciphertext and text or message but it is not work on s-bit subset character.

Each plain text block in XORed with the current output block to cipher text block and which is the encrypted form of the previous output block.

OFM mode notation uses formula,

C[i] = P[i] XOR O[i]

O[i] = Ek(O[i-1])

O [1] = E (Initial Vector)

IV

| -----| -----|

Ek () / Ek () / Ek()

|--O [1] |--O [2] |--O [3]

| | |

P [1] --XOR P [2] --XOR P [3] --XOR

| | |

C [1] C [2] C [3]

That is symmetry of the XOR operation,

$C_j = P_j \oplus O_j$,

$P_j = C_j \oplus O_j$,

$O_j = Ek(I_j)$,

$I_j = O_j\text{-}1$,

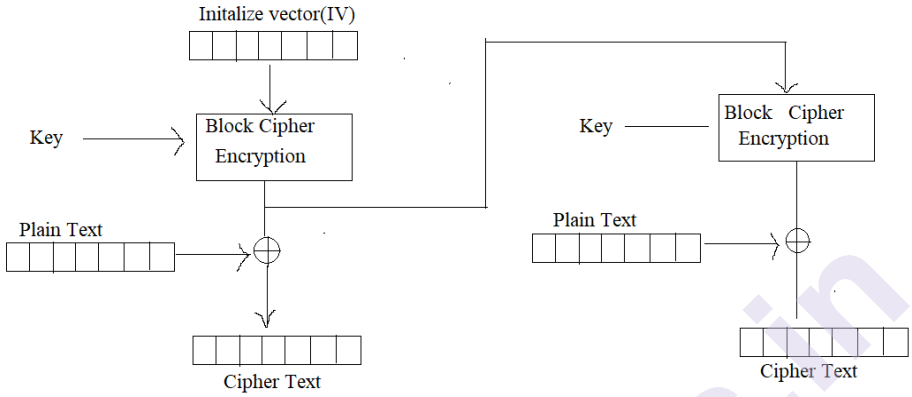
$I_0 = IV$

For encryption,

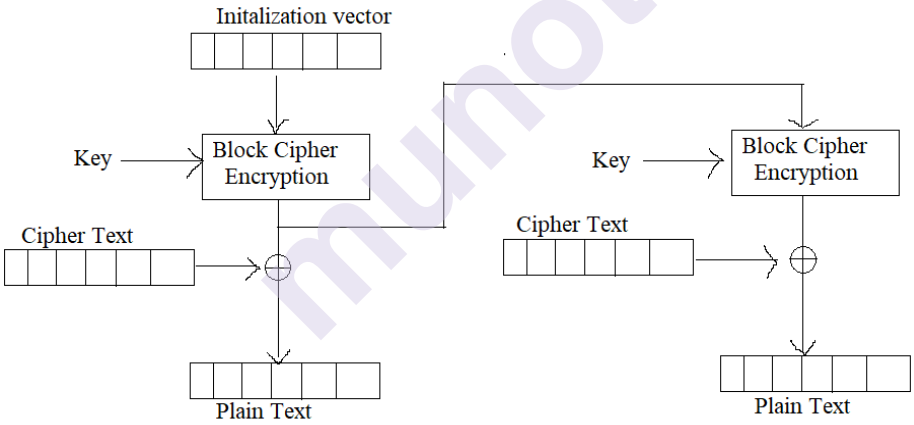
$Cj = Pj \oplus E(K, [Cj - i \otimes Pj - 1])$

For decryption,

$Pj = Cj \oplus E(K, [Cj - 1 \otimes Pj - 1])$



Output Feedback Mode Encryption



Output Feedback Mode Decryption

Advantages

1) By using this you can save our code from unknown users.

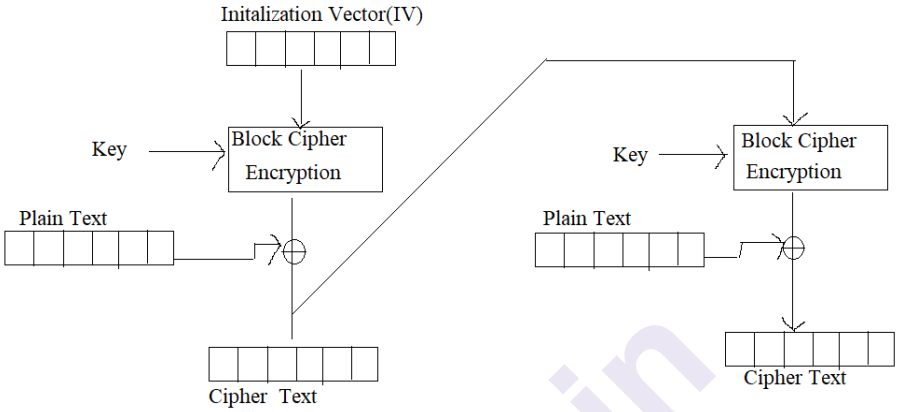
2) While converting text bit errors in transmission do not propagate.

Disadvantages

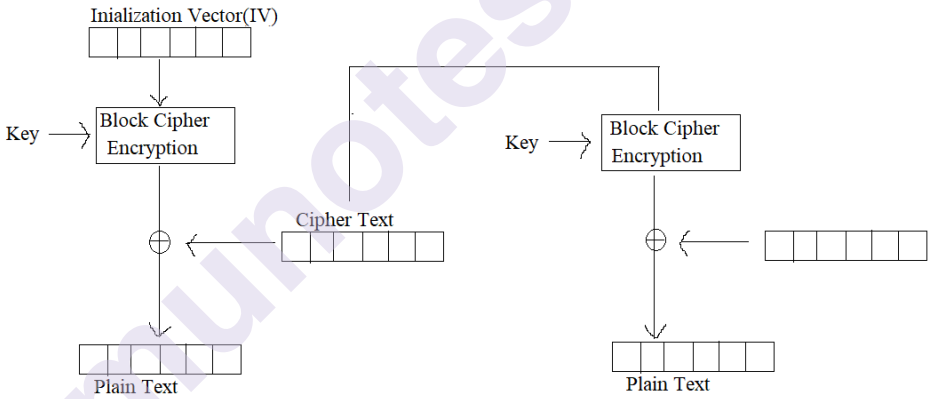
1) It is time consuming process.

4) Cipher Feedback (CFB):

It is also called as cipher feedback. It uses Initialization Vector(IV) previous ciphertext block is encrypted and the output is XORed with the current plaintext block to create the current ciphertext block. an initial vector IV is used for first encryption and output bits are divided as a set of s and b-s bits the left-hand side s bits are selected and are applied an XOR operation with plaintext bits.



Cipher Feedback Mode Encryption



Cipher Feedback Mode Decryption

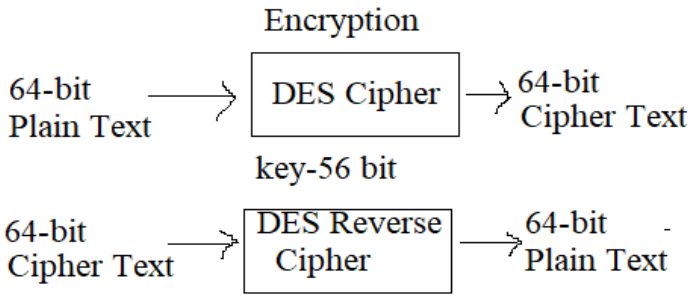
The difference between CFB and CBC is, In CFB mode if bit of data of plain text message has an error or damage due to some reasons the block of ciphertext will damaged. In CBC if one of the messages of cipher text damaged only two received plain text will damaged.

In CFB mode encryption works with only one thread. Where CBC many threads simultaneously.

3.2.2.6 Data Encryption Standard (DES):

Data Encryption Algorithm nothing but Data Encryption Standard which is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST). DES was the first standardized cipher for securing electronic communications. It is used in ECB, CBC or the CFB mode.

It is a block cipher. It encrypts data in blocks of 64 bits each. From these 64 bits used in input of DES that is plain text message of 64 bits and output of it cipher text is 64 bits. For encryption and decryption uses same algorithm with the key 56 bits. The below diagram shows the process.



Algorithm

The algorithm process breaks down into the following steps:

Step1: The process begins with the 64-bit plain text block getting handed over to an initial permutation (IP) function.

Step2: The initial permutation (IP) is then performed on the plain text.

Step3: Next, the initial permutation (IP) creates two halves of the permuted block, referred to as Left Plain Text (LPT) and Right Plain Text (RPT).

Step4: Each LPT and RPT goes through 16 rounds of the encryption process.

Step5: Finally, the LPT and RPT are re-joined, and a Final Permutation (FP) is performed on the newly combined block.

Step6: The result of this process produces the desired 64-bit ciphertext.

Initial Permutation (IP) –

It work at once in starting. It works how the transposition in IP should proceed. example consider IP replaces the first bit of the plain text block or message with the 58th bit of the plain text, the second bit with the 50th bit of the plain text block, and so on.

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

As 64 bits message are divided into two parts that is 32 bit each. And 32 bits are divided

Into 16 rounds these are 5 steps we must follow

1) Key Transformation
2) Expansion permutation
3) S-box permutation
4) P-box permutation
5) XOR and swap

Key Transformation→Expansion permutation→S-box permutation→P-box permutation→XOR and swap

3.2 7 Triple DES:

Triple DES (TDES or 3DES) is symmetric encryption algorithm that involves using DES three times to encrypt a text. While DES encrypts a block of data in 16 rounds, three times or triple DES uses 48 rounds. It is powerful with compare with DES.

3.2.8 Advanced Encryption Standard (AES):

Advanced Encryption Standard is six time faster than triple DES. It is based on 'substitution–permutation network'. In AES operations are in sequential and some of operation which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

AES is depending on the length of the key as it is variable. It uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

3.2.9 RSA Algorithm

As three scientists were found in1978 names Rivest, Shamir and Adleman so algorithm is called as RSA.It is asymmetric algorithm (means sender and receiver uses two different keys) this means this algorithm uses private and public key are used.

The RSA algorithm is set of rules which is based on the mathematical truth that it is easy to locate and multiply massive prime numbers collectively, however it's miles extremely difficult to aspect their product. The private and public keys in RSA are based totally on very big (made of a hundred or greater digits) high numbers. The algorithm itself is simple (unlike the symmetric-key cryptographic algorithms). however, the actual project within the case of RSA is the choice and era of the public and private keys.

Encryption and Decryption process in RSA Algorithms are

Step 1) Consider two prime numbers P& Q

Step 2) Calculate N=P X Q

Step 3) For Encryption key such that it is not factor of (P-1) (Q-1)

Step 4) For Decryption key Equation (D X E) mod (P-1) X(Q-1) =1

Step 5) For Encryption calculate cipher text CT from your message or plain text

$CT=PT^E \bmod N$

Where PT-Plain Text, CT-Cipher Text, E-Encryption

Step 6) send cipher text that is CT to receiver

Step 7) For decryption, calculate the plain text PT from cipher text CT as,

$PT=CT^D \bmod N$

Where PT-Plain Text, CT-Cipher Text, E-Encryption

Ex.1) Consider P=7 and Q=17

Consider P=7 and Q=17 are two large prime numbers. **(From Step No-1)**

N=P X Q **(From Step No-2)**

=7 X 17

=119

For Encryption key(P-1) (Q-1) **(From step 3)**

So, (7-1) (17-1)

=6 X 16

=96

we must choose E that is not in factors

As The factors of 96 are 2, 2, 2, 2, 2, and 3 (As 96 = 2 x 2 x 2 x 2 x 2 x 3).

So, we cannot choose 2 and 3 as encryption Key as it is factor of 96. Also, we cannot choose 8 because 2 X 2 X 2 X 2 same way we cannot choose 9 also as it contains 3 factors so we choose E as 5 because not having factor 2 and 3 so we can go with 5.

For Decryption, (D X E) mod (P-1) X(Q-1) =1**(From Step No-4)**

Put the values of E, P, Q we get,

(D X 5) mod (7-1) X (17-1) =1

(D X 5) mod 6 X 16=1

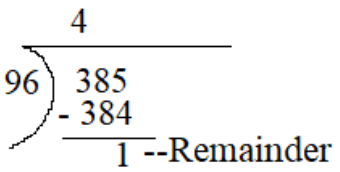
(D X 5) mod 96 =1

(As we want =1 so we have to put D value so that we get equations equal to 1)

So, we consider D=77 and put into it

(77 X 5) mod 96 = 1

385 mod 96=1



For Encryption calculate cipher text CT from your message or plain text

$CT=PT^E$ mod N **(From Step No-5)**

Consider plain text =10

$CT = 10^5$ mod 119 (PT=10, E=5, N=119)

100000 mod 119 = 40

Send 40 cipher text that is CT to receiver **(From Step No-6)**

For decryption, calculate the plain text PT from cipher text CT as,

$PT=CT^D$ mod N **(From Step No-7)**

As we are sent 40 to receiver so PT=40 and CT=119, D=77

By putting values, we get,

$PT=40^{77}$ mod 119

=10 which is original text

3.3 SUMMARY

In stream cipher text each bit or byte converted into encrypted form and again bit or byte are converted into decrypted. In Block cipher text at a time converted into encrypted form and text at a time decrypted. Different mode of operations is used for cryptographic algorithm. There are mainly four types of modes Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher feedback (CFB) and Output Feedback (OFB). Data Encryption Algorithm (DES) nothing but Data Encryption Standard which is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST). Advanced Encryption Standard (AES) is found at least six time faster than triple DES. It is based on 'substitution–permutation network'. RSA algorithm is asymmetric

algorithm. The RSA algorithm is set of rules which is based on the mathematical truth that it is easy to locate and multiply massive prime numbers collectively

3.4 REFERENCES

1) https://pdf.zlibcdn.com/dtoken/44c8a9f519cb7729d12ed9ad43161f45/Cryptography_and_network_security_by_Atul_Kahate)_5472596_(z-lib.org).pdf

2) https://www.practicalnetworking.net/series/cryptography/rsa-example/

3) http://www.crypto-it.net/eng/theory/modes-of-block-ciphers.html

4) https://www.tutorialspoint.com/cryptography/block_cipher_modes_of_operation.htm

5) https://www.geeksforgeeks.org/block-cipher-modes-of-operation/

6) https://www.cs.utexas.edu/~byoung/cs361/lecture45.pdf

7) https://www.coursera.org/lecture/symmetric-crypto/block-cipher-vs-stream-cipher-1uN46

8) https://www.techtarget.com/searchsecurity/definition/stream-cipher

9) s.uok.edu.in/Files/79755f07-9550-4aeb-bd6f-5d802d56b46d/Custom/COnfusion and Diffusion.pdf

10) https://www.includehelp.com/cryptography/mode-of-operation.aspx

11) https://www.commonlounge.com/discussion/6747358d828a45c99f61f4c09ff2f371

12) http://www.crypto-it.net/eng/theory/modes-of-block-ciphers.html

SELF-LEARNING TOPICS:

Variations of DES – 2DES and 3DES, Symmetric and Asymmetric Key Cryptography together

QUESTIONS:

1) What is mean by cipher text. Explain its type.
2) Explain type of stream cipher.
3) Difference between Stream Cipher and Block Cipher.
4) Difference between Confusion and Diffusion.
5) Explain different mode of block cipher.
6) Explain Data Encryption Standard (DES) in details.
7) Explain RSA Algorithm.

❄❄❄❄❄❄❄

4

AUTHENTICATION - I

Unit Structure

4.1 OBJECTIVE

a. To allow authorized users to access the computer and to deny access to the unauthorized users
b. To find authorized users though password, Physical identification, and Biometrics
c. verifying the identity of user or information
d. ensure that the claimants are really what they claim to be
e. avoid compromising security to an imposter

4.2 INTRODUCTION

- Authentication is the process of verifying the identity (here identity means naming, recognition or specification) of a user or information. User authentication is the process of verifying or checking the identity of a user when that user login into a computer system. Here identity means the set of qualities like distinguishing, fingering and trust that make one person or group different from others.

- A computer system does not understand something serving as a signal or suggestion. We do face-to-face communication with our friends so that they can recognize each other. Alternatively computer systems depend on data to recognize others.

- Directing who a person really is consists of two separate steps:

  a. Identification is the act of declaring who a person is.

  b. Authentication is the act of proving that confirmed identity: that the person is who? He or she says He or she is.

- The process of showing, recognizing or giving proof of who is? identification and authentication are simply and often confusing things. Identities, in particular names, are often well familiar, public, and not protected. Authentication is necessarily protected. If a person's identity is publicly known, then anyone can claim to be that person. What separates the claimant from the real person is proof by authentication.
- An authentication system may include a system using a plain-text password. It is an insecure authentication mechanism. Complex system is an authentication system like the kerberos system.
- Advanced authentication systems use thumb impression, iris image or hash values derived from data related to users.

The Difference between Identification & Authentication:

Sr. No.	Identification	Authentication
1	Identities are typically public or well known	should be private.
2	Identification is not reliable	Authentication is reliable
3	Identity is more than just person name: A person bank account number, debit card number, email address, and other things are ways by which people and processes identify you.	Authentication mechanisms use any of three qualities to confirm a user's identity: • Something the user knows • Something the user is • Something the user has

- If the information provided by the user and the information stored with the authenticating system is matched, the user is considered an authenticated user of the system.
- Authorisation also helps to control the access to various applications of the system. It provide the following information about user:
- Authorized to access particular resources in the system?
- Authorized to perform particular operations in the system?
- Authorized to perform particular operations on particular resources?

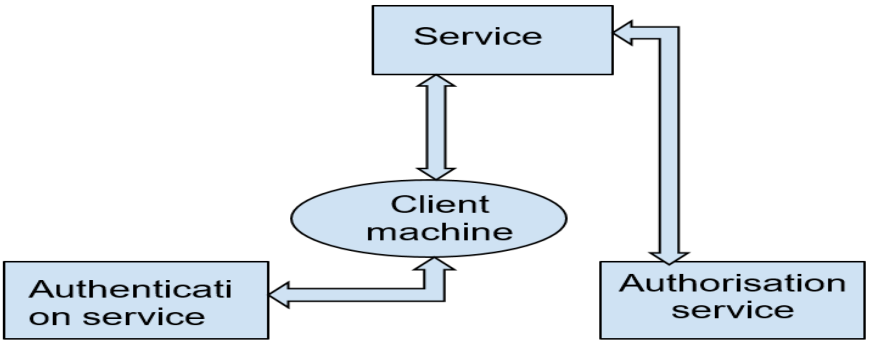


Fig. 1. Authentication V/s Authorisation

- In fig 1 graphically represents the inter-relationship between authentication and authorized system and a distinctive client server application.
- A user working on a client system first logs on to the authentication system to prove his /her identity and request to work on the server. Then the server system communicates to an authorized system to find about rights and privileges, the client user information available on the server.

Sr. No.	Authentication	Authorization
1	Authentication verifies who the user is.	Authorization determines what resources a user can access.
2	Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user.	Authorization works through settings that are implemented and maintained by the organization.
3	Authentication is the first step of a good identity and access management process.	Authorization always takes place after authentication.
4	Authentication is visible to and partially changeable by the user.	Authorization isn't visible to or changeable by the user.

5	Example: By verifying their identity, employees can gain access to an HR application that includes their personal pay information, vacation time, and 401K data.	Example: Once their level of access is authorized, employees and HR managers can access different levels of data based on the permissions set by the organization.

4.3 TYPES OF AUTHENTICATION

The list below reviews some common authentication methods used to secure modern systems.

1. Password-based authentication

- Passwords are the most common methods of authentication. Passwords can be a mixture of a string of letters, numbers, or special characters. To protect our system we need to create strong passwords that include a combination of all possible options.
- Keeping passwords the same for all accounts is prone to phishing attacks and bad hygiene that weakens effectiveness of security. An average person in this world has about 25 different online accounts, but only 54% of users use different passwords across their different accounts.
- The reality is that there are a lot of passwords to remember for a person. As an output, many people choose convenience over security. Most people use simple and easy passwords instead of creating strong & reliable passwords because they are easier to remember.
- The most important thing is that passwords have a lot of weaknesses and are not sufficient to protect our online information. Hackers can easily guess user credentials and track their activity by running through all possible combinations until they find an exact match.

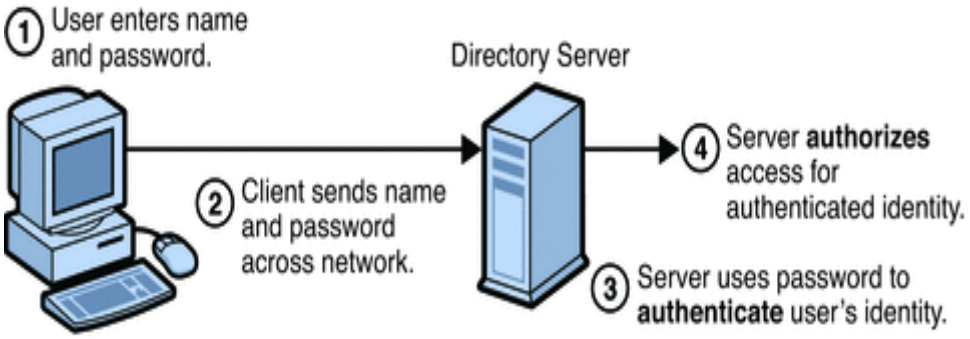


Fig. 2 Password based authentication.

2. Multi-factor authentication

- Multi-Factor Authentication (MFA) is an authentication method that requires two or more independent ways to identify or find a user. Examples include codes or OTP generated from the user's smartphone, Captcha tests, fingerprints, voice biometrics or facial recognition.

- Multi-Factor authentication methods and technologies that increase the confidence and trust of users by adding multiple layers of security. This may be a good defense against most account hacks, but it has its own pitfalls. People may lose their phones or SIM cards or due to some other problem and not be able to generate an authentication code.

Fig 3. Multi-factor authentication.

3. Certificate-based authentication

- Certificate-based authentication technologies which identify users, systems or devices by using digital certificates. A digital certificate is an electronic document based on the concept of a driver's license or a passport.

- The certificate consists of the digital identity of a user including a public key, and the digital signature of a certification authority. Digital certificates prove the holding of a public key and are issued only by a certification authority.

- Users will provide their digital certificates when they are going to sign in to a server system. The server verifies the credibility or authenticity of the digital signature and the certificate authority. The server then uses cryptography techniques to confirm that the user has a correct private key associated with the certificate.

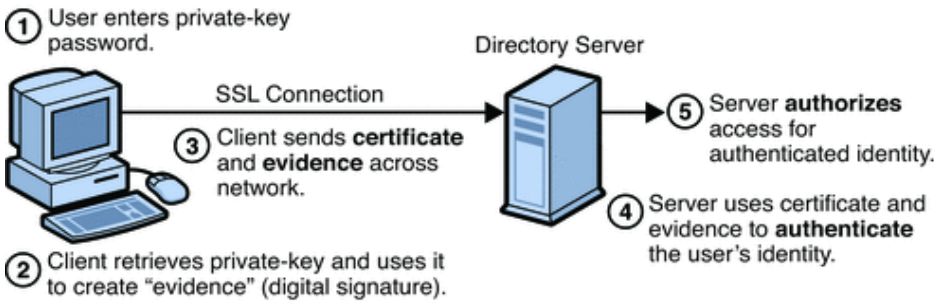


Fig. 4 Certificate-based authentication

4. Biometric authentication

- Biometrics authentication is a security check process that depends on the unique biological aspect of an individual.
- Advantages of using biometric authentication technologies:
    - Biological elements can be easily compared to authorized features stored in a database.
    - Biometric authentication controls physical access to a person when installed on gates and doors or entrances.
    - We can add biometrics features into the multi-factor authentication task.
    - Biometric authentication technologies are used by consumers, governments and private corporations including airports, military bases, and national borders to identify authorized persons.
    - The technology is increasingly acquired due to the ability to achieve a high level of security without creating resistance for the user.

Common biometric authentication methods include:

- **Facial recognition** which matches the different face characteristics of an individual trying to gain access to a valid face stored in a database. Face recognition can be at odds with when comparing faces at different angles or comparing people who look similar.
- **Fingerprint scanners which** match the distinctive patterns on an individual's personal fingerprints. Nowadays, new versions of fingerprint scanners can assess the vascular patterns in people's fingers. Fingerprint scanners are currently the most popular biometric technology for daily consumers, despite their frequent inaccuracies. These features added to iPhones.
- **Speaker Recognition, also** called as voice biometrics, examines a speaker's speech patterns for the formation of specific shapes and sound qualities. A voice-protected device usually depends on the standardized words to identify users, just like a password.

- **Eye scanners** include automation like iris recognition and retina scanners. During the Iris scan focusing a bright light towards the eye and searching for an exact match for unique patterns in the colored ring around the pupil of the eye. Then this pattern is compared to correct information stored in a database. Eye-based authentication may undergo inaccuracies if a person wears glasses or contact lenses.

Fig. 5 Biometric authentication methods

5. Token-based authentication

Token-based authentication technologies allow the users to enter their credentials once and receive a unique encrypted string of random characters in exchange. Then use the token to access protected systems in lieu of entering your credentials all over again. The digital token validates that you already have access permission.

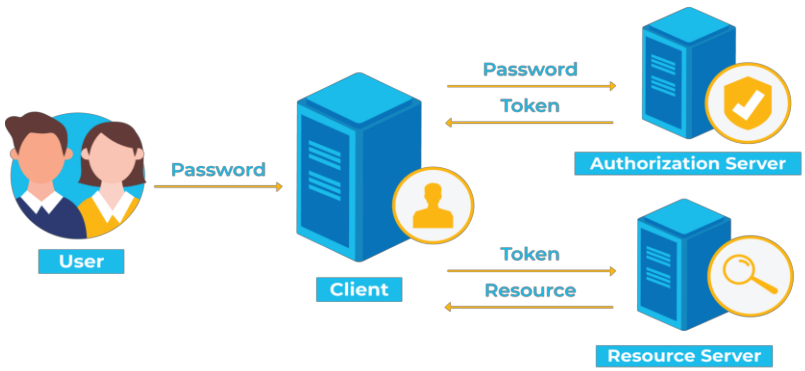


Fig. 6 Token-based authentication

4.3.4 BIOMETRIC AUTHENTICATION AND THIRD PARTY AUTHENTICATION USING KDC AND KERBEROS VERSION 5

Before study of Biometric Authentication and Third Party Authentication let first understand the concept of KDC & Kerberos V5

Key Distribution Center (KDC)

Key Distribution Center (KDC) is a central control to deal with keys for independent computers in a computer network. It is similar to the idea of the Authentication Server (AS) and Ticket Granting Server (TGS) in Kerberos. The core idea is that every node shares a unique secret key with the KDC.

Example:

When user A wants to communicate securely with user B, the following things is happens:

1. The background is that A has a shared secret key KA with KDC. Similarly, B is assumed to share a secret key KB with the KDC.
2. A sends a request to KDC encrypted with KA, which includes
   a. identities of A and B
   b. A random number R, called a nonce
3. KDC responds with a message encrypted with KA, containing
   a. One-time symmetric key KS
   b. Original request that was sent by A, for verification
   c. Plus, KS encrypted with KB and ID of A encrypted with KB.
4. A and B can now communicate by using KS for encryption

This steps shows below in a diagram

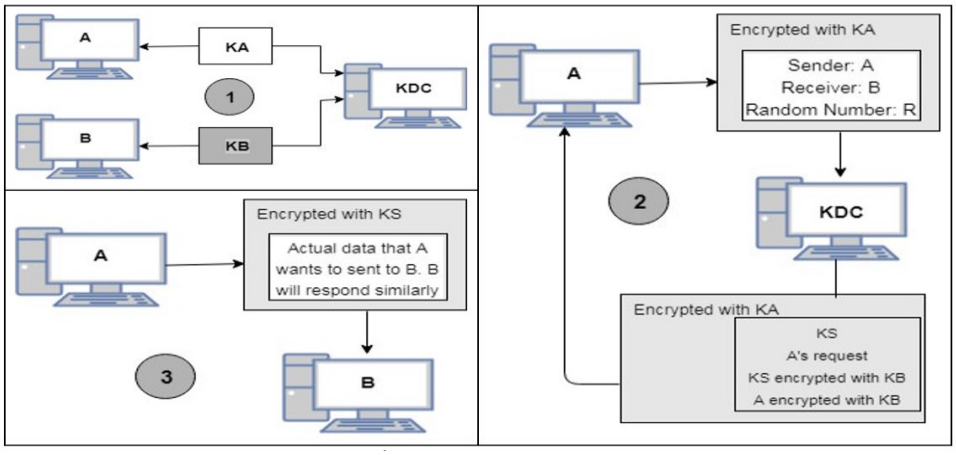


Fig. 7 KDC Concept

KDC is a single process that provides two services:

- Authentication Service (AS)

  This service issues ticket-granting tickets (TGTs) for connection to the ticket-granting service in its own domain or in any trusted domain.

- Ticket-Granting Service (TGS)

  This service issues tickets for connection to computers in its own domain.

Kerberos

Kerberos is designed for authentication in client server architecture. In this architecture, the server is not only dependent on the information given by the client but also verifies the same credential from its database. The components of kerberos are a server, clients , an authentication server and ticket granting server.

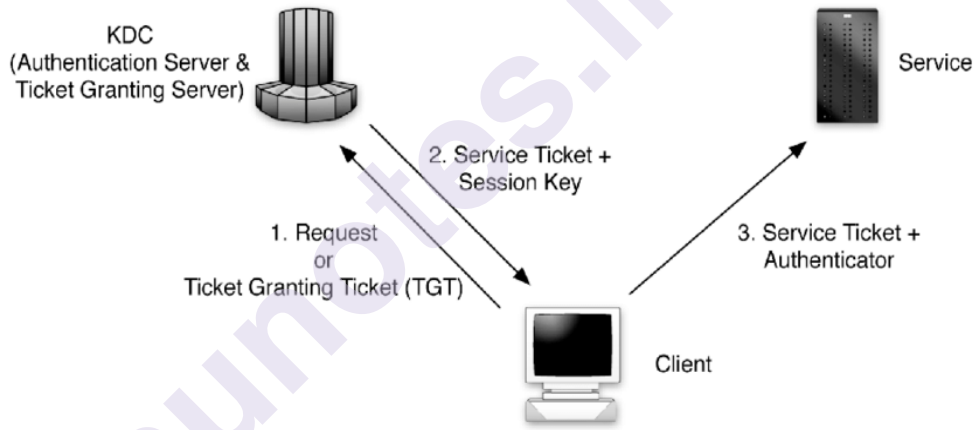


Fig. 8 Kerberos

Step 1. Client sends request for a ticket to TG service from AS

(AS contain the USer ID & Password)

Step 2. AS returns an encrypted ticket to the client.

Step 3. Client decrypts the ticket using his secret key.

Step 4. If a user wants to use the service from the server for this client must be allowed to communicate with the server.

Step 5. Client submits the ticket to TG server.

Step 6. TG server verifies the ticket for identifying the client and after successful verification provides a new ticket to the client.

Step 7. Client then submits this ticket to the server.

Step 8. Server checks the ticket and the authentication credential to ensure that it is an authenticated client or not.

Step 9. After verification, the server provides the service to the client.

Kerberos Ticket-granting System:

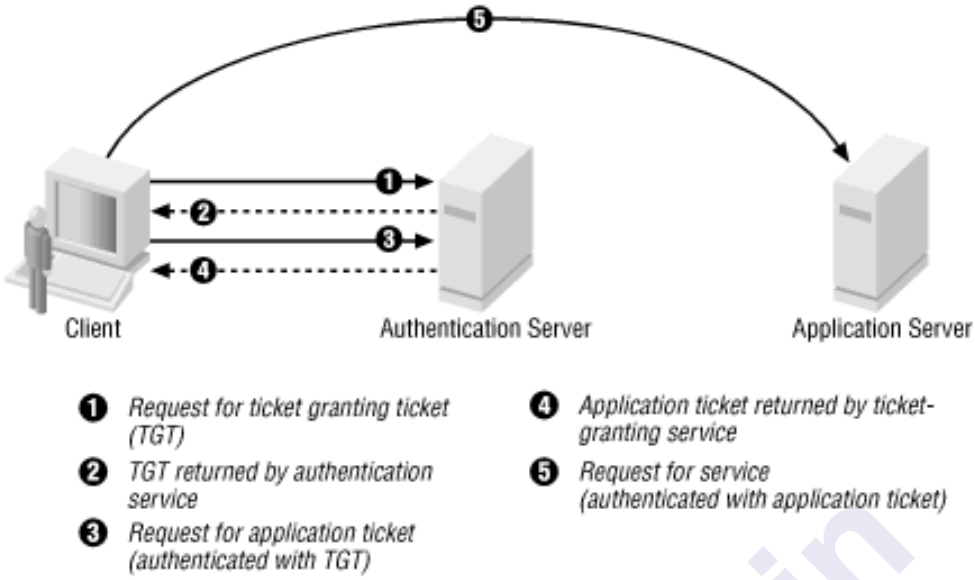


Fig. 9 Ticket Granting ticket

Ticket-Granting Server

- Ticket granting server is used to reduce the load on the authentication server. This solves the problem of password re-entry every time for a new service.
- TGS provides the ticket and session key so that users can enter their password only once.

Working of TGS:

- TGS receives a request, reads the ticket , and will validate it. If the ticket has been supplied by the AS, then the TGS has the AS secret key and can decrypt the ticket, otherwise it's potentially a forged ticket, and it will be discarded.
- The TGS then generates a ticket for the targeted service, and encrypts it using the service's secret key, then encapsulates this encrypted ticket into a response which will be itself encrypted using the client's secret key.
- The client will receive this response, will decrypt it and extract the encrypted ticket, and will send this encrypted ticket to the targeted service, which will be able to decrypt it and validate it.
- Of course, in the meantime, many checks will be done relative to the ticket validity, so one can be assured that the service is only accessible by those with the credential to do so.

Kerberos Version 5

- Version 5 of Kerberos overcomes some of the imperfections of Version 4. Version 4 requests the use of DES.

- Version 5 allows adaptability in terms of allowing the choice of other algorithms.
- Version 4 depends on IP addresses as identifiers. However, Version 5 allows the use of other types as well.

Following are key differences between Kerberos Versions 4 and 5.

Kerberos Version 4	Kerberos Version 5
DES encryptions techniques.	Any type of encryption can be employed because the encrypted text is tagged with an encryption type identifier.
"Receiver-makes-right" encoding system.	ASN.1 coding system.
For a ticket lifespan is 5 minutes, the ticket lifetime must be provided in units.	The ticket lifetime is defined as an arbitrary amount of time.
Ticket support is satisfactory	Ticket support is excellent and facilitates forwarding, renewing and postdating tickets.
Only a few IP addresses and other addresses for other sorts of network protocols are included.	Multiple IP addresses and other addresses for various network protocols are included.

It now supports forwardable, renewable, and postdatable tickets.

- Forwardable The user can use this ticket to request a new ticket, but with a different IP address Thus, a user can use his/her current credentials to get credentials valid on another machine.
- Renewable A renewable ticket can be renewed by asking the KDC for a new ticket with the extended lifetime. However, we cannot renew a ticket that has expired, we have to renew it before it expires. A renewable ticket can be renewed up until maximum renewable ticket lifetime.
- Postdatabale These are tickets which are initially invalid, and have a starting time some time in the future. To use a postdatable ticket, the user must send it back to the KDC to have it validated during the ticket's valid lifetime.

Kerberos Third-party Authentication model:

- In the Kerberos system, more than one trusted authentication server may be used (it is also called KDCs or key distribution servers). This provides third-party authentication services which are cooperative and have numerous applications.. Client obtained tickets from the trusted authentication server which can be used to provide the proof of identification for subsequent requests for service and application. This ticket is encrypted so it is secured while transmitting.
- The user wants some service so he first sends his request to an authenticated server. This request contains the user's name and the name of the service granting server that he will use.
- The user logs in to the client and requests for a TGT.
- After authentication using password and username, the initial authentication ticket is granted by the AS to the client.
- The ticket-granting service issues a ticket to the client.
- The client now submits the ticket to the particular server for the desired service.

The fig. 10 illustrates the interaction between different systems involved in the kerberos network.

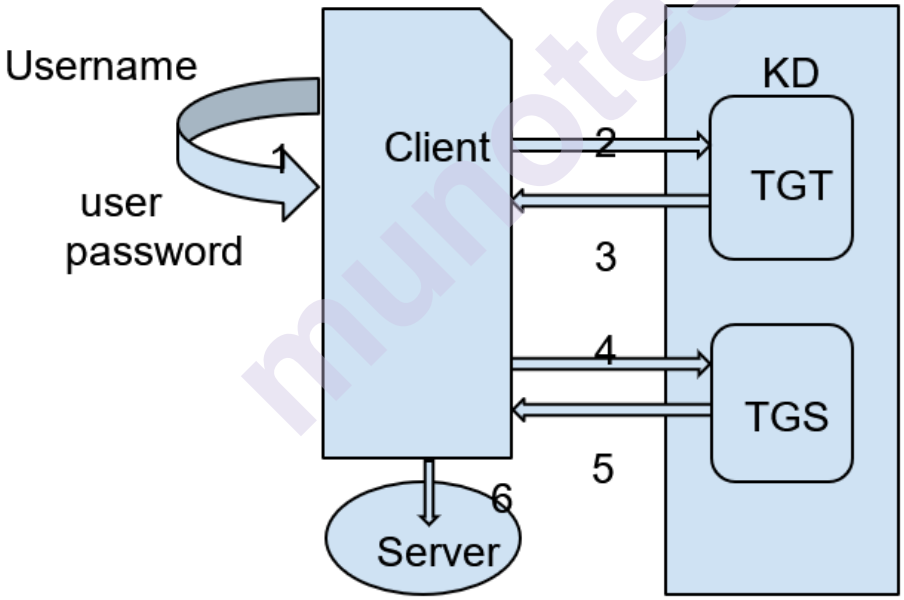


Fig. 10 Authentication Model

Kerberos Authentication Model:

- This uses a symmetric key encryption technique.
- Kerberos 4 uses DES algorithm and Kerberos 5 uses DES and IDEA algorithms.

- For more security, we can use double encryption techniques. For encryption two keys are used, ie., user password and the session key.
- The user password has a long life period and is used only for first time authentication whereas a session key has a period of 8 to 10 hours approximately and is used for requesting different services after first time authentication.
- The user first logs on the client system by using user id and password.
- Client sends the request for a ticket to the authentication server for the particular user by providing his user id to AS and not the actual password.
- The authentication server checks the user id and sends the encrypted ticket to the client.
- If the user is able to decrypt the ticket by his password then the user is considered as authenticated.
- Then the user sends the ticket to the service they want to use. If a service is able to decrypt a ticket using its own secret key, the service may presume that the user is authentic.
- In this way, without passing the password information over the insecure channel, the authentication takes place in Kerberos environment. So, it is difficult for the assailant to read the secret information about the user.

The authentication in Kerberos takes place in 6 steps as shown in Figure 11.

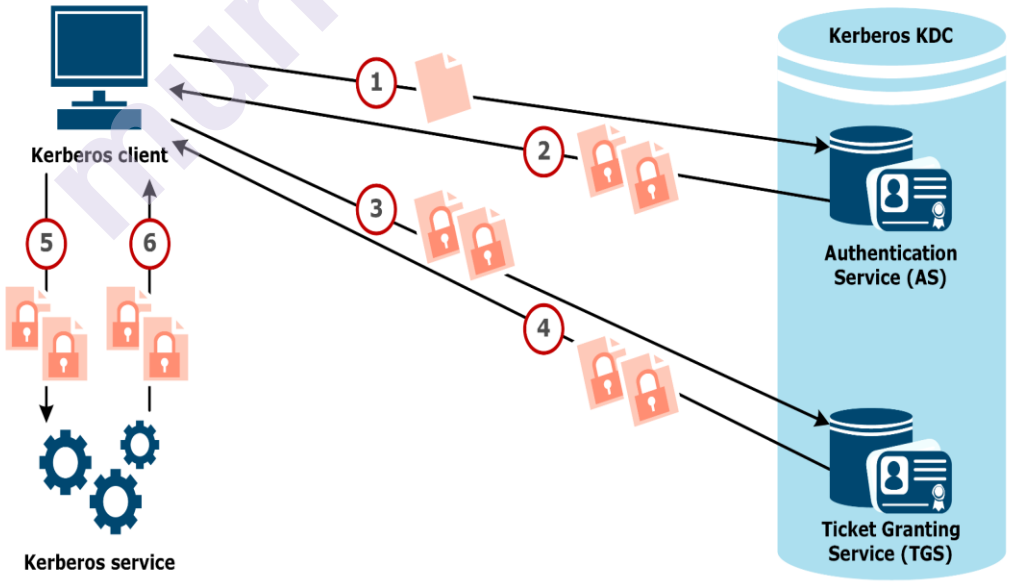


Fig. 11 Kerberos Authentication Model

1. The user first login on the client by using user is and password.

2. The client sends a request to the AS requesting a ticket for the user. Thus This request is totally unauthentIcated and it contains only user id and not the password of the user.

3. The ticketing service verifies tickets. the user's name in its database. If the user name is the database then he is an authenticated user and the ticketing service generates a unique session key for later use during the user's authenticated session. This ticket sends to the client a double-encrypted ticket-granting ticket and the session key în the form:

   Kuser(Ks, Ktgs {Ttgs, Ks}

   The client then decrypts the ticket-granting ticket using the user's password. If the client successfully decrypts the ticket using the user's password, then the user is authentic. Then the client stores the ticket TGT(Ktgs Ttgs, Ks}) for Later use.

4. Then the client sends a ticket request to ticket-granting service (TGS) for a particular service requested by the user. This request for ticket is in the form:

   TGT, Ks{request, client-IP, timestamp}}

   (where TGT = Ktgs{Ttgs, Ks})

5. The ticket-granting service decrypts the TGT using its own secret key (Ktgs) and the rest of the part of the message is decrypted by using the session key. If the ticket-granting service successfully decrypts the ticket, it gets the Following information:

   The TGT was issued by an authenticated ticketing service.

   The request for the service is from the authenticated user.

   Once the authentication is completed the TGS generates a session key and the ticket for a requested service. The TGS sends the session key and the ticket to the client machine in the form:

   Ks{Ksession, Kser{Tservice, Ksession)}

6. The client machine decrypts the service ticket using the session key (Ks) and yields the session key (Ksession) and an encrypted service ticket

   (Kser{T'service,Ksession}).

4.5 SUMMARY

- Authentication is one of the key aspects of cryptography. It can be used to guarantee that communication end-points, i.e., sender and receiver are the parties who they claim.
- The main objectives of authentication requirements are to ensure that the claimants are really what they claim to be, and avoid compromising security to an impostor.
- Authorisation is the mechanism through which a system determines what level of access to a particular authenticated user should have.
- It is used to secure the resources controlled by the system.
- The password-based authentication method refers to secret information which the user has to prove that he knows.
- In a two-factor authentication system, identification and authentication of the user take place in two different ways to establish his identity and privileges.
- Biometric authentication method uses thumb impression, iris or voices for authentication. Integrity of a message is checked by using the hash value or message digest calculated from the message.

4.6 REFERENCE FOR FURTHER READING

Atul Kahate, "Cryptography and Network Security", McGraw Hill Network Security and Cryptography: Bernard Menezes, CENGAGE Learning

4.7 UNIT END EXERCISES

1. What is authentication? What are the objectives of authentication?
2. What is the difference between authentication and authorisation?
3. List the different methods of authentication. Explain each in brief.
4. Explain the password-based authentication method.
5. What are the guidelines for choosing a password or setting up a password?
6. Discuss the weaknesses of the password-based authentication method.
7. Explain the two-factor authentication method.
8. Compare the password-based authentication with the two-factor authentication methods.
9. Discuss the weaknesses of the two-factor authentication method.
10. What is Kerberos? How does TGS work?

❄❄❄❄❄❄❄

5

AUTHENTICATION - II

Unit Structure

5.1 OBJECTIVE

a. Mutual authentication helps ensure that the data they receive is accurate and from a legitimate source, reducing the chances that an attacker has compromised their connections.

b. Authentication ensures that API requests come from a legitimate source.

c. To authenticate the identities of the client and server application to each other

5.2 INTRODUCTION

- Mutual authentication is when two or both sides of a communications channel verify each other's identity, instead of only one side verifying the other. Mutual authentication is also called a "two-way authentication process" because the process goes in both directions.

- Mutual authentication is an important application area of mutual authentication protocols. Such protocols enable communicating parties to fulfill themselves mutually about each other’s identity and to exchange session keys.

- The problem of authenticated key interchange are two issues: confidentiality and timeliness. To stop masquerade and to stop compromise of session keys, essential identification and session-key information must be communicated in the form of encryption. This requires the earlier survival of secret or public keys that can be used for this purpose.

- The second problem, timeliness, is essential because of the threat of message replays. These types of replays, at unpleasants, allow an enemy to compromise a session key or successfully impersonate

another party. At minimum, a successful replay can disrupt operations by presenting parties with messages that appear genuine but are not.

- Following are the examples of replay attacks:
  1. The simplest replay attack is one in which the enemy simply copies a message and replays it later.
  2. An enemy can replay a time stamped message within the valid time frame. If both the original and the replay arrive within the time frame, this type of incident can be logged.
  3. As with example 1, an enemy can replay a time stamped message within the valid time frame, but in addition, the enemy suppresses the original message. Thus, the repetition cannot be detected.
  4. Another attack includes a backward replay without modification. This is a reply back to the message sender. This attack is possible if symmetric encryption is used and the sender cannot easily realize the difference between messages sent and messages received on the basis of content.
- One approach to handle replay attacks is to add a sequence number to each message used during an authentication interchange. A new message is received only if its sequence number is in the proper (sequence) order. The problem with this process is that it requires each party to keep followup of the last sequence number for each claimant it has assigned with. Due to this overhead, sequence numbers are generally not used for authentication and key exchange. rather, one of the following two general approaches is used:
  1. Timestamps: Party A accepts a message as fresh only if the message consists of a timestamp that, in A's judgment, is close enough to A's knowledge of current time. This method requires that clocks among the various participants be synchronized.
  2. Challenge or response: Party A, expecting a fresh message from B, first sends B a nonce (challenge) and requires that the subsequent message (response) received from B contain the correct nonce value.

5.3 MUTUAL AUTHENTICATION APPROACHES

In mutual authentication, Party A and B both authenticate each other. Hence, we have the term mutual authentication here. This approach can also be implemented in different ways, namely shared secret, public keys, and time stamp-bascd. This is shown in Fig. 1.

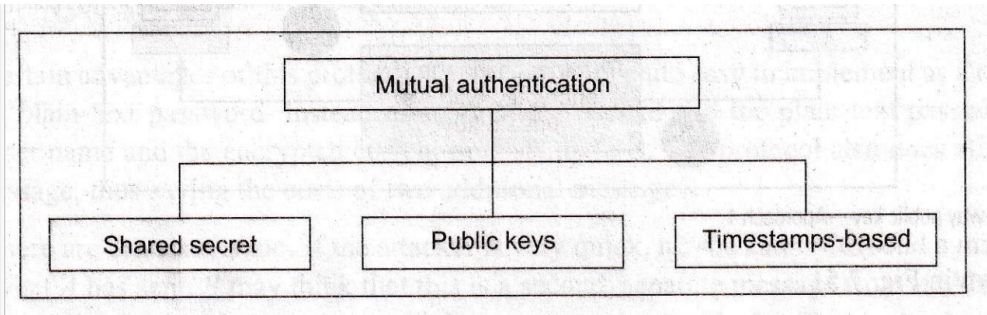


Fig. 1 Mutual authentication

1. Shared Secret

- This protocol, suppose that A and B have a shared symmetric key KAB. The protocol works as follows:
- Steps:

1. A sends her username to B.
2. B sends a random challenge R1 to A.
3. A encrypts R1 with KAB and sends it to B.
4. A sends a different random challenge R2 to B.
5. B encrypts R2 with KAB and sends it to A.

- Next, B authenticates A as before, shown in steps 2 and 3. Nevertheless, what is new is that A also authenticates B, shown in steps 4 and 5. Hence, it is mutual authentication. This is shown in Fig. 2.

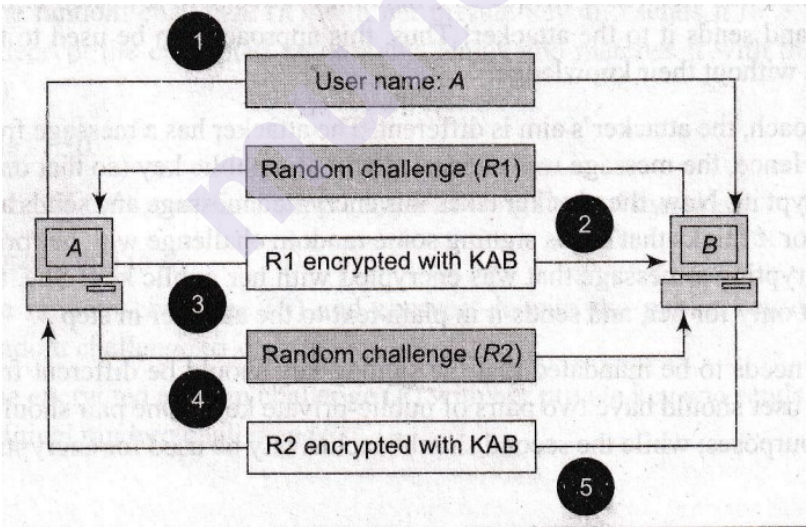


Fig. 2 Shared Secret

- Mutual authentication mainly based on a shared secret
- In This example, too many messages are exchanged, making this protocol ineffective. This will reduce this to just three messages, by putting more information in those three messages. This modified approach is described below:

1. A computer sends the user name and a random challenge (R2) to B computer.
2. B computer encrypts R2 with the shared symmetric key KAB, generates a new random challenge (R1); and sends these two to A computer.
3. A computer verifies R2, encrypts Rl with the shared symmetric key KAB; and sends it to B computer. B verifies R1. This process is shown in Fig. 3

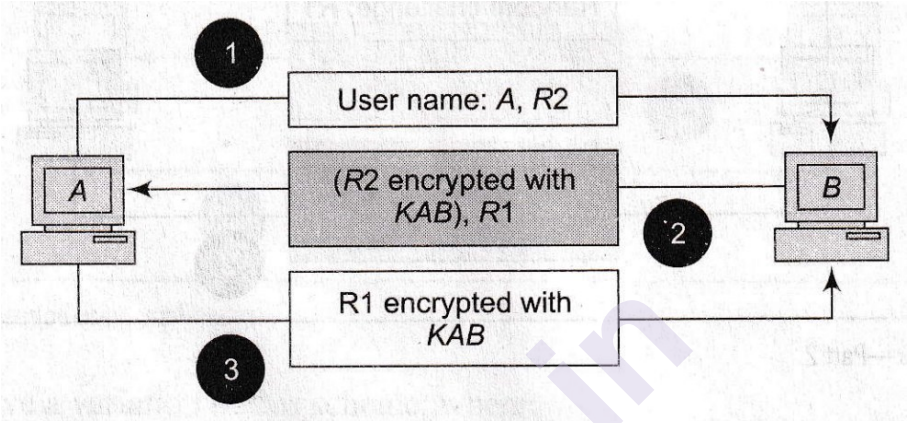


Fig. 3 Shared Secret Process

- This variety of the protocol reduces the number of messages to three. However, it suffers from a problem called reflection attack. Suppose that attacker C wants to pose as 4 to B computer. At the beginning, the attacker C starts the protocol as follows:

  1. C person sends a message to B person containing the user id of A and random challenge R2.
  2. B encrypts R2 with the shared symmetric key KAB, generates a new random challenge (R1); and sends these two to C. B person thinks he is sending these to A.

This is shown in Fig. 4

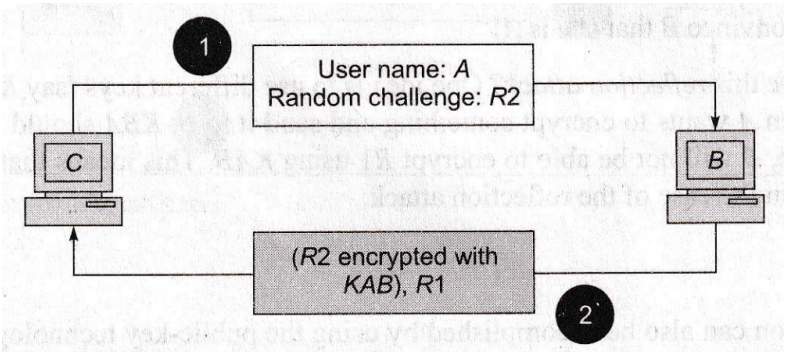


Fig. 4 Reflection attack-Part 1

- The attacker C cannot encrypt R1 with KAB. However, C has managed to have B encrypt R2.

- Next, the attacker C opens a second session with B, distinct from the first session, which is still active. Then the following things happen.

  1. sends a message to B containing the user id of A and random challenge R1.
  2. B encrypts R1 with the shared symmetric key KAB, generates a new random challenge, and sends these two to C. B thinks he is sending these to A.

This is shown in Fig. 5.

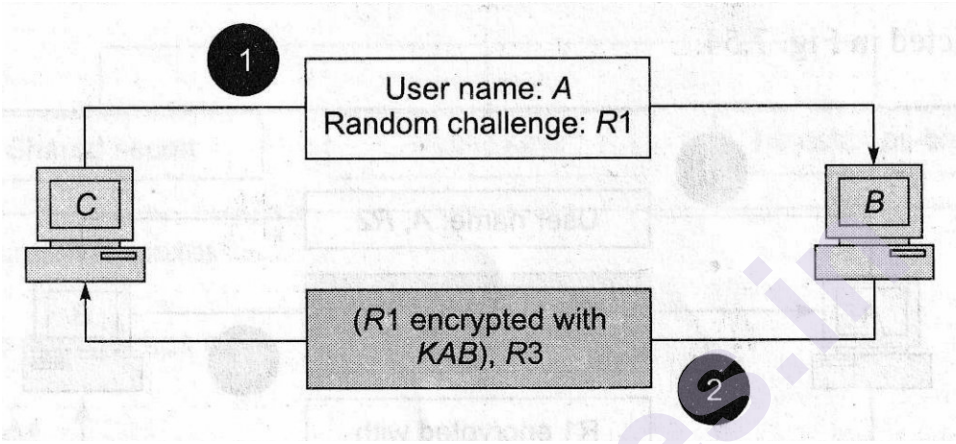


Fig. 5 Reflection attack-Part 2

- The attacker C cannot proceed with this second session, since C cannot encrypt the new random challenge R3. However, C need not proceed with this session. Instead, C can go back to her first session opened with B earlier. Remember this could not encrypt R1 with KAB in that session, and was hence waiting? Now Chas R11 is encrypted with KAB, thanks to this second session. C sends it to B and completes authentication! This is shown in Fig. 6.

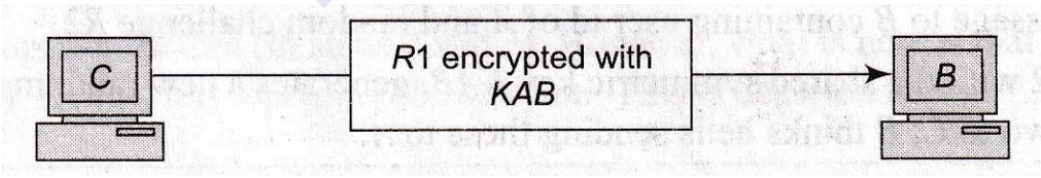


Fig. 6 Reflection attack-Part 3

- Thus, C is able to convince B that C is A! How can one resolve this reflection attack? One idea is to use different keys (say KAB andN should be used when A wants to encrypt something and send it to B. KBA should be usea se it direction. Therefore, B will not be able to encrypt Rl using KAB. This means that C cannot be misused later, as it happens in the case of the reflection attack.

2. Public Keys

Mutual authentication can also be achieved by using the public-key technology. If A and B recognize each other's public key, 3 messages are need to complete the mutual-authentication process as follows:

1. A sends her username and a random challenge (R2) encrypted with B's public Key.
2. B decrypts the random challenge (R2) with his private key. B make a new random challenge (R1) and encrypts it with A's public key. B sends these two objects (decrypted R2 and encrypted R1) to A.
3. A decrypts the random challenge (R1) with her private key and sends it to B. B verifies R1.
4. This process is shown in Fig. 7

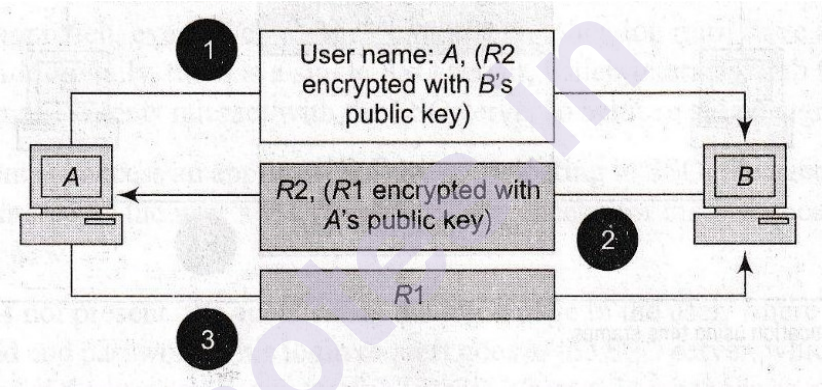


Fig. 7 Mutual authentication using public keys

As routinely, we can have a variation of this scheme, where:

1. A sends her username and R2 to B.
2. B encrypts R2 with his private key and sends it and RI to A.
3. A signs Rl and returns it back to A.

This is shown in Fig. 8.

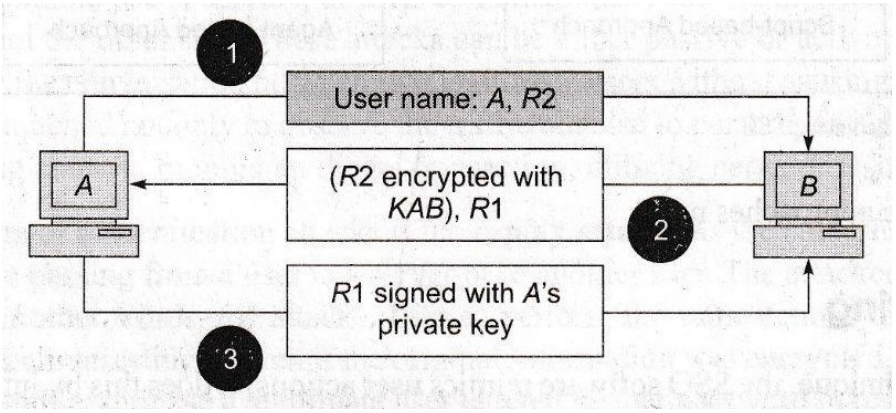


Fig. 8 Mutual authentication using public keys

3. Time Stamps

- This can reduce the mutual-authentication process to just two steps by using time stamps, instead of random numbers as challenges. This would work as follows:

  1. A sends her username and the current timestamp encrypted with a shared symmetric key (KAB) to B.
  2. B retrieves the time stamp by decrypting the above block using the KAB and adds one to the timestamp. B encrypts the result with KBA (not KAB!) er name, and sends it to A, along with his username

This approach is shown in Fig. 9.

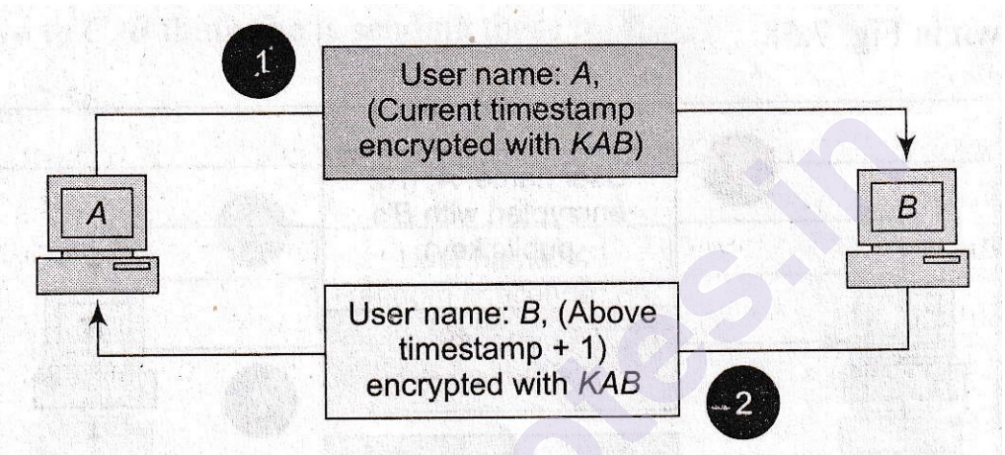


Fig. 9 Mutual authentication using time stamps

Working of mutual authentication:

The three main methods for mutually authenticating the ends of a communications channel:

1. Public key authentication:

- This method depends on public key encryption. Encryption is the process of climbing data by use of a key.
- Public key encryption uses two keys in lieu of one: a public key and a private key.
- Data encrypted with the public key is decrypted with the private key, and data encrypted with the private key is decrypted with the public key.
- In public key mutual authentication, both sides of the communication advertise a public key, and both have to prove they possess the private key that goes with their public key like someone showing a government-issued ID card to verify their name or authenticity. Each side sends a piece of data called a "signature" and encrypts that signature with their private key, and the other side decrypts the signature using the public key.

- If the signature can be decrypted properly with the public key, then the correct private key was used, and the party that sent the signature is appropriate.

2. Certificate authentication:

- This method is similar to public key authentication, except instead of just a public key, both parties have a public key certificate for authentication.
- The certificate consists of additional information that helps verify the parties name including who issued the certificate and public key, whom the certificate registers to, when the certificate expires, and so on.
- TLS certificates used for this type of mutual authentication mechanism if both sides have one.

3. Username and password:

- Despite the name, this method of mutual authentication nevertheless uses a certificate on the server side.
- The server hands over a certificate to the client, which verifies the certificate. On the other hand, it is just like typical username or password authentication, the client sends its username and password combination to the server, which verifies the credentials.

Uses of mutual authentication

- **One-way authentication** happens all the time on the Internet. Every time someone loads a website that uses HTTP, their machine authenticates the identity of the web server by checking the server's TLS certificate. Another example would be a person signing in to their account on an application in this instance, the application is authenticating the person.
- While mutual authentication eliminates some security defects and makes some types of attacks far more difficult to carry out, it adds **more time and computing power to the interchange of information**.
- It also **requires some advanced setup**. Both sides of the communication need
    - a set of credentials,
    - a public-private key pair, or a public key certificate.
- This makes mutual authentication difficult to implement for the average number of users, and this is why mutual authentication is not normally a part of TLS when someone is using a web application.
- The main use cases for mutual authentication include:

A. IoT:

1. Many IoT devices need to connect to a remote server in order to function properly.
2. They may need to connect to other IoT devices as well. IoT devices have to do so over an unsecured network (Internet).
3. Mutual authentication helps ensure that the data they receive is accurate and from a lawful source, reducing the chances that an attacker has compromised their connections.

B. API security:

1. Authentication ensures that API requests come from a legitimate source.
2. Mutual authentication is one way to make sure that an API is not accepting requests from attackers, and that an API user is not accepting spoofed API responses.

C. Zero Trust security:

1. Zero Trust is a belief that assumes any user or device could present a threat.
2. By requiring both sides of a connection to authenticate, mutual authentication ensures only legitimate users are connected to the network, server, or application. consequently, users can be assured they have connected to the true network, server, or application.

Attacks prevented by mutual authentication:

A. On-path attacks:

1. In this attack, an attacker resides in the middle of a connection between two parties.
2. The attacker heads off communications in both directions and impersonates the two ends of the conversation to each other.
3. Mutual authentication helps to stop this type of attack because the attacker will not be able to authenticate to both ends of the communication.

B. Spoofing and impersonation:

1. Attackers use these types of attacks to play a game with a server or a user into thinking they are a known and trusted party.
2. An attacker can spoof a web server to a user, or vice versa. Such attacks are a long way more difficult when both sides have to authenticate.

C. Credential theft:

1. In mutual authentication Some forms are password-based, and these could still be subject to credential theft (when an attacker steals a legitimate user's password).
2. Mutual authentication is usually public key based, credential theft is not feasible because there are no credentials to steal.

Protocols that support mutual authentication:

1. **Secure Shell Protocol (SSH):**
   a. SSH is a tunneling protocol for securely connecting to a remote server or device.
   b. SSH can use either public key authentication or certificate authentication.
   c. Different ways we can say that, it is possible to mutually authenticate in SSH with either a public key or with a public key certificate.
2. **Transport Layer Security(TLS):**
   a. TLS does not mutually authenticate both ends devices of a connection by default, it can be used for this purpose.
   b. Mutual TLS is one of the most commonly applied types of mutual authentication.
   c. In mutual TLS, both sides of a connection have a TLS certificate. Mutual TLS is commonly used for API security, IoT security, and Zero Trust security applications.

5.4 Reflection attack

- A reflection attack is an understanding of a server's security accomplished by tricking it into giving up a security code to allow a hacker to access it.
- Reflection attacks are made possible when servers use a simple protocol to authenticate guests. Adding some steps to increase security can make such attacks more difficult, pressurizing hackers to pursue other avenues of attack.
- Security professionals can assess a system to find if the security is enough for the application.

The example used to explain the concept of a reflection attack is the MIG in the middle.

- A military decides to implement a system that allows them to tell immediately if an aircraft on their radar is a good guy or a bad guy (referred to as Identify Friend from Foe (IFF) systems).

- The system they implement goes something like;

1. Alice > Bob: n <-- challenge
2. Bob > Alice: E(K, n) <-- response

Where n is a unique random nonce,

K is a pre-shared key, and

E is a suitable encryption/derivation function.

- Since only friendly aircraft know the value of the secret key K, only friendly aircraft can calculate the correct response to a given nonce n. If Bob fails to generate and send the correct response within a few seconds, Alice promptly shoots him down.
- Otherwise, Alice knows that Bob must have known the value of K in order to generate the response he sent, and that Bob must therefore be on the same team.
- This process can be repeated in reverse so that Alice and Bob are mutually authenticated.

The Reflection Attack

- An enemy aircraft can defeat the system by reflecting the challenge back to some other member of the good guys team.

1: Alice > Bob:	n	
2: Bob > Charlie:	n	<-- reflection attack
3: Charlie > Bob:	E(K, n)	<-- bob receives the correct response to Alice's challenge
4: Bob > Alice:	E(K, n)	<-- and authenticates himself to Alice

- At the 4th communication, Bob has authenticated himself to Alice who lets him fly past peacefully. Bob did this without knowledge of K, simply by sending the same challenge to someone other, hence the name reflection. Bob reflects the challenge to Charlie, who is on Alice's team and therefore knows the value of K and can generate the correct response. Charlie answers Bob's IFF challenge as he normally would, giving Bob the correct response to the original challenge.

5.5 SUMMARY

- Mutual authentication involves authentication of both parties.
- Mutual authentication is more flexible.
- Mutual authentication can be classified into shared secret , public keys and time-stamp-based.
- Reflection attack involves an attacker opening two sessions to impersonate a user.

5.6 REFERENCE FOR FURTHER READING

Atul Kahate, "Cryptography and Network Security", McGraw Hill
Network Security and Cryptography: Bernard Menezes, CENGAGE Learning

5.7 UNIT END EXERCISES

1. What is mutual authentication? What are the objectives of mutual authentication?
2. Explain the different approaches of mutual authentication?
3. Write a short note on Reflection attack?
4. What are the uses of mutual authentication?

❄❄❄❄❄❄❄

6

DIGITAL SIGNATURE

Unit Structure

6.1 CONCEPT

A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software or digital document. It's the digital equivalent of a handwritten signature or stamped seal, but it offers far more inherent security. A digital signature is intended to solve the problem of tampering and impersonation in digital communications.

Digital signatures can provide evidence of origin, identity and status of electronic documents, transactions or digital messages. Signers can also use them to acknowledge informed consent. In many countries, including the United States, digital signatures are considered legally binding in the same way as traditional handwritten document signatures.

6.1.1 How do digital signatures work?

Digital signatures are based on public key cryptography, also known as *asymmetric cryptography*. Using a public key algorithm, such as RSA (Rivest-Shamir-Adleman), two keys are generated, creating a mathematically linked pair of keys, one private and one public.

Digital signatures work through public key cryptography's two mutually authenticating cryptographic keys. The individual who creates the digital signature uses a private key to encrypt signature-related data, while the only way to decrypt that data is with the signer's public key. If the recipient can't open the document with the signer's public key, that's a sign there's a problem with the document or the signature. This is how digital signatures are authenticated. Digital signature technology requires all parties trust that the individual creating the signature has kept the private key secret. If someone else has access to the private signing key, that party could create fraudulent digital signatures in the name of the private key holder.

6.1.2 What are the benefits of digital signatures?

Security is the main benefit of digital signatures. Security capabilities embedded in digital signatures ensure a document is not altered and signatures are legitimate. Security features and methods used in digital signatures include the following:

1. **Personal identification numbers (PINs), passwords and codes.** Used to authenticate and verify a signer's identity and approve their signature. Email, username and password are the most common methods used.

2. **Asymmetric cryptography.** Employs a public key algorithm that includes private and public key encryption and authentication.

3. **Checksum.** A long string of letters and numbers that represents the sum of the correct digits in a piece of digital data, against which comparisons can be made to detect errors or changes. A checksum acts as a data fingerprint.

4. **Cyclic redundancy check (CRC).** An error-detecting code and verification feature used in digital networks and storage devices to detect changes to raw data.

5. **Certificate authority (CA) validation.** CAs issue digital signatures and act as trusted third parties by accepting, authenticating, issuing and maintaining digital certificates. The use of CAs helps avoid the creation of fake digital certificates.

6. **Trust service provider (TSP) validation.** A TSP is a person or legal entity that performs validation of a digital signature on a company's behalf and offers signature validation reports.

7. Other benefits to using digital signatures include the following:

8. **Timestamping.** By providing the data and time of a digital signature, timestamping is useful when timing is critical, such as for stock trades, lottery ticket issuance and legal proceedings.

9. **Globally accepted and legally compliant.** The public key infrastructure (PKI) standard ensures vendor-generated keys are made and stored securely. Because of the international standard, a growing number of countries are accepting digital signatures as legally binding.

10. **Time savings.** Digital signatures simplify the time-consuming processes of physical document signing, storage and exchange, enabling businesses to quickly access and sign documents.

11. **Cost savings.** Organizations can go paperless and save money previously spent on the physical resources and on the time, personnel and office space used to manage and transport them.

12. **Positive environmental impact.** Reducing paper use also cuts down on the physical waste generated by paper and the negative environmental impact of transporting paper documents.

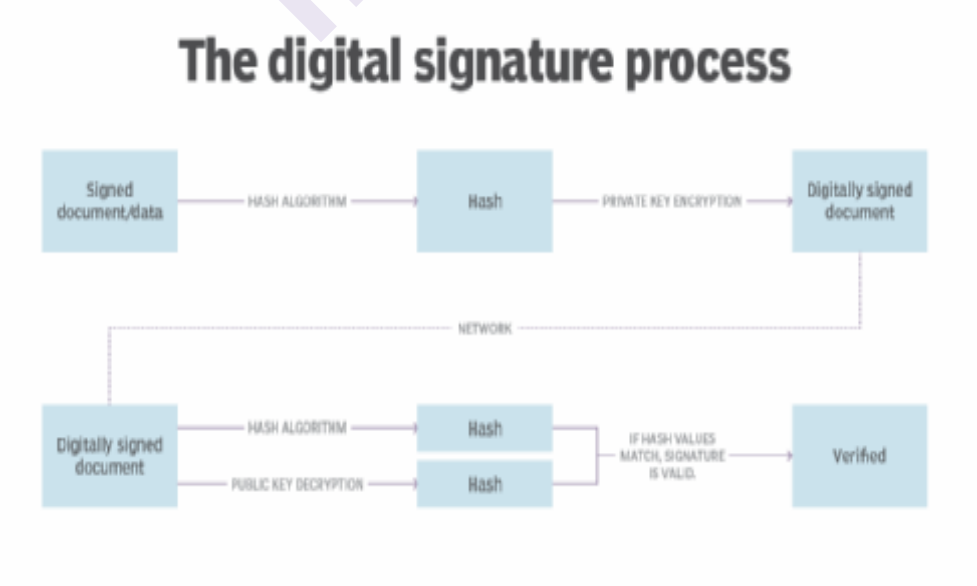
13. **Traceability.** Digital signatures create an audit trail that makes internal record-keeping easier for business. With everything recorded and stored digitally, there are fewer opportunities for a manual signee or record-keeper to make a mistake or misplace something.

How do you create a digital signature?

To create a digital signature, signing software, such as an email program, is used to provide a one-way hash of the electronic data to be signed. A hash is a fixed-length string of letters and numbers generated by an algorithm. The digital signature creator's private key is then used to encrypt the hash. The encrypted hash -- along with other information, such as the hashing algorithm -- is the digital signature. The reason for encrypting the hash instead of the entire message or document is a hash function can convert an arbitrary input into a fixed-length value, which is usually much shorter. This saves time as hashing is much faster than signing.

The value of a hash is unique to the hashed data. Any change in the data, even a change in a single character, will result in a different value. This attribute enables others to use the signer's public key to decrypt the hash to validate the integrity of the data.

If the decrypted hash matches a second computed hash of the same data, it proves that the data hasn't changed since it was signed. If the two hashes don't match, the data has either been tampered with in some way and is compromised or the signature was created with a private key that doesn't correspond to the public key presented by the signer -- an issue with authentication.



A person creates a digital signature using a private key to encrypt the signature. At the same time, hash data is created and encrypted. The recipient uses the signer's public key to decrypt the signature.

A digital signature can be used with any kind of message, whether it is encrypted or not, simply so the receiver can be sure of the sender's identity and the message arrived intact. Digital signatures make it difficult for the signer to deny having signed something as the digital signature is unique to both the document and the signer and it binds them together. This property is called *nonrepudiation*.

Digital signatures are not to be confused with digital certificates. A digital certificate is an electronic document that contains the digital signature of the issuing CA. It binds together a public key with an identity and can be used to verify that a public key belongs to a particular person or entity.

Most modern email programs support the use of digital signatures and digital certificates, making it easy to sign any outgoing emails and validate digitally signed incoming messages. Digital signatures are also used extensively to provide proof of authenticity, data integrity and nonrepudiation of communications and transactions conducted over the internet.

6.1.3 Uses for digital signatures

Industries use digital signature technology to streamline processes and improve document integrity. Industries that use digital signatures include the following:

- **Government.** The U.S. Government Publishing Office (GPO) publishes electronic versions of budgets, public and private laws, and congressional bills with digital signatures. Digital signatures are used by governments worldwide for a variety of reasons, including processing tax returns, verifying business-to-government (B2G) transactions, ratifying laws and managing contracts. Most government entities must adhere to strict laws, regulations and standards when using digital signatures. Many governments and corporations also use smart cards to ID their citizens and employees. These are physical cards endowed with a digital signature that can be used to give the cardholder access to an institution's systems or physical buildings.

- **Healthcare.** Digital signatures are used in the healthcare industry to improve the efficiency of treatment and administrative processes, to strengthen data security, for e-prescribing and hospital admissions. The use of digital signatures in healthcare must comply with the Health Insurance Portability and Accountability Act (HIPAA) of 1996.

- **Manufacturing.** Manufacturing companies use digital signatures to speed up processes, including product design, quality assurance (QA), manufacturing enhancements, marketing and sales. The use of

digital signatures in manufacturing is governed by the International Organization for Standardization (ISO) and the National Institute of Standards and Technology (NIST) Digital Manufacturing Certificate (DMC).

- **Financial services.** The U.S. financial sector uses digital signatures for contracts, paperless banking, loan processing, insurance documentation, mortgages and more. This heavily regulated sector uses digital signatures with careful attention to the regulations and guidance put forth by the Electronic Signatures in Global and National Commerce Act (E-Sign Act), state Uniform Electronic Transactions Act (UETA) regulations, the Consumer Financial Protection Bureau (CFPB) and the Federal Financial Institutions Examination Council (FFIEC).

- **Cryptocurrencies.** Digital signatures are also used in bitcoin and other cryptocurrencies to authenticate the blockchain. They are also used to manage transaction data associated with cryptocurrency and as a way for users to show ownership of currency or their participation in a transaction.

6.1.4 Digital signature tools and vendors

Digital signature tools and services are commonly used in contract-heavy industries. For example, when freelance writers sign a contract, they can agree to word count and payment, using Adobe Sign to put their e-signature on the document.

Digital and e-signature service providers include the following:

- **Adobe Sign** is designed to provide secure, legal e-signatures across all device types.

- **DocuSign standards-based services** ensure e-signatures are compliant with existing regulations. Its services include Express Signature for basic global transactions and the EU Qualified Signature, which complies with EU standards.

- **Global Sign** provides a host of management, integration and automation tools to implement PKI across enterprise environments.

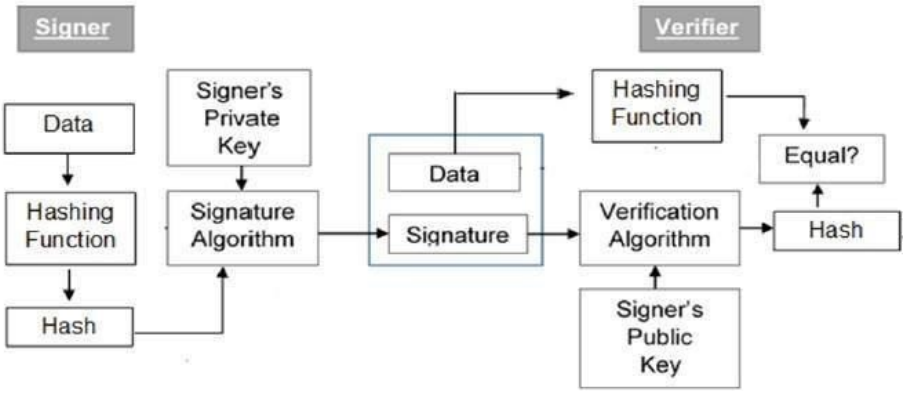
- **Sign Easy** offers an e-signing service of the same name to businesses and individuals, as well as providing application programming interfaces (APIs) for developers.

- **Sign Now**, which is part of the airSlate business cloud, provides an easy-to-use Portable Document Format (PDF) signing tool for businesses.

- **Vasco** provides its eSignLive e-signature product as a cloud service and on premises.

6.1.5 Model of Digital Signature

The digital signature scheme is based on public key cryptography. The model of digital signature scheme is depicted in the following illustration –



The following points explain the entire process in detail −

- Each person adopting this scheme has a public-private key pair.
- Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key and the public key as the verification key.
- Signer feeds data to the hash function and generates hash of data.
- Hash value and signature key are then fed to the signature algorithm which produces the digital signature on given hash. Signature is appended to the data and then both are sent to the verifier.
- Verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as output.
- Verifier also runs same hash function on received data to generate hash value.
- For verification, this hash value and output of verification algorithm are compared. Based on the comparison result, verifier decides whether the digital signature is valid.
- Since digital signature is created by 'private' key of signer and no one else can have this key; the signer cannot repudiate signing the data in future.

It should be noticed that instead of signing data directly by signing algorithm, usually a hash of data is created. Since the hash of data is a unique representation of data, it is sufficient to sign the hash in place of data. The most important reason of using hash instead of data directly for signing is efficiency of the scheme.

Signing large data through modular exponentiation is computationally expensive and time consuming. The hash of the data is a relatively small digest of the data, hence **signing a hash is more efficient than signing the entire data**.

Importance of Digital Signature

Out of all cryptographic primitives, the digital signature using public key cryptography is considered as very important and useful tool to achieve information security.

Apart from ability to provide non-repudiation of message, the digital signature also provides message authentication and data integrity. Let us briefly see how this is achieved by the digital signature −

- **Message authentication** − When the verifier validates the digital signature using public key of a sender, he is assured that signature has been created only by sender who possess the corresponding secret private key and no one else.
- **Data Integrity** − In case an attacker has access to the data and modifies it, the digital signature verification at receiver end fails. The hash of modified data and the output provided by the verification algorithm will not match. Hence, receiver can safely deny the message assuming that data integrity has been breached.
- **Non-repudiation** − Since it is assumed that only the signer has the knowledge of the signature key, he can only create unique signature on a given data. Thus the receiver can present data and the digital signature to a third party as evidence if any dispute arises in the future.

By adding public-key encryption to digital signature scheme, we can create a cryptosystem that can provide the four essential elements of security namely − Privacy, Authentication, Integrity, and Non-repudiation.

Encryption with Digital Signature

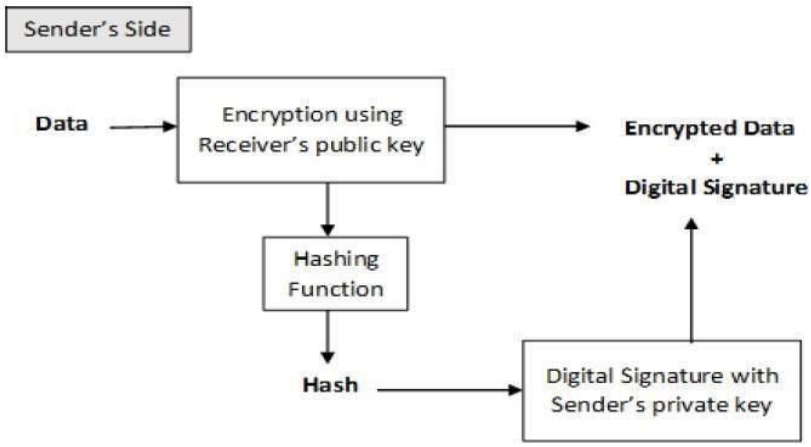
In many digital communications, it is desirable to exchange an encrypted messages than plaintext to achieve confidentiality. In public key encryption scheme, a public (encryption) key of sender is available in open domain, and hence anyone can spoof his identity and send any encrypted message to the receiver.

This makes it essential for users employing PKC for encryption to seek digital signatures along with encrypted data to be assured of message authentication and non-repudiation.

This can archive by combining digital signatures with encryption scheme. Let us briefly discuss how to achieve this requirement. There are **two possibilities, sign-then-encrypt** and **encrypt-then-sign**.

However, the crypto system based on sign-then-encrypt can be exploited by receiver to spoof identity of sender and sent that data to third party.

Hence, this method is not preferred. The process of encrypt-then-sign is more reliable and widely adopted. This is depicted in the following illustration −



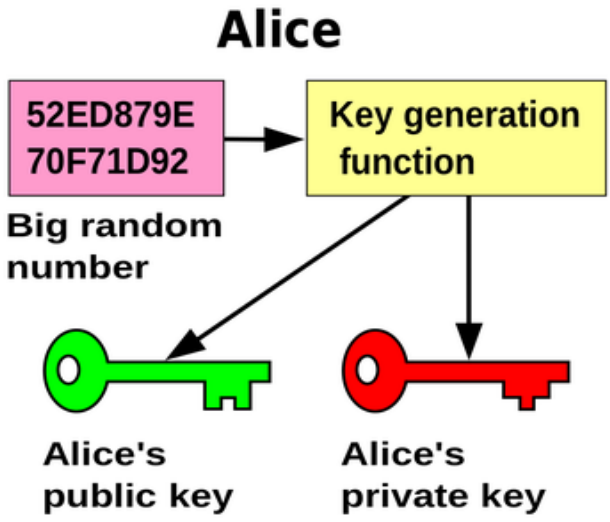
The receiver after receiving the encrypted data and signature on it, first verifies the signature using sender's public key. After ensuring the validity of the signature, he then retrieves the data through decryption using his private key.

6.2 COMPARE DIGITAL SIGNATURE WITH PUBLIC KEY CRYPTOGRAPHY

Public key cryptography: Public key cryptography is a cryptographic system that uses private/public keys. The advantage of this approach is in not requiring some sort of secure channel for the initial exchange of secret keys between communicators. This makes secure communication with strangers on open networks possible.

The **private key** is a random hexadecimal number that must be kept private by the account holder.

A **public key** is another hexadecimal number that can be shared publicly.



In most common encryption systems, the public and private keys are both generated at the same time. In others, the public key is generated from the private key. The public and private keys are associated with each other through a mathematical relationship. **However, there is no way use the public key to figure out the private key.** That is because these systems are based on math problems with no efficient solutions which can take outputs and work backwards from to get the original inputs.

RSA and Prime Numbers: One example of a hard math problem providing security for an encryption system is found in the popular RSA cryptography system. RSA uses prime numbers to ensure security. **A public-key cryptographic system needs a set of algorithms that is easy to do in one direction, but difficult to undo.** RSA uses an easy algorithm that multiplies two prime numbers.

If multiplication is easy, then the difficult part is factoring the product of the multiplying those two primes.

A **prime number** is a natural number (aka whole number used in counting) greater than 1 and can only be divided by 1 and itself. Examples of prime numbers are 3, 5, 7, 11, 13, 17, etc.

A **product** is a result of multiplying two factors. Ex: A * B = C. The factors are A and B. The product is C.

A **composite number** is a positive integer that is formed when multiplying two or more other positive integers. Thus, composite numbers are divisible by more than 1 and itself. When two primes (or any number of positive integers) are multiplied, we get a composite number.

Factoring out a number just means finding the numbers that make up the composite number.

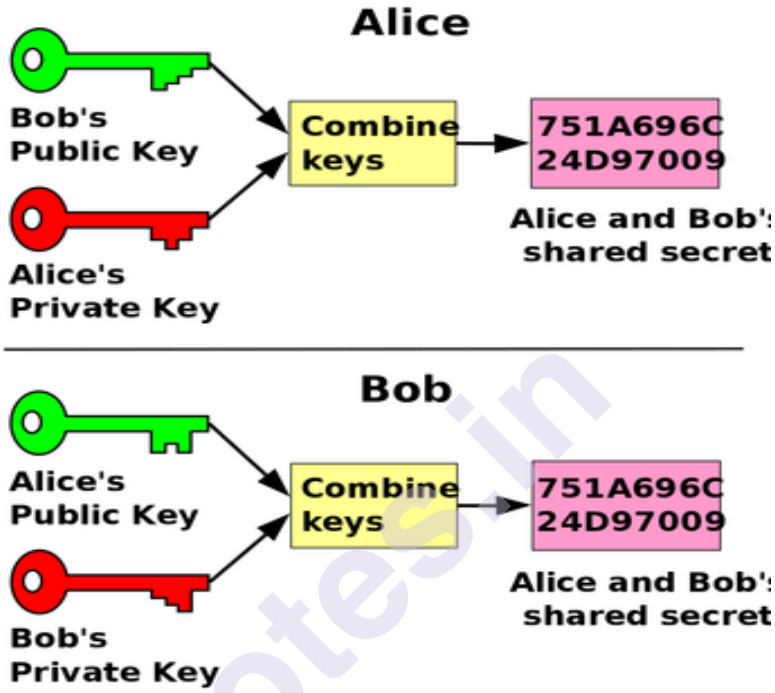
Factoring out the two prime numbers that makeup RSA's 232 digits length number will take a very long time. However, generating and checking those two primes is relatively easy.

Algorithms that have this property, easy in one direction and hard in the other, are known as trapdoor functions. Other algorithms use other types of hard math problems with this one-way property to provide security to their system. Ethereum uses something called Elliptical Curve Cryptography which will be described in a future post.

Public key cryptography with digital signatures: A digital signature with public-key cryptography securing a message is created in the following way. First, the message is digitally signed like explained above. Then, this bundle is encrypted with the sender's private key, and again with the receiver's public key

Looking at it like hypothetical function calls, it may look something like this:

public_key_of_recipient(private_key(message_hashing(message) + message + type of hashing algorithm))



Keep this a secret

Decrypting secured messages with digital signatures: When the recipient receives the information, the recipient can decrypt the outer layer with his private key. This ensures that only the receiver can read the information. Then, the receiver can decrypt the inner layer with the sender's public key. This assures that the sender was indeed the expected person. This is possible because the private and public keys are linked mathematically.

After decryption, the receiver can verify the message was not tampered with en-route by running the message through the same hashing algorithm as the sender. If they match, we have a valid message.

6.3 DIGITAL SIGNATURE SCHEME

Digital signature schemes are techniques to assure an entity's acknowledgment of having seen a certain digital message. Typically, an entity has a private key and a corresponding public key that is tied to the entity's name (Public Key Infrastructure). The entity generates a string called *signature*, which depends on the message to sign and his private key.

The fact that the entity acknowledged, that is, that he signed the message, can be verified by anyone using the entity's public key, the message, and the signature. *Data* Authentication and signature schemes are sometimes distinguished in the sense that in the latter, verification can be done by anyone at any time after the generation of the signature. Due to this property, the digital signature scheme achieves Non-Repudiation property, that is, a signer cannot later deny the fact of signing.

The conventional handwritten signature on a document is used to certify that the signer is responsible for the content of the document. The signature is physically a part of the document and while forgery is certainly possible, it is difficult to do so convincingly. Trying to mimic a handwritten signature in a digital medium leads to a difficulty since cut and paste operations can be used to create a perfect forgery. Thus, we need to have a way of signing messages digitally which is functionally equivalent to a physical signature, but which is at least as resistant to forgery as its physical counterpart.

Schemes which provide this functionality are called Digital Signature Schemes. A Digital Signature Scheme will have two components, a private signing algorithm which permits a user to securely sign a message and a public verification algorithm which permits anyone to verify that the signature is authentic. The signing algorithm needs to "bind" a signature to a message in such a way that the signature cannot be pulled out and used to sign another document, or have the original message modified and the signature remain valid. For practical reasons it would be necessary for both algorithms to be relatively fast and if small computers such as smart cards are to be used, the algorithms cannot be too computationally complex. There are many Digital Signature Schemes which meet these conditions, but we shall only investigate a few of the most popular ones.

RSA Signatures

As we have previously noted, in order for Bob to sign a message m, he raises m to his private decryption exponent mod n. This is the signature algorithm. Anyone can verify this signature by raising md to Bob's public encryption exponent mod n. This is the verification algorithm. Application of the verification algorithm to a valid signature yields the message m. The verifier must know the message m in order to be sure that this is the message that Bob signed, so in this application Bob must send the ordered pair (m, md mod n). Some care must be taken in the construction of the message to be signed in this way. For instance, if m is the instruction to Bob's bank to issue a check to Alice, then if Alice intercepts the ordered pair, she can send the same pair to Bob's bank whenever she is a little low on cash. To prevent this kind of abuse, when it matters, messages should

include dates and other such items which prevent the message from being reused.

Forgeries of Bob's signature are easy to construct. The requirement for a valid signature is that raising the second coordinate to Bob's public encryption exponent e gives the first coordinate. Frank the "forger" can take any number y, calculate x = y e mod n and send the pair (x,y). This will be verified as Bob signing the message x. Frank's problem is that he has no control over the "message" x, which will normally be just random nonsense. Without breaking the RSA crypto system, Frank has only a negligible chance of finding a meaningful message, let alone a desired message.

El-Gamal Signature Scheme

Unlike the RSA Signature scheme, which can be used as both a cryptosystem and a signature scheme, this signature scheme is designed specifically for signatures and is based on the discrete logarithm problem. As with the El-Gamal cryptosystem, computations are carried out in Z p , where p is a prime such that the discrete log problem is intractable in Z p . A generator α of Z p * is fixed, and each user selects a secret exponent a, and publishes the value β = α a mod p. If Alice wishes to sign a message m, she will first select a random secret integer k with gcd(k,p-1) = 1. She then computes r = α k mod p and then computes s = k -1 (m -ar) mod (p-1). The signature is the triple (m,r,s).

The verification algorithm compares β r r s mod p and α m mod p. Noting that from the definition of s, we have m = sk + ar mod (p-1), we see that: α m =α sk+ar = (α k) s (α a) r = r sβ r mod p.

Now suppose that Frank wants to forge Alice's signature on a message m without knowing Alice's secret exponent a. He can pick r randomly (just as Alice does) and then has to find an s so that β r r s = α m mod p. Rewritten, this amounts to solving r s = β -rα m mod p. Which is the discrete logarithm problem. On the other hand, if he first selects a random s, then he must solve the congruence β r r s = α m mod p for r. This is a problem for which no feasible solution is known and it does not seem to be related to any well studied problem such as the Discrete Log problem. There remains the possibility that Frank can choose r and s simultaneously to get a valid signature. No one has discovered a way to do this, but then again, no one has proved that it can't be done.

Unlike the RSA signature scheme, Frank cannot forge Alice's signature on "random messages" by randomly picking r and s and calculating a message m so that (m,r,s) is a valid Alice signature [to do this would require solving the discrete log problem]. However, Frank can create valid Alice signatures by selecting r,s and m simultaneously. To do this, Frank picks two integers, i and j (less

than p-1) such that gcd(j,p-1) = 1. Then Frank calculates:

$r = \alpha^i \beta^j \bmod p$

$s = -rj^{-1} \bmod (p-1)$

$m = is \bmod (p-1)$

Checking the verification algorithm, we see that:

$$\beta^r r^s = \beta^r(\alpha^i\beta^j)^s \bmod p$$
$$= \beta^r(\alpha^{is}\beta^{js}) \bmod p$$
$$= \beta^r(\alpha^{is}\beta^{-r}) \bmod p$$
$$= \alpha^m \bmod p.$$

There are some protocol failures that would compromise the El-Gamal signature scheme. The first involves the secret exponent k. Should this become known, then given a signature (m,r,s) the congruence ar = m-ks mod (p-1), has d = gcd(r,p-1) possible solutions for a. The correct one can be found by verifying that $\beta = \alpha$ a mod p. This gives Alice's secret exponent a and so breaks the system.

❄❄❄❄❄❄❄

7

PUBLIC KEY INFRASTRUCTURE

Unit Structure

7.1 PUBLIC KEY INFRASTRUCTURE

Cryptography lies at the heart of the modern business - protecting electronic communications and financial transactions, maintaining the privacy of sensitive data and enabling secure authentication and authorization. New regulations like GDPR and PSD2, the commercial pressure for digital transformation, the adoption of cloud technology and the latest trends in IoT and blockchain/DLT all help drive the need to embed cryptography into virtually every application – from toasters to core banking systems!

The good news is that modern cryptographic algorithms, when implemented correctly, are highly-resistant to attack – their only weak point is their keys. However, if a key is compromised, then it's game over! This makes such cryptographic keys one of your company's most precious assets, and they should be treated as such. The value of any key is equivalent to the value of all the data and/or assets it is used to protect.

There are three primary types of keys that need to be kept safe and secure:

1. **Symmetric keys** – typically used to encrypt bulk data with symmetric algorithms like 3DES or AES; anyone with the secret key can decrypt the data

2. **Private keys** – the secret half of public/private key pairs used in public-key cryptography with asymmetric algorithms like RSA or ECDSA; anyone with the private key can impersonate the owner of the private key to decrypt private data, gain unauthorized access to systems or generate a fraudulent digital signature that appears authentic

3. **Hash keys** – used to safeguard the integrity and authenticity of data and transactions with algorithms like HMAC-SHA256; anyone with the secret key can impersonate the originator of the data/transactions and thus modify the original data/transactions or create entirely false data/transactions that any recipient will believe is authentic

With an ever-increasing number of keys to protect, and an ever-increasing value of data being protected by those keys, not to mention the demands of PCI-DSS or GDPR, this is a challenge that nearly every business needs to face and address as a matter of urgency.

The most distinct feature of Public Key Infrastructure (PKI) is that it uses a pair of keys to achieve the underlying security service. The key pair comprises of private key and public key. Since the public keys are in open domain, they are likely to be abused. It is, thus, necessary to establish and maintain some kind of trusted infrastructure to manage these keys.

Key Management

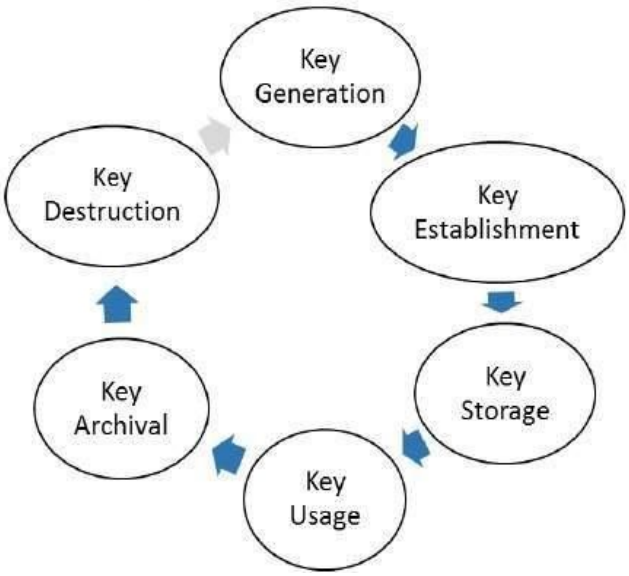
It goes without saying that the security of any cryptosystem depends upon how securely its keys are managed. Without secure procedures for the handling of cryptographic keys, the benefits of the use of strong cryptographic schemes are potentially lost.

It is observed that cryptographic schemes are rarely compromised through weaknesses in their design. However, they are often compromised through poor key management.

There are some important aspects of key management which are as follows −

Cryptographic keys are nothing but special pieces of data. Key management refers to the secure administration of cryptographic keys.

Key management deals with entire key lifecycle as depicted in the following illustration −



There are two specific requirements of key management for public key cryptography.

Secrecy of private keys. Throughout the key lifecycle, secret keys must remain secret from all parties except those who are owner and are authorized to use them.

Assurance of public keys. In public key cryptography, the public keys are in open domain and seen as public pieces of data. By default, there are no assurances of whether a public key is correct, with whom it can be associated, or what it can be used for. Thus, key management of public keys needs to focus much more explicitly on assurance of purpose of public keys.

The most crucial requirement of 'assurance of public key' can be achieved through the public-key infrastructure (PKI), a key management systems for supporting public-key cryptography.

Public Key Infrastructure (PKI)

PKI provides assurance of public key. It provides the identification of public keys and their distribution. An anatomy of PKI comprises of the following components.

- Public Key Certificate, commonly referred to as 'digital certificate'.
- Private Key tokens.
- Certification Authority.
- Registration Authority.
- Certificate Management System.

Digital Certificate

For analogy, a certificate can be considered as the ID card issued to the person. People use ID cards such as a driver's license, passport to prove their identity. A digital certificate does the same basic thing in the electronic world, but with one difference.

Digital Certificates are not only issued to people but they can be issued to computers, software packages or anything else that need to prove the identity in the electronic world.

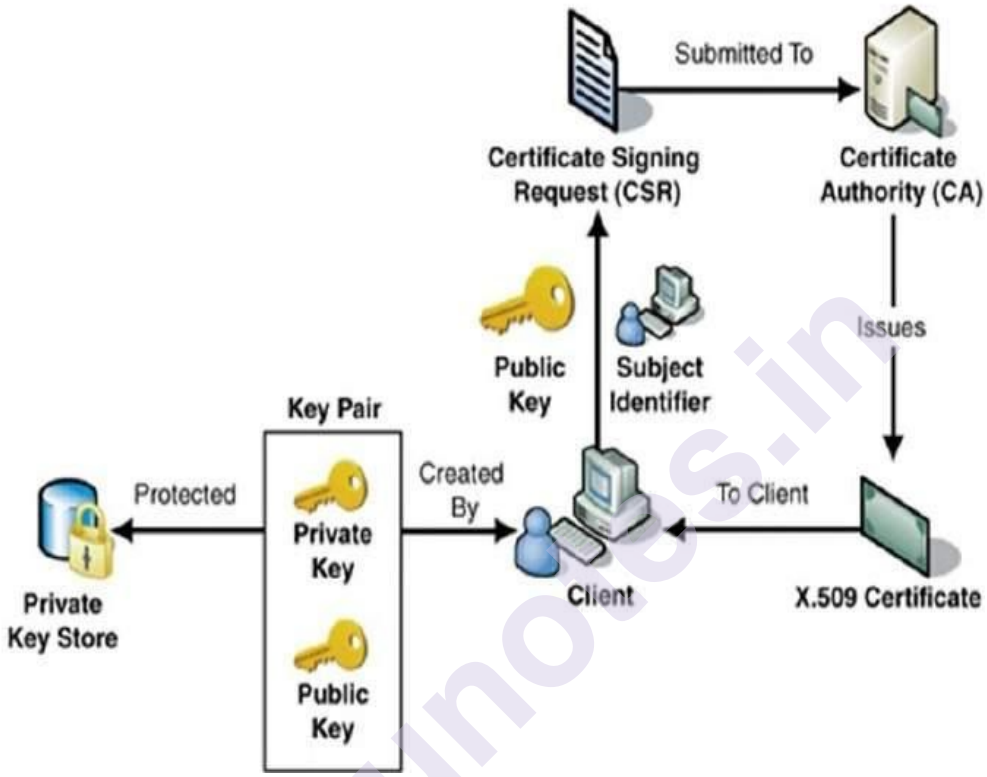
Digital certificates are based on the ITU standard X.509 which defines a standard certificate format for public key certificates and certification validation. Hence digital certificates are sometimes also referred to as X.509 certificates.

Public key pertaining to the user client is stored in digital certificates by The Certification Authority (CA) along with other relevant information such as client information, expiration date, usage, issuer etc.

CA digitally signs this entire information and includes digital signature in the certificate.

Anyone who needs the assurance about the public key and associated information of client, he carries out the signature validation process using CA's public key. Successful validation assures that the public key given in the certificate belongs to the person whose details are given in the certificate.

The process of obtaining Digital Certificate by a person/entity is depicted in the following illustration.



As shown in the illustration, the CA accepts the application from a client to certify his public key. The CA, after duly verifying identity of client, issues a digital certificate to that client.

Certifying Authority (CA)

As discussed above, the CA issues certificate to a client and assist other users to verify the certificate. The CA takes responsibility for identifying correctly the identity of the client asking for a certificate to be issued, and ensures that the information contained within the certificate is correct and digitally signs it.

Key Functions of CA

The key functions of a CA are as follows −

- **Generating key pairs** − The CA may generate a key pair independently or jointly with the client.
- **Issuing digital certificates** − The CA could be thought of as the PKI equivalent of a passport agency − the CA issues a certificate

after client provides the credentials to confirm his identity. The CA then signs the certificate to prevent modification of the details contained in the certificate.

- **Publishing Certificates** − The CA need to publish certificates so that users can find them. There are two ways of achieving this. One is to publish certificates in the equivalent of an electronic telephone directory. The other is to send your certificate out to those people you think might need it by one means or another.

- **Verifying Certificates** − The CA makes its public key available in environment to assist verification of his signature on clients' digital certificate.

- **Revocation of Certificates** − At times, CA revokes the certificate issued due to some reason such as compromise of private key by user or loss of trust in the client. After revocation, CA maintains the list of all revoked certificate that is available to the environment.

Classes of Certificates

There are four typical classes of certificate −

Class 1 − These certificates can be easily acquired by supplying an email address.

Class 2 − These certificates require additional personal information to be supplied.

Class 3 − These certificates can only be purchased after checks have been made about the requestor's identity.

Class 4 − They may be used by governments and financial organizations needing very high levels of trust.

Registration Authority (RA)

CA may use a third-party Registration Authority (RA) to perform the necessary checks on the person or company requesting the certificate to confirm their identity. The RA may appear to the client as a CA, but they do not actually sign the certificate that is issued.

Certificate Management System (CMS)

It is the management system through which certificates are published, temporarily or permanently suspended, renewed, or revoked. Certificate management systems do not normally delete certificates because it may be necessary to prove their status at a point in time, perhaps for legal reasons. A CA along with associated RA runs certificate management systems to be able to track their responsibilities and liabilities.

7.2 PRIVATE KEY MANAGEMENT

What is a private key?

A private key, also known as a *secret key*, is a variable in cryptography that is used with an algorithm to encrypt and decrypt data. Secret keys should only be shared with the key's generator or parties authorized to decrypt the data. Private keys play an important role in symmetric cryptography, asymmetric cryptography and cryptocurrencies.

A private key is typically a long, randomly or pseudo-randomly generated sequence of bits that cannot be easily guessed. The complexity and length of the private key determine how easily an attacker can execute a brute-force attack, where they try out different keys until the right one is found.

How does a private key work?

Private key encryption is also referred to as *symmetric encryption*, where the same private key is used for both encryption and decryption. In this case, a private key works as follows:

- **Generating a new private key.** Prior to encryption, generate a new key that is as random as possible; encryption software is typically used to generate private keys.
- **Securely storing the private key.** Once generated, the private key must be stored securely. Depending on the application, keys may be stored offline or on the computer used to generate, encrypt and decrypt data. Private keys may be protected with a password, encrypted or hashed for security -- or all three.
- **Key exchange.** The private key is used to decrypt, as well as to encrypt, so using it for symmetric encryption requires a key exchange to share that key securely with trusted parties authorized to exchange secured data. Cryptographic software is usually used to automate this process.
- **Key management.** Private key management is required to prevent any individual key from being used for too long. It helps to securely retire keys after their useful lifetime is reached.
- A private key is also used in asymmetric cryptography, which is also known as *public key cryptography*. In this case, the private key refers to the secret key of a public key pair. In public key cryptography, the private key is used for encryption and digital signatures. It works as follows for asymmetric cryptography:
- **Generating a public-private key pair.** Randomness is even more important for this process. Encryption application software is usually used to generate key pairs. It should require a source of randomness, such as mouse movement.

- **Securely storing the private key.** Once generated, the private key must be stored securely. Like the symmetric cryptography process, keys may be stored offline or on the computer used to generate, encrypt and decrypt data. Here, too, private keys should be protected with a password, encrypted or hashed for security.

- **Key exchange.** The private key of a public key pair should almost never be shared with others. Public key cryptography, including digital signatures, is typically used to securely share session keys used for symmetric encryption. However, other protocols for public key infrastructure are used to authoritatively share public keys between cooperating parties.

- **Using the private key.** The owner of a public key pair uses their private key for decrypting data that has been encrypted with the public key of the pair. Only the holder of the private key should be able to decrypt data encrypted with the public key. For digital signatures, the owner of the key pair uses their private key to encrypt the signature. In this way, anyone with access to the public key can decrypt the signature and verify that it was signed by the private key owner.

- **Key management.** Public key pairs are often generated with expiration dates, and key management is vital to maintaining access to data protected with a key pair. For example, an expired public key certificate, which depends on a public key pair, may cause browsers to flag access to a website as insecure. Secret keys should be stored with the highest security, and public key pairs should be managed to avoid compromise or issues related to key pair expiration.

Advantages of private encryption keys

Private key encryption provides several useful features. They include the following four benefits:

1. **More secure.** Private keys that are longer and have greater entropy, or randomness, are more secure from brute-force or dictionary attacks.

2. **Faster.** Symmetric key encryption is faster computationally than asymmetric encryption with its public-private key pairs.

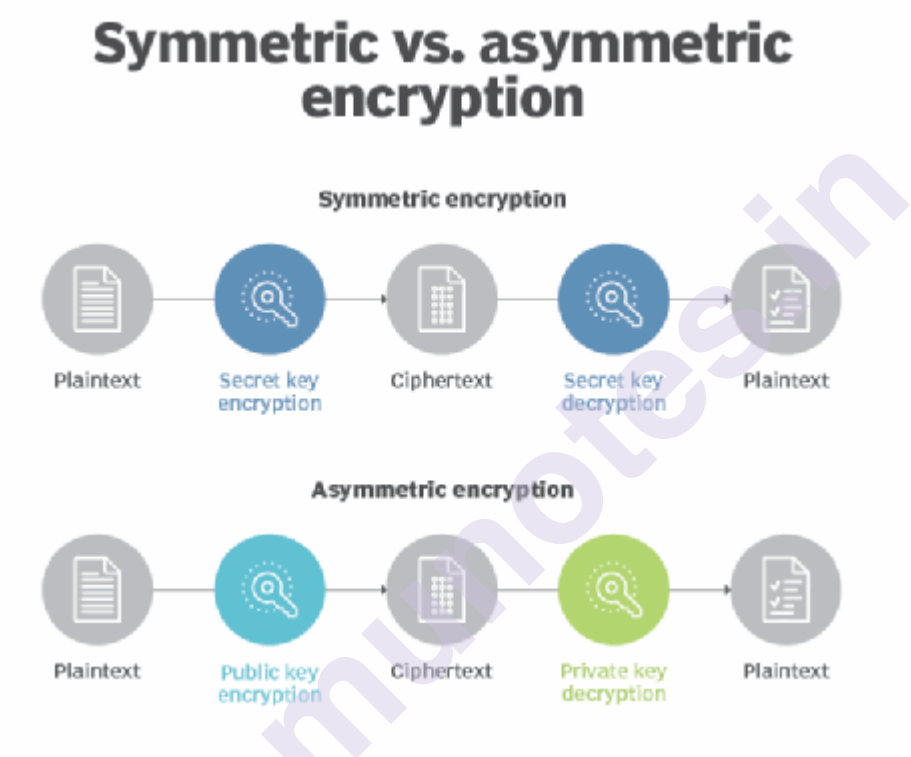
3. **Best for encryption.** Most cryptographic processes use private key encryption to encrypt data transmissions. They typically use a public key algorithm to securely share secret keys.

4. **Work for stream and block ciphers.** Secret key ciphers -- the algorithm for encrypting and decrypting data -- generally fall into one of two categories: stream ciphers or block ciphers. A block cipher applies a private key and algorithm to a block of data simultaneously, whereas a stream cipher applies the key and algorithm one bit at a time.

Private keys vs. public keys

Asymmetric cryptography, also known as *public key cryptography*, uses pairs of public and private keys. These two different but mathematically linked keys are used to transform plaintext into encrypted ciphertext or encrypted text back to plaintext.

When the public key is used to encrypt ciphertext, that text can only be decrypted using the private key. This approach enables anyone with access to the public key to encrypt a message, and only the private key holder will be able to decrypt it.



Two keys, public and private, are required to encrypt and decrypt a ciphertext encrypted with a public key algorithm. Symmetric encryption uses a single secret key.

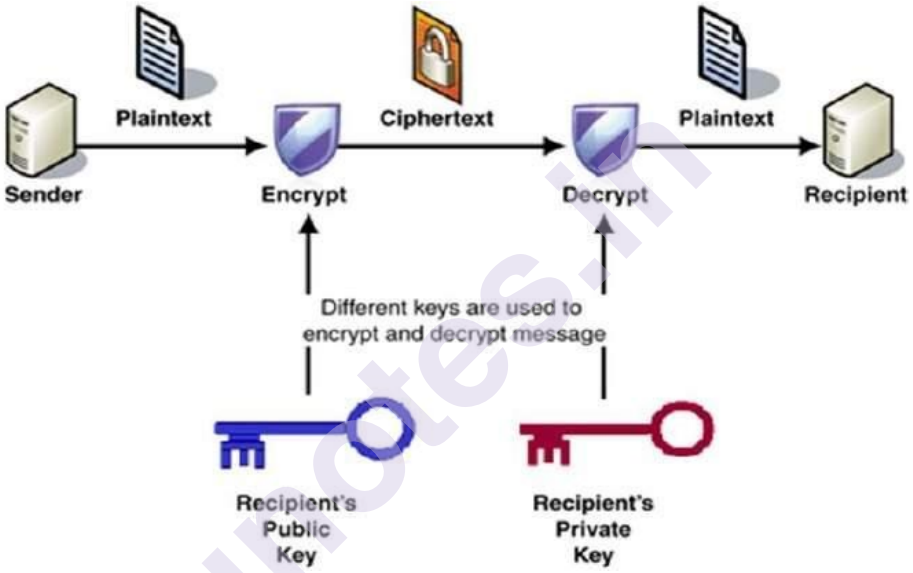
When the private key is used to encrypt ciphertext, that text can be decrypted using the public key. That ciphertext can be a component of a digital signature and used to authenticate the signature. Only the holder of the private key could have encrypted ciphertext, so if the related public key successfully decrypts it, the digital signature is verified.

The public key is made available to everyone that needs it in a publicly accessible repository. The private key is confidential and should only be accessible to the public key pair owner. In this method, whatever is encrypted with the public key requires the related private key for decryption and vice versa. Public key encryption is typically used for securing communication channels, such as email.

7.3 PUBLIC KEY CRYPTOGRAPHY STANDARDS (PKCS)

Public Key Cryptography

Unlike symmetric key cryptography, we do not find historical use of public-key cryptography. It is a relatively new concept. Symmetric cryptography was well suited for organizations such as governments, military, and big financial corporations were involved in the classified communication. With the spread of more unsecure computer networks in last few decades, a genuine need was felt to use cryptography at larger scale. The symmetric key was found to be non-practical due to challenges it faced for key management. This gave rise to the public key cryptosystems.



Public key cryptography standards (PKCS) are a group of specifications developed with the aim of accelerating the deployment of algorithms featuring two separate keys - one private and one public. PKCS were first developed by RSA Laboratories with the cooperation of security developers from around the world.

The first published release of PKCS was in 1991 as a result of the cooperation of early adaptors. The standards promote the use of cryptography techniques such as the RSA algorithm and the Schnorr signature. PKCS are a group of non-vendor dependent standards that are aimed to foster better secure communications through the use of extensive cryptography.

PKCS did not become industry standards initially because RSA retained control over them, but many of the standards were adapted by other working groups.

The standards were developed by RSA with the cooperation of industry partners which included Apple, Microsoft, Lotus, Sun, DEC and MIT.

PKCS stands for public-key cryptography standard, is a model developed by RSA laboratories in early 1990, design to standardize the public key infrastructure. Public Key Cryptography Standard provides a total of 15 standards named as a number like PKCS#1, PKCS#2, PKCS#3, ….. PKCS#15.

7.3.1 List of Public Key Cryptography Standards

There is a total of 15 Public Key cryptography standards. Let's discuss those Public Key cryptography standards one by one.

PKCS #1

The main purpose of this standard is the RSA encryption standard. This standard defines the basic rules for RSA Public Key functions, more specifically, the digital certificates. This standard also defines the syntax for the RSA private and Public Keys, which helps to choose and calculate the RSA algorithm's key pair. It also defines how digital certificates should be calculated, how the structure of the data should be signed, the format of the digital signature.

PKCS #2

The main purpose of this standard is the RSA encryption standard for message digest. This standard defines the calculation for message digest. Now PKCS#2 is merged with PKCS#1. As it merges with standard 1, it does not have an independent existence.

PKCS #3

The main purpose of this standard is the Diffie-Hellman key agreement standard. This standard defines the mechanism to implement the Diffie Hellman key agreement protocol.

PKCS #4

This Public Key cryptography standard also merged with PKCS#1, so it also does not have an independent existence.

PKCS #5

The main purpose of this standard is password-based encryption. It defines the method for encrypting an octet string using a symmetric key which is derived from the password.

PKCS #6

The main purpose of this standard is the extended certificate syntax standard. It defines the syntax for extending the attributes of the X.509 digital certificate.

PKCS #7

The main purpose of this standard is the cryptographic message syntax standard. It defines the syntax for the data, which is the resultant form of cryptographic operations, for example, digital signature and digital envelopes. This standard also provides various formatting options like messages that are only enveloped, only signed, signed.

PKCS #8

The main purpose of this standard is the private key information standard. It defines the syntax for private-key information. In other words, we can say that it defines the algorithms and attributes that are being used to generate the private key.

PKCS #9

The main purpose of this standard is to select attribute types. It defines the selected attribute types that are used in PKCS#6 extended certificates. For example, email address, unstructured address, and name.

PKCS #10

The main purpose of this standard is the certificate request syntax standard. It defines the syntax yo request the digital certificate. The certificate request contains a Distinguished name and Public Key.

PKCS #11

The main purpose of this standard is the cryptographic token interface standard. This standard is also known for Cryptok. It defines API for single-user user devices that contain information about cryptography, such as digital certificates and Public Key. These devices can perform cryptographic functions. For example, smart cards.

PKCS #12

The main purpose of this standard is personal information exchange syntax. It defines the syntax for personal identification such as digital certificates, private keys, etc. In words, we can say that this standard allows users to transfer their data from one device to another using the standard mechanism.

PKCS #13

The main purpose of this standard is the elliptic curve cryptography standard. This standard is used to deal with a new upcoming cryptographic mechanism called elliptic curve cryptography.

PKCS #14

The main purpose of this standard is the pseudo-random number generation standard. This standard defines the requirements and processes for random number generation. As random number generation is extremely used in cryptography, standardizing their generation becomes so much important.

PKCS #15

The main purpose of this standard is the cryptographic token information syntax standard. This standard defines the tokes that are used in the cryptographic process so that they can interoperate.

7.4 DIGITAL CERTIFICATE CREATION STEPS

The steps required to create a digital certificate involves three parties first the end user, second the registration authority and third is certificate

authority. The end user request for a digital certificate and the request goes to the registration authority(RA) which then assist the certificate authority(CA) to create the digital certificate. Registration authority act as a intermediate between end user and the certificate authority. It also assist in day to day task of certificate authority.

Services of Registration Authority:

- Accepting and verifying the details of new user's registration.
- User key generation.
- Backups and recovery of key.
- Certificate cancellation.

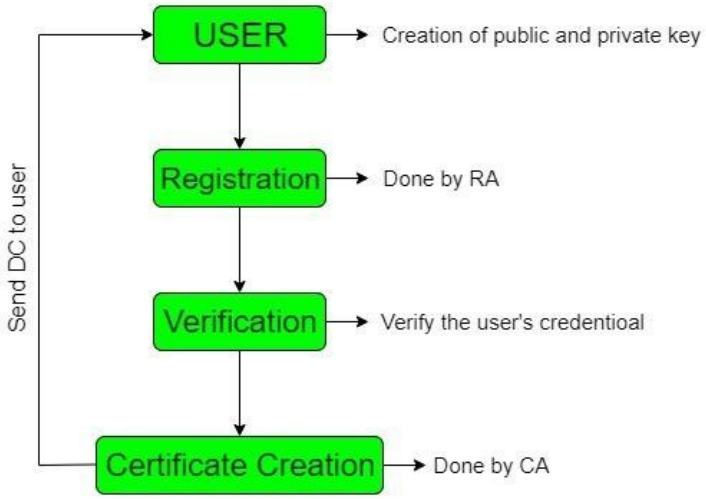
Steps for Digital Certificate Creation:

Step-1: Key generation is done by either user or registration authority. The public key which is generated is sent to the registration authority and private key is kept secret by user.

Step-2: In the next step the registration authority registers the user.

Step-3: Next step is verification which is done by registration authority in which the user's credentials are being verified by registration authority. It also checks that the user who send the public key have corresponding private key or not.

Step-4: In this step the details and sent to certificate authority by registration authority who creates the digital certificate and give it to users and also keeps a copy to itself.



7.5 X.509 CERTIFICATE

An X.509 certificate is a digital certificate based on the widely accepted International Telecommunications Union (ITU) X.509 standard, which defines the format of public key infrastructure (PKI) certificates. They are used to manage identity and security in internet communications and computer networking. They are unobtrusive and ubiquitous, and we encounter them every day when using websites, mobile apps, online documents, and connected devices.

Secure Messaging & Web Browsing

One of the structural strengths of the X.509 certificate is that it is architected using a key pair consisting of a related public key and a private key. Applied to cryptography, the public and private key pair is used to encrypt and decrypt a message, ensuring both the identity of the sender and the security of the message itself. The most common use case of X.509-based PKI is Transport Layer Security (TLS)/Secure Socket Layer (SSL), which is the basis of the HTTPS protocol, which enables secure web browsing. But the X.509 protocol is also applied to code signing for application security, digital signatures, and other critical internet protocols.

Version History

The first version of the X.509 standard was published back in 1988. Looking to formalize the rules for certificate issuance, the Telecommunication Standardization Sector of the ITU (ITU-T) developed a hierarchical system for distinguished names that followed the electronic directory service rules for X.500 and was inspired by the systems used to assign telephone numbers globally but applied to the more flexible organizational requirements of the Internet.

In 1996, version 3 of the standard provided a major update with the addition of multiple extensions that are still used today to support the expansion and new applications of internet use.

Now version 9 is the current version of the standard, having been defined in October 2019.

Additionally, the Internet Engineering Task Force (IETF) public-key infrastructure working group, known as PKIX, adapted the X.509 v3 certificate standard in the development of its own Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile standard (RFC 5280).

The Benefits of X.509 Certificates

Trust - Digital certificates allow individuals, organizations, and even devices to establish trust in the digital world. As the foundation for all digital identities, X.509 certificates are everywhere and are essential to every connected process from websites to applications to endpoint devices

and online documents. For example, without these, we wouldn't be able to trust that www.amazon.com is actually Amazon's website.

This level of trust is established both by how X.509 certificates work and by how they are issued. The key usage architecture lets certificates verify that:

A public key belongs to the hostname/domain, organization, or individual contained within the certificate

It has been signed by a publicly trusted issuer Certificate Authority (CA), like Sectigo, or self-signed.

When a certificate is signed by a trusted CA, the certificate user can be confident that the certificate owner or hostname/domain has been validated, while self-signed certificates can be trusted to a lesser extent as the owner doesn't go through any additional validation before issuance.

Scalability - An additional benefit of this certificate-based approach to identity is scalability. The PKI architecture is so scalable that it can secure billions of messages exchanged daily by organizations over their own networks and across the internet. What enables this is that public keys can be distributed widely and openly without malicious actors being able to discover the private key required to decrypt the message.

How Do X.509 Certificates Work?

The X.509 standard is based on an interface description language known as **Abstract Syntax Notation One (ASN.1)**, which defines data structures that can be serialized and deserialized in a cross-platform way. Leveraging ASN, the X.509 certificate format uses a related public and private key pair to encrypt and decrypt a message.

The Basis of Public Key Infrastructure

The public key is comprised of a string of random numbers and can be used to encrypt a message. Only the intended recipient can decipher and read this encrypted message and it can only be deciphered and read by using the associated private key, which is also made of a long string of random numbers. This private key is secret and is known only to the recipient. As the public key is published for all the world to see, public keys are created using a complex cryptographic algorithm to pair them with an associated private key by generating random numeric combinations of varying lengths so that they cannot be exploited through a brute force attack. The most common algorithms used to generate public keys are:

Rivest–Shamir–Adleman (RSA)

Elliptic curve cryptography (ECC)

Digital signature algorithm (DSA)

The key size or bit length of public keys determines the strength of protection. For example, 2048-bit RSA keys are often employed in SSL certs, digital signatures, and other digital certificates. This key length offers sufficient cryptographic security to keep hackers from cracking the algorithm. Standards organizations like the CA/Browser Forum define baseline requirements for supported key sizes.

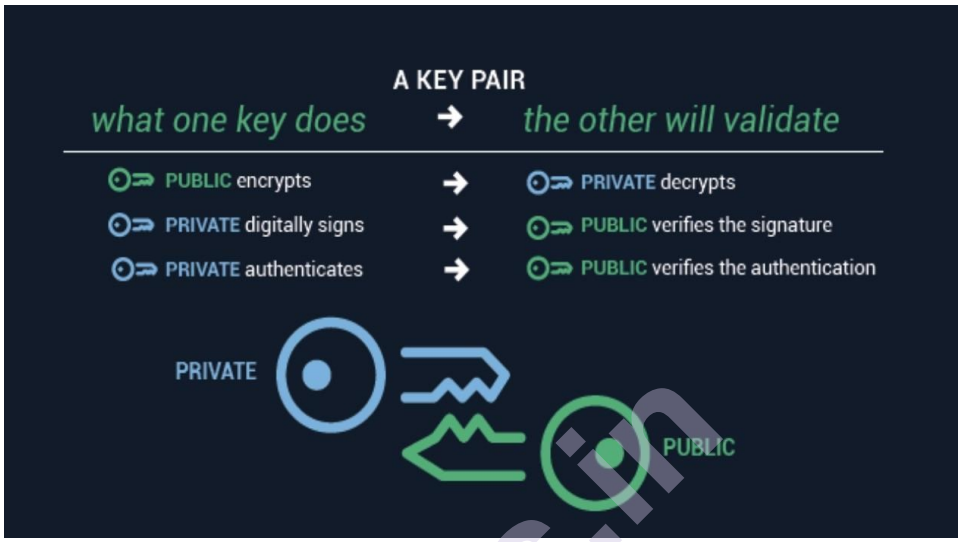


Figure: X.509 certificates use a related public and private key pair for identity authentication and security for internet communications and computer networking

Issuance Fields

X.509 certificate fields contain information about the identity that the certificate is issued to as well as the identity of the issuer CA. The standard fields include:

Version – the X.509 version that applies to the certificate

Serial number – the unique serial number identifier provided by the CA that distinguishes the certificate from others

Algorithm information – the cryptographic algorithm used by the issuer to sign the certificate

Issuer distinguished name – the name of the CA issuing the certificate

Validity period of the certificate – the start/end date and time it's valid and can be trusted

Subject distinguished name – the name of the identity the certificate is issued to

Subject public key information – the public key associated with the identity

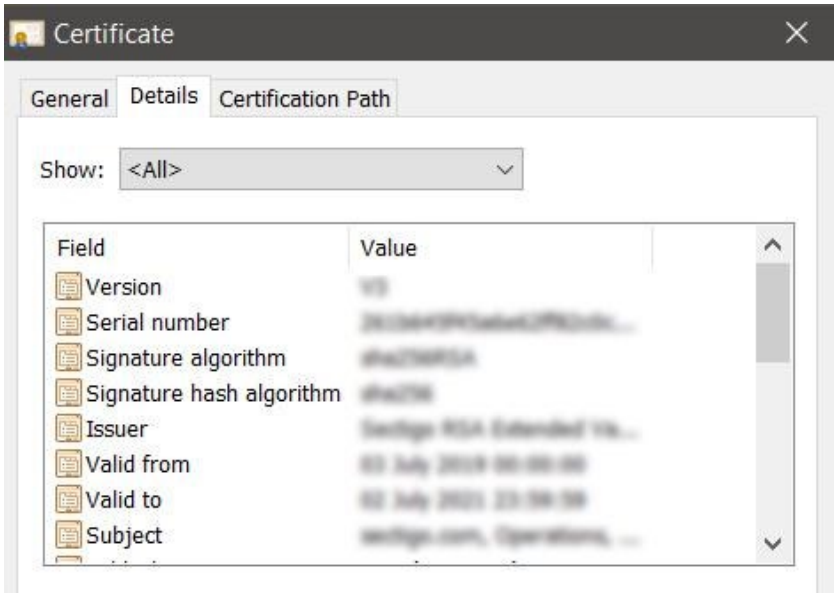


Figure: Standard certificate information fields displayed in TLS/SSL certificates

Common Digital Certificate Extensions

In addition to its standard information fields, the X.509 version 3 defined multiple extensions aimed at supporting expanded ways client applications can use the internet. Two common X.509 certificate extensions in use today are Subject Alternative Name and Key Usage.

The **Subject Alternative Name extension** allows other identities to also be associated with a certificate's public key. This could include other domains, DNS names, email addresses, and IP addresses. Based on this extension, multiple-domain certificates offered by CAs are frequently referred to as SAN certificates.

Key Usage limits the use of the keys to particular purposes such as "signing-only."

Digital Certificates Apply Hierarchical Trust Chains

To further establish the trust of an identity, multiple digital certificates are often combined to build a hierarchical chain of trust that provides a series of verification layers. As previously mentioned, each must be signed by an issuer CA as part of the X.509 verification process. The CA is named and stored in the root of the certificate. Additional intermediate certificates can be included in the trust chain and must be validated.

For example, when a web browser client reads the certificate, it must be able to follow the hierarchical path of certification including any intermediates required for validation that are recursively linked back to the root CA listed in the client's trust store, resulting in a complete chain of trust.

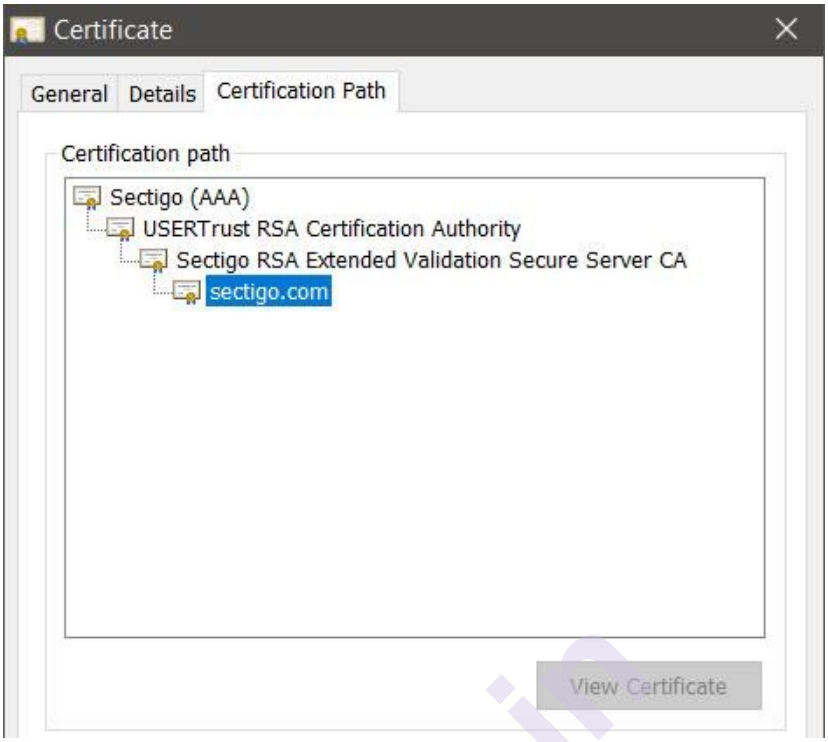


Figure: SSL/TLS certificates often combine intermediate certificates to create a hierarchical trust chain

Certificate Revocation Lists (CRLs)

The X.509 standard also defines the use of a certificate revocation list, which identifies all of the digital certificates that have been revoked by the issuing CA prior to the scheduled expiration date.

These revoked certificates should no longer be trusted.

CRLs offer a simple way to distribute information about these invalid certificates. However, their use is increasingly deprecated by popular web browsers and clients in favor of the Online Certificate Status Protocol (OCSP) and OCSP stapling, which offer complete revocation features.

PKI Certificate Encoding

One notable element not defined in the X.509 standard is how the certificate contents should be encoded to be stored in files.

However, there are two encoding schemas commonly used to store digital certificates in files:

Distinguished Encoding Rules (DER) - most common, as the schema addresses most data objects. Certificates encoded by DER are binary files and cannot be read by text editors but can be processed by web browsers and many client applications.

Privacy Enhanced Mail (PEM) is an encrypted email encoding schema that can be used to convert DER-encoded certificates into text files.

Common Applications of X.509 Public Key Infrastructure

Many internet protocols rely on X.509, and there are many applications of the PKI technology that are used every day, including Web server security, digital signatures and document signing, and digital identities.

Web Server Security with TLS/SSL Certificates

PKI is the basis for the secure sockets layer (SSL) and transport layer security (TLS) protocols that are the foundation of HTTPS secure browser connections. Without SSL certificates or TLS to establish secure connections, cybercriminals could exploit the Internet or other IP networks using a variety of attack vectors, such as man-in-the-middle attacks, to intercept messages and access their contents.

Digital Signatures and Document Signing

In addition to being used to secure messages, PKI-based certificates can be used for digital signatures and document signing.

Digital signatures are a specific type of electronic signature that leverages PKI to authenticate the identity of the signer and the integrity of the signature and the document. Digital signatures cannot be altered or duplicated in any way, as the signature is created by generating a hash, which is encrypted using a sender's private key. This cryptographic verification mathematically binds the signature to the original message to ensure that the sender is authenticated and the message itself has not been altered.

Code Signing

Code signing enables application developers to add a layer of assurance by digitally signing applications, drivers, and software programs so that end users can verify that a third party has not altered or compromised the code they receive. To verify the code is safe and trusted, these digital certificates include the software developer's signature, the company name, and timestamping.

Email Certificates

S/MIME certificates validate email senders and encrypt email contents to protect against increasingly sophisticated social engineering and spear phishing attacks. By encrypting/decrypting email messages and attachments and by validating identity, S/MIME email certificates assure users that emails are authentic and unmodified.

SSH Keys

SSH keys are a form of X.509 certificate that provides a secure access credential used in the Secure Shell (SSH) protocol. As the SSH protocol is widely used for communication in cloud services, network environments, file transfer tools, and configuration management tools, most organizations use SSH keys to authenticate identity and protect those

services from unintended use or malicious attacks. SSH keys not only improve security, but also enable the automation of connected processes, single sign-on (SSO), and identity and access management at the scale that today's businesses require.

Digital Identities

X.509 digital certificates also provide effective digital identity authentication. As data and applications expand beyond traditional networks to mobile devices, public clouds, private clouds, and Internet of Things devices, securing identities becomes more important than ever. And digital identities don't have to be restricted to devices; they can also be used to authenticate people, data, or applications. Digital identity certificates based on this standard enable organizations to improve security by replacing passwords, which attackers have become increasingly adept at stealing.

How Do I Get an X.509 Certificate?

A critical component of deploying X.509 certificates is a trusted certification authority or agent to issue certificates and publish the public keys associated with individuals' private keys. Without this trusted CA, it would be impossible for senders to know they are in fact, using the correct public key associated with the recipient's private key and not the key associated with a malicious actor intending to intercept sensitive information and use it for nefarious purposes.

Trusted, third-party CAs like Sectigo act as certificate authorities, but many enterprises and technology providers also choose to act as their own CA. They may also decide to use self-signed certificates. Either way, the certificate authority must be trusted to check and vouch for the identity of all senders whose public keys they publish, ensure that those public keys are indeed associated with the private keys of the senders, and safeguard the levels of information security within their own organization to guard against malicious attack.

Managing X.509 Certificates

One of the most critical aspects of x.509 certificates is effectively managing these certificates at scale using automation. Without great people, processes, and technology in place, companies are leaving themselves open to security breaches, outages, damage to their brand, and critical infrastructure failures. Discover how Sectigo Certificate Manager(SCM) allows you to easily manage the lifecycles of public and private digital certificates to secure every human and machine identity across the enterprise, all from a single platform.

7.6 CERTIFICATE REVOCATION

Certificate revocation is the act of invalidating a TLS/SSL before its scheduled expiration date. A certificate should be revoked immediately

when its private key shows signs of being compromised. It should also be revoked when the domain for which it was issued is no longer operational.

Certificates that are revoked are stored on a list by the CA, called the Certificate Revocation List(CRL). When a client attempts to initiate a connection with a server, it checks for problems in the certificate, and part of this check is to ensure that the certificate is not on the CRL. The CRL contains the certificates' serial number and the revocation time.

CRLs may be exhaustive, and the client that conducts the check has to parse the whole list to find (or not find) the requested site's certificate. This results in a lot of overhead, and sometimes, a certificate could be revoked within that interval. In such a scenario the client might unknowingly accept the revoked certificate.

A more recent and sophisticated method of detecting revoked certificates is the Online Certificate Status Protocol (OCSP). Here, instead of downloading and parsing the entire CRL, the client can send the certificate in question to the CA. The CA then returns the status of the certificate as "good," "revoked," or "unknown." This method involves far less overhead than CRL and is also more reliable.

The CRL file is signed by the CA to prevent tampering.

What is a digital certificate?

Digital certificates are used in the encryption process to secure communications and create trust in online transactions -- most often, by using the Transport Layer Security/Secure Sockets Layer (TLS/SSL) protocol. The certificate, which is signed by the issuing CA, also provides proof of the certificate owner's identity.

When a web browser connects to a site using TLS, its digital certificate is checked for anomalies or problems. Part of this process involves checking that the certificate is not listed in a CRL.

These checks are crucial for certificate-based transactions because they allow a user to verify the identity of the site owner and discover if the digital certificate is trustworthy.

What is a certificate revocation list?

A certificate revocation list (CRL) is a list of digital certificates that have been revoked by the issuing certificate authority (CA) before their actual or assigned expiration date.

It is a type of blocklist that includes certificates that should no longer be trusted and is used by various endpoints, including web browsers, to verify if a certificate is valid and trustworthy.

❄❄❄❄❄❄❄

8

INTEGRITY

Unit Structure

8.0 OBJECTIVES

The objective of this module is to learn the concept of Message Integrity. What are the properties of Hash Functions? Cryptographic hash functions that are used to create a message digest from a message. Message digest guarantee the integrity of the message. After that we will also study variation of this, called as MAC and HMAC. Then we will start discussing the different message digest algorithm algorithms such as MD5, SHA-512.

8.1 MESSAGE INTEGRITY

As we know the three primary security goals are confidentiality, integrity, and availability. There are some situations in which we may not need confidentiality but integrity is a must. Message/data integrity means that a message/data has not been tampered with or altered. This means we can say that message is protected from unauthorized change.

Consider an example, Leena may write a will to distribute her estate upon her death. There is no need to encrypt the will because after her death anyone can examine the will. So, the integrity of the will needs to be preserved so that the contents of the will are not to be changed.

One way to preserve the integrity of her document is through the use of a fingerprint. If Leena needs to be sure the content of her document will not be changed, she can put her fingerprint at the bottom of the document. The third person (attacker) cannot modify the contents of the document or create a false document because he cannot forge Leena's fingerprint. To ensure that the document has not been changed, Leena's fingerprint on the

document can be compared with Leena's fingerprint on file. If they are not the same, the document is not from Leena.

A variety of mechanisms are used to assure the integrity of a data unit or stream of data units. The most common approach is to use a hash function that combines all the bytes in the message with a secret key and produces a message digest that is difficult to reverse. When a message is passed through such hash function, it produces compressed image of the message that can be used as a fingerprint called as a message digest or message detection code (MDC). A modification detection code (MDC) is a message digest that can prove the integrity of the message: that message has not been changed.

A hash function H accepts a variable-length block of data M as input and produces a fixed-size hash value D = H (M).

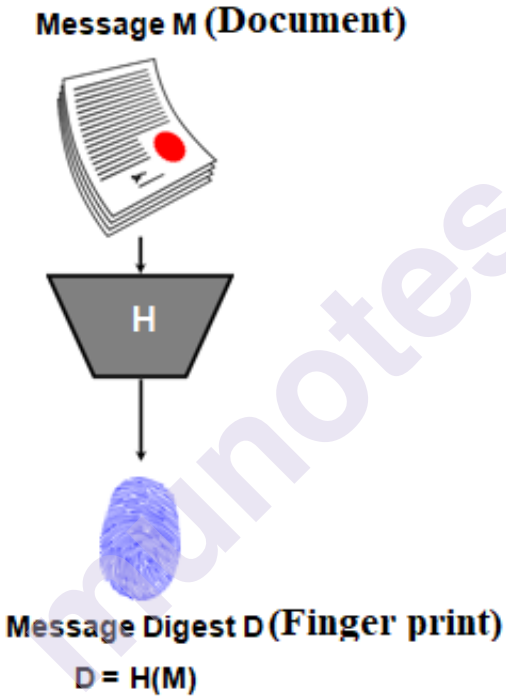


Fig. 8.1.1 Message and Digest

The two pairs (document/fingerprint) and (message/ message digest) are similar, with some differences:

- The (document/fingerprint) are physically linked together.
- (message/message digest) can be unlinked and sent separately

And the most important point is, message digest needs to be safe from any change.

The general idea to check the integrity of the message is as shown in Fig. 8.3.1.2. At the sender side while transmitting the message the initial message digest is calculated by applying the cryptographic hash function

and it's appended at the end of the message. And message along with message digest sent to the receiver. Here to check the integrity of the message, the receiver runs the cryptographic hash function again on the received message and compares the new message digest with the previous one. If both are the same, we are sure that the original message has not been changed.

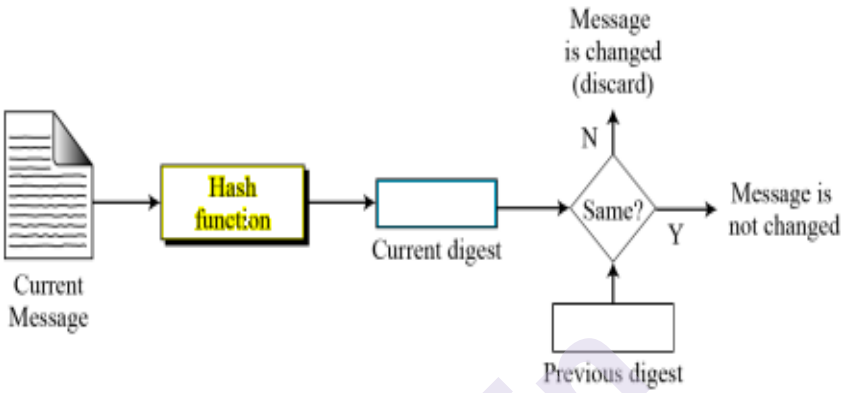


Fig. 8.1.2 Integrity Checking

8.2 HASH FUNCTION PROPERTIES

A hash function is a mathematical function that converts a numerical input value into another compressed numerical value. The hash function takes the message as input and returns a unique, short, and random-looking output.

For secure cryptographic hash function:

- Practically impossible to find the original input given the hash value
- Practically impossible to find two inputs that produce the same hash value

A hash function is an algorithm that usually takes any sized input, like a message or a file, and produces a short random output, the hash value. If you apply the hash function on the same input, we will always get the same hash value as the output. If we apply the hash function on two different inputs, we will get two different hash values as output.

Applications of Hash Function

- Message authentication
- Digital signatures
- Storing passwords
- Signatures of data for malicious behavior detection (e.g. virus, intrusion)
- Generating pseudorandom number

Properties of cryptographic hash functions:

1. **Pre-image Resistant Property(One-wayness):**

   For any given h (hash value), it is computationally infeasible to find m (message) such that H (m) = h. It is called a one-way property. In short, it is hard to find the original input message given the output hash value.

2. **Second Pre-image Resistant Property (Weak Collision Resistance):**

   For any given m1, it is computationally infeasible to find m2≠m1 with H (m2) = H (m1). It is also called a weak collision-resistant property. To break this property, the attacker is trying to find a collision. That is two input messages m1and m2 that produce the same output hash value. Importantly, the attacker cannot choose m1. They are given m1 and must find a different message m2 that produces a collision.

3. **Collision Resistant Property (Strong Collision Resistance):**

   It is infeasible to find any pair (m1, m2) such that H (m1) = H (m2). It is also called a strong collision-resistant property. The attacker tried to find a collision, to break this property. However in this case the attacker has the freedom to find any messages m1 and m2 that produce a collision. Because of this freedom, it makes it easier for the attacker to perform an attack against this property than against the Second Pre-image Resistant property.

The basic requirements for a cryptographic hash function are:

1. The input of the hash function can be of any length.
2. The output of the hash function has a fixed length.
3. H (m) is relatively easy to compute for any given m.
4. H (m) is one-way.
5. H (m) is collision-free.

8.3 MESSAGE AUTHENTICATION CODE (MAC)

Message Authentication Code (MAC) is a tag attached to a message which is used to ensure the integrity and authenticity of the message. In short, MAC is a piece of information that can be used to authenticate a message. MAC is similar to a message digest with one difference. A message digest is simply a fingerprint of a message; no cryptographic process is involved in the case of message digests. In contrast, a MAC requires that the sender and the receiver should know a shared symmetric (secret) key, which is used in the creation of the MAC. Thus, MAC involves cryptographic processing. MAC is sometimes also called a keyed hash.

MAC processing works are shown in Fig. 8.3.1.

Let us assume that sender A wants to send a message M to receiver B.

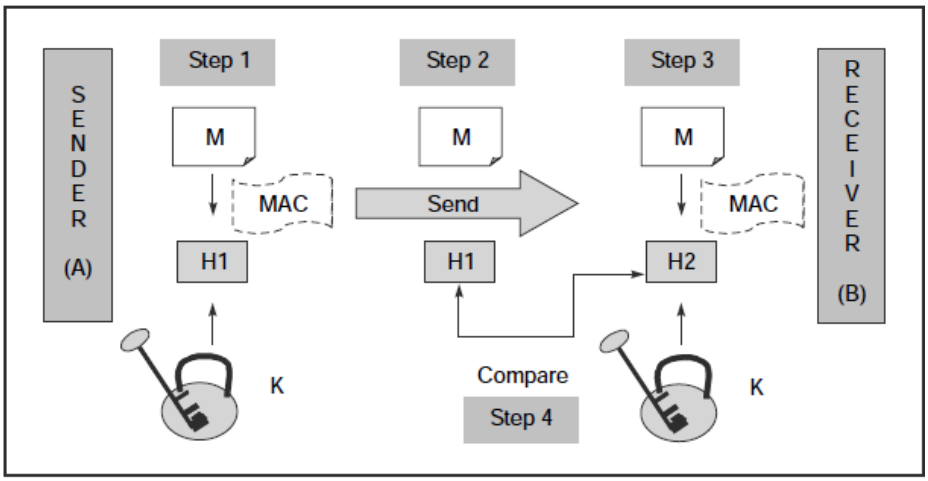


Fig 8.3.1 Message Authentication Code

1. sender A and receiver B share a secret key K, which is known by only A and B. Sender A calculate the MAC by applying key K and message M to the MAC algorithm.
2. Sender A then sends the original message M and the MAC H1 to receiver B.
3. When receiver B receives the message, B also uses K to calculate its own MAC H2 over M.
4. Receiver B now compares received H1 with computed H2. If the two match, receiver B concludes that message M has not been changed during transit. If H1 ≠ H2 then receiver B rejects the message, realizing that the message was changed during transit.

The importance of a MAC is as follows:

1. The MAC assures that message is not altered by the receiver (B), because if an attacker alters the message but does not alter the MAC then the receiver's calculation of the MAC will differ from it. There is no chance to alter the MAC by attackers since the key used in the calculation of the MAC is assumed to be known only to sender A and receiver B. As the attackers do not know the key, K and therefore, they cannot alter the MAC.
2. The receiver B is assured that the message indeed came from the correct sender A. Since only sender A and receiver B know the secret key, no one else could have calculated the MAC sent by sender A.

Though the calculation of the MAC is quite similar to an encryption process, in symmetric key cryptography the cryptographic process must be reversible. This Means the sender performs encryption and the receiver performs decryption. But in the case of MAC, both the sender and the receiver are performing the encryption process only. Thus, a MAC

algorithm need not be reversible – it is sufficient to be a one-way function (encryption) only.

Limitations of a MAC are as follows:

1. MAC does not provide Non-Repudiation. Non-repudiation is the assurance that a message originator cannot deny any previously sent messages and actions or commitments.
2. MAC can provide message authentication among pre-decided legitimate users who have shared key. This requires establishment of shared prior to use the MAC.

8.4 HASH-BASED MESSAGE AUTHENTICATION CODE (HMAC)

A Hash-based Message Authentication Code (HMAC) is a nested MAC that includes a cryptographic hash function and a secret key in deriving the message authentication code (MAC). Like any of the MAC, it is used for both data integrity and data origin authentication of the message. The idea behind HMAC is to reuse the existing message digest algorithms such as MD5, SHA-1, and SHA-256. This means it treats the message digest as a black box. It also uses the shared symmetric key to encrypt the message digest, which produces the output MAC. This is shown in Fig. 8.4.1

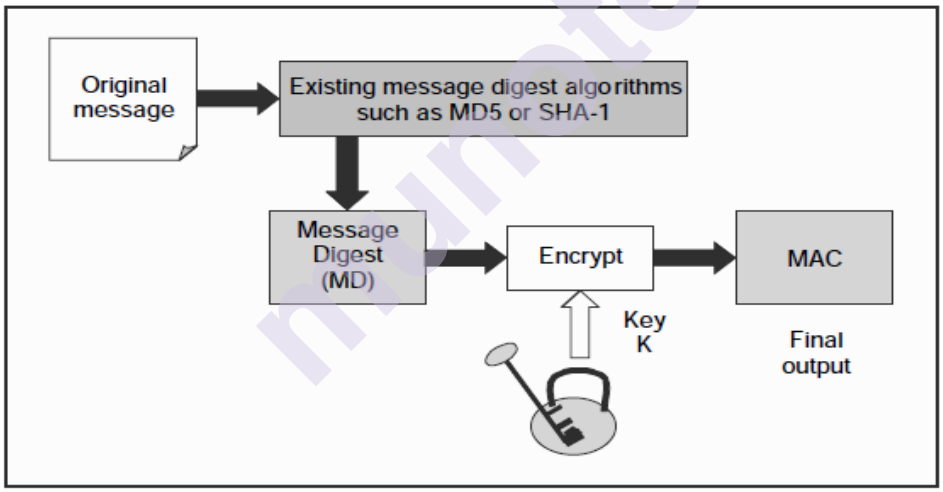


Fig. 8.4.1 HMAC concept

FTPS, SFTP, HTTPS, and other transfer protocols use HMAC. HMAC has been chosen as a mandatory security implementation for the Internet Protocol (IP) security and also used in the Secure Socket Layer (SSL) protocol, widely used on the Internet.

Internal working of HMAC:

Various variables that will be used in our HMAC discussion are as follows:

MD = Message digest/hash function used (e.g. MD5, SHA-1, etc.)

M = Input message whose MAC is to be calculated
L = Number of blocks in the message M
b = number of bits in each block
K =Shared symmetric key to be used in HMAC
ipad = A string 00110110 (36 in hexadecimal) repeated b/8 times
opad = A string 01011010 (5C in hexadecimal) repeated b/8 times
Now we will use a step-by-step approach to understand the HMAC operation.

Step 1: Make the length of the S**hared symmetric key (K) equal to b**

There are three cases, depending on the length of the key K:

Case 1: Length of K < b

Here we have to expand the key (K) to make the length of K equal to the number of bits in the original message block (i.e. b). For this, we add have to as many 0 bits as required to the left of K.

For example,

if the initial length of K = 150 bits and b = 512 bits, then we have to add 362 bits, all with a value 0, to the left of K.

Case 2: Length of K = b

Here we do not require taking any action and directly proceeding to Step 2.

Case 3: Length of K > b

Here we need to trim (reduce length) K to make the length of K equal to the number of bits in the original message block. For this, we pass K through the message digest algorithm (H) selected for this particular instance of HMAC, which will give us a key K, trimmed so that its length is equal to b. This is shown in 8.4.2

For this, we give K as input to the message digest algorithm (H) which will give us a trimmed key K, so that its length is equal to b.

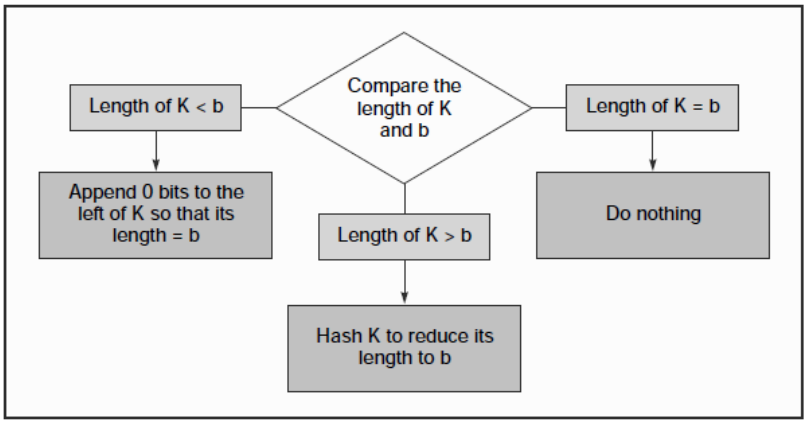


Fig. 8.4.2 Step 1 of HMAC

Step 2: XOR K with ipad to produce S1

In this step, we XOR K (the output of Step 1) and ipad to produce a variable called S1. This is shown in Fig. 8.3.4.3

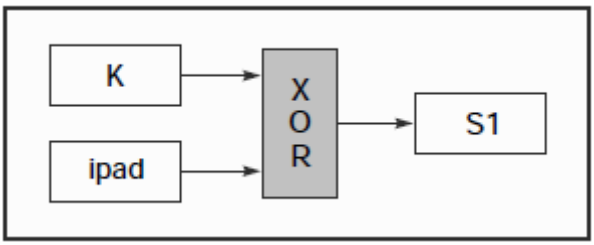


Fig. 8.4.3 Step 2 of HMAC

Step 3: Append M to S1

In this step, we now take the original message (M) and simply append it to the end of S1. This is shown in Fig. 8.3.4.4

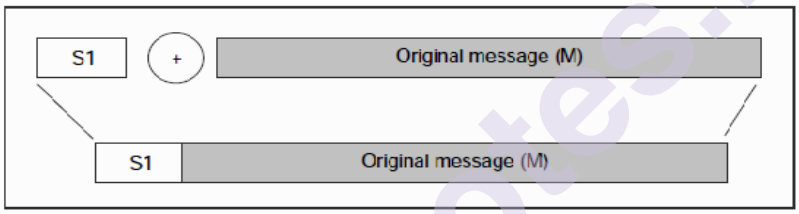


Fig. 8.4.4 Step 3 of HMAC

Step 4: Message digest algorithm

In this step, the selected message-digest algorithm (e.g. SHA-1, MD5 etc) is applied to the output of Step 3 (i.e. to the combination of S1 and M). Let us call the output of this operation H. This is shown in Fig. 8.3.4.5

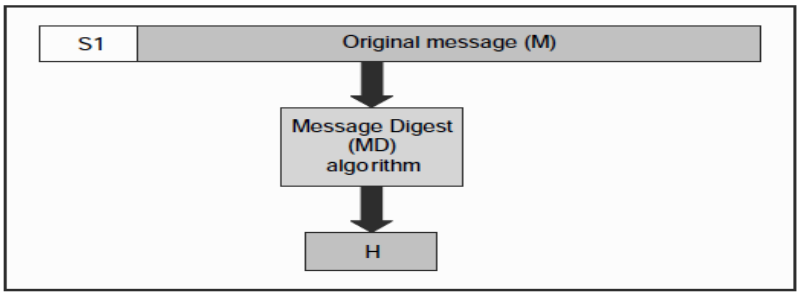


Fig. 8.4.5 Step 4 of HMAC

Step 5: XOR K with opad to produce S2

Now, we XOR K (the output of Step 1) with opad to produce a variable called S2. This is shown in Fig. 8.3.4.6

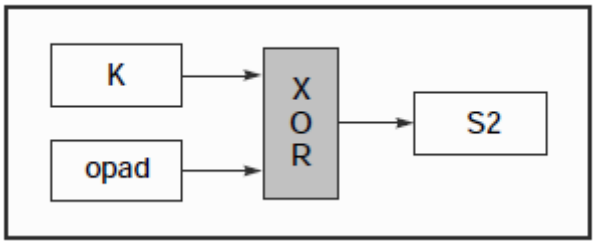


Fig. 8.4.6 Step 5 of HMAC

Step 6: Append H to S2

In this step, we take the message digest calculated in step 4 (i.e. H) and simply append it to the end of S2. This is shown in Fig. 8.3.4.7

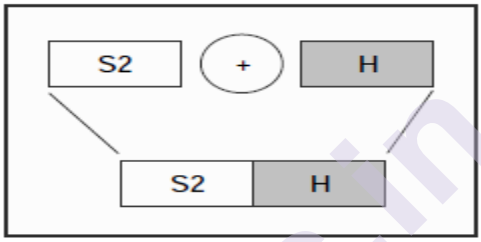


Fig. 8.4.7 Step 6 of HMAC

Step 7: Message digest algorithm

In this step, the selected message-digest algorithm (e.g. MD5, SHA-1, etc) is applied to the output of Step 6 (i.e. to the concatenation of S2 and H). This is the final MAC that we want. This is shown in Fig. 8.3.4.8

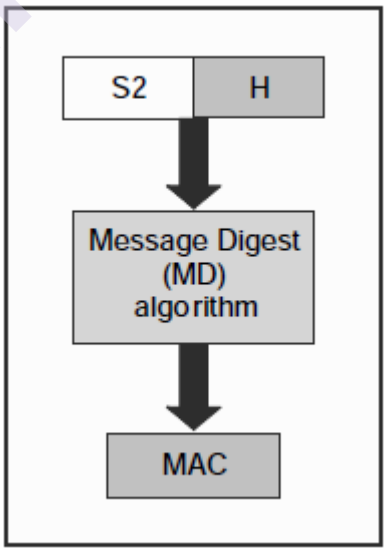


Fig. 8.4.8 Step 6 of HMAC

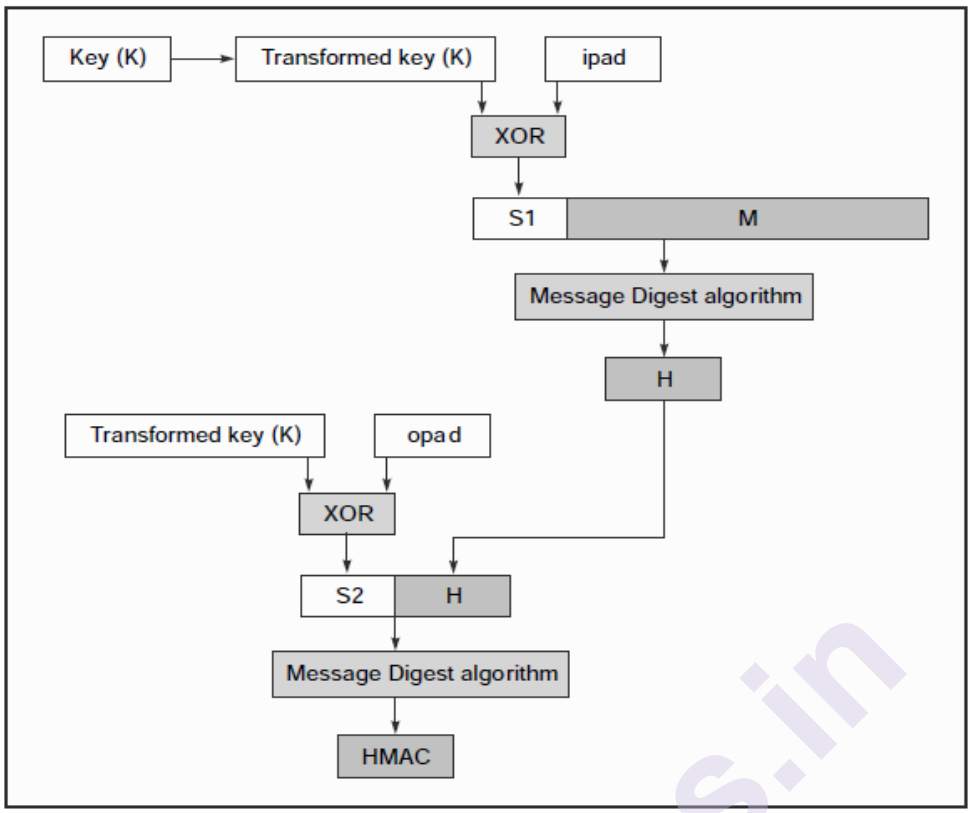


Fig. 8.4.9 Complete HMAC operation

Advantage of HMAC:

- As it uses the hashing concept twice, it is a great resistance towards cryptanalysis attacks
- HMAC consists of twin benefits of Hashing and MAC and thus is more secure than any other authentication code
- Due to the use of hash functions, HMACs are considered to be good high-performance systems.

The Disadvantage of HMAC:

- The main issue is with key exchange
- Since it uses a symmetric key for encryption of message digest, the problem of key distribution is there.
- Somehow the key-exchange problem is resolved; HMAC cannot be used if the number of receivers is greater than one.
- If multiple parties share the same symmetric key. How does a receiver know that the message was prepared and sent by the sender.
- There is also a need for periodic refreshments of keys.
- A replay of the message.
- Lack of authentication is one of the problems.

8.5 MD 5

Ron Rivest developed an **MD5 (M**essage **D**igest **5**) cryptographic hash algorithm. MD5 is quite fast and produces 128-bit message digests from a string of any length.

MD5 is the third message-digest algorithm created by Rivest. MD2, MD4, and MD5 have similar structures, but MD2 was found to be quite weak so he began working on MD3, which failed (so never released). Then Rivest Developed MD4. The MD5 algorithm is an extension of MD4, which the critical review found to be fast but potentially insecure. In comparison, MD5 is not quite as fast as the MD4 algorithm but offered much more assurance of data security. Over the years, researchers have developed potential weaknesses in MD5.

MD5 processes input text in 512-bit blocks which are further divided into 16 32-bit sub-blocks. The output of the algorithm is a set of four 32-bit blocks, which make up the 128-bit message digest.

The digest size is always 128 bits. If there is a minor change in the input string, then it generates a drastically different digest. This is essential to prevent similar hash generation as much as possible, also known as a hash collision.

Working of MD5 Algorithm:

Step 1: Padding Bits

In the MD5 algorithm, the first step is to add padding bits to the original message. The aim is here to make the length of the original message equal to a value, which is 64 bits less than an exact multiple of 512.

For example, if the length of the original message is 1200 bits, we add padding of 272 bits to make the length of the message 1472 bits.

This is because, if we add 64 to 1472, we get 1536, which is a multiple of 512 (because 1536 = 512 * 3).

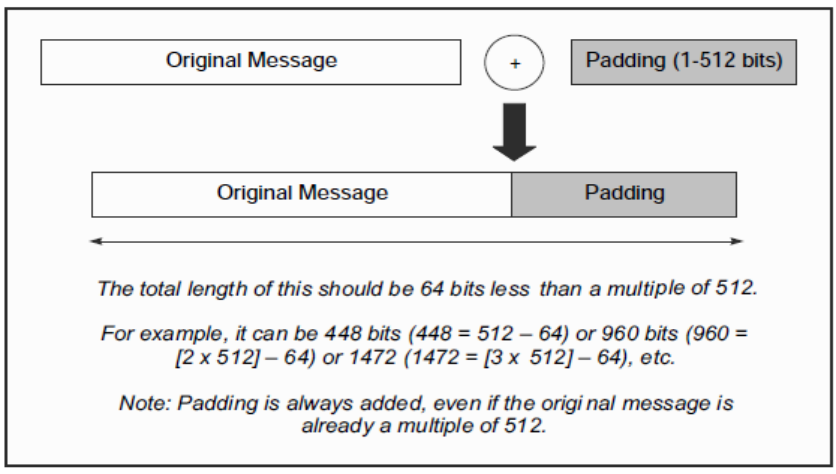


Figure 8.5.1 Padding Bits

Step 2: Append length

The next step is to calculate the length of the original message and add it to the end of the message, after padding. The length of the message is calculated; excluding the padding bits (i.e. it is the length original message).

For example, if the original message consisted of 1200 bits and we added a padding of 272 bits to make the length of the message 64 bits less than 1536 (a multiple of 512), the length is considered as 1200 and not 1472 for this step. This length of the original message is now expressed as a 64-bit value and these 64 bits are appended to the end of the original message + padding.

This is shown in Fig. 8.3.5.2

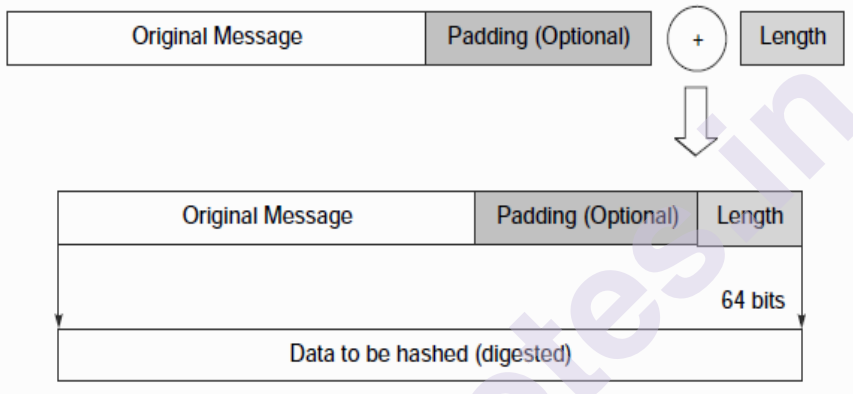


Fig 8.5.2 Append length

Note that if the length of the message exceeds 2^{64} bits then, we use only the low-order 64 bits of the length. The length of the message is now an exact multiple of 512. This now becomes the message whose digest will be calculated.

Step 3: Divide the input into 512-bit blocks

In this step, the input message is divided into blocks, each of length 512 bits.

This is shown in Fig. 8.5.3

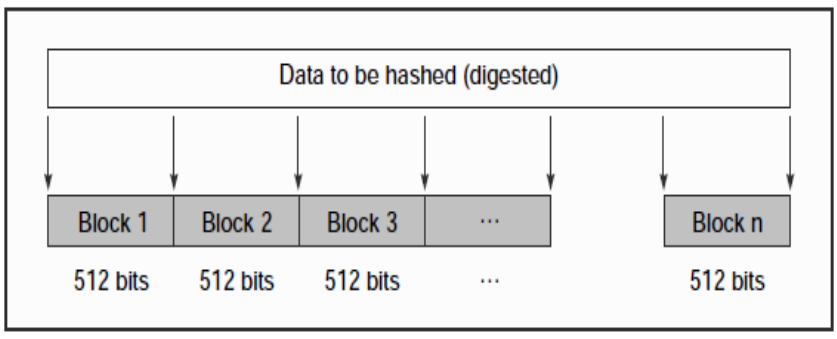


Fig. 8.5.3 Data divided into 512-bit blocks

Step 4: Initialize chaining variables

Here four variables also called **chaining variables** are initialized. They are denoted as A, B, C, and D. Each of these is a 32-bit number. The initial hexadecimal values of these chaining variables are shown in Fig. 8.5.4

A	Hex	01	23	45	67
B	Hex	89	AB	CD	EF
C	Hex	FE	DC	BA	98
D	Hex	76	54	32	10

Fig. 8.5.4 Chaining variables

Step 5: Process Each block

After all the initializations, the real algorithm begins. It is quite complicated and we shall now we will discuss it step-by-step. There is a loop that runs for as many 512-bit blocks as are in the message.

Step 5.1: Initially the four chaining variables are copied into four corresponding variables, a, b, c and d (note the smaller case). Means, we have now have a = A, b = B, c = C and d = D.

This is shown in Fig. 8.5.5.

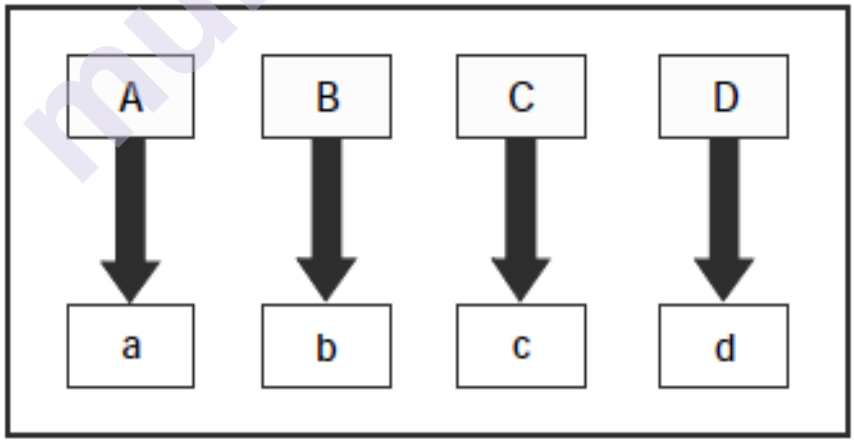


Fig. 8.5.5 Copying chaining variables into temporary variables

The algorithm considers the combination of a, b, c, and d as a 128-bit single register (which we shall call abcd). This register (abcd) is useful in the actual algorithm operation for holding final as well as intermediate results. This is shown in Fig. 8.5.6

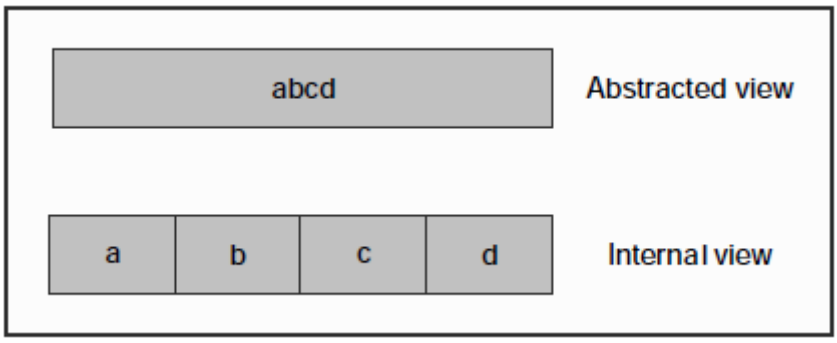


Fig. 8.5.6 Abstracted view of the chaining variables

Step 5.2*:* Current 512-bit block is divided into 16 sub-blocks. Thus, each sub-block contains now 32 bits, as shown in Fig. 8.5.7

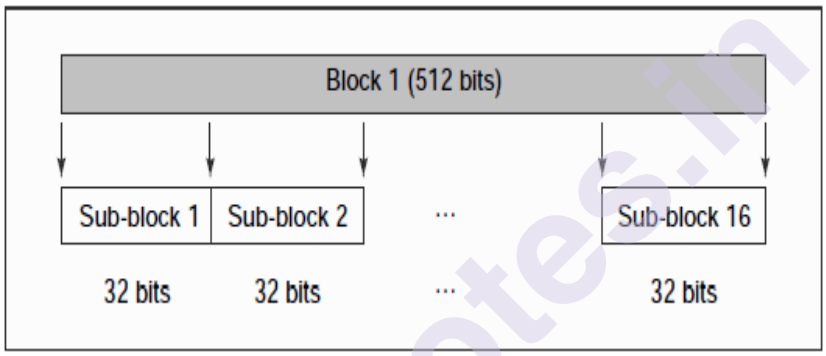


Fig. 8.5.7 Sub-blocks within a block

Step 5.3: Now, we have four *rounds*. In each round, we process all the 16 sub-blocks belonging to a block. The inputs to each round are:

(a) All the 16 sub-blocks
(b) The variables a, b, c, d
(c) Some constants, designated as t.

This is shown in Fig. 8.5.8

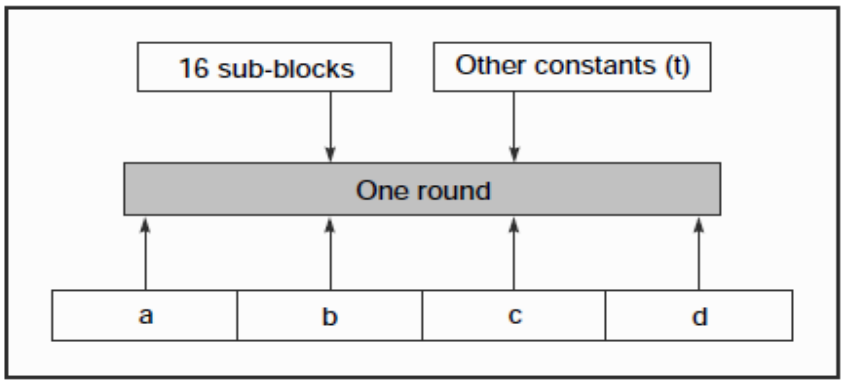


Fig. 8.5.8 Conceptual process within a round

All the four rounds vary in one major way: Step 1 of the four rounds has different processing. The other steps in all the four rounds are the same.

a) In each round, we have 16 input sub-blocks, named M[0], M[1], …, M[15] or in general, M[i], where i varies from 0 to 15.

b) Also, it is an array of constants. It contains 64 elements, with each element consisting of 32 bits. The elements of this array t as t[1], t[2], … t[64] or in general as t[k], where k varies from 1 to 64. Since there are four rounds, we use 16 out of the 64 values of t in each round.

Let us we will summarize iterations of all the four rounds:

In each case, the output of the intermediate as well as the final iteration is copied into the register abcd. Note that we have 16 such iterations in each round.

1. A process P is first performed on b, c, and d. This process P is different in all the four rounds.
2. The variable a is added to the output of the process P (i.e. to the register abcd).
3. The message sub-block M[i] is added to the output of Step 2 (i.e. to the register abcd).
4. The constant t[k] is added to the output of Step 3 (i.e. to the register abcd).
5. The output of Step 4 (i.e. the contents of register abcd) is circular-left shifted by s bits. (The value of s keeps changing, as we shall study).
6. The variable b is added to the output of Step 5 (i.e. to the register abcd).
7. The output of Step 6 becomes the new abcd for the next step.

This is shown in Fig. 8.5.9

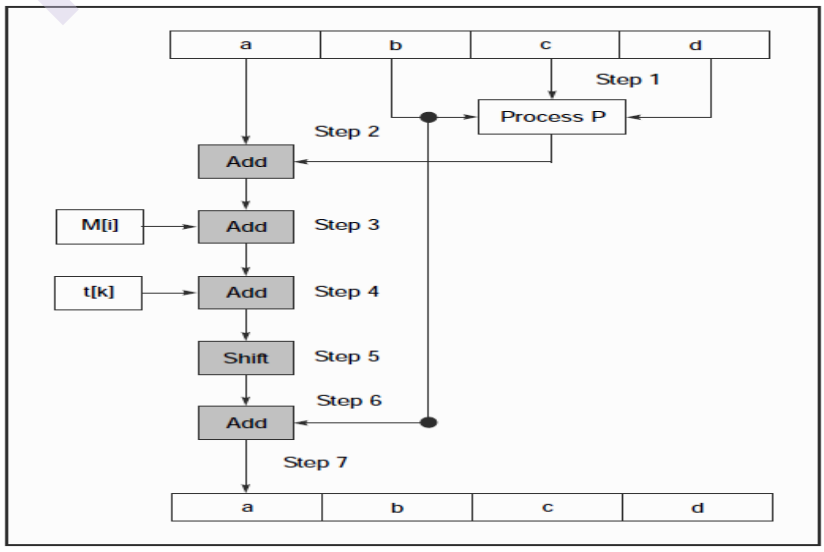


Fig 8.5.9 One MD5 operation

We can mathematically express a single MD5 operation as follows:

a = b + ((a + Process P (b, c, d) + M[i] + T[k]) <<< s)

Table 8.5 Process P in each round

Round	*Process P*
1	(b AND c) OR ((NOT b) AND (d))
2	(b AND d) OR (c AND (NOT d))
3	B XOR c XOR d
4	C XOR (b OR (NOT d))

8.6 SHA-512

SHA-512 (Secure Hash Algorithm 512) is just one of several algorithms in the Secure Hashing Algorithm (SHA) family. In 2001, SHA-512 was published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS).

SHA-512 is a hashing algorithm which takes a message of length 2128 bits and produces a message digest of length of 512 bits (64 bytes).This algorithm is commonly used for password hashing, email addresses hashing and digital record verification. SHA-512 is also used in block chain technology.

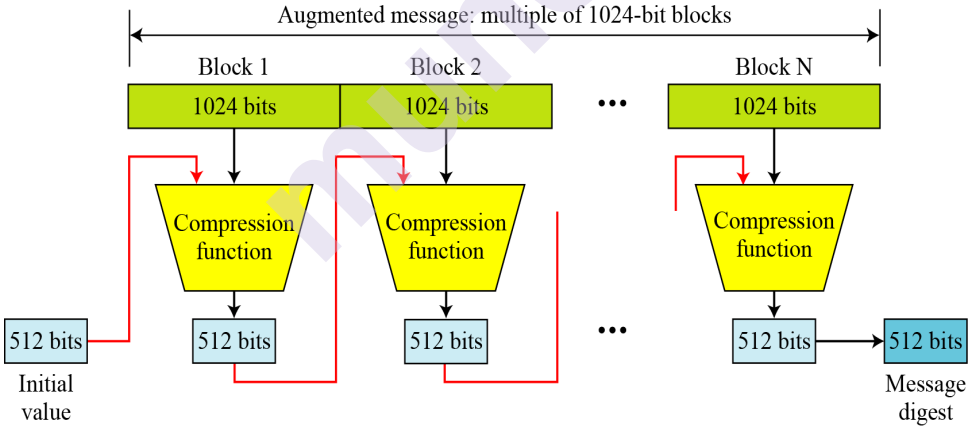


Fig 8.6.1 Structure of SHA-512

SHA-512 steps are as follows to create a message digest:

- Step 1: Append padding bits
- Step 2: Append length
- Step 3: Divide the input into 1024-bit blocks
- Step 4: Initialize hash buffer
- Step 5: Process blocks

Now we will focus on above steps one by one

Step 1: Padding

The message needs to be padded so that its length is congruent to 896 modulo 1024 [length = 896(mod 1024)]. Even if the message is in desired length already, padding is always added. Thus number of padding bits is in the range 1 to 1024. It consists of a single 1 bit followed by necessary no. of 0's (100000….00).

Step 2: Append length

Here we add 128 bits consisting of the length of the original message (before padding in Step 1). The length of the message is taken in hexadecimal format for adding in this 128 bit pad. Thus the length of the block becomes of length 1024 (= 896 + 128).

(|M| + |P| + 128) = 0 mod 1024 -> |P| = (- |M| - 128) MOD 1024

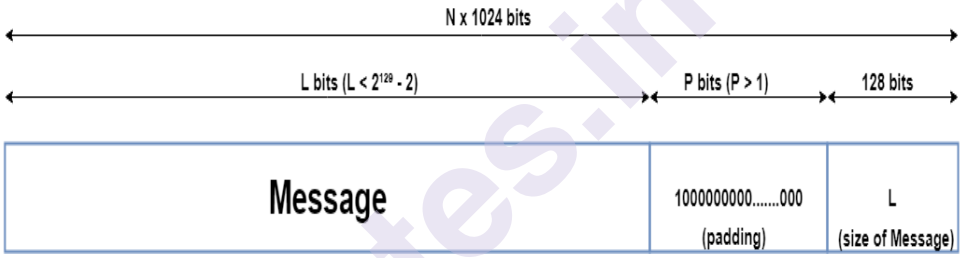


Fig 8.6.2 Length Field and Padding

Example: What is the number of padding bits if the length of the original message is 2590 bits?

We can calculate the number of padding bits as follows:

|P| = (- 2590| - 128) MOD 1024= -2718 MOD 1024 =354

The padding consists of one 1 followed by 353 0's.

Step 3: Divide the input into 1024-bit blocks

The input message is now divided into blocks, each of length 1024 bits. These blocks become the input to the message digest processing logic.

Step 3: Initialize Hash Buffer

A 512-bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as eight 64-bit registers (a, b, c, d, e, f, g, h). These registers are initialized to the following 64- bit integers (hexadecimal values) which were obtained by taking the first sixty-four bits of the fractional parts of the square roots of the first eight prime numbers(2,3,5,7,11,13,17,19). These values are called the Initial Vectors (IV):

a = 6A09E667F3BCC908 e = 510E527FADE682D1

b = BB67AE8584CAA73B d = A54FF53A5F1D36F1

c = 3C6EF372FE94F82B f = 9B05688C2B3E6C1F

g = 1F83D9ABFB41BD6B h = 5BE0CDI9137E2179

Step 5: Process blocks

Now the actual algorithm begins. Here also, the steps are quite similar to those in MD5. The combination of a-h, called as abcdefgh will be considered as a single register for storing the temporary intermediate and final results. Then the current 1024-bit block divided into 16 sub-blocks, each consisting of 64 bits.

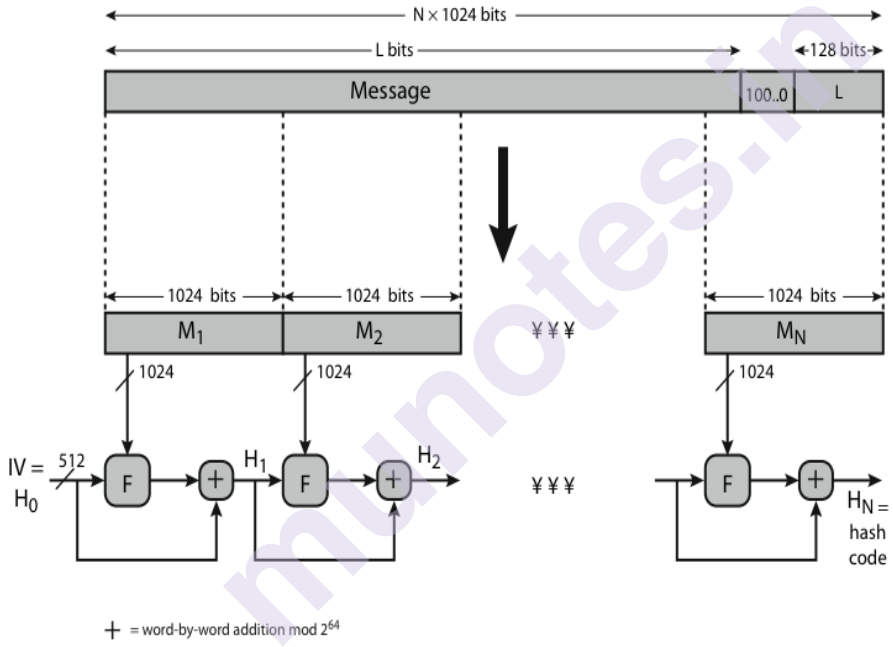


Figure 12.1 Message Digest Generation Using SHA-512

Fig 8.6.3 Message Digest creation using SHA-512

SHA-512 consists of 80 rounds, labeled F in Figure 3.3.6.3. Each round takes the current 1024-bit block, the register abcdefgh and a constant K[t] (where t= 0 to 79) as the three input values as shown in Figure 3.3.6.4. It then updates the contents of the register abcdefgh using SHA-512 algorithm steps. The operation of a single round is as shown in Fig 3.3.6.5.

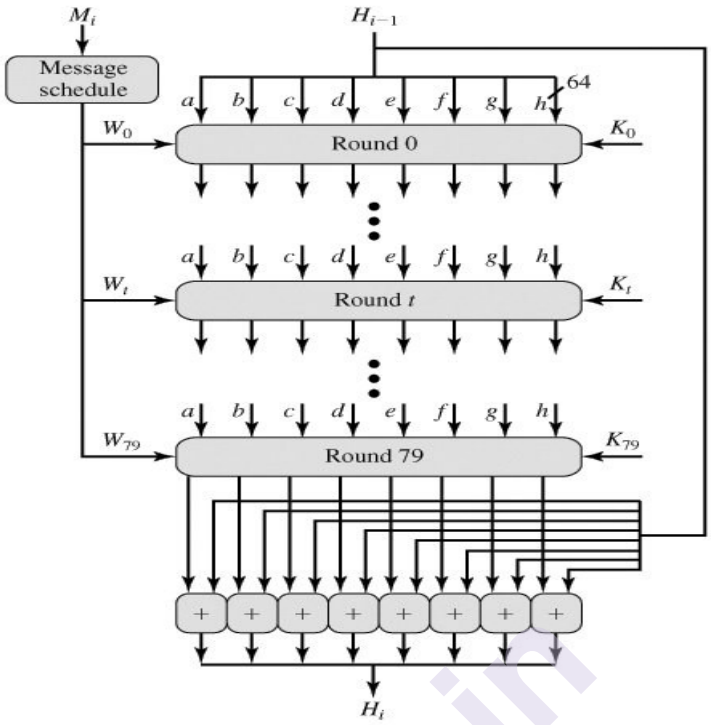


Fig 3.6.4 SHA-512 Round Function

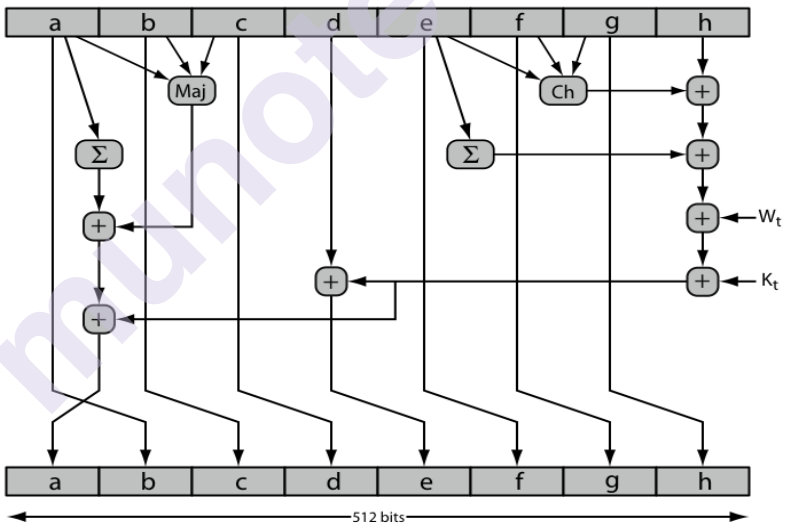


Figure 12.3 Elementary SHA-512 Operation (single round

Fig 8.6.5 Single SHA-512 iteration

Each 64-bit word shuffled along one place, and in some cases manipulated using a series of simple logical functions (ANDs, NOTs, ORs, XORs, ROTates), in order to provide the avalanche & completeness properties of the hash function.

Each round consists of the following operations:

Temp1=h+ Ch(e,f,g)+ ∑(ei for i= 1 to 512)+ Wt+ Kt

Temp2= ∑(ai for i= 1 to 512)+ Maj(a,b,c)
a = Temp1+Temp2
b=a
c=b
d=c
e=d+Temp1
f=e
g=f
h=g
Where:
t = Round Number
Ch(e,f,g) = (e AND f) XOR (NOT e AND g)
Maj(a,b,c) = (a AND b) XOR (a AND c) XOR (b AND c)
∑(a) = ROTR(a,28) XOR ROTR(a,34) XOR ROTR(a,39)
∑(e) = ROTR(e,14) XOR ROTR(e,18) XOR ROTR(e,41)
ROTR(x) = Circular right shift, i.e. rotation, of the 64-bit array x by the specified number of bits
+ = addition modulo 2^64
Kt = a 64-bit additive constant
Wt = a 64-bit word derived from the current 512-bit input block.

Six of the eight words of the output of the round function involve simply permutation (b, c, d, f, g, h) by means of rotation. This is indicated by shading in Figure 3.3.6.5 Only two of the output words (a, e) are generated by substitution. Word e is a function of input variables d, e, f, g, h, as well as the round word Wt and the constant Kt. Word a is a function of all of the input variables, as well as the round word Wt and the constant Kt.

Calculation of Wt

The 64-bit word values for Wt are derived from 1024 -bit message using certain mappings. Wt is used in each of the 80 rounds of each block where t = (0 to 79). Each wt is of length 64 bits.

Wt is calculated as follows:

- First 16 Wt's (0 to 15) are taken as it is from the message (16 x 64=1024).
- The remaining values are defined as a function of the earlier values using ROTates, SHIFTs and XORs as shown in Fig. 3.3.6.6.

Rest of the Wt's are calculated using the formula –

Wt= ∂1(x) [W (t-2)] + W (t-7) + ∂0(x) [W (t-15)] + W (t- 16).
Where, ∂0(x) = ROTR(x, 1) XOR ROTR(x, 8) XOR SHR(x, 7)
∂1(x) = ROTR(x, 19) XOR ROTR(x, 61) XOR SHR(x, 6).

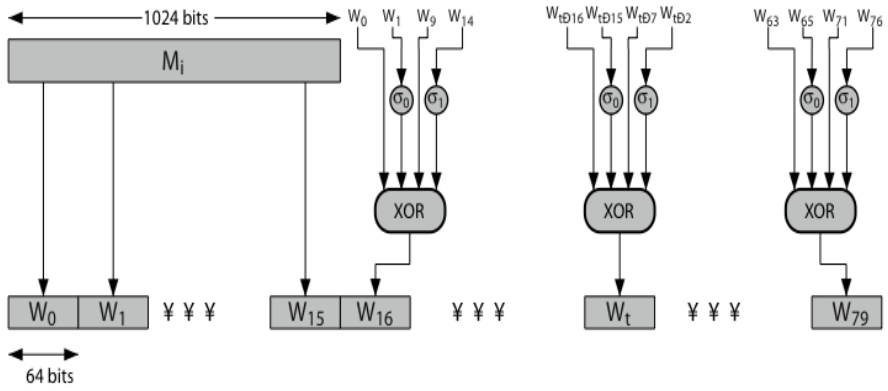


Figure 12.4 Creation of 80-word Input Sequence for SHA-512 Processing of Single Block

Fig. 8.6.6 Creation of 80 word input sequence for SHA-512 Processing of Single Block

8.7 SUMMARY

In this chapter we studied the concept message integrity and message digest, the requirements of cryptographic hash functions, the properties of hash function, and discussed the concept of MAC and HMAC. HMAC is a message digest that involves encryption. Here we also discussed MD5 and SHA-512. MD5 is considered to be vulnerable to attacks now. And SHA-512 is the latest algorithm in SHA family.

8.8 MULTIPLE CHOICE QUSETION ANSWERS

i. Message authentication code is also known as _____

a) key code
b) hash code
c) keyed hash function
d) message key hash function

Ans : c

ii. In SHA-512, the message is divided into blocks of size ___ bits for the hash computation.

a) 1024
b) 512
c) 256
d) 1248

Ans : a

iii. Which attack requires the least effort/computations?

a) Pre-image

b) Second Pre-image

c) Collision

d) All required the same effort

Ans : c

iv. The _____ criterion states that it must be extremely difficult or impossible to create the message if the message digest is given.

a) one-wayness

b) weak-collision-resistance

c) strong-collision-resistance

d) none of the above

Ans : a

v. MD5 produces_____ bits hash data.

a. 112

b. 128

c. 150

d. 160

Ans : b

8.9 TRUE OR FALSE

i. A collision occurs if we have x=y and H(x) = H(y). **--- False**

ii. SHA-512 produces a hash value of 512 bits. ---**True**

iii. A function that is second pre-image resistant is also collision resistant. **---False**

iv. The output of MD5 is 32 hexa digest characters. **----True**

8.10 SAMPLE QUESTIONS

1. What are the requirements of the cryptographic hash functions?
2. Compare and contrast HMAC and CMAC.
3. Explain steps in MD5 algorithm.
4. Explain steps in SHA-512 algorithm

8.11 LIST OF REFERENCES

1. https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm
2. https://medium.com/@zaid960928/cryptography-explaining-sha-512-ad896365a0c1
3. https://sandilands.info/crypto/HashFunctionsandMACs.html#x27-13300016.3
4. Atul Kahate, “Cryptography and Network Security”, McGraw Hill
5. Cryptography and Network Security: Principles and Practice, William Stallings
6. Cryptography and Network Security, Behrouz A Forouzan

❄❄❄❄❄❄❄

9

INTERNET AND WEB SECURITY

Unit Structure

9.0 OBJECTIVES

- Understand the security issues to be solved in the Internet and web
- Gain the knowledge about email security
- Cognize about the working principle and security techniques of web services
- Manage the browser attacks
- Elucidate the data from the website or the client

9.1 AN OVERVIEW

9.1.1 Internet Security and standards

The terms Internet and World Wide Web are often used interchangeably, but they are not exactly the same thing; the internet refers to the global communication system, including hardware and infrastructure, while the Tim Berners-Lee, a British scientist, invented the World Wide Web (WWW) in 1989, while working at CERN. The Web was originally conceived and developed to meet the demand for automated information-sharing between scientists in universities and institutes around the world. web is one of the services communicated over the internet.

Internet Explained

The internet originated with the U.S. government, which began building a computer network in the 1960s known as ARPANET. In 1985, the U.S. National Science Foundation (NSF) commissioned the development of a university network backbone called NSFNET.

The system was replaced by new networks operated by commercial internet service providers in 1995. The internet was brought to the public on a larger scale at around this time. Since then, the Internet has grown and evolved over time to facilitate services like:

- Email, Web-enabled audio/video conferencing services, Online movies and gaming
- Data transfer/file-sharing, often through File Transfer Protocol (FTP), , Instant messaging, Internet forums, Social networking, Online shopping, Financial services

As a global network responsible for vast amounts of data transfer and process facilitation, the Internet is constantly evolving.

It is governed by agencies like the Internet Assigned Numbers Authority (or IANA) that establish universal protocols.

Defining groups like the Internet Engineering Task Force (IETF) and World Wide Web Consortium (W3C) continue to work on standards and universal approaches.

Some of the other authorities about Internet are:

Internet Architecture Board (IAB): A technical advisory group of the ISOC, chartered by the ISOC Trustees to provide oversight of Internet architecture and protocols and, in the context of Internet Standards.

Internet Control Message Protocol (ICMP): An Internet Standard protocol (RFC 792) that is used to reporterror conditions during IP datagram processing and to exchange other information concerning the state of the IP network.

- Internet Security Association and Key Management Protocol (ISAKMP): An Internet IPsec protocol [R2408] to negotiate, establish, modify, and delete security associations, and to exchange key generation and authentication data, independent of the details of any specific key generation technique, key establishment protocol, encryption algorithm, or authentication mechanism.

Computer security: Shall be classified as given below:

- *Data confidentiality***:** Private or confidential information is not made available or disclosed to unauthorized individuals.
- *Privacy:* Individuals control or influence about their information
- *Data integrity:* Information and programs are changed only in a specified and authorized manner.
- *System integrity:* A system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.
- *Availability:* Systems work promptly and service is not denied to authorized users.

9.1.2 Web Services Security

Today web services being deployed are distributed process that process XML (eXtensibleMarkup Language) encoded SOAP (Simple Object Access Protocol) messages sent over HTTP(Hypertext Transfer Protocol) and described using WSDL(Web Services Description Language).

As the web services are loosely coupled software components, they are published, located, and invoked across the web. A web service comprises of several operations like web service creation, publishing, discovering, locating, and passing messages. Each operation in the architecture of web services takes a SOAP package containing a list of input parameters, fulfills a certain task, and returns the result in an output SOAP package. Large enterprises are increasingly relying on web services as methodology for large-scale software development and sharing of services within and outside the organization. There are number of security challenges in this regard and the challenges are overcome by the following techniques.

- *Cryptography:* Protects communications from disclosure or modification by using encryption or digital signatures
- *Authentication* :Using passwords, tokens, public key certificates,or secret keys
- *Authorization*: Rights to use resources
- *Security association: T*o establish trust between client and target components

9.1.3 Challenges of Computer security

- Computer security is not as simple as it appears. The requirements are explicit, techniques are complex.
- In developing a particular security mechanism or algorithm, one must always consider potential attacks on those security features.
- It is necessary to decide where to use the security mechanisms.
- People are very reluctant in investing for security measures.
- Security requires consistent monitoring and enhancement which is difficult in today's rapid changing world.

9.2 INTERNET AND WEB SECURITY

Internet security refers to securing communication over the internet. It includes specific security protocols such as: i) Internet Security Protocol (IPSec) ii)Secure Socket Layer (SSL)

Threats

Internet security threats impact the network, data security and other internet connected systems. Cyber criminals have evolved several techniques to threat privacy and integrity of bank accounts, businesses, and organizations.

Following are some of the internet security threats:

Mobile worms, Malware, PC and Mobile ransomware, Large scale attacks like Stuxnet that attempts to destroy infrastructure.

Hacking as a Service, Spam, Phishing are also the threats for the internet security.

9.2.1 Web Security

Virtually all businesses, most government agencies, scientific research organizations, educational institutions and many individuals now have Web sites. The number of individuals and companies with Internet access is expanding rapidly and all of these have graphical Web browsers. As a result, businesses are enthusiastic about setting up facilities on the Web for electronic commerce. But the reality is that the Internet and the Web are extremely vulnerable to compromises of various sorts. As businesses wake up to this reality, the demand for secure Web services grows.

Web Security Considerations

The Internet is two way. Unlike traditional publishing environments, even electronic publishing systems involving teletext, voice response, or fax-back, the Web is vulnerable to attacks on the Web servers over the Internet.

The Web is increasingly serving as a highly visible outlet for corporate and product information and as the platform for business transactions. Reputations can be damaged and money can be lost if the Web servers are subverted.

The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets.

Although Web browsers are very easy to use, Web servers are relatively easy to configure and manage, and Web content is increasingly easy to develop, the underlying software is extraordinarily complex. This complex software may hide many potential security flaws.

A Web server can be exploited as a launching pad into the corporation's or agency's entire computer complex. Once the Web server is subverted, an attacker may be able to gain access to data and systems not part of the Web itself but connected to the server at the local site

Casual and untrained users are common clients for Web based services. Such users are not necessarily aware of the security risks that exist and do not have the tools or knowledge to take effective countermeasures.

Web Security Threats

Threats are categorized in terms of passive and active attacks.

Passive attacks include eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted.

Active attacks include impersonating another user, altering messages in transit between client and server, and altering information on a Web site.

Another way to classify Web security threats is in terms of the location of the threat: Web server, Web browser, and network traffic between browser and server.

The Threats on the Web is shown in the Table 9.1

	Threats	Consequences	Countermeasures
Integrity	• Modification of user data • Trojan horse browser • Modification of memory • Modification of message traffic in transit	• Loss of information Compromise of machine Vulnerability to all other threats	Cryptographic checksums
Confidentiality	• Eavesdropping on the net • Theft of info from server • Theft of data from client • Info about network configuration • Info about which client talks to server	• Loss of information • Loss of privacy	Encryption, Web proxies
Denial of Service	• Killing of user threads • Flooding machine with bogus requests • Filling up disk or memory • Isolating machine by DNS attacks	• Disruptive • Annoying • Prevent user from getting work done	Difficult to prevent
Authentication	• Impersonation of legitimate users • Data forgery	• Misrepresentation of user • Belief that false information is valid	Cryptographic techniques Integrity

Table 9.1 Threats on the Web

Web Traffic Security Approaches

i) Secure Sockets Layer (SSL) Protocolis an encryption-based Internet security protocol that protects confidentiality and integrity of data.

ii) Transport Layer Security (TLS) Protocol is widely used for the privacy and security of data over the internet. It uses a pseudo-random algorithm to generate the master secret key used for the encryption between the protocol client and protocol server.

iii) Secure Hyper Text Transfer (SHTT) Protocol is designed to secure internet communication such as establishing strong passwords, setting up a firewall etc.

iv) Secure Electronic Transaction (SET) Approach assures the security and integrity of electronic transactions made using credit cards.

9.2.2 SSL -Secure Sockets Layer protocol

SSL was introduced in 1994 and Netscape originated SSL.

Architecture of SSL

SSL is using TCP to provide a reliable end-to-end secure service. SSL is two layers of protocols, as illustrated in Figure 9.1

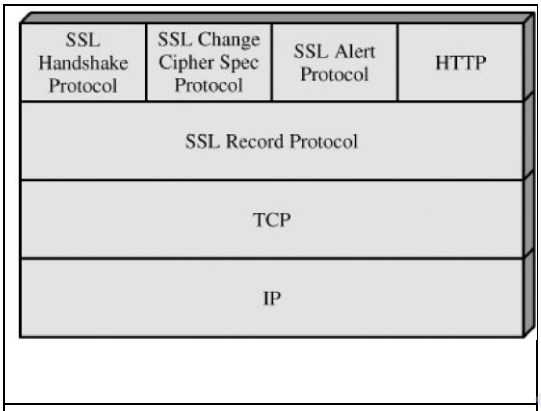


Fig 9.1 SSL Protocol Stack

Two important SSL concepts are:

SSL Session: It is an association between a client and a server. Sessions are created by the Handshake Protocol. Shares set of cryptographic security parameters among multiple connections.

SSL Connection: It is a transport that provides a suitable type of service. Here it is peer-to-peer relationships. The connections are transient. Every connection is associated with one session.

A session state and connection state is defined by the following parameters (Table 9.2)

	Session State		***Connection State***	
S. No.	**Parameter**	**Description**	**Parameter**	**Description**
1	***Session identifier***	An arbitrary byte sequence chosen by the server to identify an active or resumable session state	***Server and client random***	Byte sequences that are chosen by the server and client for each connection.
2	***Peer certificate***	An X509.v3 certificate of the peer	***Server write MAC secret***	Secret key used in MAC operations on data sent by the server
3	***Compression method***	The algorithm used to compress data prior to encryption	***Client write MAC secret***	The secret key used in MAC operations on data sent by the client
4	***Cipher spec***	Specifies the bulk data encryption algorithm like AES and a hash	***Server write key***	Data encrypted by the server and decrypted by the client

		algorithm like MD5 used for MAC calculation		
5	***Master secret***	48-byte secret shared between the client and server	***Client write key***	Data encrypted by the client and decrypted by the server
6	***Is resumable***	A flag indicating whether the session can be used to initiate new connections	***Initialization vectors***	When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key
7	---	---	***Sequence numbers***	Each party maintains separate sequence numbers for transmitted and received messages for each connection

The SSL Record Protocol provides basic security services to higher-layer protocols like i) The Handshake Protocol ii) The Change Cipher Spec Protocol iii) The Alert Protocol

i) The Handshake Protocol

The most complex part of SSL is the Handshake Protocol. This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record. The Handshake Protocol is used before any application data is transmitted.

The Handshake Protocol consists of a series of messages exchanged by client and server.

the format shown in Figure 9.2. Each message has three fields:

1 byte	3 bytes	>=0 bytes
Message Type	Message Length	Content

Fig 9.2 Handshake Protocol

Message Type Parameters:

hello_request, client_hello, server_hello, certificate, server_key_exchange, certificate_request, server_done , certificate_verify , client_key_exchange, finished

ii) The Change Cipher Spec Protocol

The Change Cipher Spec Protocol is one of the three SSL-specific protocols that use the SSL Record Protocol, and it is the simplest. This protocol consists of a single message (Figure 9.3), which consists of a single byte with the value 1. The sole purpose of this

message is to cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection

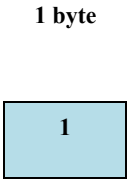


Fig. 9.3 Change Cipher Spec Protocol

iii) Alert Protocol

It is used to convey SSL-related alerts to the peer entity. Each message in this protocol consists of two bytes (Figure 9.4).

The first byte takes the value warning(1) or fatal(2) to convey the severity of the message. Alerts that are always fatal are unexpected_message, bad_record_mac, decompression_failure, handshake_failure, illegal_parameter:

The second byte contains a code that indicates the specific alert. First, we list those

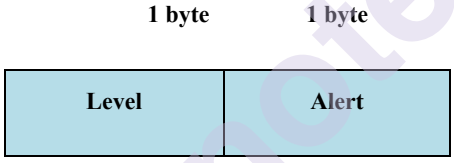


Fig. 9.5 Alert Protocol

9.2.3, Internet Protocol Security (IPSec)

An organization can ensure secure networking by implementing security at the IP level.

Three functional areas of IPSec are authentication, confidentiality, and key management.

- The authentication mechanism assures that a received packet was, transmitted by the party identified as the source in the packet header. It also ensures that the packet has not been altered in transit.
- The confidentiality facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties.
- The key management facility is concerned with the secure exchange of keys.

One way to provide Web security is to use IP security (IPsec) (Figure 9.6). The advantage of using IPsec is that it is transparent to end users and applications and provides a general-purpose solution

Applications of IPSec

Secure branch office connectivity over the Internet: A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.

Secure remote access over the Internet: An end user whose system is equipped with IPsecurity protocols can make a local call to an Internet service provider (ISP) and gain secureaccess to a company network. This reduces the cost of toll charges for traveling employees and telecommuters.

Establishing extranet and intranet connectivity with partners: IPSec can be used tosecure communication with other organizations, ensuring authentication and confidentiality andproviding a key exchange mechanism.

Enhancing electronic commerce security: Even though some Web and electronic commerceapplications have built-in security protocols, the use of IPSec enhances that security.

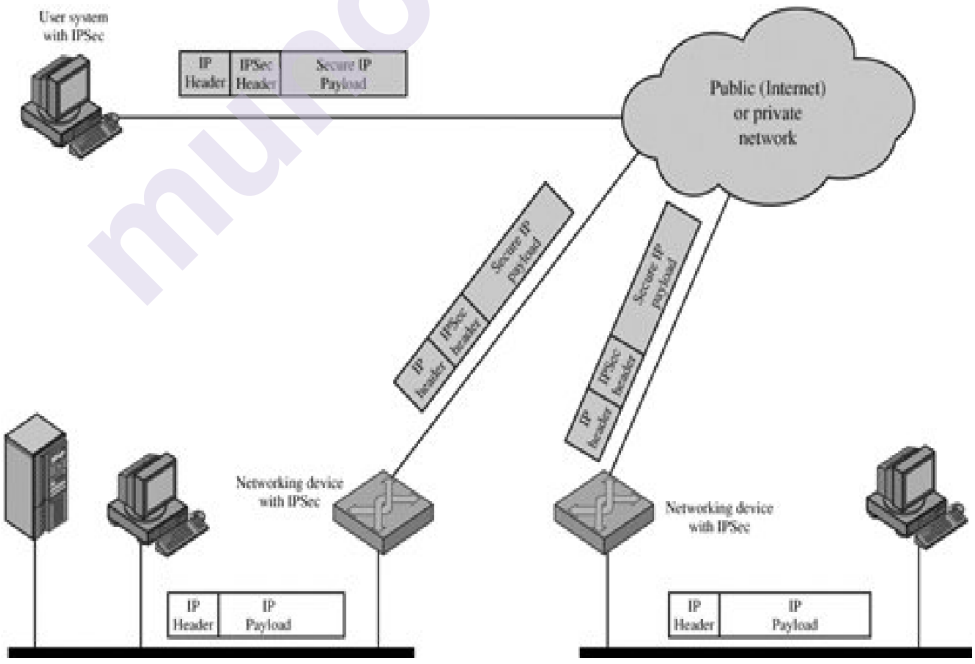


Fig. 9.6 IP Sec Scenario

IP Security Architecture

IPSec Documents: The important documents

- RFC 2401: An overview of a security architecture
- RFC 2402: Description of a packet authentication extension to IPv4 and IPv6
- RFC 2406: Description of a packet encryption extension to IPv4 and IPv6
- RFC 2408: Specification of key management capabilities

Support for these features is mandatory for IPv6 and optional for IPv4.

The extension header for authentication is known as the Authentication header

Header for encryption is known as the Encapsulating Security Payload (ESP) header.

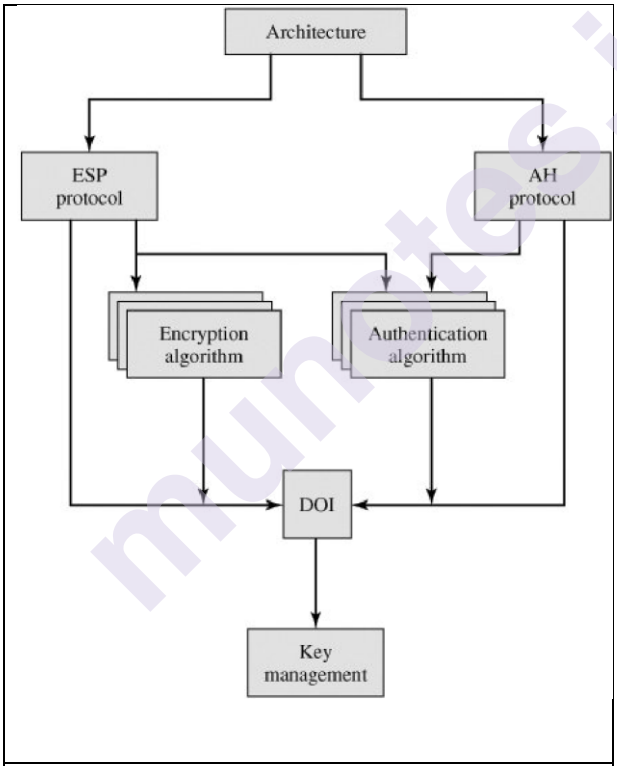


Fig 9.7 IP Sec Document Overview

Additional drafts published by the IP Security Protocol Working Group set up by the IETF (RFC 2401) are: (Fig 9.7)

- Architecture
- Encapsulating Security Payload (ESP)
- Authentication Header (AH)
- Encryption Algorithm
- Authentication Algorithm

- Key Management
- Domain of Interpretation (DOI)

IPSec Services

IPSec provides security services at the IP layer by enabling a system to select required security protocols. The services are

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets (a form of partial sequence integrity)
- Confidentiality (encryption)
- Limited traffic flow confidentiality

Security Associations (SA)

An association is a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it. Security services are afforded to an SA for the use of AH or ESP, but not both.

A security association is uniquely identified by three parameters:

- Security Parameters Index (SPI): A bit string assigned to this SA and having local significance only.
- IP Destination Address: Unicast addresses are allowed; this is the address of the destination end point of the SA.
- Security Protocol Identifier: This indicates whether the association is an AH or ESP security association.

SA Parameters

A security association is defined by the following parameters:

Sequence Number Counter, Sequence Counter Overflow,Anti-Replay Window, AH Information, Encapsulating Security Payload (ESP) Information, Lifetime of This Security Association, IPSec Protocol Mode and Path MTU:

SA Selectors

The means by which IP traffic is related to specific SAs is the nominal Security Policy Database (SPD). SPD contains entries, that defines a subset of IP traffic and points to an SA for that traffic these selectors are used to filter outgoing traffic in order to map it into a particular SA.

Outbound processing obeys the following general sequence for each IP packet:

1. Compare the values of the appropriate fields in the packet against the SPD to find a matching SPD entry, which will point to zero or more SAs.
2. Determine the SA if any for this packet and its associated SPI.
3. Do the required IPSec processing (i.e., AH or ESP processing).

The following selectors determine an SPD entry:

- Destination IP Address:
- Source IP Address:
- UserID:
- Data Sensitivity Level
- Transport Layer Protocol
- Source and Destination Ports:

Transport and Tunnel Modes

Both AH and ESP support two modes of use: transport and tunnel mode.

Transport Mode

Transport mode provides protection primarily for upper-layer protocols.

Tunnel Mode

Tunnel mode provides protection to the entire IP packet. AH or ESP fields are added to the IP packet, the entire packet plus security fields is treated as the payload of new "outer" IP packet with a new outer IP header.

9.2.4. EMAIL SECURITY-

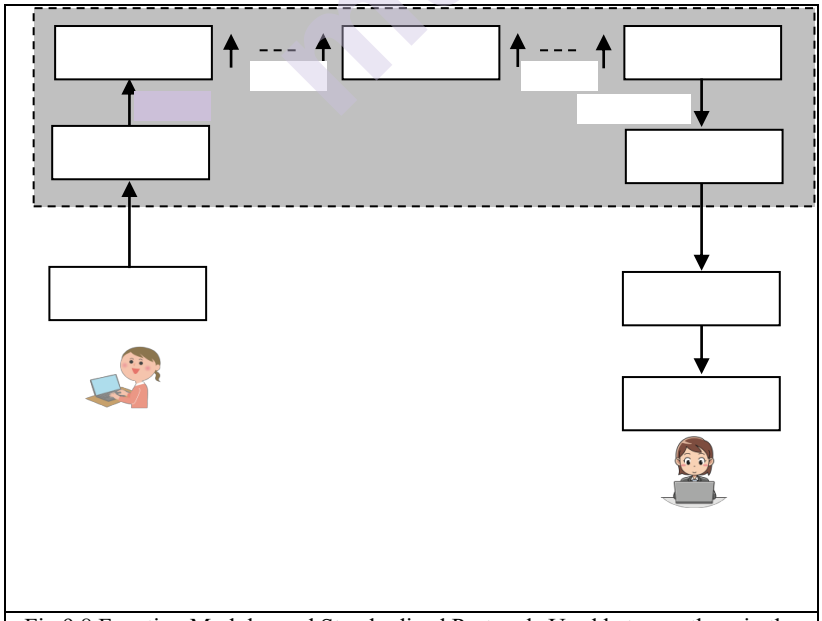


Fig 9.8 Function Modules and Standardized Protocols Used between them in the Internet Mail Architecture

In todays internet world, electronic mail (Email) is the most heavily used network-based application. Users, send email to others who are connected directly or indirectly to the Internet, regardless of host operating system or communications suite. With the explosively growing reliance on email, there grows a demand for authentication and confidentiality services. Two approaches in this regard are i) Pretty Good Privacy (PGP) and ii) S/MIME.

Email Components

Key components of the Internet mail architecture, includes . (Fig 9.8)

- *Message User Agent (MUA):* Operates on behalf of user actors and user applications. It is their representative within the email service.
- *Mail Submission Agent (MSA):* Accepts the message submitted by an MUA and enforces the policies of the hosting domain and the requirements of Internet standards.
- *Message Transfer Agent (MTA):* Relays mail for one application-level hop. An MTA also adds trace information to the message header.
- *SMTP* is used between MTAs and between an MTA and an MSA or MDA.
- *Mail Delivery Agent (MDA):* Responsible for transferring the message from the MHS to the MS.
- *Message Store (MS):* An MUA can employ a long-term MS. An MS can be located on a remote server or on the same machine as the MUA.

Email Protocols

Two types of protocols are used for transferring email.

i) SMTP - Used to move messages through the Internet from source to destination.
ii) IMAP and POP - Used to transfer messages between mail servers

9.2.4.1 Pretty Good Privacy (PGP)

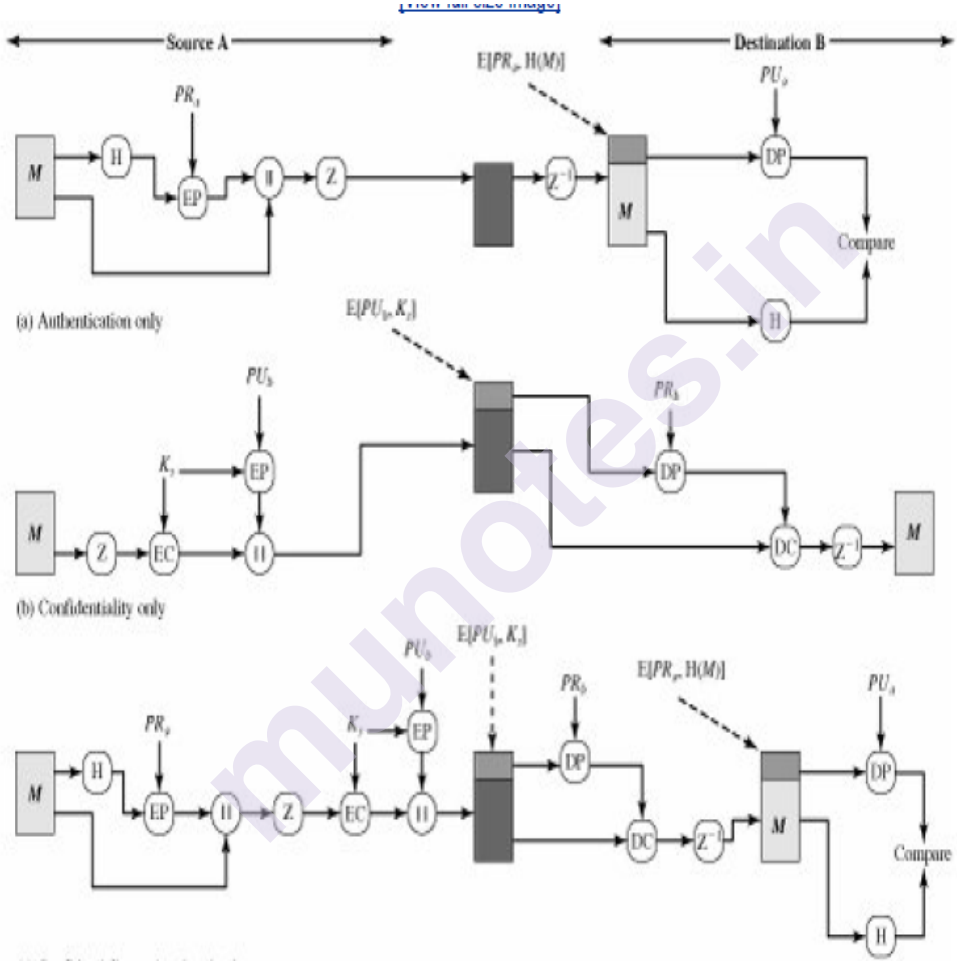
Phil Zimmermann, developed PGP to provide a confidentiality and authentication service that can be used for electronic mail and file storage applications. PGP features:

1. Uses best available cryptographic algorithms
2. Integrated these algorithms into aindependent ,general-purpose application
3. Source code and its documentation, freely available via theInternet,

Notation

Ks =session key

PRa=private key of user A,
PUa=public key of user A
EP = public-key encryption
DP = public-key decryption
EC = symmetric encryption
DC = symmetric decryption
H = hash function



= concatenation

Z = compression using ZIP algorithm
R64 = conversion to radix 64 ASCII format

Operational Description

The actual operation of PGP consists of five services:Authentication, Confidentiality, Compression, E-mail compatibility, and Segmentation (Table 9.3).

Function	Algorithms	Used Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed, for storage or transmission, using ZIP.
Email compatibility	Radix 64 conversion	To provide transparency for email applications, an encrypted message may be converted to an ASCII string using radix 64 conversion.
Segmentation	--	To accommodate maximum message size limitations, PGP performs segmentation and reassembly

Table 9.3 Operational description

Authentication

Figure 9.9 illustrates the digital signature service provided by PGP. The sequence is as follows:

1. The sender creates a message.
2. SHA-1 is used to generate a 160-bit hash code of the message.
3. The hash code is encrypted with RSA using the sender's private key, and the result is prependedto the message.
4. The receiver uses RSA with the sender's public key to decrypt and recover the hash code.
5. The receiver generates a new hash code for the message and compares it with the decryptedhash code. If the two match, the message is accepted as authentic.

Confidentiality

The 64-bit cipher feedback (CFB) mode is used. Each symmetric key is used only once. That is, a new key is generated as a random 128-bit number for each message. To protect the key, it is encrypted with the receiver's public key. Figure 9.10 illustrates the sequence, which can be described as follows:

1. The sender generates a message and a random 128-bit number to be used as a session key fort his message only.
2. The message is encrypted, using CAST-128 (or IDEA or 3DES) with the session key.
3. The session key is encrypted with RSA, using the recipient's public key, and is prepended to the message.
4. The receiver uses RSA with its private key to decrypt and recover the session key.
5. The session key is used to decrypt the message.

Compression

PGP compresses the message after applying the signature but before encryption. This has the benefit of saving space both for e-mail transmission and for file storage.

The placement of the compression algorithm, indicated by Z for compression and Z-1 for decompression in Figure 9.11, is critical.

1. The signature is generated before compression for two reasons:
    a. One can store only the uncompressed message together with the signature for future verification.
    b. PGP algorithm is not deterministic; various implementations of the algorithm achieve different trade-offs in running speed versus compression ratio and, as a result, produce different compressed forms.
2. Message encryption is applied after compression to strengthen cryptographic security. Because the compressed message has less redundancy than the original plaintext, cryptanalysis is more difficult.

E-mail Compatibility

When PGP is used, at least part of the block to be transmitted is encrypted. PGP provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII characters to support the compatability.

Segmentation and Reassembly

E-mail facilities often are restricted to a maximum message length. To overcome the issue, PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail.

Cryptographic Keys and Key Rings

PGP makes use of four types of keys:

i) One-time session symmetric keys
ii) Public keys

iii) Private keys

iv) Passphrase-based symmetric keys Three separate requirements can be identified with respect to these keys:

1. Generating unpredictable session keys is needed.
2. Users have multiple public-key/private-key pairs
3. A file has to be maintained for public/private key pairs correspondents of public keys

We examine each of these requirements in turn.

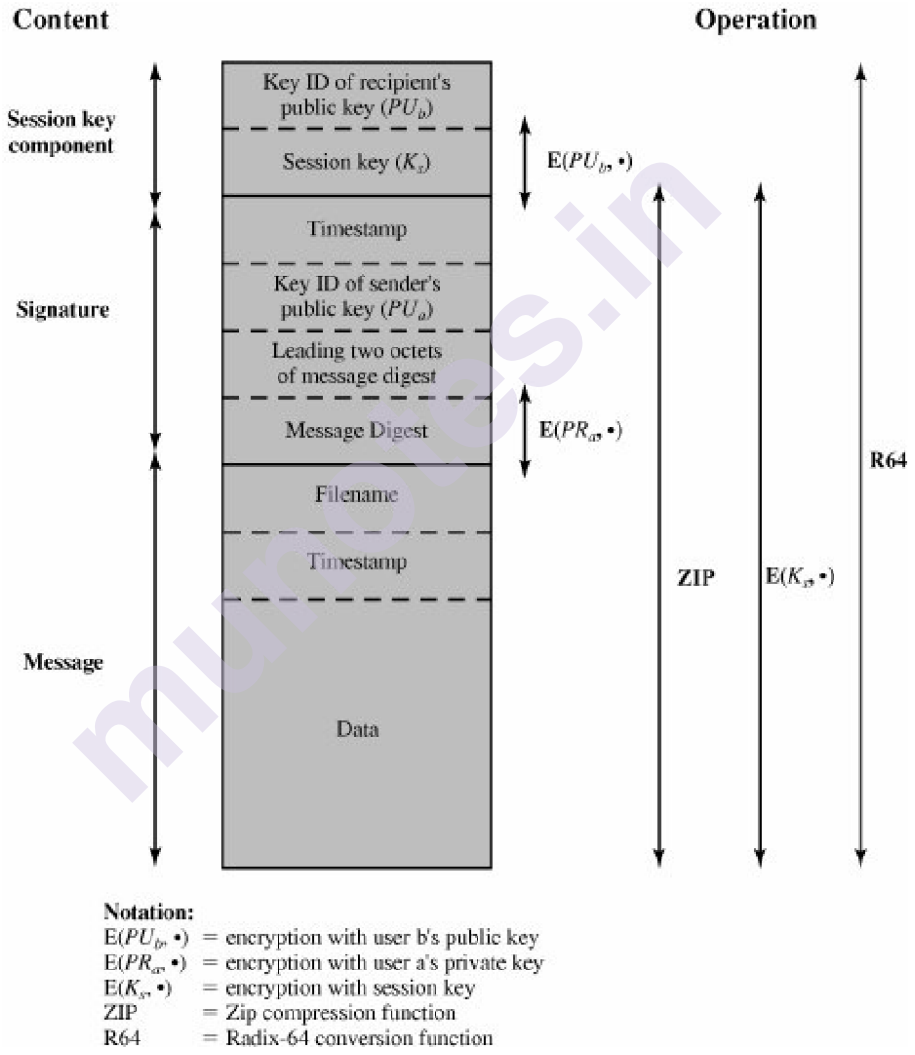


Figure 9.12. General Format of PGP Message (from A to B)

The **message component** includes the actual data to be stored or transmitted, as well as a filename and a timestamp that specifies the time of creation.

The **signature component** includes the following:

Timestamp and Message digest: The digest is calculated over the signature time stamp concatenated with the data portion of the message component.

Leading two octets of message digest: To determine if the correct public key was used to decrypt the message digest for authentication

- **Key ID of sender's public key:** Identifies the public key that should be used to decrypt the message digest

The **session key component** the identifier of the recipient's public key that was used by the sender to encrypt the session key.

Key Rings

Two key IDs of PGP messages are both confidentiality and authentication. These keys need to be stored and organized in a systematic way for efficient and effective use by all parties. The scheme used in PGP is to provide a pair of data structures at each node, one to store the public/private key pairs owned by that node and one to store the public keys of other users known at this node. These data structures are referred to, respectively, as the private-key ring and the public-key ring.

Figure 15.4 shows the general structure of a **private-key ring**. We can view the ring as a table, inwhich each row represents one of the public/private key pairs owned by this user. Each row contains thefollowing entries:

- **Timestamp:** The date/time when this key pair was generated.
- **Key ID:** The least significant 64 bits of the public key for this entry.
- **Public key:** The public-key portion of the pair.
- **Private key:** The private-key portion of the pair; this field is encrypted.
- **User ID: U**ser's e-mail address to choose to associate a different name with each pair or to reuse the same User ID more than once.

Figure 9.13. General Structure of Private- and Public-Key Rings

Private-Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
. . .	. . .	. . .	. . .	. . .
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(P_i), PR_i)$	User i
. . .	. . .	. . .	. . .	. . .

Public-Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
T_i	$PU_i \bmod 2^{64}$	PU_i	$trust\_flag_i$	User i	$trust\_flag_j$		
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .

* = field used to index table

The private-key ring can be indexed by either User ID or Key ID;

Figure 9.13 also shows the general structure of a **public-key ring**.

9.2.4.2 Email Attacks

For both organizations and individuals, email is both pervasive and especially open to a wide range of security threats. In general terms, email security threats can be classified as follows:

- ***Authenticity-related threats:*** Could result in unauthorized access to an enterprise'semail system.
- ***Integrity-related threats:*** Could result in unauthorized modification of emailcontent.
- ***Confidentiality-related threats:*** Could result in unauthorized disclosure ofsensitive information.
- ***Availability-related threats*****:** Could prevent end users from being able to sendor receive email.

Sender Policy Framework (SPF): Uses the Domain Name System (DNS) to allow domain owners to create records that associate the domain name with a specific IP address range of authorized message senders.

DomainKeys Identified Mail (DKIM): Enables an MTA to sign selected headers and the body of a message.

Domain-based Message Authentication, Reporting, and Conformance (DMARC)

9.3 WEB APP VERSUS WEB SERVICE CONCEPT

9.3.1 Web App

A web application (Web App) consists of one or more web pages that are created usually by using a various number of web programming and scripting languages. These pages contain a combination of static and/or dynamic contents including text, images, and code that can be run on servers or web browsers. Users can access these pages using their web browsers. Web applications can reside on servers that have the ability to handle user requests and to provide back the required responses. Besides, they can use multiple servers in the network in order to deliver their functionalities. Generally, users are not aware that the required task requested by them might be distributed across multiple servers

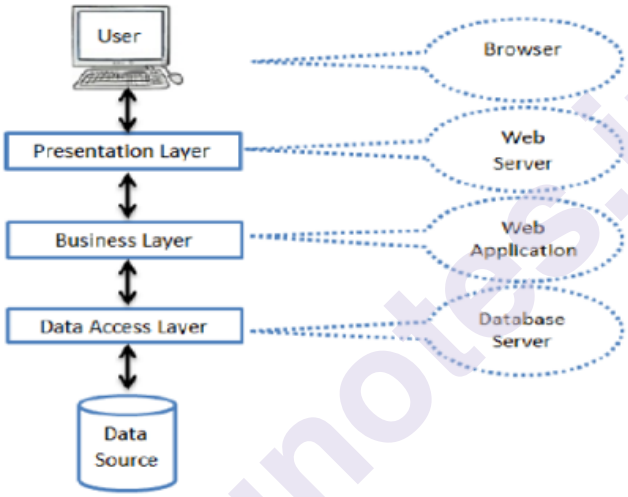
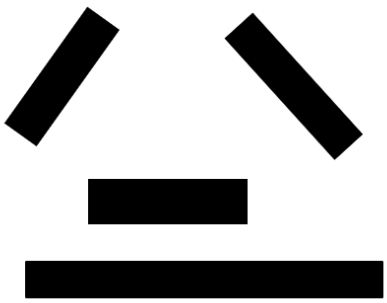


Fig. 1.14 Web App Building Block

9.3.2 Web Service

A web service is a number of independent functional components that allows different machines to interact and collaborate with each other through a network to achieve a common goal. Building Block of Web Services s given in the fig 9.15 .



Here service provider publishes their product / functionality in the registry through Web service Description Language(WSDL). The Consumer of the servicees find the required service from the registry through WSDL. Finally the consumer is bound with the provider through Simple Object accesss Protocol(SOAP).

9.3.3 Web App Vs Web Service

Web applications

- It is Human-oriented
- Accessed by web browsers
- Developed using browser-oriented programming, scripting, and styling languages/frameworks alongside server ones
- Development and Usability is Easy
- User Interface is Provided with Centralized structured view
- Fully Interoperable where as not having integrated infrastructure control
- Users are not involved in updating the application while the complexity is low

Web Services

- It is Machine-oriented
- Accessed by services, applications, and systems
- Developed using standard programming languages
- Development and Usability is comparatively difficult
- User interfaces not available where as having distributed structured view
- Interaction models are available
- Uses SOAP, WSDL, and UDDI to build the blocks
- Both Synchronous and asynchronous operation modes are available
- Having Integrated infrastructure control
- Clients are involved in the updating process
- Reusability is the major advantage of web services
- The complexity level is comparatively high

9.4 WS-SECURITY

The WS-Security initiative defines a single security model that abstracts security services

Unifies formerly the dissimilar security technologies such as PKI and Kerberos.

Operational Steps

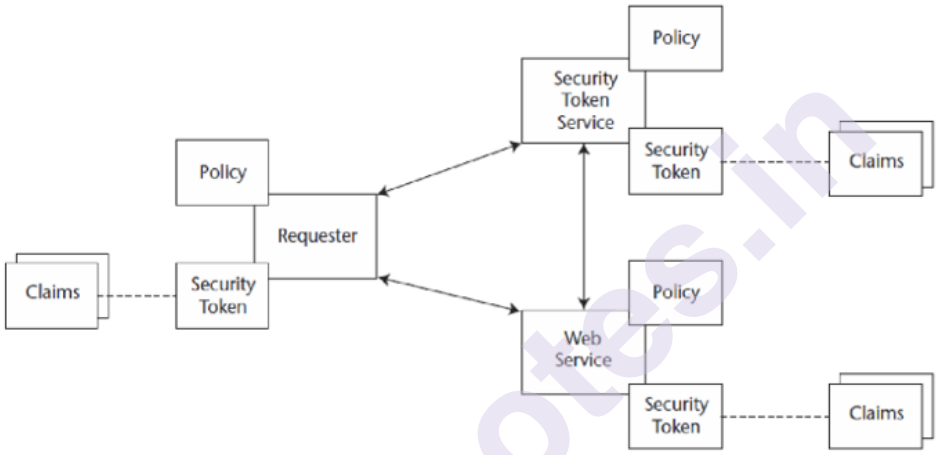
i) In WS-Security the requester requests resources from the Web Service.

ii) The Web Service requires proof of some claims before satisfying the request.

iii) These claims could be an identity or a permission.

iv) If the requester has the needed proof, it will be sent to the Web Service in a security token.

v) If the requester does not have the proof, the service provider will try to get the proof from a security token service, which is also a Web Service.



The joint effort will result in several specifications. The initial specifications are:

WS-Security. How to attach signature and encryption information as well as security tokens to SOAP messages

WS-Policy. How to specify the security requirements and capabilities of Web Services nodes

WS-Trust : How to establish trust in a Web Services environment, either directly or indirectly using a security token service

WS-Privacy: How to specify the privacy policies in place and privacy preferences Additional specifications are:

WS-SecureConversation: How to authenticate the subscriber, the provider, or both **WS-Federation :** How to support federation

WS-Authorization : How to specify and manage access control policies

Functionality

- The WS-Security specification addresses single-message, end-to-endsecurity.
- WS-Security fills in some of the gaps left when XML Signatureand XML Encryption are used with SOAP
- Aclaim is astatement made about a subject by the subject or another party. It can assert an identity,
- A role, or the ownership of a key.
- A security token is a collection of claims

Security Element

- The security element iscontained in the SOAP message header and is targeted at a specific *role*
- The *Security* element contains all claims or other message security information thatis relevant to the *role*. Claims can include the sender's identity
- Other message securityinformation includes *Signature* elements and *EncryptedKey*elements for encryption.
- When *EncryptedKey* is in the *Security* element, it must contain a *ReferenceList*

Structure

The *Security* element can contain several types of subelements. They are *UsernameToken*, *BinarySecurityToken*, *SecurityTokenReference*, *KeyInfo*, *Signature*, *ReferenceList*, and *EncryptedData*

Example

The followinga SOAP message with a WS-Security *Security*header, signature, and encrypted content

<? Xml version='1.0' ?>
<env:Envelopexmlns:env="http://www.w3.org/2002/06/soap-envelope"
xmlns:wsse="http://schmas.xmlsoap.org/ws/2002/04/secext"
xmlns:sig="http://www.w3.org/2002/02/xmldsig#"
xmlns:enc="http://www.w3.org/2002/03/xmlenc#">
<env:Header>
<wsse:Securityenv:role=http://www.w3.org/2002/06/soapenvelope/
role/next env:mustUnderstand="true">
<wsse:BinarySecurityToken
...
</wsse:BinarySecurityToken
<sig:Signature>
...
</sig:Signature>
<enc:EncryptedKey>

```
...
</enc:EncryptedKey>
</wsse:Security>
</env:Header>
<env:Body>
<enc:EncryptedData>
...
</enc:EncryptedData>
</env:Body>
</env:Envelope>
```

9.5 SOAP WEB SERVICE

The major transport and description elements of Web services are SOAP, WSDL and UDDI

A Web service is a software system that supports interoperable machine-to-machine interaction.

- Interaction means that more than one application is involved.
- Interoperable means that applications can operate with oneanother without sharing the same platform, operatingsystem, programming language, etc.

Web services are largely delivered by SOAP(Simple Object Access Protocol), WSDL (Web Services DescriptionLanguage), and UDDI (Universal Description, Discovery, and Interoperability).

These three build on XML (the meta language for the representation)and HTTP, the transport protocol

In overview,

- SOAP defines a uniform way of passing XML-encoded data.
- WSDL allows service providers to specify what a web service can do, where it resides, and how to invoke it.
- UDDI provides a mechanism for clients to dynamically find otherWeb services.

In SOAP messages, the name of the service request and the inputparameters take the form of XML elements.

WSDL describes the data types and structures for Web services, and tells how to map them into the messages that are exchanged.

UDDI provides a repository for Web-services descriptions. An UDDI registry can be searched on various criteria to find all kinds of services offered by businesses.

SOAP is the standard messaging protocol used by Web services. SOAP's primary application is inter application communication. SOAP codifies the use of XML as an encoding scheme for request and response parameters using HTTP as a means for transport.

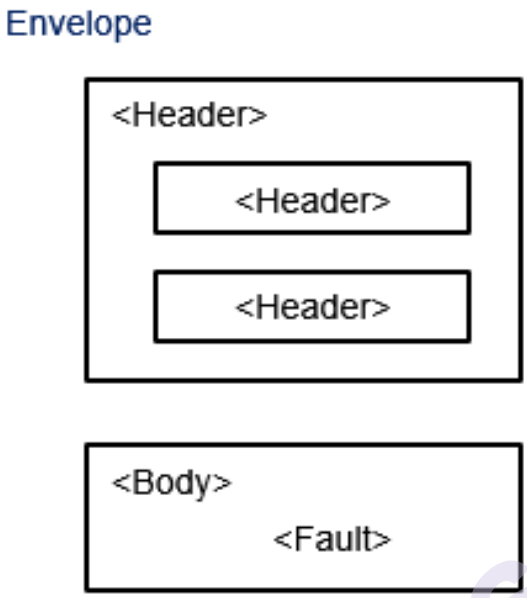


Fig. 9.17 Structure of SOAP Message

SOAP covers the following four main areas:

- A message format for one-way communication describing how a message can be packed into an XML document.
- A description of how a SOAP message should be transported using HTTP (for Web-based interaction) or SMTP (for e-mail-based interaction).
- A set of rules that must be followed when processing a SOAP message and a simple classification of the entities involved in processing a SOAP message.
- A set of conventions on how to turn an RPC call into a SOAP message and back.

WSDL was created to describe the formats and protocols of a Web service in a uniform way.

WSDL elements describe the data and the operations to be performed on it. Both are described in terms of XML schemas.

WSDL is usually used with SOAP.

In order to communicate, both sender and receiver must have access to the same XML schema.

9.6 SAML ASSERTION

9.6.1 What Is SAML?

SAML is a specification (OASIS 2002) that defines a standard way to represent authentication, attribute, and authorization information that may be used in a distributed environment by disparate applications. All the security services that comply with the SAML specification will be able to interpret the security data sent from one security service to another security service.

The heart of the SAML specification is the XML Schema that defines the representation of security data, which can be used as part of a general solution to pass the security context between applications. This representation of security data is an assertion by a trusted third-party security service that the activity of authentication, attribute retrieval, or authorization is correct as represented. For example, the authentication assertion is a representation by a third party that the subject of the assertion, the security principal, has been authenticated. As long as the target trusts this third party, it can accept the assertion as true and can accept the principal named by the authentication assertion as authenticated. , SAML is designed to work with other specifications, such as the digital signature specification (XML Signature Syntax and Processing, W3C, 2002) or the HTTP and SOAP specifications created by the World Wide Web Consortium (W3C).

SAML Containers: These are as assertions about authentication, attributes, and authorization. SAML uses XML as its language. The use of XML makes it fit for the security needs of Web Services.

The SAML assertion divided into two general areas.

i) Common to all SAML assertions Contains items like the version number, the security principal involved in the transaction, some required conditions, and an optional advice field.

ii) Statements

Actuals about authentication, attributes, or authorization.

9.6.2 Common Portion of an Assertion

Subject by SAML: Each assertion has a set of data that is common to all assertions. Which contains the identity of the security principal of this assertion.

The subject can have a domain and a name.

Subject confirmation: An alternate way of identifying the subject and/or as a means by which the target can confirm the authentication of the subject of the assertion

9.6.3 Statements

The top-level statement portion of an assertion is an abstract element.

A valid assertion must contain one of the three statements defined by SAML, authentication, attribute, or authorization.

These will make up the concrete representation of the abstract element Statement Abstract Type.

Abstract statement element can be used as an extension point,

Authentication Statement

The authentication statement is derived from the abstract Subject Statement Abstract Type that, in turn, is derived from an abstract Statement Abstract Type. In the common portion of the assertion we have already assigned to the assertion a particular Subject, stated who the issuer is, and signed the assertion.

<element name=”AuthenticationStatement”
type=”saml:AuthenticationStatementType”/>
<complexType name=”AuthenticationStatementType”>
<complexContent>
<extension base=”saml:SubjectStatementAbstractType”>
<sequence>
<element ref=”saml:SubjectLocality” minOccurs=”0”/>
<element ref=”saml:AuthorityBinding” minOccurs=”0”
maxOccurs=”unbounded”/>
</sequence>
<attribute name=”AuthenticationMethod” type=”anyURI”
use=”required”/>
<attribute name=”AuthenticationInstant” type=”dateTime”
use=”required”/>
</extension>
</complexContent>
</complexType>

Attribute Statement

The attribute statement returns the attributes that the issuer of the assertion asserts are associated with the Subject identified in the common portion of the assertion. The schema definition of an attribute is:

<element name=”AttributeStatement”
type=”saml:AttributeStatementType”/>
<complexType name=”AttributeStatementType”>
<complexContent>
<extension base=”saml:SubjectStatementAbstractType”>
<sequence>
<element ref=”saml:Attribute” maxOccurs=”unbounded”/>

</sequence>
</extension>
</complexContent>
</complexType>

The attribute element contains the AttributeValues as shown below:

<element name=”Attribute” type=”saml:AttributeType”/>
<complexType name=”AttributeType”>
<complexContent>
<extension base=”saml:AttributeDesignatorType”>
<sequence>
<element ref=”saml:AttributeValue” maxOccurs=”unbounded”/>
</sequence>
</extension>
</complexContent>
</complexType>

Authorization Statement

SAML authorization deals with conveying the decision on whether some action or actions may be performed on some resource.

The infrastructure for authorization may be complex, SAML does define a few additional constructs that can be involved in an authorization decision

These are a Policy Enforcement Point (PEP) : Responsible for enforcing the results of an authorization decision.

Poliy Decision Point (PDP): The authorization decision is carried out in this.

In order to satisfy an authorization request the PEP makes a request on the PDP, passing authentication and/or attribute assertions as evidence that the PDP can use to make an authorization decision. A fragment of the authorization statement is presented below:

<element name=”AuthorizationDecisionStatement”
type=”saml:AuthorizationDecisionStatementType”/>
<complexType name=”AuthorizationDecisionStatementType”>
<complexContent>
<extension base=”saml:SubjectStatementAbstractType”>
<sequence>
<element ref=”saml:Action” “maxOccurs =”unbounded”/>
<element ref=”saml:Evidence” minOccurs=”0/>
</sequence>
<attribute name=”Resource” type=”anyURI” use=”required”/>
<attribute name=”Decision”type=”saml:DecisionType “use=”required”/>
</extension>
</complexContent>
</complexType>

9.7 BROWSER ATTACKS

Web Browser Attacks

Web browser attacks are pretty typical of Web-based applications in general.

The attacks can be summarized as follows:

- Hijacking :This is a man-in-the-middle attack in which the attacker takes over the session.
- Replay : This is a man-in-the-middle attack in which sent data is repeated (replayed) leading to various results.
- Spread of malcode (viruses, worms, and so on): The scripting nature of Web browsers makes them prime targets for the spread of malcode.
- Running dangerous executables on the host : In some cases, the browser may permit executables to run on the host workstation. This can be very risky.
- Accessing host files: Certain attacks allow the browser to send files to an attacker. These files may contain personal information, such as banking data, or system information, such as passwords.
- Theft of private information— Browsers are at risk of disclosing sensitive information to strangers on the Internet. This information may be used in identity theft or to conduct a social engineering attack

Hijacking attack

Session hijacking occurs when an HTTP session is observed and captured by a network sniffer.

The attacker modifies the captured traffic to allow the attacker to take the place of the client.

All future traffic in the session is now channeled between the Web server and the attacker.

The hijacking is usually done after the legitimate user has authenticated to the Web server. The fig 9.18 explains that

1. A valid user does some web activity that results in their acquiring a Cookie.
2. The Cookie is stolen or captured by an attacker.
3. The Cookie is transmitted with the attacker's attempt to access the application. The Cookie authenticates the attacker as a valid user. The attacker gets access to the application.

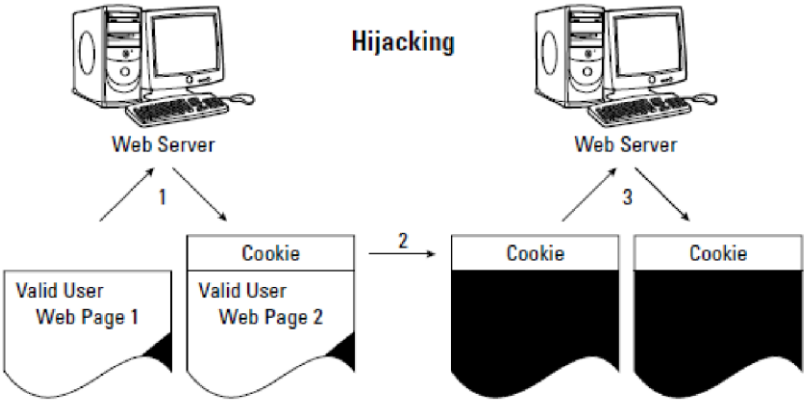


Figure 1.18 Hijacking attempt to exploit this

weak method of maintaining state.

Replay attack

Session replay occurs when an HTTP session is captured by a network sniffer. Some aspect of the session is then modified (certain replays, such as transferring bank funds, may not require modifications). The modified session is then fed back onto the network. If the replay is successful, the Web server will believe the replayed traffic to be legitimate and respond accordingly.

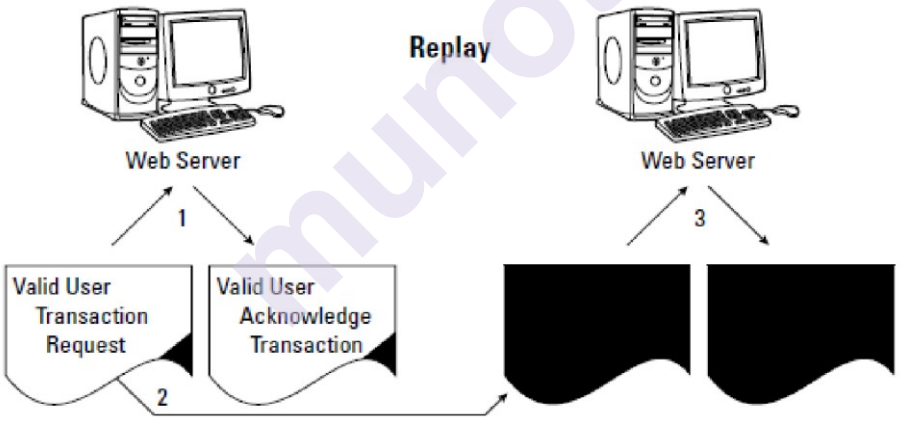


Fig. 1.19 Relay attack

1. A valid user does some web activity such as "Transfer $5,000 from account A to account B". There may or may not be a cookie.
2. The web page holding the transaction request is stolen or captured by an attacker.
3. The web page is re-transmitted. The transaction is repeated - an additional $5,000 is transferred. The attacker can re-transmit numerous times.
4. Depending on whether the attacker had to do spoofing, the final acknowledgment transaction may go back to the valid user's IP address where it is dropped because no session is open.

9.8 WEB ATTACKS TARGETING USERS

- Denial of Service
- Phishing Attack
- Brute-Force
- SQL Injection
- Eavesdropping Attacks
- Birthday Attacks

Denial of Service

- A DDoS uses system's resources so that the system cannot answer service requests.
- ADDoS attack is additionally an attack on system's resources, it is performed from an oversized range of different host machines that are infected by malicious code controlled by the hacker.
- If the attacked resource belongs to a business contender, then the profit to the offender is also real enough.
- Another purpose of a DoS attack may be to require a system offline in order that a distinct reasonably attack may be launched.

Types of DoS and DDoS attacks: transmission control protocol SYN flood attack, teardrop attack, smurf attack, ping-of-death attack and botnets.

Phishing Attack:

Phishing attack is that observe of causing emails that seem to be from sure sources with the goal of gaining personal data or influencing users.

It combines social engineering and technical trickery. It may involve an attachment to an email that masses malware onto the computer.

Brute-Force:

Brute-force positive identification estimation suggests that employing a random approach by attempting totally different passwords and hoping that one work

Some logic may be applied by attempting passwords associated with the person's name, job title, hobbies or similar things.

To repeat an encrypted file that contains the passwords, apply the identical coding to a lexicon of ordinarily used passwords, and compare the results.

SQL Injection:

SQL injection has become a standard issue with database driven websites. It happens once a wrong doer executes a SQL question to the info via the input file from the consumer to server.

SQL commands are inserted into data-plane input so as to run predefined SQL commands.

Eavesdropping Attacks:

Eavesdropping attacks occur through the interception of network traffic.

By eavesdropping, an offender will get passwords, Mastercard numbers and different wind that a user could be causing over the network.

Passive eavesdropping: A hacker detects the data by paying attention to the message transmission within the network

Birthday Attacks:

Birthday attacks are created against hash algorithms to verify the integrity of a message, code or digital signature.

A message processed by a hash perform produces a message digest (MD) of fastened length, freelance of the length of the input message; this MD unambiguously characterizes the message.

9.9 OBTAINING USER OR WEBSITE DATA

Spoofing

IP spoofing is used by an intruder to convince a system that it is communicating with a known, trusted entity to provide the intruder with access to the system.

IP spoofing involves an alteration of a packet at the TCP level, which is used to attack Internet-connected systems that provide various TCP/IP services.

The attacker sends a packet with an IP source address of a known, trusted host instead of its own IP source address to a target host.

The user may accept the packet and act upon it.

Port scanning

A cracker can use scanning software to determine which hosts are active and which are down to avoid wasting time on inactive hosts.

A port scan can gather data about a single host or hosts within a subnet A scan can be implemented using the Ping utility.

After determining which hosts and associated ports are active, the cracker will initiate different types of probes on the active ports.

Examples of probes are Gathering information from the Domain Name System (DNS), Determining the network services that are available, such as e-mail, FTP, and remote logon and Determining the type and release of the operating system

Dumpster diving

Dumpster diving involves the acquisition of information that is discarded by an individual or organization.

Information found in trash is very valuable to a cracker. Because the discarded information may include technical manuals, password lists, telephone numbers, and organization charts.

It is important to note that one requirement for information to be treated as a trade secret is that the information be protected and not revealed to any unauthorized individuals.

9.10 SUMMARY

As a global network responsible for vast amounts of data transfer and process facilitation, the Internet is constantly evolving and thus the security of the data gets vital importance. As the internet protocols and techniques are responsible for the security of information the concepts like SSL and IPSec are discussed.

The mail transfer requires security and privacy inevitably. The protocol PGP regarding email secures the transformation of messages and its contents.

The web applications and web services are used in the business world for data transformation and the security of these are discussed in this chapter.

9.11 REFERENCES

1. Cryptography and Network Security, Behrouz A Forouzan
2. Cryptography and Network Security: Principles and Practice, William Stallings
3. Computer Security :William Stallings , Edition 6
4. Network Security bible, Eric Cole
5. Cryptography And Information Security, V. K. Pachghare
6. Mastering Web Services Security, Bret Hartman Donald J. Flinn Konstantin Beznosov Shirley Kawamoto, Wiley Publishing,2003
7. https://www.geeksforgeeks.org/types-of-internet-security-protocols/
8. http://docs.oasis-open.org/security/saml/v2.0/saml-schema-assertion-2.0.xsd
9. https://www.techopedia.com/definition/2419/internet#what-does-internet-mean
10. https://datatracker.ietf.org/doc/html/rfc4949

9.12 BIBLIOGRAPHY

1. J. Ghayathri J, Dr. S. PannirSelvam , Selection of Paramount Web Service using Sentient QoS Parameters, International Journal of Computer Applications (0975 – 8887) ,National Conference on Research Issues in Image Analysis & Mining Intelligence (NCRIIAMI-2015)

2. PatilShital, RaosahebChavan, Web Browser Security: Different Attacks Detection and Prevention Techniques, International Journal of Computer Applications (0975 – 8887) Volume 170 – No.9, July 2017

3. QusayIdrees Sarhan,Idrees S Gawdan, Web Applications and Web Services: A Comparative Study, Science Journal of University of Zakho · March 2018

9.13 UNIT END EXERCISES

1. What are the standards of internet?
2. Give the architecture of web service
3. Explain the concept of WS Security
4. Discuss the security functions of SSL
5. Explain the functionality of IPSec.
6. Compare Web services with web application
7. Discuss the structure and components of email
8. Elucidate the working principle of PGP
9. Write short note on Internet security
10. List the attacks on web browsers.

❄❄❄❄❄❄❄

10

FIREWALL

Unit Structure

10.0 OBJECTIVES

After this chapter, you should be able to understand the following concepts:

1. List the key characteristic of firewalls.
2. Explain the role of firewalls as part of a computer and network security strategy
3. Understand how to control the interface between private and public network
4. Understand the different kinds of firewall
5. Know about the different types of attacks of firewall
6. Classify different types of configurations
7. Know about its functionality and limitations

10.1 INTRODUCTION

The dramatic rise and progress of the Internet has opened possibilities that no one would have thought of. We can connect any computer in the world to any other computer, no matter how far the two are located from each other. This can be a nightmare for network support staff, which is left with a very difficult job of trying to protect the corporate networks from a variety of attacks.

Most corporations have large amounts of valuable and confidential data in their networks. Leaking of this critical information to competitors can be a great setback.

Internet connectivity is no longer optional for organizations. The information and services available are essential to the organization. Moreover, individual users within the organization want and need Internet access, and if this is not provided via their LAN, they will use dial-up capability from their PC to an Internet service provider (ISP). However, while Internet access provides benefits to the organization, it enables the outside world to reach and interact with local network assets.

This creates a threat to the organization. While it is possible to equip each workstation and server on the premises network with strong security features, such as intrusion protection, this is not a practical approach. Consider a network with hundreds or even thousands of systems, running a mix of various versions of UNIX, plus Windows. When a security flaw is discovered, each potentially affected system must be upgraded to fix that flaw. The alternative, increasingly accepted, is the firewall.

A firewall can be simple a router that is used to filter the packets or a complex multi-computer, multi-router solution that performs filtering of packets along with application-level proxy services. A firewall is essentially a router or a group of routers and computers to enforce access control between two networks.

A firewall can be through of as a pair of mechanisms: allow, which permits traffic and deny, which blocks traffic. There are some firewalls which emphasize on blocking traffic, while others emphasize on permitting traffic.

Apart from the danger of the insider information leaking out, there is a great danger of the outside elements like viruses and worms entering a corporate network to create havoc.

Firewalls are the first line of defence between the internal network and untrusted networks like the Internet. A firewall is a combination of software and hardware used to maintain security of a private network by applying security policies at two or more network boundaries. Firewalls are incorporated into a wide variety of networked devices to filter traffic and lower the risk that malicious packets travelling over the public internet can impact the security of private network. First introduced conceptually in the late 1980s in a whitepaper from Digital Equipment Corporation, “firewalls” provided a then new and important function to the rapidly growing networks of the day.

The design goals include

All traffic from inside to outside a network must be pass through a firewall.

Only authorized traffic will be allowed to pass from a firewall.

The firewall itself must be strong enough, so as to render attacks on it useless.

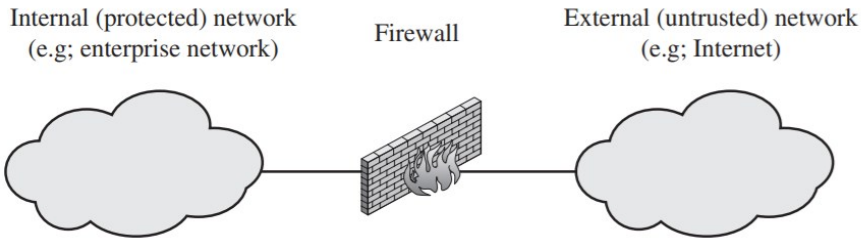


Figure 10.1 Firewall

Firewall design principles

Internet connectivity is no longer an option for most organizations. However, while internet access provides benefits to the organization, it enables the outside world to reach and interact with local network assets. This creates the threat to the organization. While it is possible to equip each workstation and server on the premises network with strong security features, such as intrusion protection, this is not a practical approach. The alternative, increasingly accepted, is the firewall.

The firewall is inserted between the premise network and internet to establish a controlled link and to erect an outer security wall or perimeter. The aim of this perimeter is to protect the premises network from internet-based attacks and to provide a single choke point where security and audit can be imposed. The firewall can be a single computer system or a set of two or more systems that cooperate to perform the firewall function.

10.2 FIREWALL CHARACTERISTICS

The characteristics of firewall are

- Service Control
- Direction Control
- User Control
- Behaviour Control

- **Service Control:** Determines the types of Internet services that can be accessed by the network user. The inbound or outbound traffic may be filtered based on the basis of IP address and TCP port number. It can be implemented by proxy software or host on the server software.
- **Direction control**: Determines the direction such as inbound or outbound in which particular service requests are allowed to flow through the Firewall.
- **User control**: Controls access to a service according to the user. Each user will have a ACLs indicates their level of access. Based on ACL the user traffic may allowed or denied. This feature is typically applied to users inside the private network to control outbound traffic. It may also be applied to incoming traffic from external users, but it needs authentication technique

- **Behavior control**: It makes use of statistical data to control the traffic. Controls how particular services are used (e.g. filter e-mail to eliminate spam), or it may enable external access to only a portion of the information on a local Web server.

10.3 TYPES OF FIREWALLS

There are three types of firewall namely,

1) Packet Filtering
2) Application Gateways
3) Circuit-level Gateways

1) Packet Filtering:

Firewalls filter each packet that attempt to enter or leave a private network and either accept or reject them depending on the predefined set of filter rules which is based on based on pattern-matching. Figure 5.8 shows the placement of packet filtering firewall which is between the public and private network.

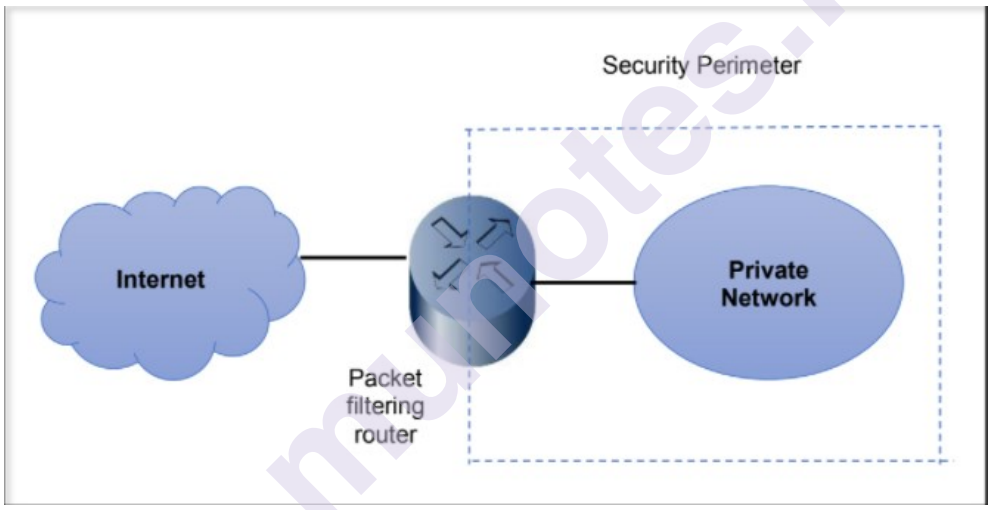


Figure 10.2 Packet Filtering Firewall

A packet filter performs the following functions.

1) Receive each packet as it arrives.
2) Pass the packet through a set of rules, based on the contents of the IP and transport header fields of the packet. If there is a match with one of the set rules, decide whether to accept or discard the packets based on that rule.
3) If there is no match with any rule, take the default action. The default can be discard all packets or accept all packets.

An advanced type of packet filter is called as **stateful packet filter or dynamic packet filter**. This packet filter allows the examination of packets based on the current state of the network. It adapts itself to the current exchange of information, unlike the normal packet filters.

Allow incoming TCP packets only if they are responses to the outgoing TCP packets that have gone through our network. Dynamic packet filter has to maintain a list of the currently open connections and outgoing packets in order to deal with this rule. So, it is called dynamic or stateful.

This type of firewall combines the speed of packet filters with the enhanced security of stored session information typified by proxies. While traffic is being forwarded through the firewall, stateful inspections of the packets create slots in session flow tables.

These tables contain source and destination IP addresses, port numbers, and TCP protocol information. Before traffic can travel back through the firewall, stateful inspections of the packets are cross-referenced to the session flow tables for an existing connection slot. If a match is found in the tables, the packets are forwarded; otherwise, the packets are dropped or rejected.

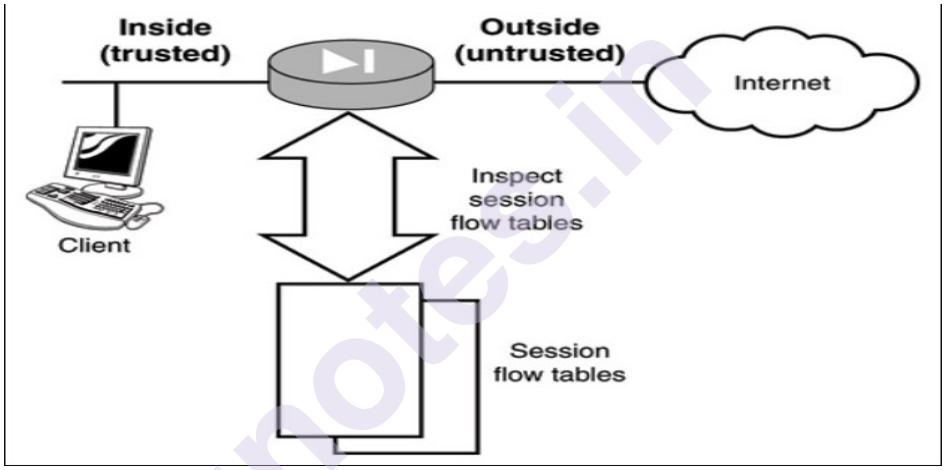


Figure 10.3 Stateful inspection

Stateless Packet Filter:

Stateless packet filtering firewalls are perhaps the oldest and most established firewall option. While they're less common today, they do still provide functionality for residential internet users or service providers who distribute low-power customer-premises equipment (CPE). They protect users against malware, non-application-specific traffic and harmful applications. If users host servers for multi-player video games, email or live-streamed videos, for example, they often must manually configure firewalls if they plan to deviate from default security policies. Manual configurations allow different ports and applications through the packet filter.

Advantages of Packet Filter Firewall:

It is simplicity

Packet filters are very fast in their operating speed.

Packet filter is transparent to the user.

Disadvantages:

Setting up the packet filter rule correctly is difficult.

Most packet filter firewalls do not support advanced user authentication schemes.

They are generally vulnerable to attacks such as layer address spoofing.

Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application-specific vulnerabilities or functions.

Because of the limited information available to the firewall, the logging functionality present in packet filter firewalls is limited.

2) Application Gateway Firewall:

It is also called as proxy server. This is because it acts like a proxy and decides about the flow of application-level traffic. Application gateway work as follows:

1. An internal user contacts the application gateway using a TCP/IP application, such as HTTP or TELNET.
2. The application gateway asks the user about the remote host with which the user wants to set up a connection for actual communication. The application gateway also asks for the user id and password required to access the services of the application gateway.
3. The user provides this information to the application gateway.
4. The application gateway now access the remote host on behalf of the user and passes the packets of the user to the remote host.
5. The application gateway acts like a proxy of the actual end user and delivers packets from the user to the remote host and vice versa.

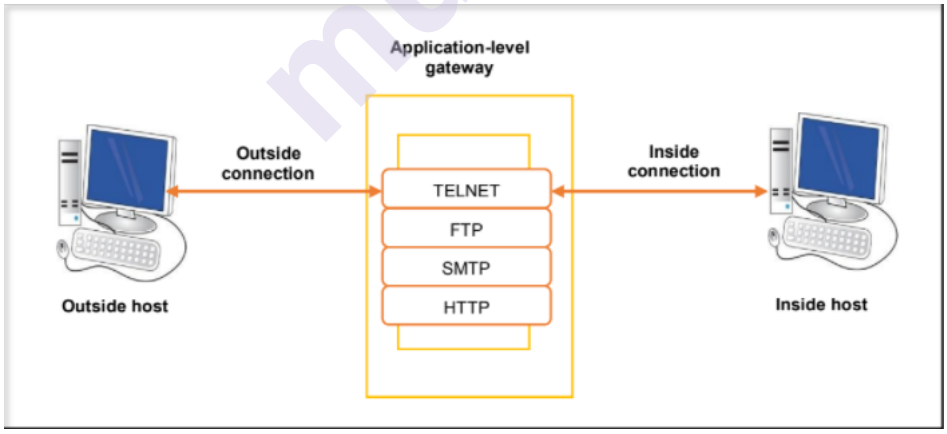


Figure 10.4 Application Gateway Firewall

Advantages:

Application gateway is more secure than packet filter because it examines every packet against a number of rules.

It is easy to log and audit all incoming traffic at the application level

Disadvantage:

Application gateway is the additional processing overhead on each connection-.

3) Circuit level Gateway:

Provides TCP and UDP connection security and works at session layer OSI model. It monitors the TCP data packet handshaking between packets to ensure the session is legitimate and fulfilment of firewall rules and policies.

Circuit level gateway can be a stand-alone system or it can be a specified function performed by an application-level gateway for certain applications. A Circuit level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outer host. Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed.

A typical use of Circuit level gateways is a situation in which the system administrator trusts the internal users. The gateway can be configured to support application level or proxy service on inbound connections and circuit level functions for outbound connections.

Figure 9.5 shows the placement of circuit-level gateway firewall.

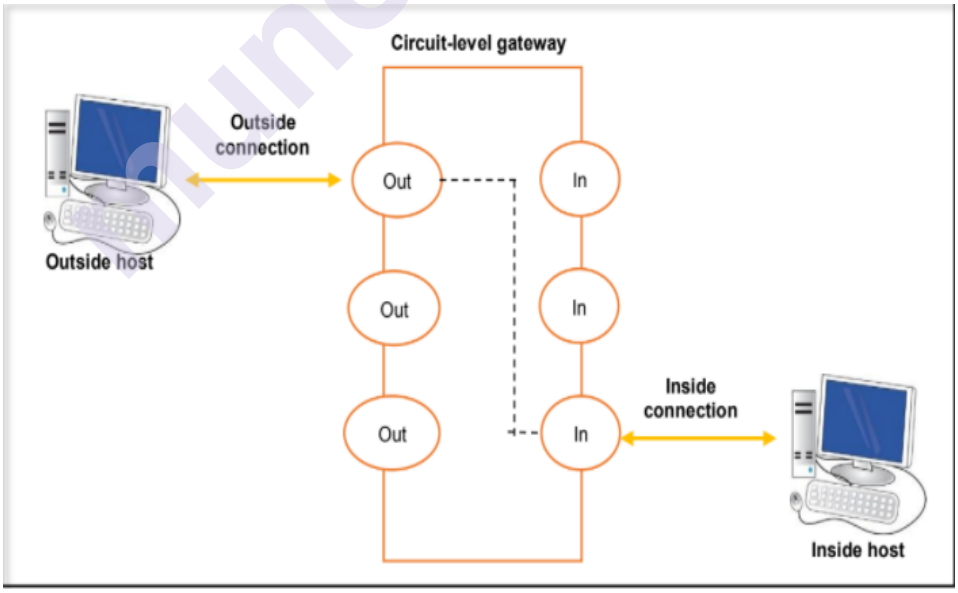


Figure 10.5 Circuit level Gateway

The SOCKS server is an example of the real-life implementation of a circuit gateway. It is client-server application. The SOCKS client runs on the internal hosts and the SOCKS server runs on the firewall.

10.4 ATTACKS OF PACKET FILTER

IP address spoofing: The intruder transmits packets from the outside with a source IP address field containing an address of an internal host. The attacker hopes that the use of a spoofed address will allow penetration of systems that employ simple source address security, in which packets from specific trusted internal hosts are accepted. The countermeasure is to discard packets with an inside source address if the packet arrives on an external interface.

Source routing attacks: The source station specifies the route that a packet should take as it crosses the Internet, in the hopes that this will bypass security measures that do not analyze the source routing information. The countermeasure is to discard all packets that use this option.

Tiny fragment attacks: The intruder uses the IP fragmentation option to create extremely small fragments and force the TCP header information into a separate packet fragment. This attack is designed to circumvent filtering rules that depend on TCP header information. Typically, a packet filter will make a filtering decision on the first fragment of a packet.

All subsequent fragments of that packet are filtered out solely on the basis that they are part of the packet whose first fragment was rejected. The attacker hopes that the filtering router examines only the first fragment and that the remaining fragments are passed through.

A tiny fragment attack can be defeated by enforcing a rule that the first fragment of a packet must contain a predefined minimum amount of the transport header. If the first fragment is rejected, the filter can remember the packet and discard all subsequent fragments.

10.5 BASTION HOST

A bastion host is a system identified by the firewall administrator as a critical strong point in the network's security. Typically, the bastion host serves as a platform for an application-level or circuit level gateway. Common characteristics of a bastion host include the following:

- The bastion host hardware platform executes a secure version of its operating system, making it a trusted system.
- Only the services that the network administrator considers essential are installed on the bastion host. These include proxy applications such as Telnet, DNS, FTP, SMTP, and user authentication.
- The bastion host may require additional authentication before a user is allowed access to the proxy services. In addition, each proxy service may require its own authentication before granting user access.
- Each proxy is configured to support only a subset of the standard application's command set

Each proxy is configured to allow access only to specific host systems. This means that the limited command/feature set may be applied only to a subset of systems on the protected network.

- Each proxy maintains detailed audit information by logging all traffic, each connection, and the duration of each connection. The audit log is an essential tool for discovering and terminating intruder attacks.
- Each proxy module is a very small software package specifically designed for network security. Because of its relative simplicity, it is easier to check such modules for security flaws. For example, a typical UNIX mail application may contain over 20,000 lines of code, while a mail proxy may contain fewer than 1000.
- Each proxy is independent of other proxies on the bastion host. If there is a problem with the operation of any proxy, or if a future vulnerability is discovered, it can be uninstalled without affecting the operation of the other proxy applications. Also, if the user population requires support for a new service, the network administrator can easily install the required proxy on the bastion host.
- A proxy generally performs no disk access other than to read its initial configuration file. This makes it difficult for an intruder to install Trojan horse sniffers or other dangerous files on the bastion host.
- Each proxy runs as a non privileged user in a private and secured directory on the bastion host.

10.6 FIREWALL CONFIGURATION

In practical implementations, a firewall is usually a combination of packet filters and application gateways. Based on this, there are three possible configurations of firewalls as shown in figure.

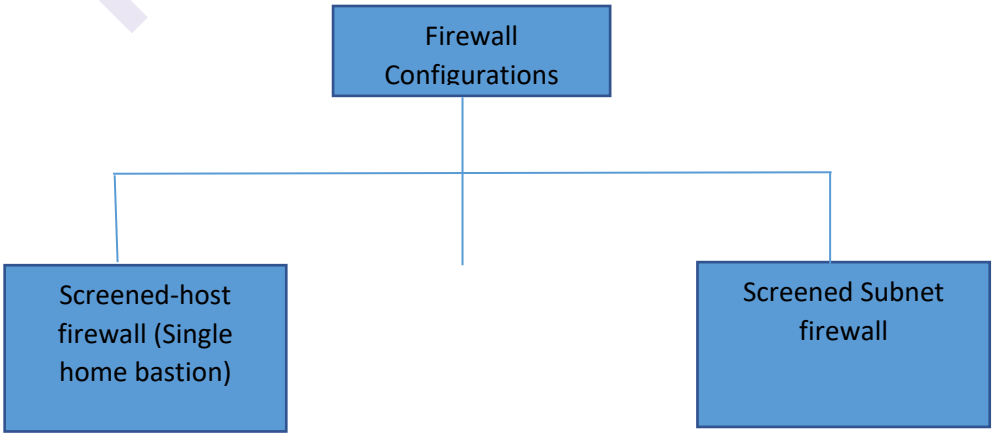


Figure 10.6 Firewall Configuration

Screened Host Firewall, Single Homed Bastion:

In the screened host firewall, a firewall set up consists of two parts: a packet-filtering router and an application gateway.

Their purposes are as follows.

The packet filter ensures that incoming traffic (i.e., form the Internet to the corporate network) is allowed only if it is designed for the application gateway, by examining the destination address field of every incoming IP packet. Similarly, it also ensures that the outgoing traffic is allowed only if it is originating from the application gateway, by examining the source address field of every outgoing IP packet.

The application gateway performs authentication and proxy functions.

The bastion host performs authentication and proxy functions. This configuration has greater security than simply a packet filtering router or an application-level gateway alone, for two reasons:

This configuration implements both packet level and application-level filtering, allowing for considerable flexibility in defining security policy.

An intruder must generally penetrate two separate systems before the security of the internal network is compromised.

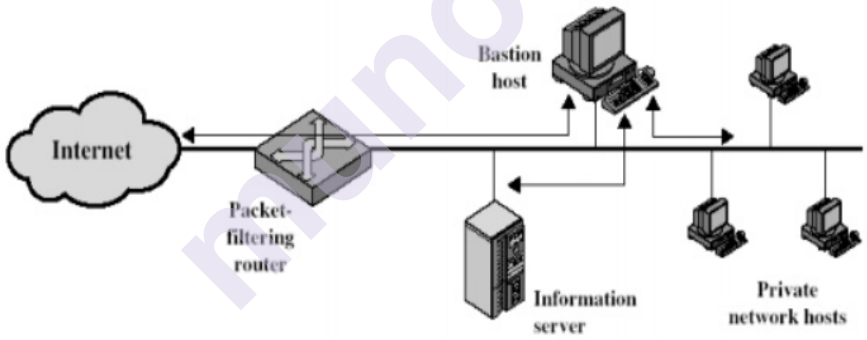


Figure 10.7 Screened-host firewall, Single homed bastion

Screened host firewall, Dual-Homed Bastion:-

To overcome the drawback of a screened host firewall, single-homed bastion configuration another type of configuration, called as screened host firewall, dual-homed bastion. In the previous configuration, if the packet filtering router is compromised, traffic could flow directly through the router between the internet and the other hosts on the private network. This configuration physically prevents such a security break. The internal hosts are protected.

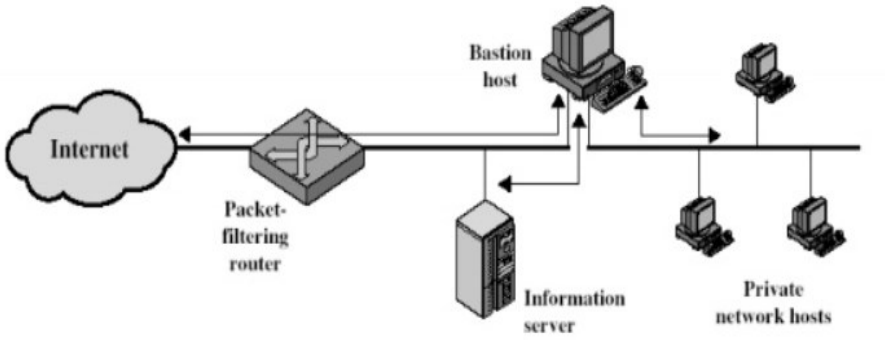


Figure 10.8 Dual-homed bastion

Screened Subnet Firewall:-

It offers the highest security among the possible firewall configurations. It is an improvement over the previous scheme of screened host firewall, dual-homed bastion. Two packet filtering routers are used, one between the bastion host and internet and one between the bastion host and the internal network.

This configuration creates an isolated sub-network, which may consist of simply the bastion host but may also include one or more information servers and modems for dial-in capability. Typically, both the internet and the internal network have access to hosts on the screened subnet, but traffic across the screened subnet is blocked. This configuration offers several advantages:

- There are now three levels of defense to prevent intruders.
- The outside router advertises only the existence of the screened subnet to the internet; therefore, the internal network is invisible to the internet.
- Similarly, the inside router advertises only the existence of the screened subnet to the internal network; therefore, the systems on the internal network cannot construct direct routes to the internet.

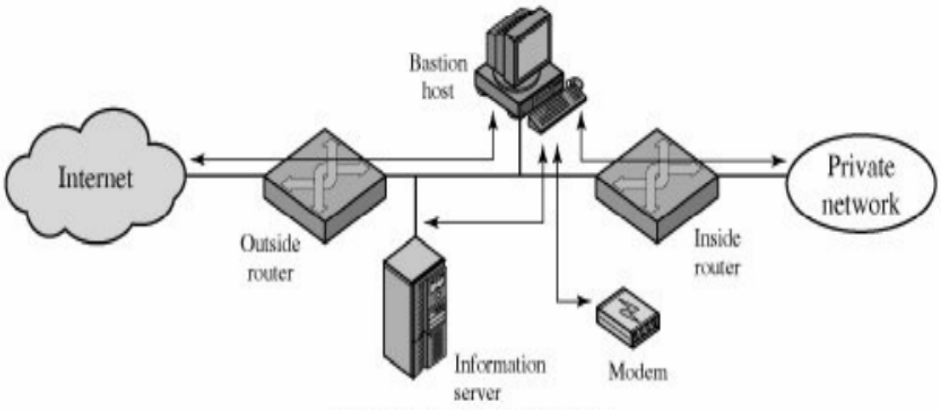


Figure 10.9 Screened subnet firewall

10.7 LIMITATIONS OF FIREWALL

We must note that although a firewall is an extremely useful security measure for an organization, it does not solve all the practical security problems. The main limitations of a firewall can be listed as follows.

1) **Insider's intrusion:**
   As we know, a firewall system is designed to prevent outside attacks. Therefore, if an inside user attacks the internal network in some way, the firewall cannot prevent such an attack. The firewall does not protect against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.

2) **Direct internet traffic:**
   A firewall must be configured very carefully. It is effective only if it is the only entry-exit point of an organization's network. If instead, the firewall is one of the entry-exit points, user can bypass the firewall and exchange information with the internet via the other entry-exit points. This can open up the possibilities of attacks on the internal networks through those points. Such situation can not be handled by firewall.

3) **Virus attacks:**
   A firewall cannot protect internal network from virus threats. Because of the variety of operating systems and applications supported inside the perimeter, it would be impractical and perhaps impossible for the firewall to scan all incoming files, e-mail, and messages for viruses.

10.8 SUMMARY

This chapter provided an in-depth overview of firewalls and their roles in protecting the corporate network and also design principles of firewall. There are three main types of firewalls: packet filters, application gateways, and circuit-level gateways. Corporate networks can be attacked from outside or internal information can be leaked out. Encryption cannot prevent outside attackers from attacking a network. A firewall should be placed between a corporate network and the outside world. A firewall is a special type of router, which applies rules for allowing or stopping traffic.

10.9 BIBLIOGRAPHY

Atul Kahate, "Cryptography and Network Security", McGraw Hill

Behrouz A Forouzan, Cryptography and Network Security

William Stallings, Cryptography and Network Security: Principles and Practice

Mark Rhodes-Ousley, The complete reference Information Security

10.10 EXERCISES

1. What are the limitations of a firewall?
2. List the characteristics of a good firewall implementation.
3. What is an application-level gateway?
4. What is a circuit-level gateway?
5. What is the difference between a packet filtering firewall and a stateful inspection firewall?
6. What are some weaknesses of a packet filtering firewall?
7. What information is used by a typical packet filtering firewall?
8. How is Screened host firewall, Dual-homed bastion different from screened host firewall, Single-homed bastion?
9. How is a circuit gateway different from an application gateway?
10. What is the disadvantage of a screened host firewall, Single-homed bastion?
11. Study at least one real-life firewall product. Study its features with reference to the theory introduced in this chapter.

❄❄❄❄❄❄❄

11

INTRUSION

Unit Structure

11.1 OBJECTIVE

- Understand the concept of IDS & two major categorizations by feature model and by location.
- Understand the behavior of authorized user and intruder
- Understand the approaches for intrusion detection.
- Identifying the attacks in a system or network.
- To analyze the data for possible attacks.

11.2 INTRODUCTION

- Network security is becoming increasingly important in the modern World, where The Internet is an essential part of human life. Due to large extensive use of the Internet, the risks and chances of attacks are also increased. So, it is necessary to protect our system from different attacks.
- The mechanism, which is used to protect our system from different attacks is called intrusion detection systems (IDS).
- The process of recognizing the attacks in a system or computer network is called intrusion detection.
- An intrusion is a deliberate or unauthorized activity or action that attempts to access or manipulate the information or compromise the security of the systems to make them unreliable or unusable.
- An intruder is a kind of a person who is responsible for intrusions. It may be a person from inside the network, i.e, legitimate user of the network or from outside the network.

- In computer networks a firewall is used to prevent such types of attacks on the network. Firewall governs a set of rules and it protects those attacks, which are defined in advance as a rule. So, a firewall does not protect the new attack coming from the attacker, as the rule is not defined.
- In this instance, IDS is useful to detect new attacks. But it is impractical to prevent all the types of attacks. An IDS collects all the information from inside as well as from outside the network and examines this information to identify whether there is intrusion or not.
- Intrusion detection is non-identical from intrusion prevention. Intrusion detection means the process of observing, analyzing & examining the incoming and outgoing traffic and it collects the data. Then, it analyzes the data for possible attacks. Intrusion is the process of identifying the attacks and it attempts to block the detected possible incidents. Intruders are the person who has unauthorized access to the network.

11.3 WHAT IS INTRUSION?

An intrusion is any activity that is designed to compromise your data or information security. This can be through more threatened and prevalent formats like ransomware or unintentional data violation by employees or others attached to computer networks.

An intrusion may include any of the following:

1. Malware or ransomware
2. Unauthorized Access to a system
3. DDOS attacks
4. Cyber-enabled equipment destruction
5. Accidental employee security breaches
6. Untrustworthy users both team members and those outside of your organization
7. Social engineering attacks such as phishing campaigns and other ways of tricking users with seemingly legitimate communication

11.4 INTRUDERS:

- Intruders are the person who has unauthorized access to the network.
- This is one of the two most publicized threats to security is the intruder & the other is viruses, frequently referred to as a hacker or cracker.
- There are different type of intruders:

    1. Masquerader: An individual person who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account. Generally it is an outsider.

2. Misfeasor: A allowable user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such types of access but misuses his or her privileges.
3. Clandestine user: It refers to the user who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection. It may be an insider or outsider.

- The masquerader is mostly from an outsider; the misfeasor generally is an insider; and the clandestine user can be either an outsider or an insider. Intruder attacks range from the warm hearted to the serious. At the benign end of the scale, there are many people who simply wish to explore the internet and see what is out there. At the serious end are individuals or a person who are intercepting to read special rights data, perform unauthorized modifications to data, or disrupt the system. lists the following examples of intrusion:

1. Performing a remote root compromise of an e-mail server
2. Defacing a Web server
3. Guessing and cracking passwords.
4. Copying a database containing credit card numbers.
5. Viewing sensitive data, including payroll records and medical information, without authorization or login.
6. Running a packet sniffer application on a workstation to capture usernames and passwords.
7. Using a permission error on an anonymous FTP server to give out pirated software and music files Dialing into an unsecured modem and gaining internal network access.
8. Present as an executive, calling the help desk, resetting the executive's email password, and learning the new password.
9. Using an unattended, logged-in workstation without permission.

11.5 INTRUSION DETECTION:

- Intrusion detection means to detect or find the vulnerabilities exploited against the computer system or against any application running on the computer system.
- Intrusion detection system helps in providing the information about such vulnerabilities to the network administrator and helps him in preparing some system to protect such attacks or deal with such vulnerabilities.
- It also includes collection of all information by monitoring the network traffic and the suspicious activities in the network.
- It also collects the information about these vulnerabilities from different sources and analyzes the same.

- Many people think that a firewall is Sufficient to protect their network and can recognise the attacks on the network and block the intrusions. But the fact is that the firewall works just like a tene home.
- It restricts the access only to the designated points on the network, but the whole network cannot be secured using a firewall.
- Firewall cannot detect the new au on the network. This detection of new attacks is done by IDS.
- Intrusion detection provides the following functions:

  1. Monitoring and analyzing both the user and the system activities
  2. Analyzing system configurations and vulnerabilities
  3. Assessing the integrity of system and files
  4. Analyzing the traffic pattern based on knowing attack patterns
  5. Analyzing abnormal activity patterns
  6. Tracking the policy violations by the user
  7. Doing audit of the operating system

- The intrusion detection system can be divided into following two types depending on the architecture:

1. Network intrusion detection system (NIDS):

It works on the network and performs an analysis of all the traffic passing on the entire subnet. Every packet is monitored and if the attack is identified or some abnormal behavior is observed, then the alert can be sent to the administrator.

2. Host intrusion detection system (HIDS):

It works off the host, monitors the system events and audits the event logs. It then takes a snapshot of the existing system files and compares it with the previous snapshot available. If any of these files are found modified or deleted, then the alert is sent to the administrator.

- An intrusion-detection system obtains information related to an information system to perform a diagnosis on the security status of the latter. The objective is to discover breaches of security, attempted breaches, or open vulnerabilities that could lead to potential breaches. A classic intrusion-detection system is shown in Figure 1.

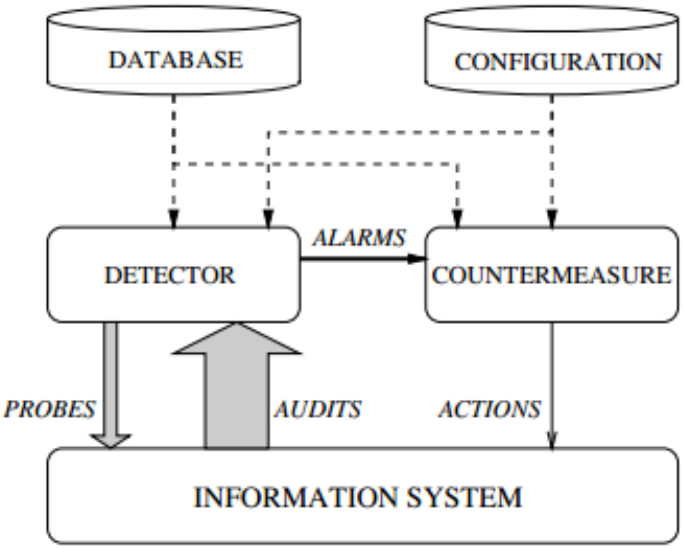


Fig. 1 Simple IDS

Characteristics of intrusion-detection systems:

1. **Performance:** The performance of an intrusion-detection system is the rate at which audit events are processed. If the performance of the intrusion-detection system is weak, then real-time detection is not possible.
2. **Completeness:** Completeness is the property of an intrusion-detection system to detect all attacks. imperfection occurs when the intrusion-detection system fails to detect any types of attack. This measure is much more difficult to evaluate than the others because it is impossible to have a worldwide knowledge about attacks or exploit of privileges. Let see more two additional properties

    - **Fault tolerance:**
        - An intrusion-detection system should be resistant to attacks, especially denial-of-service-type attacks
        - This is important because most intrusion-detection systems run above commercially available operating systems or hardware, which are known to be vulnerable to attacks.
    - **Timeliness:**
        - To perform and propagate its analysis as quickly as possible to enable the security officer to react before much damage has been done, and also to prevent the attacker from subverting the audit source or the intrusion-detection system itself.

- This implies more than the measure of performance because it not only encompasses the intrinsic processing speed of the intrusion-detection system, but also the time required to propagate the information and react to it.

11.6 BEHAVIOR OF AUTHORIZED USER AND INTRUDER

- Inevitably, the best intrusion prevention system will fail. A system's second important is intrusion detection, and this has been the focus of much research in recent trends. This interest is inspired by a number of considerations, including the following:

    1. If an intrusion is detected rapidly & on or before, the intruder can be identified and removed from the system before any damage is done or any data is being compromised. Even if the recognition is not adequately timely to prevent the intruder, the sooner that the intrusion is detected, the less the amount of damage and the more quickly that recovery can be achieved.

    2. Intrusion detection permits the collection of information related to intrusion techniques that can be used to toughen the intrusion prevention facility. Intrusion detection depends on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be determined. Of course, it is not expected that there will be a breakable, exact distinction between an attack by an intruder and the normal use of resources by an authorized user.

- Figure 1 shows a very abstract term, the nature of the task confronting the designer of an intrusion detection system. In spite of the fact that the typical behavior of an intruder differs from the typical behavior of an authorized user, there is an overlap in these behaviors. Consequently, a loose interpretation of intruder behavior, which will snatch more intruders, will also lead to a number of "false positives," or authorized users identified as intruders. In contrast, trying to limit false positives by a tight explanation of intruder behavior will lead to an increase in false negatives, or intruders not identified as intruders. Hence, there is an component of compromise and art in the practice of intrusion detection.

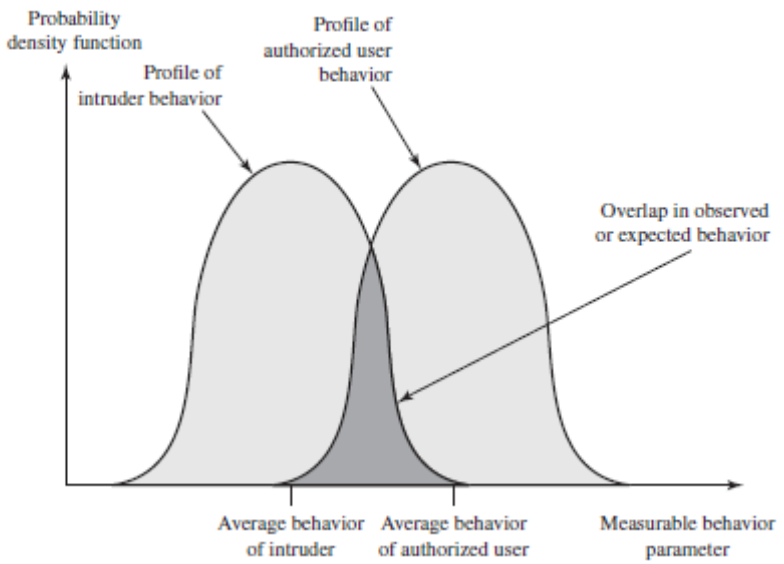


Fig. 2 Behavior of Authorized user and Intruder

11.7 APPROACHES FOR INTRUSION DETECTION: STATISTICAL ANOMALY DETECTION AND RULE BASED DETECTION.

1. Statistical anomaly detection:

- Involves the collection of data relating to the behavior of legitimate users over a period of time. Then statistical tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.

  a. **Threshold detection:** This approach involves defining thresholds, independent of the user, for the frequency of occurrence of various events.

  b. **Mean and standard deviation:** The confidence interval for the abnormality is computed using the comparison of event measures and the mean and standard of a profile deviation.

  c. **Multivariate model:** This model considers computing the correlation between different event measures with respect to the profile expectations.

  d. **Markov process model:** This model considers the types of events with respect to the state variables in a state transition matrix.

  e. **Profile based:** A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.

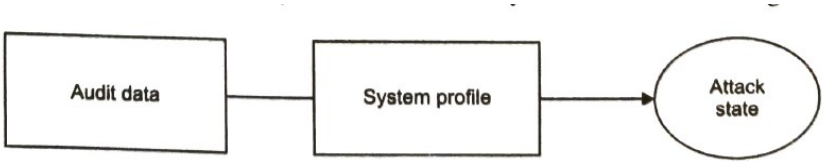


Fig. 2 Anomaly-based detection system

- **Advantage of Anomaly-based detection:**
  - It is possible to detect new or unknown attacks.
  - Accuracy is more.
  - Internal attacks can be detected easily.
- **Disadvantages of anomaly-based detection system**
  - False negatives are more.
  - It is expensive.
  - Accuracy is less.
- **Limitations**
  - The performance of statistics-based IDS depends on the data representation. if it contains some irrelevant data, then IDS may fail to identify an unknown attack.
  - The performance of this approach depends on the threshold. Is it set too low or too high, then it affects false negatives and false positives.

2. Rule-based detection:

- Involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.
  - **a. Anomaly detection:** Rules are developed to detect deviation from previous usage patterns.
  - **b.** Penetration identification: An expert system approach that searches for suspicious behavior.
- In terms of the types of attackers listed earlier, statistical anomaly detection is effective against masqueraders. On the other hand, such techniques may be unable to deal with misfeasors. For Such attacks, rule-based approaches may be able to recognize events and sequences that, in context, reveal penetration.

11.8 AUDIT RECORD AND AUDIT RECORD ANALYSIS

- A fundamental tool for intrusion detection is the audit record. Some record of ongoing activity by users must be maintained as input to an intrusion detection system. Basically, two plans are used:

- Native audit records: Virtually all multiuser operating systems include accounting software that collects information on user activity. The advantage of using this information is that no additional collection software is needed. The disadvantage is that the native audit records may not contain the needed information or may not contain it in a convenient form.
- Detection-specific audit records: A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system. One advantage of such an approach is that it could be made vendor independent and ported to a variety of systems. The disadvantage is the extra overhead involved in having, in effect, two accounting packages running on a machine.

- **Example:**

  A good example of detection-specific audit records is one developed by Dorothy Denning. Each audit record contains the following fields:

  - Subject: Initiators of actions. A subject is typically a terminal user but might also be a process acting on behalf of users or groups of users.All activity arises through commands issued by subjects. Subjects may be grouped into different access classes, and these classes may overlap.
  - Action: Operation performed by the subject on or with an object; for example, login, read, perform I/O, execute.
  - Object: Receptors of actions. Examples include files, programs, messages, records, terminals, printers, and user- or program-created structures. When a subject is the recipient of an action, such as electronic mail, then that subject is considered an object. Objects may be grouped by type. Object granularity may vary by object type and by environment. For example, database actions may be audited for the database as a whole or at the record level.
  - Exception-Condition: Denotes which, if any, exception condition is raised on return.
  - Resource-Usage: A list of quantitative elements in which each element gives the amount used of some resource (e.g., number of lines printed or displayed, number of records read or written, processor time, I/O units used, session elapsed time).
  - Time-Stamp: Unique time-and-date stamp identifying when the action took place. Most user operations are made up of a number of elementary actions.

For example, a file copy involves the execution of the user command, which includes doing access validation and setting up the copy, plus the read from one file, plus the write to another file. Consider the command

COPY GAME.EXE TO <Libray>GAME.EXE

issued by Smith to copy an executable file GAME from the current directory to the directory. The following audit records may be generated: In this case, the copy is aborted because Smith does not have write permission to .

- The decomposition of a user operation into elementary actions has three advantages:

  1. Because objects are the protectable entities in a system, the use of elementary actions enables an audit of all behavior affecting an object. Thus, the system can detect attempted subversions of access controls and can detect successful subversions by noting an abnormality in the set of objects accessible to the subject.
  2. Single-object, single-action audit records simplify the model and the implementation.
  3. Because of the simple, uniform structure of the detection-specific audit records, it may be relatively easy to obtain this information or at least part of it by a straightforward mapping from existing native audit records to the detection-specific audit records.

Smith	execute	<Library>COPY.EXE	0	CPU = 00002	11058721678
Smith	read	<Smith>GAME.EXE	0	RECORDS = 0	11058721679
Smith	execute	<Library>COPY.EXE	write-viol	RECORDS = 0	11058721680

Fig. 3 detection-specific audit records

- In this case, the copy is aborted because Smith does not have write permission to. The decomposition of a user operation into elementary actions has three advantages:

  1. Because objects are the protectable entities in a system, the use of elementary actions enables an audit of all behavior affecting an object. Thus, the system can detect attempted subversions of access controls (by noting an abnormality in the number of exception conditions returned) and can detect successful subversions by noting an abnormality in the set of objects accessible to the subject.
  2. Single-object, single-action audit records simplify the model and the implementation.
  3. Because of the simple, uniform structure of the detection-specific audit records, it may be relatively easy to obtain this information or at least part of it by a straightforward mapping from existing native audit records to the detection-specific audit records.

11.9 SUMMARY

- The System which is used to protect our system from different attacks is called intrusion detection systems (IDS).
- The Process of identifying the attacks in a system or network is called intrusion detection.
- An intrusion is a deliberate or authorized activity or action that attempts to access or manipulate the information or compromise the security of the system to make it unreliable or unusable.
- An intruder is a person who is responsible for intrusion. There are different types of intruders, masquerader, misfeasor, and clandestine user.
- The intrusion detection system can be divided into two types depending on the architecture. network intrusion detection system & Host intrusion detection system.

11.10 REFERENCE FOR FURTHER READING

1. AtulKahate, "Cryptography and Network Security", McGraw Hill
2. Network Security and Cryptography: Bernard Menezes, CENGAGE Learning

11.11 UNIT END EXERCISES

1. List and briefly define three classes of intruders
2. What is the difference between statistical anomaly detection and rule-based intrusion detection?
3. What is intrusion?
4. What are the different types of intruders? Explain in brief.
5. Explain the different categories of intrusion detection systems in brief.
6. Why is intrusion detection system required?
7. Classify the intrusion detection system.
8. Explain anomaly detection.
9. What is misuse-based detection? Explain in brief.
10. Explain model-based intrusion detection.

❄❄❄❄❄❄❄

12

DATABASE AND OS SECURITY

Unit Structure

Self learning topics:

Cryptographic Toolkits, Denial of Service attack

12.0 LEARNING OBJECTIVES

After studying this chapter, you should be able to:

- Understand the unique need for database security, separate from ordinary computer security measures.
- Compare and contrast different approaches to database access control.
- Explain how inference poses a security threat in database systems.

- Understand the Security in operating systems.
- Understand the Security Features of Ordinary Operating Systems.
- Discuss various ways of structuring an operating system.
- Learn the topic rootkit and its examples.

12.1 INTRODUCTION TO DATABASE AND OPERATING SYSTEM

DATABASE: Data is a valuable entity that must have to be firmly handled and managed as with any economic resource. So, some part or all of the commercial data may have tactical importance to their respective organization and hence must have to be kept protected and confidential. In this chapter, you will learn about the scope of database security. There is a range of computer-based controls that are offered as countermeasures to these threats.

- Database is a collection of data and set of rules that organize the data by specifying certain relationships among data.
- Through these rules, the user describes a logical format for the data.
- The user interacts with the data base through a program called a database manager or database management system (DBMS) informally known as a front end.

Advantages of Using Databases

- *Shared Access* – so that many users can use one common, centralized set of data.
- *Minimal Redundancy* – so that individual users do not have to collect and maintain their own sets of data.
- *Data consistency* – so that a change to a data value affects all users of the data value.
- *Data integrity* – so that data values are protected against accidental or malicious incorrect changes.
- *Controlled access* – so that only authorized users allowed to view or modify

OPERATING SYSTEM: Operating system security (OS security) is the process of ensuring OS integrity, confidentiality and availability. OS security refers to specified steps or measures used to protect the OS from threats, viruses, worms, malware or remote hacker intrusions. OS security encompasses all preventive-control techniques, which safeguard any computer assets capable of being stolen, edited or deleted if OS security is compromised.

12.2 WHAT IS DATABASE SECURITY?

Database security refers to the collective measures used to protect and secure a database or database management software from illegitimate use and malicious cyber threats and attacks.

Database security procedures are aimed at protecting not just the data inside the database, but the database management system and all the applications that access it from intrusion, misuse of data, and damage.

It is a broad term that includes a multitude of processes, tools and methodologies that ensure security within a database environment.

Database security must **address** and protect the following:

- The **data** in the database
- The database management system (**DBMS**)
- Any associated **applications**
- The physical database **server** and/or the virtual database server and the underlying hardware
- The computing and/or **networ**k infrastructure used to access the database

12.2.1 SECURITY REQUIREMENTS OF DATABASES

The basic security requirements of database systems are not unlike those of other computing systems. The basic problems access control, exclusion of spurious data, authentication of users, and reliability has appeared in many contexts.

Following is a list of requirements for database security.

- ***Physical database integrity*:** The data of a database are immune from physical problems, such as power failures, and someone can reconstruct the database if it is destroyed through a catastrophe.
- ***Logical database integrity:*** The structure of the database is preserved. With logical integrity of a database, a modification to the value of one field does not affect other fields.
- ***Element integrity*:** The data contained in each element are accurate.
- ***Auditability:*** It is possible to track who or what has accessed (or modified) the elements in the database.
- ***Access control:*** A user is allowed to access only authorized data, and different users can be restricted to different modes of access (such as read or write).
- ***User authentication:*** Every user is positively identified, both for the audit trail and for permission to access certain data.
- ***Availability:*** Users can access the database in general and all the data for which they are authorized.

12.2.2 SENSITIVE DATA

Sensitive data is data that should not made public. **Sensitive data is defined as any information that is protected against unwarranted disclosure.** Protection of data may be required for legal or ethical reasons, for issues pertaining to personal privacy, or for proprietary considerations. Determining which data items are sensitive depends on the individual database and the underlying meaning of the data.

- **Confidential Data** is the most sensitive classification and LSU students, faculty and staff are required by law to protect it. Examples of confidential data include:
  - Social Security Numbers
  - Credit Card Numbers
  - Health Records
  - Financial Records
  - Student Records
- **Private Data** is not considered confidential, but reasonable effort should be made so that it does not become readily available to the public. Examples of private data include:
  - Research Data
  - Personal Contact Data
  - Proprietary information
  - LSU ID (i.e. 89 number)
- **Public Data** is suitable for public consumption and protection of the data is at the discretion of the owner. Examples of public data include:
  - Public budget data
  - Employee contact data
  - Departmental Websites

12.2.3 THREATS TO DATABASES

- **Loss of integrity.** Database integrity refers to the requirement to **protect** information from **incorrect changes**. Integrity is lost if unauthorized changes are made to the data as a result of deliberate or accidental actions. If the loss of system or data integrity is not resolved, further use of the infected system or corrupted data may lead to **inaccuracies**, **fraud or erroneous decisions**.
- **Loss of availability.** Database availability means the accessibility of objects to a user or program that has a **legal right** to these data objects.
- **Loss of confidentiality.** Database confidentiality refers to the protection of data from unauthorized disclosure. Unauthorized, unanticipated, or unintentional disclosure could result in loss of public confidence, embarrassment, or legal action against the organization.

12.3 CONTROL MEASURES

To protect databases from threats, there are general types of control measures:

1. Database Access control
2. Inference control
3. Flow control
4. Encryption

12.3.1 DATABASE ACCESS CONTROL

Database access controls to a database systems is based on the granting and revoking of privileges. A privilege allows a user to create or access (that is read, write or modify) a database object or to execute a DBMS utility. The DBMS keeps track of how these privileges are granted to users and possibly revoked, and ensures that at all times only users with necessary privileges can access an object.

Database access control is a method of allowing access to company's sensitive data only to those people (database users) who are allowed to access such data and to restrict access to unauthorized persons.

It includes two main components: **authentication** and **authorization**.

Authentication is a method of verifying the identity of a person who is accessing your database.

Authorization determines whether a user should be allowed to access the data or make the transaction he's attempting.

Access control systems come in three variations:

1. Discretionary Access Control (DAC),
2. Mandatory Access Control (MAC),
3. Role Based Access Control (RBAC).

1. DISCRETIONARY ACCESS CONTROL (DAC)

Discretionary Access Control is a type of access control system that holds the business owner responsible for deciding **which people** are **allowed** in a specific location, physically or digitally.

DAC is the least restrictive compared to the other systems, as it essentially allows an individual complete control over any objects they own, as well as the programs associated with those objects.

The drawback to Discretionary Access Control is the fact that it gives the end user complete control to set security level settings for other users and the permissions given to the end user are inherited into other programs they use which could potentially lead to malware being executed without the end user being aware of it.

2. MANDATORY ACCESS CONTROL (MAC),

With **mandatory access control**, this security policy is centrally controlled by a security policy **administrator**; users do not have the ability to override the policy and, for example, grant access to files that would otherwise be restricted.

By contrast, discretionary access control (DAC), which also governs the ability of subjects to access objects, allows users the ability to make policy decisions and/or assign security attributes.

MAC-enabled systems allow policy administrators to implement organization-wide security policies. Under MAC (and unlike DAC), users cannot override or modify this policy, either accidentally or intentionally. This allows security administrators to define a central policy that is guaranteed (in principle) to be enforced for all users (ie. **military** institutions).

3. ROLE-BASED ACCESS CONTROL (RBAC)

Role-based access control (RBAC) is a policy-neutral access-control mechanism defined around **roles** and **privileges**. The components of RBAC such as role-permissions, user-role and role-role relationships make it simple to perform user assignments.

RBAC addresses many needs of commercial and government organizations. RBAC can be used to facilitate administration of **security** in **large organizations** with hundreds of users and thousands of permissions. Although RBAC is different from MAC and DAC access control frameworks, it can enforce these policies without any complication.

12.3.2 INFERENCE CONTROL

Inference *is a way to infer or derive sensitive data from non-sensitive data.*

Inference, as it relates to database security, is the process of performing authorized queries and deducing unauthorized information from the legitimate responses received. The inference problem arises when the combination of a number of data items is more sensitive than the individual items, or when a combination of data items can be used to infer data of a higher sensitivity. Figure illustrates the process. The attacker may make use of nonsensitive data as well as metadata.

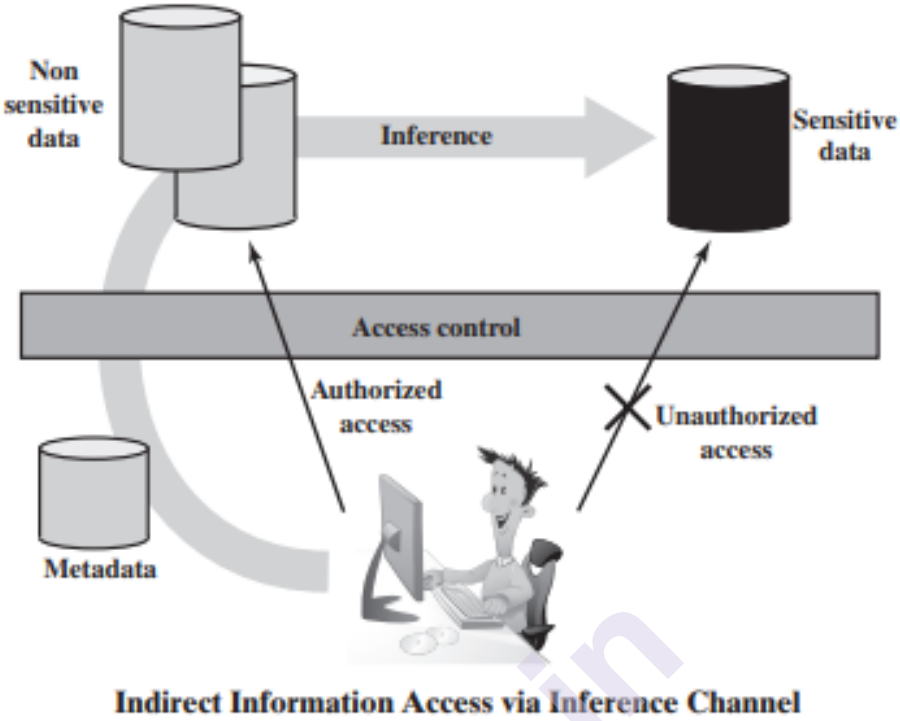


Fig. 12.1 inference channel

Metadata refers to knowledge about correlations or dependencies among data items that can be used to deduce information not otherwise available to a particular user. The information transfer path by which unauthorized data is obtained is referred to as an **inference channel**.

Statistical databases are used to provide statistical information or summaries of values based on various criteria.

For example, a database for population statistics may provide statistics based on age groups, income levels, household size, education levels, and other criteria. Statistical database users such as government statisticians or market research firms are allowed to access the database to retrieve statistical information about a population **but not to access the detailed confidential information about specific individuals**. Security for statistical databases must ensure that information about individuals cannot be accessed.

In general terms, there are two approaches to dealing with the threat of disclosure by inference:

- **Inference detection during database design**: This approach removes an inference channel by altering the database structure or by changing the access control regime to prevent inference. Examples include removing data dependencies by splitting a table into multiple tables or using more fine-grained access control roles in an RBAC scheme. Techniques in this category often result in unnecessarily stricter access controls that reduce availability.

- **Inference detection at query time**: This approach seeks to eliminate an inference channel violation during a query or series of queries. If an inference channel is detected, the query is denied or altered.

12.3.3 FLOW CONTROL

Distributed systems encompass a lot of data flow from one site to another and also within a site. Flow control prevents data from being transferred in such a way that it can be accessed by unauthorized agents. A flow policy lists out the channels through which information can flow. It also defines security classes for data as well as transactions. Prevents information from flowing in such a way that it reaches unauthorized users. Suitable for database over multiuser system or network.

Flow control checks that information contained in some data objects does not flow (explicitly or implicitly) into less protected objects.

- A clearance for a security class can be assigned to each application program.
- Like a DB user, each application program is subjected to the same read/write restrictions.

12.3.4 DATA ENCRYPTION

Suppose we communicate data, but our data falls into the hands of a nonlegitimate user. In this situation, by using encryption we can disguise the message so that even if the transmission is diverted, the message will not be revealed.

Database encryption is the process of converting **data**, within a **database**, in plain text format into a meaningless cipher text by means of a suitable algorithm.

Database decryption is converting the meaningless cipher text into the original information using keys generated by the **encryption** algorithms.

It enhances security and privacy when access controls are bypassed, because in cases of data loss or theft, encrypted data cannot be easily understood by unauthorized persons.

12.4 SECURITY IN OPERATING SYSTEMS

In this chapter we explore the role of the operating system in security. Although operating systems are crucial for implementing separation and access control, they are not invulnerable, and therefore compromise of an operating system can lead to security failure. Furthermore, user's objects can be commingled with code and data for applications and support routines, and operating systems are limited in their ability to separate and protect these resources.

In this chapter a brief overview of operating system designs. We continue by examining aspects of operating system design that enhance security. Finally, we consider rootkits, the most serious compromise of an operating system; with such an exploit the attacker undermines the entire operating system and thus all the security protections it is expected to provide.

The operating system is the fundamental controller of all system resources—which makes it a primary target of attack, as well.

When the operating system initializes at system boot time, it initiates tasks in an orderly sequence, such as, first, primitive functions and device drivers, then process controllers, followed by file and memory management routines and finally, the user interface. To establish security, early tasks establish a firm defense to constrain later tasks. Primitive operating system functions, such as interprocess communication and basic input and output, must precede more complex structures such as files, directories, and memory segments, in part because these primitive functions are necessary to implement the latter constructs, and also because basic communication is necessary so that different operating system functions can communicate with each other. Antivirus applications are usually initiated late because they are add-ons to the operating system; still, antivirus code must be in control before the operating system allows access to new objects that might contain viruses. Clearly, prevention software can protect only if it is active before the malicious code.

But what if the malware embeds itself *in* the operating system, such that it is active before operating system components that might detect or block it? Or what if the malware can circumvent or take over other parts of the operating system? This sequencing leads to an important vulnerability: Gaining control before the protector means that the protector's power is limited. In that case, the attacker has near-complete control of the system: The malicious code is undetectable and unstoppable. Because the malware operates with the privileges of the root of the operating system, it is called a rootkit. Although embedding a rootkit within the operating system is difficult, a successful effort is certainly worth it. We examine rootkits later in this chapter. Before we can study that class of malware, we must first consider the components from which operating systems are composed.

12.4.1 OPERATING SYSTEM STRUCTURE

Every operating system has its own internal structure in terms of file arrangement, memory management, storage management, etc., and the entire performance of the system depends on its structure. The internal structure of operating system provides an idea of how the components of the operating system are interconnected and blended into kernel. This section discusses various system structures that have evolved with time.

Some approaches of Operating System are:

1. Simple Structure
2. Monolithic Systems
3. Layered Systems

4. Microkernels
5. Client-Server Model
6. Virtual Machines
7. Exokernels

1. Simple Structure:

OS started as small, simple, and limited systems and then grew beyond their original scope. MS- DOS is an example of such a system. Early operating systems were developed with an elementary approach without much concern about the structure. In this approach, the structure of the operating systems was not well-defined. The operating systems were monolithic, written as a collection of procedures where each procedure is free to call any other procedure. An example of operating systems designed with this approach is MS-DOS. Initially, MS-DOS was designed as a small-size and simple system, and with limited scope, but grew beyond its scope with time. It was designed with the idea of providing more functionality within less space; therefore, it was not carefully divided into modules. Figure shows the structure of the MS-DOS system.

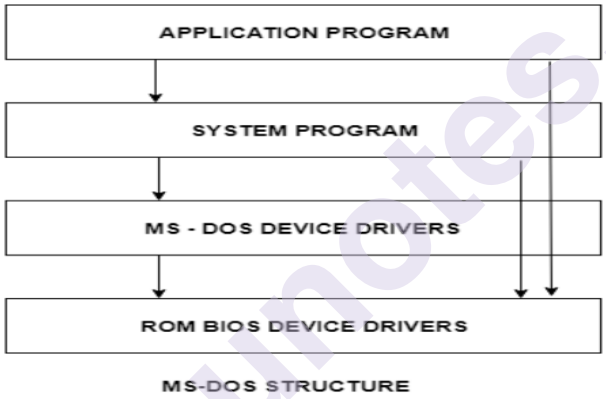


Fig 12.2 MS-DOS Structure

Though MS-DOS has a limited structuring, there is no clear separation between the different interfaces and level of functionality. For example, application programs can directly call the basic I/O routines to read/write data on disk instead of going through a series of interfaces. This exemption makes the MS-DOS system susceptible to malicious programs that may lead to system crash. Moreover, due to the lack of hardware protection and dual-mode operation in the Intel 8088 system (for which MS-DOS system was developed), the base hardware was directly accessible to the application programs.

2. Monolithic System

In the monolithic approach the entire operating system runs as a single program in kernel mode. The operating system is written as a collection of procedures, linked together into a single large executable program. Each procedure in the system is free to call any other process. Being able to call any procedure makes the system very efficient. No information

hiding every procedure is visible to every other procedure. Eg MS DOS and LINUX.

This organization suggests a basic structure for the operating system:

- Main Function- invokes requested service procedure
- Service Procedures- carry out system calls
- Utility functions- Help service procedures to perform certain tasks

Disadvantage:

1. Difficult and complicated structure.
2. A crash in any of these procedures will take down the entire operating system

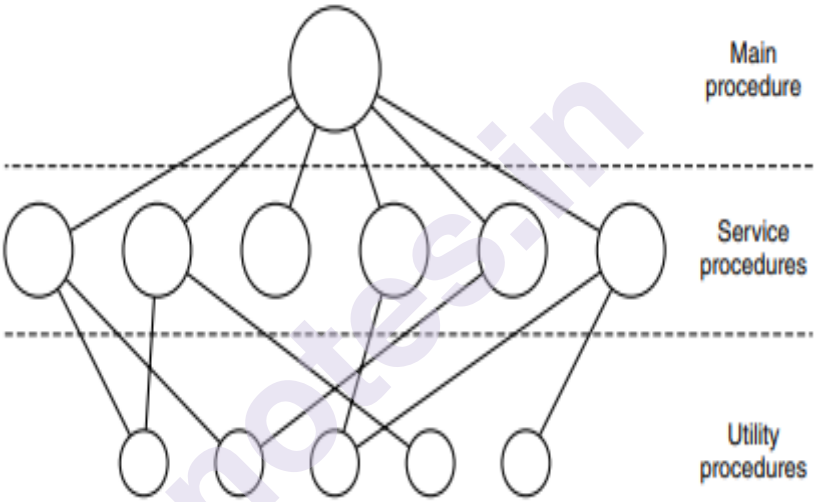


Fig. 12.3 Simple Structuring model for a monolithic system

3. Layered Structure

In the layered approach, the operating system is organized as a hierarchy of layers with each layer built on the top of the layer below it. The topmost layer is the user interface, while the bottommost layer is the hardware. Each layer has a well defined function and comprises data structures and a set of routines. The layers are constructed in such a manner that a typical layer (say, layer n) is able to invoke operations on its lower layers and the operations of layer n can be invoked by its higher layers.

The same concept of layered approach was also implemented by MULTICS with concentric rings. The procedures in out rings are supposed to make a system call to access the process in the inner ring.

The diagram reflects the structure of the operating system with following details

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

Structure of the THE operating system.

Fig. 12.4 Structure of the THE Operating System

Layer 0 dealt with allocation of the processor, switching between processes when interrupts occurred or timers expired.

Layer 1 did the memory management. It allocated space for processes in main memory

Layer 2 handled communication between each process and the operator console

Layer 3 took care of managing the I/O devices and buffering the information streams

Layer 4 was where the user programs were found.

Layer 5 the system operator process was located.

TRAP instruction whose parameters were carefully checked for validity before the call was allowed to proceed.

4. Microkernels

In the mid-1980s, researchers at Carnegie Mellon University developed an operating system called Mach that modularized the kernel using the microkernel approach. This method structures the operating system by removing all nonessential components from the kernel and implementing them as system and user-level programs. The result is a smaller kernel.

Microkernel structure focuses on making the kernel smaller by reducing the non essential components from the kernel. These non essential components are placed in user space. The basic idea behind the microkernel design is to achieve high reliability by splitting the operating system up into small, well-defined modules. The microkernel—runs in kernel mode. The main function of microkernel is to provide a communication facility between the client program and various services that are also running in user space. All new services are added to the user space and the kernel doesn't need to be modified.

Microkernel provides high security and reliability as most of the services are running in user space, if a service fails the rest operating system remains untouched

Disadvantage: Performance decrease due to increased system function overhead.

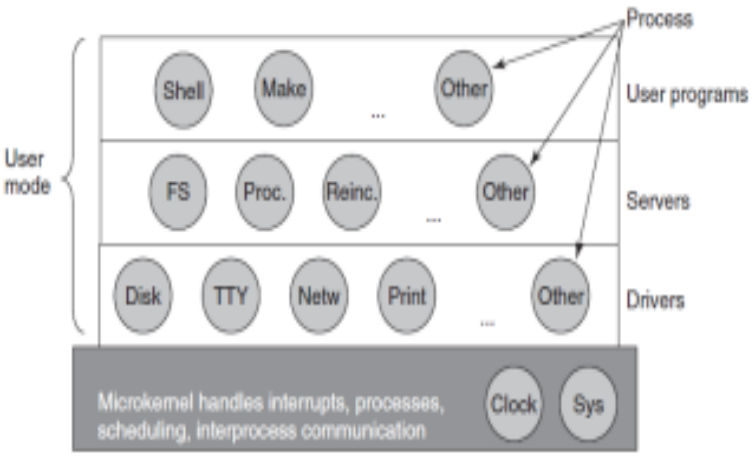


Fig. 12.5 Microkernel structure

5. Modular structure or approach:

It is considered as the best approach for an OS. It involves designing of a modular kernel. The kernel has only set of core components and other services are added as dynamically loadable modules to the kernel either during run time or boot time. It resembles layered structure due to the fact that each kernel has defined and protected interfaces but it is more flexible than the layered structure as a module can call any other module. For example Solaris OS is organized as shown in the figure.

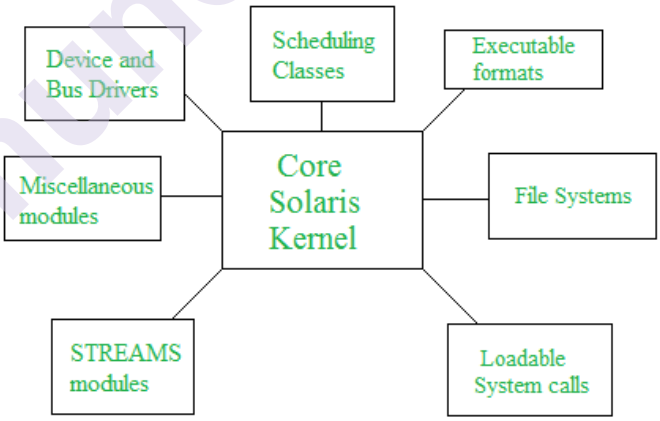


Fig. 12.6 Modular structure

6. Client Server System

The servers, each of which provides some service, and the clients, which use these services. This model is known as the client-server model. Since clients communicate with servers by sending messages, the clients need not know whether the messages are handled locally on their own machines, or whether they are sent across a network to servers on a remote machine. As far as the client is concerned: requests are sent and replies

come back. Thus the client-server model is an abstraction that can be used for a single machine or for a network of machines.

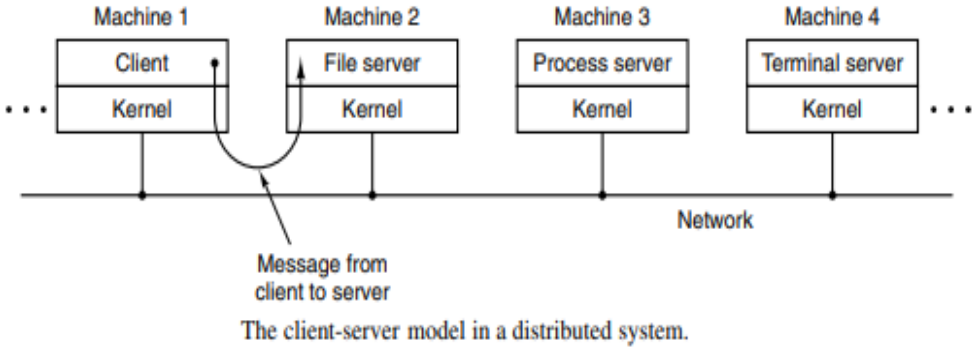


Fig. 12.7 Client Server System

7. Virtual Machines

Virtual machine is nothing but the identical copy of the bare hardware including CPU, disks, I/O devices, interrupts, etc. It allows each user to run operating system or software packages of his choice on a single machine thereby creating an illusion that each user has its own machine.

The virtual machine operating system (VMOS) creates several virtual machines by partitioning the resources of the real machine. The operating system uses the CPU scheduling and virtual memory concept to create an appearance that each running process has its own processor as well own virtual memory (see Figure 1.14). The spooling and file system are used to create illusion of each user having own card reader and line printer.

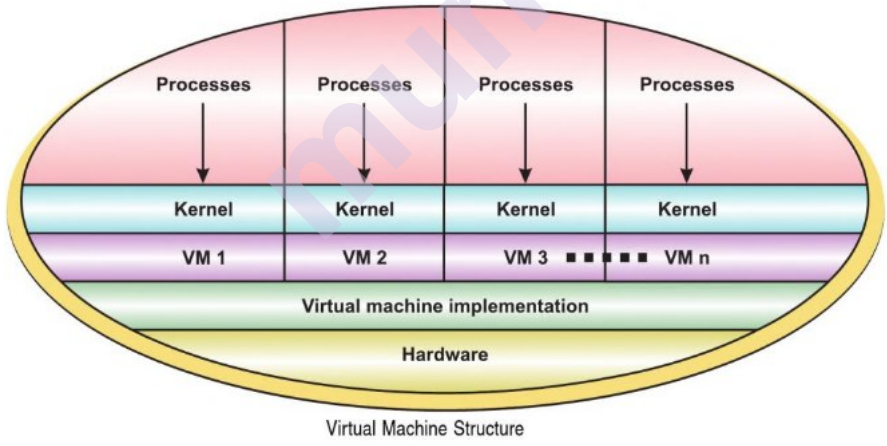


Fig. 12.8 Virtual Machine Structure

The virtual memory approach provides the following benefits: Using virtual machines does not result in an extra overhead and performance degradation as each virtual machine has same architecture as that of a real machine.

Generally, while developing the operating system, the normal functioning of the current system is to be halted. However, by using the virtual machine system, each system programmer can be provided with his own

virtual machine for system development. Thus, there is no need to interrupt the normal system operation. The VMOS keeps the virtual machines isolated from one another. This results in the protection of system resources.

8. Exokernel

Exokernel runs in the bottom layer of kernel mode. Its job is to allocate resources to virtual machines and then check attempts to use them to make sure no machine is trying to use somebody else's resources.

Rather than cloning the actual machine, as is done with virtual machines, another strategy is partitioning it, giving each user a subset of the resources. At the bottom layer, running in kernel mode, is a program called the exokernel. Its job is to allocate resources to virtual machines and then check attempts to use them to make sure no machine is trying to use somebody else's resources. Example: VM/360

The advantage of the Exo-kernel scheme is that it saves a layer of mapping. In the other designs, each virtual machine thinks it has its own disk, with blocks running from 0 to some maximum, so the virtual machine monitor must maintain tables to remap disk addresses (and all other resources).With the exokernel, this remapping is not needed.

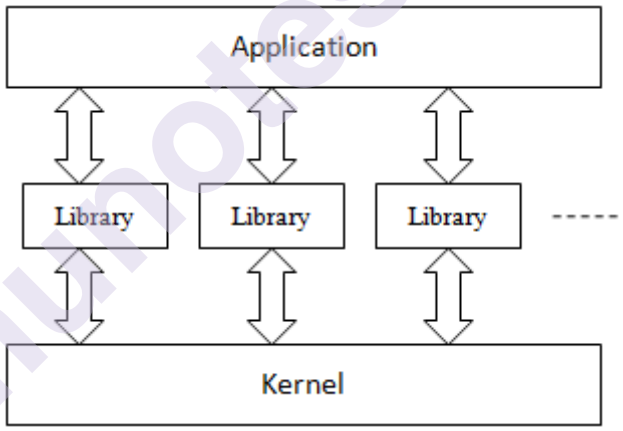


Fig. 12.9 Exokernel Structure

12.4.2 SECURITY FEATURES OF ORDINARY OPERATING SYSTEMS

A multiprogramming operating system performs several functions that relate to security.

To see how, examine Figure which illustrates how an operating system interacts with users, provides services, and allocates resources.

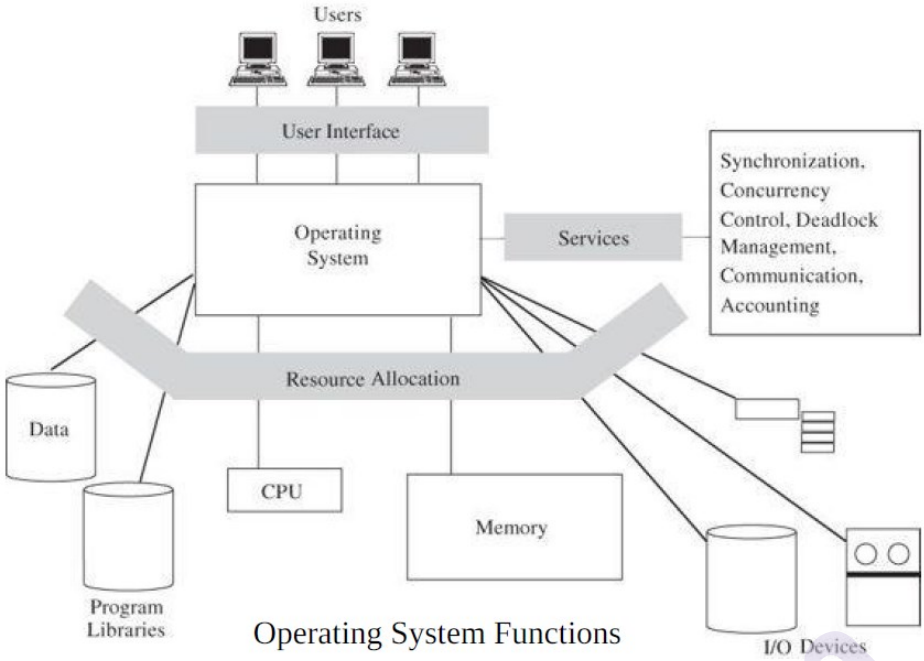


Fig. 12.10 Ordinary Operating System

We can see that the system addresses several particular functions that involve computer security:

- ***Enforced sharing***. Resources should be made available to users as appropriate.

- Sharing brings about the need to guarantee integrity and consistency. Table lookup, combined with integrity controls such as monitors or transaction processors, is often used to support controlled sharing.

- ***Interprocess communication and synchronization***. Executing processes sometimes need to communicate with other processes or to synchronize their accesses to shared resources. Operating systems provide these services by acting as a bridge between processes, responding to process requests for asynchronous communication with other processes or synchronization. Interprocess communication is mediated by access control tables.

- ***Protection of critical operating system data.*** The operating system must maintain data by which it can enforce security. Obviously, if these data are not protected against unauthorized access (read, modify, and delete), the operating system cannot provide enforcement. Various techniques (including encryption, hardware control, and isolation) support protection of operating system security data.

- ***Guaranteed fair service.*** All users expect CPU usage and other service to be provided so that no user is indefinitely starved from receiving service. Hardware clocks combine with scheduling disciplines to provide fairness. Hardware facilities and data tables combine to provide control.

- ***Interface to hardware.*** All users access hardware functionality. Fair access and controlled sharing are hallmarks of multitask operating systems (those running more than one task concurrently), but a more elementary need is that users require access to devices, communications lines, hardware clocks, and processors. Few users access these hardware resources directly, but all users employ such things through programs and utility functions. Hardware interface used to be more tightly bound into an operating system's design; now, however, operating systems are designed to run on a range of hardware platforms, both to maximize the size of the potential market and to position the operating system for hardware design enhancements.

- ***User authentication.*** The operating system must identify each user who requests access and must ascertain that the user is actually who he or she purports to be. The most common authentication mechanism is password comparison.

- ***Memory protection.*** Each user's program must run in a portion of memory protected against unauthorized accesses. The protection will certainly prevent outsiders' accesses, and it may also control a user's own access to restricted parts of the program space. Differential security, such as read, write, and execute, may be applied to parts of a user's memory space. Memory protection is usually performed by hardware mechanisms, such as paging or segmentation.

- ***File and I/O device access control.*** The operating system must protect user and system files from access by unauthorized users. Similarly, I/O device use must be protected. Data protection is usually achieved by table lookup, as with an access control matrix.

- ***Allocation and access control to general objects.*** Users need general objects, such as constructs to permit concurrency and allow synchronization. However, access to these objects must be controlled so that one user does not have a negative effect on other users. Again, table lookup is the common means by which this protection is provided.

You can probably see security implications in many of these primitive operating systems functions. Operating systems show several faces: traffic director, police agent, preschool teacher, umpire, timekeeper, clerk, and housekeeper, to name a few. These fundamental, primitive functions of an operating system are called **kernel** functions, because they are basic to enforcing security as well as the other higher-level operations an operating system provides. Indeed, the operating system kernel, which we describe shortly, is the basic block that supports all higher-level operating system functions.

12.4.3 OPERATING SYSTEM TOOLS TO IMPLEMENT SECURITY FUNCTIONS

In this section we consider how an operating system actually implements the security functions for general objects of unspecified types, such as files, devices, or lists, memory objects, databases, or sharable tables.

Operating systems implement both the underlying tables supporting access control and the mechanisms that check for acceptable uses.

Important operating system function related to the access control function is **audit**: a log of which subject accessed which object when and in what manner. Auditing is a tool for reacting after a security breach, not for preventing one. If critical information is leaked, an audit log may help to determine exactly what information has been compromised and perhaps by whom and when. Such knowledge can help limit the damage of the breach and also help prevent future incidents by illuminating what went wrong this time.

Audit logs show what happened in an incident; analysis of logs can guide prevention of future successful strikes.

An operating system cannot log every action because of the volume of such data. This is too much data impedes timely and critical analysis.

Virtualization

Another important operating system security technique is virtualization, providing the appearance of one set of resources by using different resources. If you present a plate of cookies to a group of children, the cookies will likely all disappear. If you hide the cookies and put them out a few at a time you limit the children's access. Operating systems can do the same thing.

Virtual Machine

Suppose one set of users, call it the A set, is to be allowed to access only A data, and different users, the B set, can access only B data. We can implement this separation easily and reliably with two unconnected machines. But for performance, economic, or efficiency reasons, that approach may not be desirable. If the A and B sets overlap, strict separation is impossible.

Another approach is **virtualization**, in which the operating system presents each user with just the resources that class of user should see. To an A user, the machine, called a **virtual machine**, contains only the A resources. It could seem to the A user as if there is a disk drive, for example, with only the A data. The A user is unable to get to or even know of the existence of B resources, because the A user has no way to formulate a command that would expose those resources, just as if they were on a separate machine.

Virtualization: presenting a user the appearance of a system with only the resources the user is entitled to use.

Virtualization has advantages other than for security. With virtual machines, an operating system can simulate the effect of one device by using another. So, for example, if an installation decides to replace local disk devices with cloud-based storage, neither the users nor their programs need make any change; the operating system virtualizes the disk drives by covertly modifying each disk access command so the new commands retrieve and pass along the right data.

Hypervisor

A **hypervisor**, or **virtual machine monitor**, is the software that implements a virtual machine. It receives all user access requests, directly passes along those that apply to real resources the user is allowed to access, and redirects other requests to the virtualized resources.

Virtualization can apply to operating systems as well as to other resources. Thus, for example, one virtual machine could run the operating system of an earlier, outdated machine. Instead of maintaining compatibility with old operating systems, developers would like people to transition to a new system. However, installations with a large investment in the old system might prefer to make the transition gradually; to be sure the new system works, system managers may choose to run both old and new systems in parallel, so that if the new system fails for any reason, the old system provides uninterrupted use. In fact, for a large enough investment, some installations might prefer to never switch. With a hypervisor to run the old system, all legacy applications and systems work properly on the new system.

A hypervisor can also support two or more operating systems simultaneously. Suppose you are developing an operating system for a new hardware platform; the hardware will not be ready for some time, but when it is available, at the same time you want to have an operating system that can run on it. Alas, you have no machine on which to develop and test your new system. The solution is a virtual machine monitor that simulates the entire effect of the new hardware. It receives system calls from your new operating system and responds just as would the real hardware. Your operating system cannot detect that it is running in a software-controlled environment.

This controlled environment has obvious security advantages: Consider a law firm working on both defense and prosecution of the same case. To install two separate computing networks and computing systems for the two teams is infeasible, especially considering that the teams could legitimately share common resources (access to a library or use of common billing and scheduling functions, for example). Two virtual machines with both separation and overlap support these two sides effectively and securely.

Sandbox

A concept similar to virtualization is the notion of a sandbox. As its name implies, a **sandbox** is a protected environment in which a program can run and not endanger anything else on the system.

Sandbox: an environment from which a process can have only limited, controlled impact on outside resources

The original design of the Java system was based on the sandbox concept, skillfully led by Li Gong. The designers of Java intended the system to run code, called applets, downloaded from untrusted sources such as the Internet. Java trusts locally derived code with full access to sensitive system resources (such as files). It does not, however, trust downloaded remote code; for that code Java provides a sandbox, limited resources that cannot cause negative effects outside the sandbox. The idea behind this design was that web sites could have code execute remotely (on local machines) to display complex content on web browsers.

Java compilers and a tool called a byte code verifier ensure that the system executes only well-formed Java commands. A class loader utility is part of the virtual machine monitor to constrain untrusted applets to the safe sandbox space. Finally, the Java Virtual Machine serves as a reference monitor to mediate all access requests. The Java runtime environment is a kind of virtual machine that presents untrusted applets with an unescapable bounded subset of system resources.

Unfortunately, the original Java design proved too restrictive; people wanted applets to be able to access some resource outside the sandbox. Opening the sandbox became a weak spot, as you can well appreciate. A subsequent release of the Java system allowed signed applets to have access to most other system resources, which became potential and soon actual security vulnerability. Still, the original concept showed the security strength of a sandbox as a virtual machine.

Honeypot

A final example of a virtual machine for security is the honeypot. A **honeypot** is a faux environment intended to lure an attacker. Usually employed in a network, a honeypot shows a limited (safe) set of resources for the attacker; meanwhile, administrators monitor the attacker's activities in real time to learn more about the attacker's objectives, tools, techniques, and weaknesses, and then use this knowledge to defend systems effectively.

Honeypot: system to lure an attacker into an environment that can be both controlled and monitored

Cliff Stoll and Bill Cheswick both employed this form of honeypot to engage with their separate attackers. The attackers were interested in sensitive data, especially to identify vulnerabilities (presumably to exploit

later). In these cases, the researchers engaged with the attacker, supplying real or false results in real time.

As a best security practice, the honeypots aimed at diverting attacks from the Internet should be run on a dedicated physical machine, which is not connected to your real production network or, has a firewall between the two. The honeynet is typically placed in the DMZ or perimeter network. Another approach is to place a honeypot on your internal network to detect attacks that come from insiders.

12.5 ROOTKIT

In the UNIX operating system root is the identity of the most powerful user, owning sensitive system resources such as memory and performing powerful actions such as creating users and killing processes. The identity root is not normally a user with login credentials; instead it is the name of the entity (subject) established to own and run all primitive system tasks (and these tasks create the remaining user identities such as admin and ordinary users). Thus, compromising becoming a task with root privilege is a hacker's ultimate goal because from that position the hacker has complete and unrestricted system control.

Root: most privileged subject (in a UNIX system)

As you have seen, there are two types of attackers: those who craft new attacks and those who merely execute someone else's brainchild. The latter far outnumber the former, but the new attacks are especially troublesome because they are new, and therefore unknown to protection tools and response teams. People who execute attack code from someone else are sometimes pejoratively called "script kiddies" because they simply execute someone else's attack script or package. An attack package that attains root status is called a rootkit. In this section we look at rootkits to see how the power of root can be used to cause serious and hard-to-eradicate harm.

Rootkit: Tool or script that obtains privileges of root

12.5 .1 Phone Rootkit

Researchers at Rutgers University demonstrated an ability to load a rootkit onto a mobile phone. The operating system of a mobile phone is rather simple, although Smartphone with their rich functionality demand a more complex operating system to support a graphical user interface, downloadable applications, and files of associated data. The complexity of the operating system led to more opportunities for attack and, ultimately, a rootkit. Rootkits can exist on any operating system; the Rutgers researchers chose to investigate this platform because it is relatively simple and many users forget or are unaware it is an operating system that can be compromised. The points in this research apply equally to operating systems for more traditional computers.

In one test, the researchers demonstrated a rootkit that could turn on a phone's microphone without the owner's knowing it happened. In such a case, the attacker would send an invisible text message to the infected phone, telling it to place a call and turn on the microphone; imagine the impact of such an attack when the phone's owner is in a meeting on which the attacker wants to eavesdrop.

In another demonstration, these same researchers displayed a rootkit that responds to a text query by relaying the phone's location as furnished by its GPS receiver. This would enable an attacker to track the owner's whereabouts.

In a third test, the researchers showed a rootkit that could turn on power-hungry capabilities such as the Bluetooth radio and GPS receiver to quickly drain the battery. People depend on cell phones for emergencies. Imagine a scenario in which the attacker wants to prevent the victim from calling for help, for example, when the attacker is chasing the victim in a car. If the phone's battery is dead, the cell phone cannot summon help.

The worst part of these three attacks is that they are effectively undetectable: The cell phone's interface seems no different to the user who is unaware of danger. The rootkit can thus perform actions normally reserved for the operating system but does so without the user's knowledge. A rootkit is a variation on the virus theme. A rootkit is a piece of malicious code that goes to great lengths not to be discovered or, if discovered and removed, to reestablish itself whenever possible. The name rootkit refers to the code's attempt to operate as root, the ultra-privileged user of a Unix system, so-named because the most critical and fundamental parts of the Unix operating system are called root functions. Put yourself in the mind of an attacker. If you want persistency, you want an attack that is really difficult to detect so your victim cannot find and try to eradicate your code.

Two conditions can help you remain undiscovered: your code executing before other programs that might block your execution and you're not being detected as a file or process. You can achieve these two goals together. Being in control early in the system boot cycle would allow you to control the other system defenses instead of their controlling you. If your code is introduced early enough, it can override other normal system functions that would detect its presence.

12.5.2 SONY XCP ROOTKIT

A computer security expert named Mark Russinovich developed a rootkit revealer, which he ran on one of his systems. Instead of using a high-level utility program like the file manager to inventory all files, Russinovich wrote code that called the NTQueryDirectoryObject function directly. Summing the file sizes in his program, he compared the directory size against what the file manager reported; a discrepancy led him to look further. He was surprised to find a rootkit. On further investigation he determined the rootkit had been installed when he loaded and played a Sony music CD on his computer.

Princeton University researchers Edward Felten and Alex Halderman extensively examined this rootkit, named XCP (short for extended copy protection).

What XCP Does

The XCP rootkit was installed (automatically and without the user's knowledge) from the Sony music CD to prevent a user from copying the tunes, while allowing the CD to be played as audio. To do this, it includes its own special music player that is allowed to play the CD. But XCP interferes with any other access to the protected music CD by garbling the result any other process would obtain in trying to read from the CD. The rootkit scrambled the result so that it was meaningless as music and passed that uninterruptable result to the calling application.

The rootkit has to install itself when the CD is first inserted in the PC's drive. To do this, XCP depends on a "helpful" feature of Windows: With the "autorun" feature, Windows looks on each newly inserted CD for a file with a specific name, and if it finds that, it opens and executes the file without the user's involvement. (The file name can be configured in Windows, although it is autorun.exe by default.) You can disable the autorun feature. XCP has to hide from the user so that the user cannot just remove or disable it. So the rootkit does as we just described: It blocks display of any program whose name begins with sys (which is how it is named). Unfortunately for Sony, this feature concealed not just XCP but any program beginning with sys from any source, malicious or not. So any virus writer could conceal a virus just by naming it sysvirus-1, for example.

Sony did two things wrong: First, as we just observed, it distributed code that inadvertently opens an unsuspecting user's system to possible infection by other writers of malicious code. Second, Sony installs that code without the user's knowledge, much less consent, and it employs strategies to prevent the code's removal.

Patching the Penetration

Why "penetrate and patch" was abandoned as a security strategy? Among other reasons, the pressure for a quick repair sometimes leads to shortsighted solutions that address the immediate situation and not the underlying cause: Fixing one fault often causes a failure somewhere else. Sony's uninstaller itself opened serious security holes. It was presented as a web page that downloaded and executed the uninstaller. But the programmers did not check what code they were executing, and so the web page would run any code from any source, not just the intended uninstaller. And worse, the code to perform downloads and installations remained on the system even after XCP was uninstalled, meaning that the vulnerability persisted. (In fact, Sony used two different rootkits from two different sources and, remarkably, the uninstallers for both rootkits had this same vulnerability.) How many computers were infected by this rootkit? Nobody knows for sure.

Security researcher Dan Kaminsky found 500,000 references in DNS tables to the site the rootkit contacts, but some of those DNS entries could support accesses by hundreds or thousands of computers. How many users of computers on which the rootkit was installed are aware of it? Again nobody knows, nor does anybody know how many of those installations might not yet have been removed. An interesting analysis of this situation, examining how digital rights management (copy protection for digital media such as music CDs) leads to requirements similar to those for a malicious code developer. The full potential range of rootkit behavior as a way of determining how to defend against them. Automatic software updates, antivirus tools, spyware, even applications all do things without the user's express permission or even knowledge. They also sometimes conspire against the user: Sony worked with major antivirus vendors so its rootkit would not be detected, because keeping the user uninformed was better for all of them, or so Sony and the vendors thought.

12.5.3 TDSS Rootkits

TDSS is the name of a family of rootkits, TDL-1 through (currently) TDL-4, based on the Alureon rootkit, code discovered by Symantec in September 2008. The TDSS group originated in 2008 with TDL-1, a relatively basic rootkit whose main function seemed to be collecting and exfiltrating personal data. TDL-1 seemed to have stealth as its major objective, which it accomplished by several changes to the Windows operating system.

First, it installed filter code in the stack of drivers associated with access to each disk device. These filters drop all references to files whose names begin with "tdl," the file name prefix TDL uses for all its files. With these filters, TDL-1 can install as many files as it requires, anywhere on any disk volume. Furthermore, the filters block direct access to any disk volume, and other filters limit access to network ports, all by installation of malicious drivers, the operating system routines that handle communication with devices. The Windows registry, the database of critical system information, is loaded with entries to cause these malicious drivers to reload on every system startup.

The TDL-1 rootkit hides these registry values by modifying the system function NTEnumerateKey, used to list data items (keys) in the registry. The modification replaces the first few bytes of the system function with a jump instruction to transfer to the rootkit function, which skips over any rootkit keys before returning control to the normal system function. Modifying code by inserting a jump to an extension is called splicing, and a driver infected this way is said to have been hooked. Splicing: a technique allowing third-party code to be invoked to service interrupts and device driver calls.

In early 2009, the second version, TDL-2 appeared. Functionality and operation were similar to those of TDL-1, the principal difference being that the code itself was obscured by scrambling, encrypted, and padded with nonsense data such as words from Hamlet. Later that year, the TDSS developers unleashed TDL-3. Becoming even more sophisticated, TDL-3 implemented its own file system so that it could be completely independent of the regular Windows functions for managing files using FAT (file allocation table) or NTFS (NT file system) technology [DRW09]. The rootkit hooked to a convenient driver, typically atapi.sys, the driver for IDE hard disk drives, although it could also hook to the kernel, according to Microsoft's Johnson. At this point, TDSS developers introduced command-and-control servers with which the rootkit communicates to receive work assignments and to return data collected or other results.

TDL-3 also began to communicate by using an encrypted communications stream, effectively preventing analysts from interpreting the data stream. All these changes made the TDSS family increasingly difficult to detect. Network World estimated that in 2009, 3 million computers were controlled by TDSS, more than half of which were located in the United States. These controlled computers are sold or rented for various tasks, such as sending spam, stealing data, or defrauding users with fake antivirus tools. But TDL-3 is not the end of the line.

A fourth generation, TDL-4, appeared in autumn 2010. This version circumvented the latest Microsoft security techniques. TDL-4 follows the path of other TDSS rootkits by hooking system drivers to install itself and remain undetected. But during this time, Microsoft's 64-bit Windows software implemented a cryptographic technique by which a portion of each driver is encrypted, using a digital signature. Basically, Microsoft's digital signatures let it verify the source and integrity of kernel-level code each time the code is to be loaded (ordinarily at system boot time). TDL-4 changes a system configuration value Load Integrity Check Policy so that the unsigned rootkit is loaded without checking. TDL-4 infects the master boot record (MBR) and replaces the kernel debugger (kdcom.dll) that would ordinarily be available to debug kernel-level activity. The replaced debugger returns only safe values (meaning those that do not reveal TDL-4), making it difficult for analysts to investigate the form and function of this rootkit. The sophistication of the TDSS family is amazing, as is its ability to adapt to system changes such as code integrity checking.

12.6 LET US SUM UP

This chapter covered the Security related to database, Database access control and inference. We studied the Security in operating systems. Structure of the operating system has evolved with time. Most common ones includes monolithic, layered, microkernel etc.

In this chapter we studied how an operating system actually implements the security functions for general objects of unspecified types, such as files, devices, or lists, memory objects, databases, or sharable tables. A multiprogramming operating system performs several functions that relate to security. For best security practices, the operating systems and applications running on VMs should be secured in the same way you would secure them on individual physical machines. Virtualization should be only one of many tools in your security arsenal. Rootkits to see how the power of root can be used to cause serious and hard-to-eradicate harm.

12.7 LIST OF REFERENCES

Reference Books:	**Reference Name**
1	Security in Computing fifth edition Charles P. Pfleeger Shari Lawrence Pfleeger Jonathan Margulies
Reference No	**Reference Name**
1	https://link.springer.com/content/pdf/10.1007%2F978-1-4302-6383-8_16.pdf
2	docs.oracle.com/cd/B19306_01/server.102/b14220/security.htm
3	https://www.w3.org/Security/security-resource
4	https://www.sophos.com/en-us/labs/security-threat-report.aspx
5	https://www.tutorialspoint.com/cryptography/data_integrity_in_cryptography.htm
6	https://www.unf.edu/public/cop4610/ree/Notes/PPT/PPT8E/CH15-OS8e.pdf

12.8 UNIT END EXERCISE

1. Explain Security requirements of database.
2. Define term: a) Database b)Sensitive data c)Rootkits
3. Explain Database access control and its types.
4. Explain with neat diagram Inference Control.
5. Explain the micro kernel approach of Operating System design.
6. Explain client-server model.
7. List various Operating Systems. Explain any two.
8. Describe threats to databases.
9. Describe purpose of inference control

10. Write short notes on the following:
    (a) Exokernel
    (b) Virtual machines
    (c) Layered Structure
11. Explain Security Features of Ordinary Operating Systems.
12. Explain any two Operating System Tools to Implement Security Functions.
13. What is Rootkit? Give Example.
14. Write short notes on the following:
    a. Phone Rootkit
    b. Sony XCP Rootkit
    c. TDSS Rootkits

❄❄❄❄❄❄❄