UNIT I

CHAPTER 1

Introduction to Information Security

University Prescribed Syllabus

Foundations of Security, Computer Security Concepts, The OSI security architecture, Security Attacks, Security Services, Security Mechanisms, A Model of Network Security.

1.1 FOUNDATIONS OF SECURITY

UQ. What are different attributes of information security ? Explain each in detail.
(SPPU - Q. 1(a), Dec. 2015, 10 Marks)

UQ. What are the heads of information security? Discuss in detail.
(SPPU - Q. 1(a), Dec. 2016, 10 Marks)

UQ. Enlist and explain needs of information security.
(SPPU - Q. 2(b), May 2018, 9 Marks)

UQ. List and explain various elements of information security. **(SPPU - Q. 1(a), May 2019, 5 Marks)**

In this era of information, organizations are highly dependent on information systems. Computer data often travels from one computer to another, leaving its secure physical environment. Once the data is out of hand, people with bad intention could modify or forge the data, either for amusement or for their own benefit.

Cryptography can reformat and transform the data, making it safer on its trip between computers. The technology is based on the essentials of secret codes, augmented by modern mathematics that protects our data in powerful ways.

Certain terms related to security are defined below:

1.1.1 Computer Security

- Measures and controls that ensure confidentiality, integrity, and availability of the information processed and stored by a computer.
- It is a generic name for the collection of tools designed to protect data and to thwart hackers.

This definition introduces three key objectives of computer security :

(a) **Confidentiality :** This term refers to two definitions that are related:

- **Data Confidentiality :** Ensures the private or sensitive information is not made available to or revealed to unauthorized people.
- **Piracy :** Assures that people have discretion over what information about them is gathered and processed, as well as who has access to it and to whom it is revealed.

(b) **Integrity :** This word encompasses two terms that are intertwined:

- **Data Integrity :** Ensures that information (both stored and transmitted packets) and programs are modified only in the ways that are defined and allowed.
- **System Integrity :** Assures that a system performs its intended purpose without being harmed by intentional or unintentional unauthorized manipulation.

(c) **Availability :** Assures that systems are up and running quickly, and that approved users are not denied service.

1.1.2 Network Security

Network security is the process of taking physical and software preventative measures to protect the underlying networking infrastructure from unauthorized access, misuse, malfunction, modification, destruction, or improper disclosure, thereby creating a secure platform for computers, users and programs to perform their permitted critical functions within a secure environment.

1.1.3 Internet Security

- Internet security is a branch of computer security which comprises various security measures exercised for ensuring the security of transactions done online.
- In the process, the internet security prevents attacks targeted at browsers, network, operating systems, and other applications.

1.2 COMPUTER SECURITY CONCEPTS

GQ. Enlist security goals. Discuss their significance.

GQ. A secure e-voting system is to be designed. Discuss the security goals that must be met and enlist mechanisms for the same.

The CIA Triad is a benchmark model in information security designed to govern and evaluate how an organization handles data when it is stored, transmitted, or processed.

Each attribute of the triad represents a critical component (goals) of information security. The CIA triad is depicted in Fig. 1.2.1.

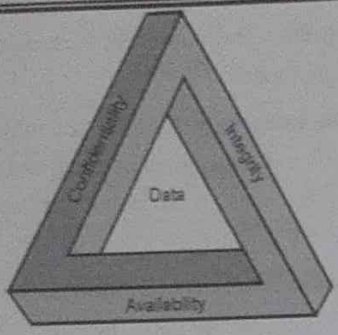


Fig. 1.2.1 : CIA

(1) **Confidentiality :** Data should not be accessed or read without authorization. It ensures that only authorized parties have access. Attacks against Confidentiality are disclosure attacks.

(2) **Integrity :** Data should not be modified or compromised in anyway. It assumes that data remains in its intended state and can only be edited by authorized parties. Attacks against Integrity are alteration attacks.

(3) **Availability :** Data should be accessible upon legitimate request. It ensures that authorized parties have unimpeded access to data when required. Attacks against Availability are destruction attacks.

To make information available to those who need it and who can be trusted with it, organizations use authentication and authorization.

(4) **Authentication** is proving that a user is the person he or she claims to be. That proof may involve something the user knows (such as a password), something the user has (such as a "smartcard"), or something about the user that proves the person's identity (such as a fingerprint).

(5) **Authorization** is the act of determining whether a particular user (or computer system) has the right to carry out a certain activity, such as reading a file or running a program.

Authentication and authorization go hand in hand. Users must be authenticated before carrying out the activity they are authorized to perform. Security is strong when the means of authentication cannot later be refuted – the user cannot later deny that he or she performed the activity. This is known as **nonrepudiation**.

These concepts of information security also apply to the term information security; that is, internet users want to be assured that

- they can trust the information they use
- the information they are responsible for will be shared only in the manner that they expect
- the information will be available when they need it
- the systems they use will process information in a timely and trustworthy manner

In addition, information assurance extends to systems of all kinds, including large-scale distributed systems, control systems, and embedded systems, and it encompasses systems with hardware, software, and human components. The technologies of information assurance address system intrusions and compromises to information.

1.3 THE OSI SECURITY ARCHITECTURE

UQ. Explain OSI Security architecture.
(SPPU - Q. 1(a), May 2012, 6 Marks, Q. 1(b), Dec. 2012, Dec. 2015, 8 Marks)

- To get a sense of how system security is established about, we must know the generally accepted architecture of cyber security setups.
- The Open System Interconnect (OSI) security architecture was designated by the ITU-T (International Telecommunication Union - Telecommunication).
- The ITU-T decided that their standard "X.800" would be the ISO security architecture.
- This standardized architecture defines security requirements and specifies means by which these requirements might be satisfied.
- The OSI architecture focuses on security attacks, mechanisms, and services as shown in Fig. 1.3.1.

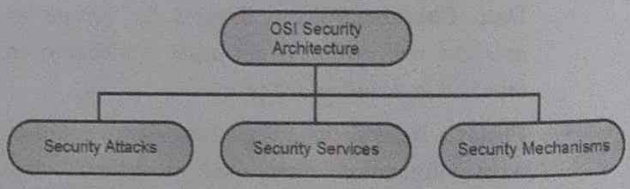


Fig. 1.3.1 : OSI Security Architecture

(1) **Security attacks :** An attack is when the security of a system is compromised by some action of a perpetrator. Attacks could be either active attacks or passive attacks.

(2) **Security mechanisms :** A mechanism that is designed to detect, prevent, or recover from a security attack.

(3) **Security services :** A service that enhances the security of the data processing systems and the information transfers of an organization. The services make use of one or more security mechanisms to provide the service.

1.4 SECURITY ATTACKS

UQ. Define an active attack. Explain any two active attacks with example.
(SPPU - Q. 1(c), May 2012, 6 Marks)

UQ. List and briefly define categories of passive and active security attacks.
(SPPU - Q. 2(a), Dec. 2012, 8 Marks)

UQ. Explain replay, modification of message and denial of service attacks.
(SPPU - Q. 2(a), Dec. 2013, 6 Marks)

UQ. What is masquerade? Discuss it with suitable example. Is it active attack? Justify your answer.
(SPPU - Q. 2(a), Dec. 2014, 8 Marks)

- Any activity that jeopardizes the security of an organization's information is referred to as **an attack**.
- These attacks are generally classified into four categories as:

(1) **Interception** : It is an attack on confidentiality. An adversary can compromise the network to get unauthorized access to node or data stored within it. The main purpose is to eavesdrop on the information carried in the messages.

(2) **Fabrication** : It is an attack on authentication. This gives threats to message authenticity.

(3) **Modification :** It means that a party without any authorization, not only accesses the data but tampers the data. This threatens message integrity. The main purpose is to create confusion or mislead the parties involved in the communication protocol. This is usually aimed at the network layer and the application layer.

(4) **Interruption :** It is an attack on the availability of the network, for example physical nodes capturing, corruption of message, malicious code insertion etc. The main purpose [4] is to launch denial-of-service (DoS) attacks.

- The security attacks can be further categorized as passive attacks and active attacks.
- A **passive** attack tries to learn or use knowledge from the system without causing any damage to the system's resources.
- An **active** attack tries to change the system's resources or disrupt its activity.
- Fig. 1.4.1 shows the classification of attacks with relation to security goals.

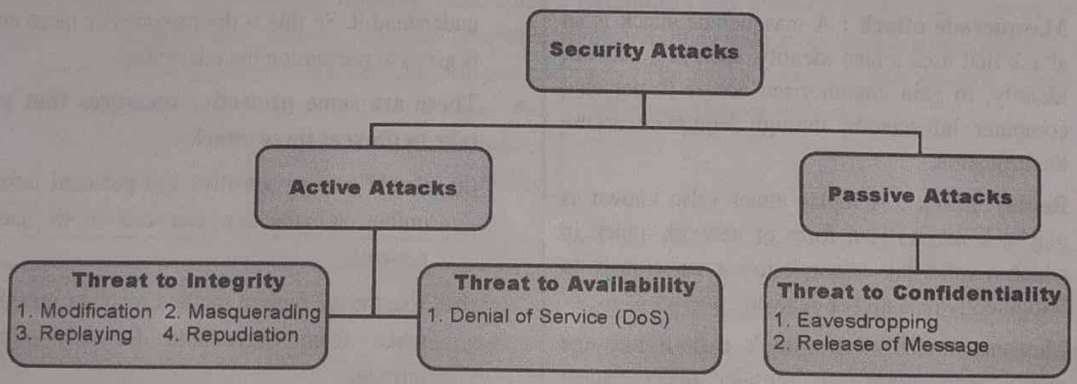


Fig. 1.4.1 : Classification of Attacks with relation to Security Goals

1.4.1 Active Attacks

- Active attacks are attacks in which the hacker attempts to change or transform the content of messages or information.
- These attacks are a threat to the integrity and availability of the system.
- Due to these attacks, systems get damaged, and information can be altered.
- The prevention of these attacks is difficult due to their high range of physical and software vulnerabilities.
- The damage that is done with these attacks can be very harmful to the system and its resources.
- The good thing about this type of attack is that the victim is notified about the attack. So, instead of prevention, the paramount importance is laid on detecting the attack and restoration of the system from the attack.
- An active attack typically requires more effort and generally have more difficult implication.
- Some protective measures that can be taken against this kind of attack are:
  (a) Making use of one time passwords helps in authenticating the transactions between two parties.
  (b) A random session key can be generated, which will be valid for only one transaction. This will help in preventing the attacker from retransmitting the original information after the actual session ends.
- **Active attacks are further divided into five types**
  (i) Masquerade attack : A masquerade attack is an attack that uses a fake identity, such as a network identity, to gain unauthorized access to personal computer information through legitimate access identification.
  (ii) Replay attack : A replay attack (also known as playback attack) is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed.
  (iii) Message modification attack : In a message modification attack, an intruder alters packet header addresses to direct a message to a different destination or modify that data on a target machine.
  (iv) Repudiation attack : A repudiation attack occurs when the user denies the fact that he or she has performed a certain action or has initiated a transaction. A user can simply deny having knowledge of the transaction or communication and later claim that such transaction or communication never took place.
  (v) Denial-of-service attack : A denial-of-service (DoS) is any type of attack where the attackers (hackers) attempt to prevent legitimate users from accessing the service. In a DoS attack, the attacker usually sends excessive messages asking the network or server to authenticate requests that have invalid return addresses.

1.4.2 Passive Attacks

- Passive attacks are the ones in which the attacker observes all the messages and copy the content of messages or information. They focus on monitoring all the transmission and gaining the data.
- The attacker does not try to change any data or information he gathered. Although there is no potential harm to the system due to these attacks, they can be a significant danger to your data's confidentiality.
- Unlike the Active attacks, these are difficult to detect as it does not involve alteration in data or information. Thus, the victim doesn't get any idea about the attack. Although it can be prevented using some encryption techniques.
- In this way, at any time of transmission, the message is in indecipherable form, so that hacker could not understand it. So this is the reason why more emphasis is given to prevention than detection.
- **There are some protective measures that you can take to prevent these attacks.**
  (a) Avoid posting sensitive and personal information online as attackers can use it to hack your network.
  (b) Use the encryption method for your messages and make them unreadable for any unintended intruder.

- **Passive attacks are further divided into two types:**

  (i) **Eavesdropping** : Eavesdropping is the unauthorized real-time interception of a private communication, such as a phone call, instant message, video conference or fax transmission. The term eavesdrop derives from the practice of actually standing under the eaves of a house, listening to conversations inside. It is sometimes called as snooping.

  (ii) **Traffic analysis :** Traffic analysis is a special type of inference attack technique that looks at communication patterns between entities in a system. "Traffic analysis" is the process of intercepting and examining messages in order to deduce information from patterns in communication.

1.4.3 Active Attack Vs Passive Attack

The Table 1.4.1 briefs about the comparison between active and passive attacks.

Table 1.4.1 : Comparison between Active Attack and Passive Attack

Sr. No.	Active Attack	Passive Attack
1.	In active attacks, modification of messages is done.	In passive attacks, the information remains unchanged.
2.	The active attack causes damage to the integrity and availability of the system.	Passive attacks cause damage to data confidentiality.
3.	In active attacks, attention is given to detection.	In passive attacks, attention is given to prevention.
4.	The resources can be changed in active attacks.	Passive attacks have no impact on the resources.
5.	The active attack influences the system services.	The information or data is acquired in passive attacks.
6.	In active attacks, information is gathered through passive attacks to attack the system.	Passive attacks are achieved by collecting confidential information such as private chats and passwords.
7.	Active attacks are challenging to be prohibited.	Passive attacks are easy to prevent.
8.	**Types** : Masquerade, Replay, Modification, Denial of Service (DoS)	**Types :** Eavesdropping, Traffic Analysis
9.	**Examples :** The attacker is inserting his data into the original data stream. Man-in-the-middle attack where the attacker sits between both parties communicating and replacing their messages with his message. In other words, both parties believe that they are talking to each other, but in reality, they are talking to the attacker.	**Examples :** The attackers try to scan a device to find vulnerabilities such as weak operating system or open ports. The hackers analyze and monitor a website's traffic to see who is visiting it.

1.5 SECURITY SERVICES

UQ. List and explain OSI security services. **(SPPU - Q. 1(A), May 2013, 8 Marks)**

UQ. What are different categories of security services defined by x.800? Discuss each in detail. **(SPPU - Q. 1(a), Dec. 2014, 8 Marks)**

The International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) has identified five services that are linked to the security goals and attacks we discussed in the previous sections. Fig. 1.5.1 shows the categorization of those five common services.

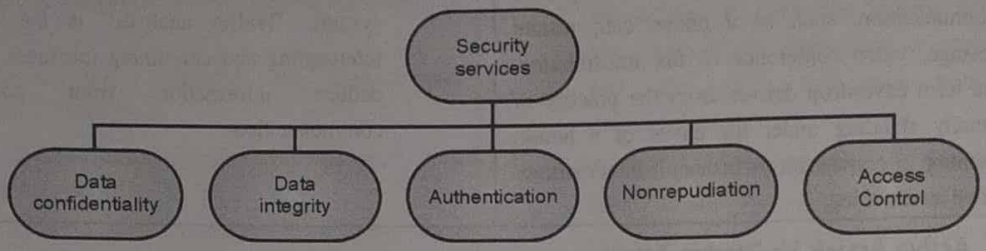


Fig. 1.5.1 : Security Services

(1) Data Confidentiality

The safeguarding of data against unauthorized disclosure. There are four types of confidentiality services defined by ITU-T standard.

(a) **Connection Confidentiality :** All user data on a connection link is protected.

(b) **Connectionless Confidentiality :** All user data is protected in a single data block.

(c) **Selective-Field Confidentiality :** Selected fields within user data on a connection or in a single data block are kept confidential.

(d) **Traffic-Flow Confidentiality :** The safeguarding of data obtained from traffic flow observation.

(2) Data Integrity

The guarantee that data obtained is exactly as it was submitted by a legitimate source (i.e., contain no modification, insertion, deletion, or replay). There are five types of integrity services defined by ITU-T standard.

(a) **Connection Integrity with Recovery :** Maintains the integrity of all user data on a connection link by detecting any alteration, addition, deletion, or replay of any data within an entire data sequence and attempting to recover it.

(b) **Connection Integrity without Recovery :** Maintains the integrity of all user data on a connection link by detecting any alteration, addition, deletion, or replay of any data within an entire data sequence without attempting to recover it.

(c) **Selective-Field Connection Integrity :** Determines whether selected fields in the user data of a data block transmitted over a connection link have been changed, added, removed, or replayed.

(d) **Connectionless Integrity :** Provides integrity for a single connectionless data block which can provide data change detection. A restricted version of replay detection may also be available.

(e) **Selective-Field Connectionless Integrity :** Sets the integrity of the fields selected in a single connectionless data block; takes the form of determining whether or not the fields selected had been changed.

(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.

(a) **Peer Entity Authentication**: When used in conjunction with a logical connection, it provides assurance that the people connected are who they say they are.

(b) **Data-Origin Authentication**: Provides assurance that the source of obtained data is as stated in a connectionless transfer.

(4) Nonrepudiation

Provides security against the denial of participation in all or part of a conversation by one of the individuals participating in the communication. There are two types of nonrepudiation services defined by ITU-T standard.

(a) **Nonrepudiation, Origin :** The message was sent by the stated party, according to the evidence.

(b) **Nonrepudiation, Destination :** The message was received by the stated party, according to the evidence.

(5) **Access Control :** The act of preventing unauthorized access to a resource (i.e., 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).

1.6 SECURITY MECHANISMS

UQ. Explain the following OSI security mechanisms: Digital signature, Access control, Data integrity, Authentication exchange.

(SPPU - Q. 1(B), May 2013, 10 Marks)

UQ. What is mechanism in security? Discuss any one mechanism in detail.

(SPPU - Q. 2(b), Dec. 2015, 8 Marks)

ITU-T also recommends some security mechanisms to provide the security. Fig. 1.6.1 gives the taxonomy of these mechanisms.

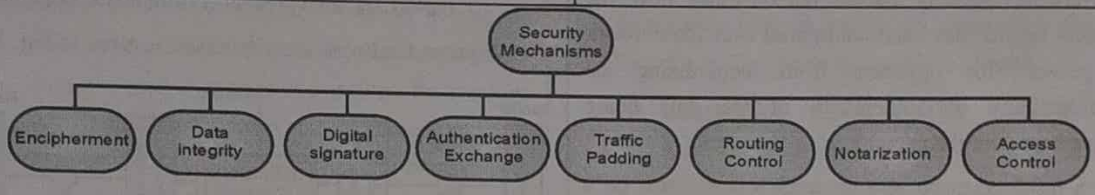


Fig. 1.6.1 : Security Mechanisms

(1) **Encipherment :** The application of mathematical algorithms to transform data into a form that is difficult to understand. An algorithm and zero or more encryption keys are used to convert the data and then recover it.

(2) **Data Integrity :** Various methods for ensuring the integrity of a data unit or a stream of data units.

(3) **Digital Signature :** Data appended to, or a cryptographic transformation of, a data unit that allows a receiver to prove the data unit's source and integrity while protecting against forgery (e.g., by the recipient).

(4) **Authentication Exchange :** A system for ensuring an entity's identity through the exchange of information.

(5) **Traffic Padding :** Bits are inserted into gaps in a data stream in order to thwart traffic analysis attempts.

(6) **Routing Control :** Allows for the selection of specific physically safe routes for specific data, as well as routing changes, particularly when a security breach is suspected.

(7) **Notarization :** The use of a trustworthy third party to ensure that a data exchange maintains those properties.

(8) **Access Control :** Various methods for enforcing resource access privileges.

1.7 RELATION BETWEEN SERVICES AND MECHANISMS

Table 1.7.1 shows the relationship between the security services and security mechanism.

Table 1.7.1 : Relation between Security Services and Security Mechanisms

Services	Mechanisms							
	Encipherment	Data Integrity	Digital Signature	Authentication Exchange	Traffic Padding	Routing Control	Notarization	Access Control
Data Confidentiality	Y					Y		
Data Integrity	Y	Y	Y					

Services	Mechanisms							
	Encipherment	Data Integrity	Digital Signature	Authentication Exchange	Traffic Padding	Routing Control	Notarization	Access Control
Authentication	Y		Y	Y			Y	
Nonrepudiation		Y	Y					Y
Access Control								

1.8 A MODEL OF NETWORK SECURITY

- A Network Security Model demonstrates how the security service has been configured over the network to prevent the opponent from jeopardizing the confidentiality or authenticity of the data being transmitted over the network.
- For a message to be sent or receive there must be a sender and a receiver. Both the sender and receiver must also be mutually agreeing to the sharing of the message.
- Now, the transmission of a message from sender to receiver needs a medium i.e. **Information channel** which is an **Internet** service.
- A logical route is defined through the network (Internet), from sender to the receiver and using the **communication protocols** (e.g. TCP/IP, etc.) both the sender and the receiver established communication.
- Any security service would have the **three components** discussed below:
  (a) **Transformation** of the information, which has to be sent to the receiver. So, that any opponent present at the information channel is unable to read the message. This indicates the **encryption** of the message. It also includes the addition of code during the transformation of the information which will be used in verifying the identity of the authentic receiver.
  (b) Sharing of the **secret information** between sender and receiver of which the opponent must not any clue. Yes, we are talking of the **encryption key** which is used during the encryption of the message at the sender's end and also during the decryption of message at receiver's end.
  (c) There must be a **trusted third party** which should take the responsibility of **distributing the secret information** (key) to both the communicating parties and also prevent it from any opponent.
- A general network security model is given in Fig. 1.8.1.

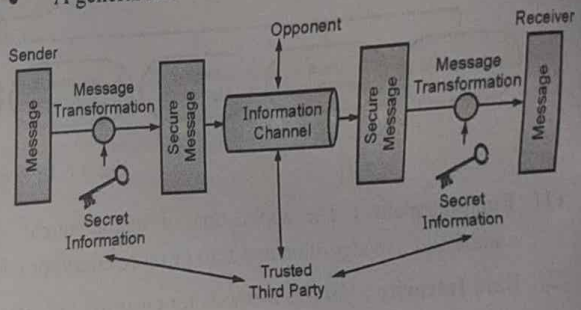


Fig. 1.8.1 : Network Security Model

- The network security model presents the two communicating parties sender and receiver who mutually agrees to exchange the information. The sender has information to share with the receiver.
- But sender cannot send the message on the information cannel in the readable form as it will have a threat of being attacked by the opponent. So, before sending the message through the information channel, it should be transformed into an unreadable format.
- **Secret information** is used while transforming the message which will also be required when the message will be retransformed at the recipient side. That's why a trusted third party is required which would take the responsibility of distributing this secret information to both the parties involved in communication. So, considering this general model of network security, one must consider the following four tasks while designing the security model.

(1) To **transform** a readable message at the sender side into an unreadable format, an appropriate algorithm should be designed such that it should be difficult for an opponent to crack that security algorithm.

(2) Next, the network security model designer is concerned about the **generation of the secret information** which is known as a **key**. This secret information is used in conjunction with the security algorithm in order to transform the message.

(3) Now, the secret information is required at both the ends, sender's end and receiver's end. At sender's end, it is used to encrypt or transform the message into unreadable form and at the receiver's end, it is used to decrypt or retransform the message into readable form. So, there must be a **trusted third party** which will distribute the secret information to both sender and receiver. While designing the network security model designer must also concentrate on **developing the methods** to distribute the key to the sender and receiver. An appropriate methodology must be used to deliver the secret information to the communicating parties without the interference of the opponent.

(4) It is also taken care that the **communication protocols** that are used by the communicating parties should be supporting the security algorithm and the secret key in order to achieve the security service.

- These attackers who attack the system that is accessible through the internet fall into two categories:
  - **(a) Hacker :** The one who is only interested in penetrating into your system. They do not cause any harm to your system; they only get satisfied by getting access to your system.
  - **(b) Intruders :** These attackers intend to do damage to your system or try to obtain the information from the system which can be used to attain financial gain.
- The attacker can place a logical program on the system through the network which can affect the software on the system. This leads to two kinds of risks :
  - **(a) Information threat :** This kind of threats modifies data on the user's behalf to which actually user should not access. Like enabling some crucial permission in the system.
  - **(b) Service threat :** This kind of threat disables the user from accessing data on the system.
- These kinds of threats can be introduced by launching **worms** and **viruses** and many more like this on the system.
- Attack with worms and viruses are the software attack that can be introduced to the system through the internet.
- The network security model to secure one's system is shown in the Fig. 1.8.2.

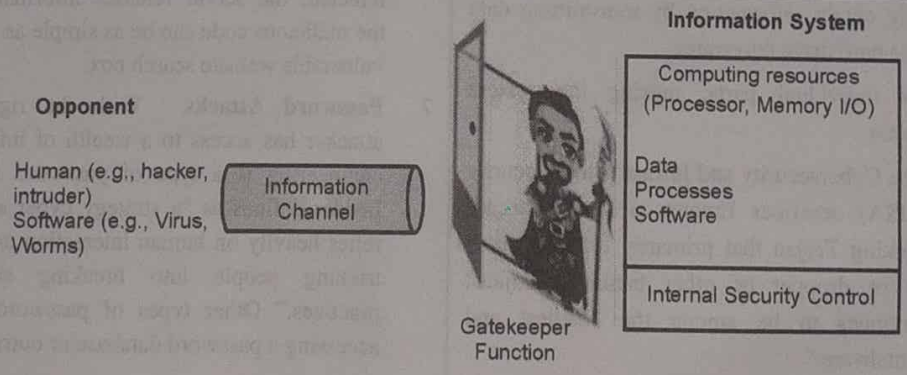


Fig. 1.8.2 : Network Access Security Model

- There are two ways to secure one's system from attacker of which the first is to introduce the **gatekeeper function**.
- Introducing gatekeeper function means introducing **login-id** and **passwords** which would keep away the unwanted access.
- In case the unwanted user gets access to the system, the second way to secure the system is introducing **internal control** which would detect the unwanted user trying to access the system by analyzing system activities.
- This second method we call as **antivirus** which we install on our system to prevent the unwanted user from accessing the computer system through the internet.

1.8.1 Threats in Network Security

UQ. What are threats? Explain the different categories of threat. (SPPU - Q. 1(a), Dec. 2013, 6 Marks)

A *security threat* is a malicious act that aims to corrupt or steal data or disrupt an organization's systems or the entire organization. IT professionals should have an in-depth understanding of the following types of security threats.

1. **Malware :** Malware is malicious software such as spyware, ransomware, viruses and worms. Malware is activated when a user clicks on a malicious link or attachment, which leads to installing dangerous software. Cisco reports that malware, once activated, can:
   - Block access to key network components (ransomware)
   - Install additional harmful software
   - Covertly obtain information by transmitting data from the hard drive (spyware)
   - Disrupt individual parts, making the system inoperable
2. **Emotet** : The Cybersecurity and Infrastructure Security Agency (CISA) describes Emotet as "an advanced, modular banking Trojan that primarily functions as a downloader or dropper of other banking Trojans. Emotet continues to be among the costliest and destructive malware."
3. **Denial of Service :** A denial of service (DoS) is a type of cyber attack that floods a computer or network so it can't respond to requests. A distributed DoS (DDoS) does the same thing, but the attack originates from a computer network. Cyber attackers often use a flood attack to disrupt the "handshake" process and carry out a DoS. Several other techniques may be used, and some cyber attackers use the time that a network is disabled to launch other attacks. A botnet is a type of DDoS in which millions of systems can be infected with malware and controlled by a hacker, according to Jeff Melnick of Netwrix, an information technology security software company. Botnets, sometimes called zombie systems, target and overwhelm a target's processing capabilities. Botnets are in different geographic locations and hard to trace.
4. **Man in the Middle :** A man-in-the-middle (MITM) attack occurs when hackers insert themselves into a two-party transaction. After interrupting the traffic, they can filter and steal data, according to Cisco. MITM attacks often occur when a visitor uses an unsecured public Wi-Fi network. Attackers insert themselves between the visitor and the network, and then use malware to install software and use data maliciously.
5. **Phishing** : Phishing attacks use fake communication, such as an email, to trick the receiver into opening it and carrying out the instructions inside, such as providing a credit card number. "The goal is to steal sensitive data like credit card and login information or to install malware on the victim's machine," Cisco reports.
6. **SQL Injection** : A Structured Query Language (SQL) injection is a type of attack that results from inserting malicious code into a server that uses SQL. When infected, the server releases information. Submitting the malicious code can be as simple as entering it into a vulnerable website search box.
7. **Password Attacks** : With the right password, a attacker has access to a wealth of information. Social engineering is a type of password attack that Data Insider defines as "a strategy cyber attackers use that relies heavily on human interaction and often involves tricking people into breaking standard security practices." Other types of password attacks include accessing a password database or outright guessing.

Descriptive Questions

Q. 1 Enlist security goals. Discuss their significance.

Q. 2 A secure e-voting system is to be designed. Discuss the security goals that must be met and enlist mechanisms for the same.

Q. 3 Distinguish between passive and active security attacks. Name some passive attacks. Name some active attacks.

Q. 4 Write short note on eavesdropping.

Q. 5 List and explain security services and security mechanisms in detail.

Q. 6 Explain network security model in detail with neat diagram.

...Chapter Ends

❑❑❑

UNIT II

CHAPTER 2

Symmetric Key Cryptography

University Prescribed Syllabus

Classical Encryption Techniques : Stream Ciphers, Substitution Techniques : Caesar Cipher, Mono alphabetic Ciphers, Play fair Cipher, Hill Cipher, Poly alphabetic Ciphers, Transposition Techniques, Block Ciphers and Data Encryption standards, 3 DES, Advanced Encryption standard.

M	F	H	I/J	K
U	N	O	P	Q
Z	V	W	X	Y
E	L	A	R	G
D	S	T	B	C

2.1 CLASSICAL ENCRYPTION TECHNIQUES

2.1.1 Terminologies in Encryption Techniques

This section describes the various terms used in encryption techniques.

(1) **Plaintext :** It is the raw data that must be safeguarded during transmission from sender to receiver. This is also often referred to as the message.

(2) **Ciphertext :** It is the scrambled version of the plaintext that results from applying the encryption algorithm (and the encryption key) to the plaintext. It is sometimes referred to as cryptogram.

(3) **Enciphering or encryption :** The process of converting from plaintext to ciphertext is called enciphering or encryption.

(4) **Deciphering or decryption :** The process of restoring the plaintext from the ciphertext is called deciphering or decryption.

(5) **Cryptography :** The term cryptography means secret writing. **Cryptography** is a method of protecting information and communications through the use of codes, so that only those for whom the information is intended can read and process it.

(6) **Cryptanalysis :** The field of cryptanalysis encompasses techniques for deciphering a message without knowing the encoding information. Cryptanalysis is what the general public refers to as "breaking the code."

(7) **Cryptology :** The areas of cryptography and cryptanalysis together are called **cryptology**.

2.1.2 Symmetric Cipher Model

The symmetric encryption method has five components as shown in Fig. 2.1.1

(1) **Plaintext** : This is the original, understandable message or data that is fed as input into the algorithm.

(2) **Encryption algorithm** : On the plaintext, the encryption algorithm performs various substitutions and transformations.

(3) **Secret key** : The encryption algorithm also takes the secret key into account. The trick is a value that isn't affected by the plaintext or the algorithm. Depending on the particular key used at the time, the algorithm can generate a different result. The algorithm's precise substitutions and transformations are determined by the key.

(4) **Ciphertext :** This is the output code, which is scrambled (unintelligible). It depends on the plaintext and the secret key. Two different keys generate two different ciphertexts for the same message.

(5) **Decryption algorithm :** It is the reverse execution of encryption algorithm. It takes the ciphertext and the secret key and produces the original plaintext.

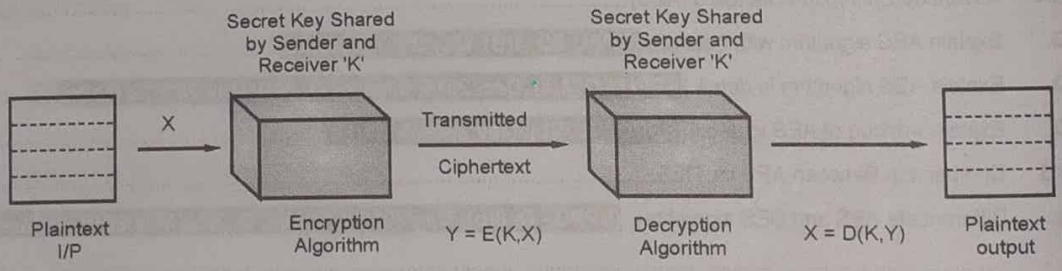


Fig. 2.1.1 : Symmetric Encryption Model

NOTES

2.1.3 Encryption Requirements

Traditional encryption must meet two conditions in order to be secure :

(1) A strong encryption algorithm is needed.

At the very least, an adversary who understands the algorithm and has access to one or more ciphertexts would be unable to decode the ciphertext or deduce the key. Even if he or she has a number of ciphertexts and the plaintext for each ciphertext, the opponent should be unable to decrypt ciphertexts or discover the key in a stronger form.

(2) Both the sender and the receiver must have received and held copies of the secret key in a safe manner. All communication using this key is readable if someone discovers the key and understands the algorithm.

2.1.4 Why not keep the Encryption Algorithm Secret ?

- We assume that decrypting a message using just the ciphertext and the encryption/decryption algorithm is impractical.
- This means that only the key must be kept hidden, not the algorithm.
- Because of this property of symmetric encryption, low-cost chip implementations of data encryption algorithms are commonly available and are used in a variety of products.
- The most significant security issue with symmetric encryption is keeping the key secret.

2.1.5 Symmetric Cryptosystem Model

The Fig. 2.1.2 depicts the main components of a symmetric encryption scheme.

- A source produces a message in plaintext, $X = [X_1, X_2, ..., X_M]$.
- A key of the form $K = [K_1, K_2, ..., K_J]$ is generated for encryption.
  - If the key is created at the source of the message, it must also be sent to the destination through a secure channel.
  - A third party could also generate the key and deliver it securely to both the source and the destination.
- The encryption algorithm generates the ciphertext $Y = [Y_1, Y_2, ..., Y_N]$ using the message X and the encryption key K as input.
- The encryption procedure is as follows : **Y = E(K, X).** This notation indicates that Y is generated as a function of the plaintext X using encryption algorithm E, with the basic function determined by the value of the key K.
- The transformation can be inverted by the intended receiver with the key: **D = X(K, Y)**
- When an opponent sees Y but does not have access to K or X, he or she can try to recover X, K, or both. The adversary is believed to be aware of the encryption (E) and decryption (D) algorithms.

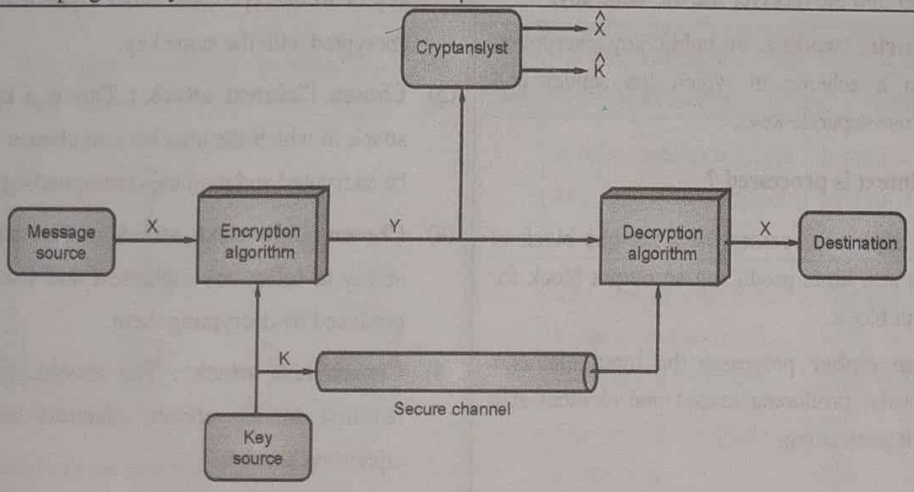


Fig. 2.1.2 : Symmetric Cryptosystem Model

- Either of the following options is available to the opponent:
  - If the opponent is only interested in this letter, recover X by producing a plaintext approximation $\hat{X}$.
- If the opponent is interested in being able to read future messages, recover K by producing an approximation $\hat{K}$

2.1.6 Cryptography

Three independent dimensions are used to describe cryptographic systems:

(1) Type of operations for transforming plaintext to ciphertext

Two fundamental principles drive all encryption algorithms:

- **Substitution :** Each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element.
- **Transposition :** Elements in the plaintext are rearranged.

The most essential factor is that no data is lost (all operations are reversible). Multiple phases of substitutions and transpositions occur in product systems.

(2) Number of keys used

- The scheme is referred to as **symmetric**, single-key, secret-key, or conventional encryption if both the sender and the receiver use the same key.
- **Asymmetric**, two-key, or public-key encryption refers to a scheme in which the sender and receiver use separate keys.

(3) How the plaintext is processed ?

- A **block cipher** processes the input one block of elements at a time, producing an output block for each input block.
- A **stream cipher** processes the input elements continuously, producing output one element at a time, as it goes along.

2.1.7 Cryptanalysis and Brute-Force Attack

The goal of an encryption system attack is to recover the key in use rather than simply the plaintext of a single ciphertext. There are two broad approaches to attacking a traditional encryption scheme:

- **Cryptanalysis :** Cryptanalysis is a process of finding weaknesses in cryptographic algorithms and using these weaknesses to decipher the ciphertext without knowing the secret key.
- **Brute-force attack :** The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained.

If any form of attack succeeds in deducing the key, all future and previous messages encrypted with that key will be vulnerable.

The following attacks can refer to either of the two classes (all forms of attack assume the attacker knows the encryption algorithm):

(1) Ciphertext-only attack : In this attack the attacker knows only the ciphertext to be decoded. The attacker will try to find the key or decrypt one or more pieces of ciphertext (only relatively weak algorithms fail to withstand a ciphertext-only attack).

(2) Known plaintext attack : The attacker has a collection of plaintext-ciphertext pairs and is trying to find the key or to decrypt some other ciphertext that has been encrypted with the same key.

(3) Chosen Plaintext attack : This is a known plaintext attack in which the attacker can choose the plaintext to be encrypted and read the corresponding ciphertext.

(4) Chosen Ciphertext attack : The attacker has the ability to select any ciphertext and study the plaintext produced by decrypting them.

(5) Chosen text attack : The attacker has the abilities required in the chosen plaintext and the chosen ciphertext attacks.

2.2 SYMMETRIC KEY AND ASYMMETRIC KEY CRYPTOGRAPHY

UQ. What is ciphering? Explain any two ciphering techniques with suitable example.
(SPPU - Q. 4(b), Dec. 2015, 8 Marks, Q. 4(b), Dec. 2016, 8 Marks)

UQ. Differentiate public key and private key cryptography with suitable example.
(SPPU - Q. 5(a), Dec. 2015, 8 Marks)

UQ. What is cryptography? Discuss different types of cryptography in short.
(SPPU - Q. 3(a), Dec. 2016, 8 Marks)

2.2.1 Symmetric Key Cryptography

- Also called symmetric key encryption or private key encryption or private key cryptography. In symmetric key cryptography, the same key is used for both encrypting and decrypting the messages.
- It doesn't scale very well because the secret key must not be lost or shared with unauthorized parties, or else they can read the message.
- Symmetric key encryption algorithms can use either block ciphers or stream ciphers. With block ciphers, a number of bits (in chunks) is encrypted as a single unit.
- For instance, AES uses a block size of 128 bits with options for three different key lengths – 128, 192, or 256 bits.
- Although there are key management issues with symmetric key encryption, its faster and functions without a lot of overheads on network or CPU resources.
- Therefore, it's often used in combination with asymmetric key encryption. Fig. 2.2.1 depicts the working of symmetric key encryption.

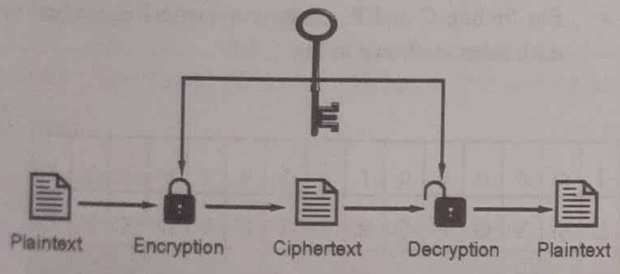


Fig. 2.2.1 : Symmetric Key Encryption

2.2.2 Asymmetric Key Cryptography

- Also called asymmetric key encryption or public key encryption or public key cryptography. Asymmetric key encryption uses a pair of related keys - a public and a private key.
- The public key, which is accessible to everyone, is what's used to encrypt a plaintext message before sending it. To decrypt and read this message, one need to hold the private key.
- The public and the private keys are mathematically related, but the private key cannot be derived from it.
- In asymmetric key encryption, the private key is only shared with the key's initiator since its security needs to be maintained. Fig. 2.2.2 depicts the working of asymmetric key encryption.

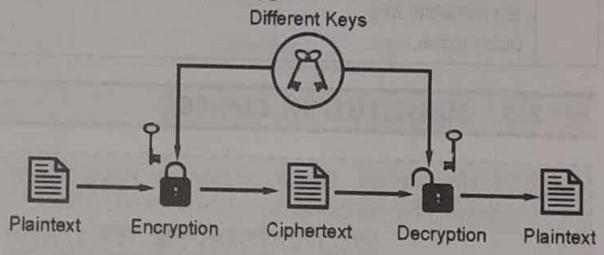


Fig. 2.2.2 : Asymmetric Key Encryption

2.2.3 Symmetric Vs Asymmetric Key Cryptography

Table 2.2.1 compares the features of symmetric and asymmetric key cryptography.

Table 2.2.1 : Symmetric Vs Asymmetric Key Cryptography

Sr. No.	Symmetric Key Cryptography	Asymmetric Key Cryptography
1.	It only requires a single key for both encryption and decryption.	It requires two key one to encrypt and the other one to decrypt.
2.	The size of cipher text is same or smaller than the original plain text.	The size of cipher text is same or larger than the original plain text.
3.	The encryption process is very fast.	The encryption process is slow.

Sr. No.	Symmetric Key Cryptography	Asymmetric Key Cryptography
4.	It is used when a large amount of data is required to transfer.	It is used to transfer small amount of data.
5.	It only provides confidentiality.	It provides confidentiality, authenticity and non-repudiation.
6.	Examples: AES, DES, Triple DES and RC4	Examples: RSA, Diffie-Hellman, Elliptical Curve Cryptography and El Gamal Algorithm
7.	In symmetric key encryption, resource utilization is low as compared to asymmetric key encryption.	In asymmetric key encryption, resource utilization is high.

2.3 SUBSTITUTION CIPHERS

UQ. Briefly define Caesar cipher, monoalphabetic cipher, play fair cipher and transposition cipher.
(SPPU - Q. 2(b), Dec. 2012, 8 Marks)

- Substitution ciphers encrypt the plaintext by swapping each letter or symbol in the plaintext by a different letter or symbol as directed by the key.
- These plaintext units may be individual letters or characters, letter pairs, triplets, or other combinations.
- Substitution ciphers may replace only the letters of the standard alphabet with ciphertext, or apply substitutions to spaces and punctuation marks as well.

2.3.1 Monoalphabetic Ciphers

- In monoalphabetic substitution, a character (or a symbol) in the plaintext is always replaced by the same character (or a symbol) in the ciphertext irrespective of its position in the plaintext.
- The relationship between a symbols in the plaintext to a symbol in the ciphertext is always one-to-one.
- For example, if the algorithm says that letter A in the plaintext is replaced by letter D in the ciphertext, then every letter A is replaced by letter D.

2.3.2 Additive Cipher

- The additive cipher is the simplest monoalphabetic cipher. The additive cipher is also called a **shift cipher or Caesar cipher.**
- Julius Caesar used the shift cipher (additive cipher) technique to communicate with his officers. For this reason, the shift cipher technique is called the Caesar cipher.
- The Caesar cipher is a kind of replacement (substitution) cipher, where each alphabet of plain text is replaced by an alphabet three places down the line.
- Let's take an example to understand the Caesar cipher, suppose we are shifting with 3, then A will be replaced by D, B will be replaced by E, C will be replaced by F, D will be replaced by G, and this process continues until the entire plain text is finished.
- A Caesar cipher is a weak method of cryptography. It can be easily hacked. It means the message encrypted by this method can be easily decrypted.
- The general formula of encryption using Additive cipher is :

$$C = (P + K) \bmod 26$$

- The general formula of decryption using Additive cipher is: $P = (C - K) \bmod 26$.
- If in any case during decryption, P value becomes negative, then add 26 in the negative value.
- Here, C denotes the letter in ciphertext, P denotes the letter in plaintext, K is the shift value (3 in case of Caesar Cipher). The value of K can range from 0 to 25.
- For finding C and P, assign a numerical equivalent to each letter as shown in Fig. 2.3.1.

Plaintext →	a	b	c	d	e	F	g	H	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Fig. 2.3.1 : Numerical Equivalent of Each Letter

Ex. 2.3.1 : Use the Caesar cipher to encrypt and decrypt the message "COMPUTER".

☑ **Soln. :**

▶ **Encryption**

We apply the encryption algorithm to the plaintext, character by character.

$$C = (P + K) \bmod 26$$

For Caesar cipher, $K = 3$

Plaintext: C → 02	Encryption: (02 + 3) mod 26	Ciphertext: 05 → F
Plaintext: O →14	Encryption: (14 + 3) mod 26	Ciphertext: 17 → R
Plaintext: M →12	Encryption: (12 + 3) mod 26	Ciphertext: 15 → P
Plaintext: P → 15	Encryption: (15 + 3) mod 26	Ciphertext: 18 → S
Plaintext: U → 20	Encryption: (20 + 3) mod 26	Ciphertext: 23 → X
Plaintext: T → 19	Encryption: (19 + 3) mod 26	Ciphertext: 22 → W
Plaintext: E → 04	Encryption: (04 + 3) mod 26	Ciphertext: 07 → H
Plaintext: R → 17	Encryption: (17 + 3) mod 26	Ciphertext: 20 → U

The result is "FRPSXWHU".

▶ **Decryption**

We apply the decryption algorithm to the ciphertext, character by character.

$$P = (C - K) \bmod 26$$

For Caesar cipher, $K = 3$

Ciphertext : F → 05	Decryption: (05 – 3) mod 26	Plaintext: 02 → C
Ciphertext : R → 17	Decryption: (17 – 3) mod 26	Plaintext: 14 → O
Ciphertext : P → 15	Decryption: (15 – 3) mod 26	Plaintext: 12 → M
Ciphertext : S → 18	Decryption: (18 – 3) mod 26	Plaintext: 15 → P
Ciphertext : X → 23	Decryption: (23 – 3) mod 26	Plaintext: 20 → U
Ciphertext : W → 22	Decryption: (22 – 3) mod 26	Plaintext: 19 → T
Ciphertext : H → 07	Decryption: (07 – 3) mod 26	Plaintext: 04 → E
Ciphertext : U → 20	Decryption: (20 – 3) mod 26	Plaintext: 17 → R

The result is "COMPUTER".

Ex. 2.3.2 : Use the additive cipher with key = 15 to encrypt and decrypt the message "HELLO".

☑ **Soln. :**

▶ **Encryption**

We apply the encryption algorithm to the plaintext, character by character.

$$C = (P + K) \bmod 26$$

Given key $K = 15$

Plaintext: H → 07	Encryption: (07 + 15) mod 26	Ciphertext: 22 → W
Plaintext: E → 04	Encryption: (04 + 15) mod 26	Ciphertext: 19 → T
Plaintext: L → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: L → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: O →14	Encryption: (14 + 15) mod 26	Ciphertext: 03 → D

The result is "WTAAD".

Note that the cipher is monoalphabetic because two instances of the same plaintext character (L's) are encrypted to the same character (A).

Decryption

We apply the decryption algorithm to the plaintext, character by character.

$$P = (C - K) \bmod 26$$

Given key $K = 15$

Ciphertext: W → 22	Decryption: (22 – 15) mod 26	Plaintext: 07 → H
Ciphertext: T → 19	Decryption: (19 – 15) mod 26	Plaintext: 04 → E
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: –15 = –15 + 26 = 11 → L
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: –15 = –15 + 26 = 11 → L
Ciphertext: D → 03	Decryption: (03 – 15) mod 26	Plaintext: –12 = –12 + 26 = 14 → O

The result is "HELLO".

2.3.2(A) Multiplicative Ciphers

- The general formula of encryption using Multiplicative cipher is: $C = (P \times K) \bmod 26$.
- The general formula of decryption using Multiplicative cipher is: $P = (C \times K^{-1}) \bmod 26$.

Algorithm to find multiplicative inverse

The integer 'a' in Z_n has a multiplicative inverse if and only if $\gcd(n, a) \equiv 1 \pmod n$

To find multiplicative inverse of b in Z_n when n and b are given and gcd(n, b) = 1.

```
r1 ← n; r2 ← b;
t1 ← 0; t2 ← 1;
while (r2 > 0)
{
    q ← r1 / r2;
    r ← r1 – q x r2;
    r1 ← r2; r2 ← r;
    t ← t1 – q x t2;
    t1 ← t2; t2 ← t;
}
if (r1 = 1) then b^-1 ← t1
```

Ex. 2.3.3 : Find the multiplicative inverse of 7 in Z_{26}.

☑ **Soln. :** Given $r_1 = 26$ and $r_2 = 7$

Q	r_1	r_2	r	t_1	t_2	t
3	26	7	5	0	1	– 3
1	7	5	2	1	– 3	4
2	5	2	1	– 3	4	– 11
2	2	1	0	4	– 11	26
	1	0		– 11	26	

The gcd(26, 7) is 1, which means the multiplicative inverse of 7 exist. The above algorithm gives $t_1 = -11$. The multiplicative inverse is $(-11) \bmod 26 = (-11 + 26) \bmod 26 = 15 \bmod 26 = 15$.

Thus, the multiplicative inverse of 7 in Z_{26} is 15.

The table of multiplicative inverses existing in Z_{26} is given below.

Inverses mod 26												
b	1	3	5	7	9	11	15	17	19	21	23	25
b^{-1}	1	9	21	15	3	19	7	23	11	5	17	25

Ex. 2.3.4 : Use the multiplicative cipher with key = 7 to encrypt and decrypt the message "HELLO".

☑ **Soln. :**

▶ **Encryption**

We apply the encryption algorithm to the plaintext, character by character.

$$C = (P \times K) \bmod 26$$

Given key $K = 7$

Plaintext: H → 07	Encryption: (07 × 7) mod 26	Ciphertext: 23 → X
Plaintext: E → 04	Encryption: (04 × 7) mod 26	Ciphertext: 02 → C
Plaintext: L → 11	Encryption: (11 × 7) mod 26	Ciphertext: 25 → Z
Plaintext: L → 11	Encryption: (11 × 7) mod 26	Ciphertext: 25 → Z
Plaintext: O →14	Encryption: (14 × 7) mod 26	Ciphertext: 20→ U

The result is "XCZZU".

▶ **Decryption**

We apply the decryption algorithm to the plaintext, character by character.

$$P = (C \times K^{-1}) \bmod 26$$

Given key $K = 7$

Multiplicative inverse of 7 is 15.

Therefore, $K^{-1} = 7^{-1} = 15$

Ciphertext: X → 23	Decryption: (23 × 15) mod 26	Plaintext: 07 → H
Ciphertext: C → 02	Decryption: (02 × 15) mod 26	Plaintext: 04 → E
Ciphertext: Z → 25	Decryption: (25 × 15) mod 26	Plaintext: 11 → L
Ciphertext: Z → 25	Decryption: (25 × 15) mod 26	Plaintext: 11 → L
Ciphertext: U → 20	Decryption: (20 × 15) mod 26	Plaintext: 14 → O

The result is "HELLO".

2.3.2(B) Affine Cipher

(1) The affine cipher is a combination of both the additive and multiplicative ciphers with a pair of key.

(2) The first key is used with the multiplicative cipher and the second key is used with the additive cipher.

(3) The general formula of encryption using affine cipher is : $C = (P \times K_1 + K_2) \bmod 26$.

(4) The general formula of decryption using affine cipher is : $P = ((C - K_2) \times K_1^{-1}) \bmod 26$.

(5) Here, K_1^{-1} is the multiplicative inverse of K_1.

Ex. 2.3.5 : Use the affine cipher to encrypt and decrypt the message "HELLO" with key pair (7, 2) in modulus26.

Soln. :

Encryption

We apply the encryption algorithm to the plaintext, character by character.

$$C = (P \times K_1 + K_2) \bmod 26$$

Given $K_1 = 7$ and $K_2 = 2$

Plaintext: H → 07	Encryption: (07 x 7 + 2) mod 26	Ciphertext: 25 → Z
Plaintext: E → 04	Encryption: (04 x 7 + 2) mod 26	Ciphertext: 04 → E
Plaintext: L → 11	Encryption: (11 x 7 + 2) mod 26	Ciphertext: 01 → B
Plaintext: L → 11	Encryption: (11 x 7 + 2) mod 26	Ciphertext: 01 → B
Plaintext: O →14	Encryption: (14 x 7 + 2) mod 26	Ciphertext: 22 → W

The result is "ZEBBW".

Decryption

We apply the decryption algorithm to the plaintext, character by character.

$$P = ((C - K_2) \times K_1^{-1}) \bmod 26$$

Given $K_1 = 7$ and $K_2 = 2$

Multiplicative inverse of 7 is 15.

Therefore, $K^{-1} = 7^{-1} = 15$

Ciphertext: Z → 25	Decryption: ((25 – 2) × 15) mod 26	Plaintext: 07 → H
Ciphertext: E →04	Decryption: ((04 – 2) × 15) mod 26	Plaintext: 04 → E

Ciphertext: B → 01	Decryption: ((01 – 2) × 15) mod 26 = (–15 + 26) mod 26 = 11 mod 26	Plaintext: 11 → L
Ciphertext: B → 01	Decryption: ((01 – 2) × 15) mod 26 = (–15 + 26) mod 26 = 11 mod 26	Plaintext: 11 → L
Ciphertext: W → 22	Decryption: ((22 – 2) × 15) mod 26	Plaintext: 14 →O

The result is "Hello"

Ex. 2.3.6 : Using Affine Cipher, encrypt the plaintext "SECURITY" with key pair (5, 2).

☑ **Soln. :** We apply the encryption algorithm to the plaintext, character by character.

$$C = (P \times K_1\ 0 + K_2) \bmod 26$$

Given $K_1 = 5$ and $K_2 = 2$

Plaintext: S → 18	Encryption: (18 x 5 + 2) mod 26	Ciphertext: 14 → O
Plaintext: E → 04	Encryption: (04 x 5 + 2) mod 26	Ciphertext: 22 → W
Plaintext: C → 02	Encryption: (02 x 5 + 2) mod 26	Ciphertext: 12 → M
Plaintext: U → 20	Encryption: (20 x 5 + 2) mod 26	Ciphertext: 24 → Y
Plaintext: R →17	Encryption: (17 x 5 + 2) mod 26	Ciphertext: 09 → J
Plaintext: I → 08	Encryption: (08 x 5 + 2) mod 26	Ciphertext: 16 → Q
Plaintext: T → 19	Encryption: (19 x 5 + 2) mod 26	Ciphertext: 19 → T
Plaintext: Y → 24	Encryption: (24 x 5 + 2) mod 26	Ciphertext: 18 → S

The result is "OWMYJQTS".

2.3.3 Polyalphabetic Ciphers

UQ. What is cryptography? Explain polyalphabetic ciphering with suitable example.

(SPPU - Q. 3(a), May 2015, 8 Marks)

- In polyalphabetic substitution, each occurrence of a character may have a different substitution character.
- The relationship between a character in the plaintext to a character in the ciphertext is one-to-many. For example, "A" could be enciphered as "B" in the beginning of the text, but as "D" at the middle.
- In polyalphabetic cipher, we need to have a key stream $K = (K_1, K_2, K_3, \ldots)$ in which K_i is used to encipher the i^{th} character in the plaintext to create the i^{th} character in the ciphertext.

2.3.4 Autokey Cipher

- The key in the autokey cipher is a stream of subkeys, each of which is used to encrypt the plaintext character it corresponds to.
- The first subkey is a secretly agreed-upon value among the communicating parties. The value of the first plaintext character is the second subkey (between 0 and 25).
- The value of the second plaintext character is the third subkey, and so on.
- Given plaintext $P = P_1P_2P_3\ldots$ and key $K = (K_1, P_1, P_2, \ldots)$

  Encryption : $C_i = (P_i + K_i) \bmod 26$

  Decryption: $P_i = (C_i - K_i) \bmod 26$
- The cipher's name, autokey, implies that the subkeys are generated automatically during the encryption process from the plaintext cipher characters.

Ex. 2.3.7 : Encrypt the message "ATTACK IS TODAY" using autokey cipher with key = 12. Ignore the space between words.

☑ **Soln. :** Encryption is done character by character. Each character in the plaintext is first replaced by its integer value. The first subkey is added to create the first ciphertext character. The rest of the key is created as the plaintext characters are read.

▶ **Encryption : $C_i = (P_i + K_i) \bmod 26$**

Plaintext	A	T	T	A	C	K	I	S	T	O	D	A	Y
P's Values	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream	12	00	19	19	00	02	10	08	18	19	14	03	00
C's Values	12	19	12	19	02	12	18	00	11	07	17	03	24
Ciphertext	**M**	**T**	**M**	**T**	**C**	**M**	**S**	**A**	**L**	**H**	**R**	**D**	**Y**

The result is "MYMTCMSALHRDY".

Note : The cipher is polyalphabetic because three occurrences of "A" in the plaintext are encrypted differently. The three occurrences of "T" are also encrypted differently.

2.3.5 Playfair Cipher

UQ. Construct a Playfair matrix with the key largest.

Construct a Playfair matrix with the key occurrence. Make a reasonable assumption about how to treat redundant letters in the key.

Using the following Playfair matrix encrypt the message: Must see you over Cadogan West: Coming at once.

(SPPU - Q. 2(b), May 2012, 10 Marks)

M	F	H	I/J	K
U	N	O	P	Q
Z	V	W	X	Y
E	L	A	R	G
D	S	T	B	C

- The Playfair cipher was the first practical digraph substitution cipher.
- The scheme was invented in 1854 by Charles Wheatstone, but was named after Lord Playfair who promoted the use of the cipher.
- The technique encrypts pairs of letters (digraphs), instead of single letters as in the simple substitution cipher.
- The Playfair is significantly harder to break since the frequency analysis used for simple substitution ciphers does not work with it. Frequency analysis can still be undertaken, but on the 25*25 = 625 possible digraphs rather than the 25 possible monographs. Frequency analysis thus requires much more ciphertext in order to work.
- It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the Australians during World War II. This was because Playfair is reasonably fast to use and requires no special equipment.
- It initially creates a key-table of 5x5 matrix.
- The matrix contains alphabets that act as the key for encryption of the plaintext. Note that any alphabet should not be repeated. Another point to note that there are 26 alphabets and we have only 25 blocks to put a letter inside it. So, J is always combined with I.

☞ Playfair Cipher Encryption Rules

(1) First, split the plaintext into **digraphs** (pair of two letters).

If the plaintext has the odd number of letters, append the letter **Z** at the end of the plaintext. It makes the plaintext of **even.** For example, the plaintext **MANGO** has five letters. So, it is not possible to make a digraph. So, we will append a letter **Z** at the end of the plaintext, i.e. **MANGOZ.**

(2) If any letter appears twice (side by side), put **X** at the place of the second occurrence.

Suppose, the plaintext is **COMMUNICATE**, then its digraph becomes **CO MX MU NI CA TE**. Similarly, the digraph for the plaintext **JAZZ** will be **JA ZX ZX**, and for plaintext **GREET**, the digraph will be **GR EX ET.**

(3) To determine the cipher (encryption) text, first, build a 5 × 5 key-matrix or key-table and fill it with the letters of alphabets, as directed below:

- Fill the first row (left to right) with the letters of the given keyword (say, **ATHENS**). If the keyword has duplicate letters (if any) avoid them. It means a letter will be considered only once. After that, fill the remaining letters in alphabetical order. Let's create a 5 × 5 key-matrix for the keyword **ATHENS**.

Note that in the below matrix any letter is not repeated. The letters in the first row (in bold) represent the keyword and the remaining letters sets in alphabetical order.

A	**T**	**H**	**E**	**N**
S	B	C	D	F
G	I/J	K	L	M
O	P	Q	R	U
V	W	X	Y	Z

(4) There may be the following three conditions :

(i) If a pair of letters (digraph) appears in the same row : In this case, replace each letter of the digraph with the letters immediately to their right. If there is no letter to the right, consider the first letter of the same row as the right letter. Suppose, Z is a letter whose right letter is required, in such case, T will be right to Z.

(ii) If a pair of letters (digraph) appears in the same column : In this case, replace each letter of the digraph with the letters immediately below them. If there is no letter below, wrap around to the top of the same column. Suppose, **X** is a letter who's below letter is required, in such case, **H** will be below X.

(iii) If a pair of letters (digraph) appears in a different row and different column : If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. The order is important – the first encrypted letter of the pair is the one that lies on the same row as the first plaintext letter. For example, 'BQ' will be encrypted as 'CP', 'DV' will be encrypted as 'SY'.

☞ Playfair Cipher Decryption

The decryption procedure is the same as encryption but the steps are applied in **reverse** order. For decryption cipher is symmetric (move left along rows and up along columns). The receiver of the plain text has the same key and can create the same key-table that is used to decrypt the message.

Ex. 2.3.8 : Encrypt "COMMUNICATE" with Playfair Cipher using key "COMPUTER".

☑ **Soln. :**

(1) First, split the plaintext into digraph as **CO MX MU NI CA TE.**

(2) Construct a 5*5 key-matrix. In our case, the key is **COMPUTER**.

C	**O**	**M**	**P**	**U**
T	**E**	**R**	A	B
D	F	G	H	I/J
K	L	N	Q	S
V	W	X	Y	Z

(3) Now, we will traverse in key-matrix pair by pair and find the corresponding encipher for the pair.

- The first digraph is **CO**. The pair appears in the same row. In this case, replace each letter of the digraph with the letters immediately to their right. **CO** gets encipher into **OM**.
- The second digraph is **MX**. The pair appears in the same column. In this case, replace each letter of the digraph with the letters immediately below them. **MX** gets encipher into **RM**.
- The third digraph is **MU**. The pair appears in the same row. In this case, replace each letter of the digraph with the letters immediately to their right. **MU** gets encipher into **PC**.

- The fourth digraph is **NI**. The pair appears in different rows and different columns. If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. **NI** gets encipher into **SG**.
- The fifth digraph is **CA**. The pair appears in different rows and different columns. The pair appears in different rows and different columns. If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. **CA** gets encipher into **PT**.
- The sixth digraph is **TE**. The pair appears in the same row. In this case, replace each letter of the digraph with the letters immediately to their right. **TE** gets encipher into **ER**.

Therefore, the plaintext **COMMUNICATE** gets encipher (encrypted) into **OMRMPCSGPTER.**

Ex. 2.3.9 : Encrypt "THIS IS THE FINAL EXAM" with Playfair Cipher using the key "GUIDANCE".

☑ **Soln. :**

(1) First, split the plaintext into digraph as **TH IS IS TH EF IN AL EX AM**.

(2) Construct a 5*5 key-matrix. In our case, the key is **GUIDANCE**.

G	U	I	D	A
N	C	E	B	F
H	K	L	M	O
P	Q	R	S	T
V	W	X	Y	Z

(3) Now, we will traverse in key-matrix pair by pair and find the corresponding encipher for the pair.

- The first digraph is **TH**. The pair appears in different rows and different columns. If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. **TH** gets encipher into **PO**.
- The second digraph is **IS**. The pair appears in different rows and different columns. If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. **IS** gets encipher into **DR**.
- The third digraph is **IS**. The pair appears in different rows and different columns. If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. **IS** gets encipher into **DR**.
- The fourth digraph is **TH**. The pair appears in different rows and different columns. If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. TH gets encipher into PO.
- The fifth digraph is EF. In this case, replace each letter of the digraph with the letters immediately to their right. EF gets encipher into BN.
- The sixth digraph is IN. The pair appears in different rows and different columns. If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. IN gets encipher into GE.
- The seventh digraph is AL. The pair appears in different rows and different columns. If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. AL gets encipher into IO.
- The eighth digraph is EX. The pair appears in the same column. In this case, replace each letter of the digraph with the letters immediately below them. EX gets encipher into LI.
- The ninth digraph is AM. The pair appears in different rows and different columns. If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. **AM** gets encipher into **DO**.

Therefore, the plaintext **THIS IS THE FINAL EXAM** gets encipher (encrypted) into **PODRDRPOBNGEIOLIDO.**

UEx. 2.3.10 (SPPU - Q. 2(b), May 2012,4 Marks)

Construct a Playfair matrix with the key largest.

☑ **Soln. :**

L	A	R	G	E
S	T	B	C	D
F	H	I/J	K	M
N	O	P	Q	U
V	W	X	Y	Z

UEx. 2.3.11 (SPPU - Q. 2(b), May 2012, 4 Marks)

Construct a Playfair matrix with the key occurrence. Make a reasonable assumption about how to treat redundant letters in the key.

☑ **Soln. :**

O	C	U	R	E
N	A	B	D	F
G	H	I/J	K	L
M	P	Q	S	T
V	W	X	Y	Z

UEx. 2.3.12 (SPPU - Q. 2(b), May 2012, 4 Marks)

Using the following Playfair matrix encrypt the message: Must see you over Cadogan West: Coming at once.

☑ **Soln. :**

The given 5*5 playfair matrix is:

M	F	H	I/J	K
U	N	O	P	Q
Z	V	W	X	Y
E	L	A	R	G
D	S	T	B	C

First, split the plaintext into digraph as

MU ST SE EY OU OV ER CA DO GA NW ES TC OM IN GA TO NC EZ

The encrypted message is:

UZ TB DL GZ PN NW LG TG TU ER OV LD BD UH FP ER HW QS FE

2.3.5(A) Vigenere Cipher

- The Vigenère cipher is an example of a polyalphabetic substitution cipher. A polyalphabetic substitution cipher is similar to a monoalphabetic substitution except that the cipher alphabet is changed periodically while enciphering the message. This makes the cipher less vulnerable to cryptanalysis using letter frequencies.
- Blaise de Vigenère developed what is now called the Vigenère cipher in 1585. He used a table known as the Vigenère square, to encipher messages as shown in Table 2.3.1.
- In addition to the plaintext, the Vigenère cipher also requires a keyword, which is repeated so that the total length is equal to that of the plaintext.
- To encrypt, pick a letter in the plaintext and its corresponding letter in the keyword, use the keyword letter and the plaintext letter as the row index and column index, respectively, and the entry at the row-column intersection is the letter in the ciphertext.
- For example, the first letter in the plaintext is **M** and its corresponding keyword letter is **H**. This means that the row of **H** and the column of **M** are used, and the entry **T** at the intersection is the encrypted result.
- The Vigenère cipher uses a different strategy to create the key stream. The key stream is a repetition of an initial secret key stream of length 'm', where $1 \leq m \leq 26$.
- The Vigenère key stream does not depend on the plaintext characters; it depends only on the position of the character in the plaintext.
- The Vigenère cipher can be seen as combinations of 'm' additive ciphers.
- The general formula of encryption using Vigenère cipher is: $C_i = (P_i + K_i) \bmod 26$. The general formula of decryption using Vigenère cipher is: $P_i = (C_i - K_i) \bmod 26$
- The Vigenère cipher do not preserve the frequency of characters, however, the intercepted ciphertext can be deciphered by finding the length of the key and finding the key itself.

Ex. 2.3.13 : Use the Vigenère cipher with keyword "HEALTH" to encipher the message "LIFE IS FULL OF SURPRISES".

☑ **Soln. :** The general formula of encryption using Vigenère cipher is:

$$C_i = (P_i + K_i) \bmod 26$$

Given keyword : HEALTH

Plaintext: LIFEISFULLOFSURPRISES

Plaintext	L	I	F	E	I	S	F	U	L	L	O	F	S	U	R	P	R	I	S	E	S
P's Values	11	08	05	04	08	18	05	20	11	11	14	05	18	20	17	15	17	08	18	04	18
Key Stream	H	E	A	L	T	H	H	E	A	L	T	H	H	E	A	L	T	H	H	E	A
K's Values	07	04	00	11	19	07	07	04	00	11	19	07	07	04	00	11	19	07	07	04	00
C's Values	18	12	05	15	01	25	12	24	11	22	07	12	25	24	17	00	10	15	25	08	18
Ciphertext	**S**	**M**	**F**	**P**	**B**	**Z**	**M**	**Y**	**L**	**W**	**H**	**M**	**Z**	**Y**	**R**	**A**	**K**	**P**	**Z**	**I**	**S**

The result is "SMFPBZMYLWHMZYRAKPZIS"

Table 2.3.1 : Vigenere Square

KEY \ PLAINTEXT	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Ex. 2.3.14 : Use the Vigenère cipher with keyword "DECEPTIVE" to encipher the message "WE ARE DISCOVERED SAVE YOURSELF".

☑ **Soln. :** The general formula of encryption using Vigenère cipher is: $C_i = (P_i + K_i) \bmod 26$

Given keyword : DECEPTIVE

Plaintext : WEAREDISCOVEREDSAVEYOURSELF

Plaintext	W	E	A	R	E	D	I	S	C	O	V	E	R	E	D	S	A	V	E	Y	O	U	R	S	E	L	F
P's Values	22	04	00	17	04	03	08	18	02	14	21	04	17	04	03	18	00	21	04	24	14	20	17	18	04	11	05
Key Stream	D	E	C	E	P	T	I	V	E	D	E	C	E	P	T	I	V	E	D	E	C	E	P	T	I	V	E
K's Values	03	04	02	04	15	19	08	21	04	03	04	02	04	15	19	08	21	04	03	04	02	04	15	19	08	21	04
C's Values	25	08	02	21	19	22	16	13	06	17	25	06	21	19	22	00	21	25	07	02	16	24	06	11	12	06	09
Ciphertext	Z	I	C	V	T	W	Q	N	G	R	Z	G	V	T	W	A	V	Z	H	C	Q	Y	G	L	M	G	J

The result is "ZICVTWQNGRZGVTWAVZHCQYGLMGJ".

2.3.5(B) Hill Cipher

UQ. Explain Hill ciphering developed by Lester Hill in detail with suitable example.

(SPPU - Q. 1(b), Dec. 2014, 10 Marks)

UQ. Using Hill Cipher encrypt the message 'ESSENTIAL'. The key for encryption is 'ANOTHERBZ'.

(SPPU - Q. 1(b), May 2019, 5 Marks)

- Hill Cipher in cryptography was invented and developed in 1929 by Lester S. Hill, a renowned American mathematician. Hill Cipher represents a polygraphic substitution cipher that follows a uniform substitution across multiple levels of blocks.
- Here, polygraphic substitution cipher defines that Hill Cipher can work seamlessly with digraphs (two-letter blocks), trigraphs (three-letter blocks), or any multiple-sized blocks for building a uniform cipher.
- Hill Cipher is based on a particular mathematical topic of linear Algebra and the sophisticated use of matrices in general, as well as rules for modulo arithmetic.
- The way Hill Cipher works is explained below:
  (1) Treat every letter in the plaintext message as a number such that A = 00, B = 01, ..., Z = 25.
  (2) Organize the plaintext message as a matrix of numbers based on the above conversion. For example, if the plaintext is ATT. Based on the above step, we know that A = 00, T =19. Therefore, our plaintext would look as follows: $\begin{pmatrix} 00 \\ 19 \\ 19 \end{pmatrix}$
  (3) Now, the plaintext matrix is multiplied by a matrix of randomly chosen keys. The key matrix consists of size n x n, where n is the number of rows in the plaintext. For example, we take the following matrix:
  $$\begin{pmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 17 & 7 \end{pmatrix}$$
  (4) Now, multiply the two matrices as shown below:
  $$\begin{pmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 17 & 7 \end{pmatrix} \times \begin{pmatrix} 00 \\ 19 \\ 19 \end{pmatrix} = \begin{pmatrix} 171 \\ 57 \\ 456 \end{pmatrix}$$

(5) Now compute a modulo 26 value of the above matrix. That is, take the remainder after dividing the above matrix values by 26.

$$\begin{pmatrix} 171 \\ 57 \\ 456 \end{pmatrix} \bmod 26 = \begin{pmatrix} 15 \\ 05 \\ 14 \end{pmatrix}$$

(6) Now, translate the numbers to alphabets. 15 = P, 05 = F, 14 = O. Therefore, the ciphertext is "PFO".

(7) For decryption, take the ciphertext matrix and multiply it by the inverse of original key matrix.

(8) After this take modulo 26 of this matrix.

(9) Now, translate the numbers to alphabets. You will get the original plaintext back successfully.

- Hill cipher is vulnerable to the known-plaintext attack. This is because it is linear due to the possibility to compute smaller factors of the matrices, work on them individually, and then join them back as and when they are ready.

Ex. 2.3.15 : Use a Hill cipher to encipher the message "WE LIVE IN AN INSECURE WORLD".

Use the following key: $K = \begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix}$

☑ **Soln. :** The key matrix consists of size 2×2, where 2 is the number of rows in the plaintext. Hence, we divide the given plaintext in matrix of size 1×2 as below.

$$\begin{pmatrix} W \\ E \end{pmatrix}, \begin{pmatrix} L \\ I \end{pmatrix}, \begin{pmatrix} V \\ E \end{pmatrix}, \begin{pmatrix} I \\ N \end{pmatrix}, \begin{pmatrix} A \\ N \end{pmatrix}, \begin{pmatrix} I \\ N \end{pmatrix}, \begin{pmatrix} S \\ E \end{pmatrix}, \begin{pmatrix} C \\ U \end{pmatrix}, \begin{pmatrix} R \\ E \end{pmatrix}, \begin{pmatrix} W \\ O \end{pmatrix}, \begin{pmatrix} R \\ L \end{pmatrix}, \begin{pmatrix} D \\ Z \end{pmatrix}$$

Now organize the plaintext message as a matrix of numbers.

$$\begin{pmatrix} 22 \\ 04 \end{pmatrix}, \begin{pmatrix} 11 \\ 08 \end{pmatrix}, \begin{pmatrix} 21 \\ 04 \end{pmatrix}, \begin{pmatrix} 08 \\ 13 \end{pmatrix}, \begin{pmatrix} 00 \\ 13 \end{pmatrix}, \begin{pmatrix} 08 \\ 13 \end{pmatrix}, \begin{pmatrix} 18 \\ 04 \end{pmatrix}, \begin{pmatrix} 02 \\ 20 \end{pmatrix}, \begin{pmatrix} 17 \\ 04 \end{pmatrix}, \begin{pmatrix} 22 \\ 14 \end{pmatrix}, \begin{pmatrix} 17 \\ 11 \end{pmatrix}, \begin{pmatrix} 03 \\ 25 \end{pmatrix}$$

Now, multiply each plaintext matrix with the key matrix and perform modulo 26 operations on the product.

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 22 \\ 04 \end{pmatrix} \bmod 26 = \begin{pmatrix} 74 \bmod 26 \\ 138 \bmod 26 \end{pmatrix} = \begin{pmatrix} 22 \\ 8 \end{pmatrix} = \begin{matrix} W \\ I \end{matrix}$$

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 11 \\ 08 \end{pmatrix} \bmod 26 = \begin{pmatrix} 49 \bmod 26 \\ 111 \bmod 26 \end{pmatrix} = \begin{pmatrix} 23 \\ 7 \end{pmatrix} = \begin{matrix} X \\ H \end{matrix}$$

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 21 \\ 04 \end{pmatrix} \bmod 26 = \begin{pmatrix} 71 \bmod 26 \\ 133 \bmod 26 \end{pmatrix} = \begin{pmatrix} 19 \\ 03 \end{pmatrix} = \begin{matrix} T \\ D \end{matrix}$$

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 08 \\ 13 \end{pmatrix} \bmod 26 = \begin{pmatrix} 50 \bmod 26 \\ 131 \bmod 26 \end{pmatrix} = \begin{pmatrix} 24 \\ 01 \end{pmatrix} = \begin{matrix} Y \\ B \end{matrix}$$

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 00 \\ 13 \end{pmatrix} \bmod 26 = \begin{pmatrix} 26 \bmod 26 \\ 91 \bmod 26 \end{pmatrix} = \begin{pmatrix} 00 \\ 13 \end{pmatrix} = \begin{matrix} A \\ N \end{matrix}$$

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 08 \\ 13 \end{pmatrix} \bmod 26 = \begin{pmatrix} 50 \bmod 26 \\ 131 \bmod 26 \end{pmatrix} = \begin{pmatrix} 24 \\ 01 \end{pmatrix} = \begin{matrix} Y \\ B \end{matrix}$$

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 18 \\ 04 \end{pmatrix} \bmod 26 = \begin{pmatrix} 62 \bmod 26 \\ 118 \bmod 26 \end{pmatrix} = \begin{pmatrix} 10 \\ 14 \end{pmatrix} = \begin{matrix} K \\ O \end{matrix}$$

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 02 \\ 20 \end{pmatrix} \bmod 26 = \begin{pmatrix} 46 \bmod 26 \\ 150 \bmod 26 \end{pmatrix} = \begin{pmatrix} 20 \\ 20 \end{pmatrix} = \begin{matrix} U \\ U \end{matrix}$$

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 17 \\ 04 \end{pmatrix} \bmod 26 = \begin{pmatrix} 59 \bmod 26 \\ 113 \bmod 26 \end{pmatrix} = \begin{pmatrix} 7 \\ 9 \end{pmatrix} = \begin{matrix} H \\ J \end{matrix}$$

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 22 \\ 14 \end{pmatrix} \bmod 26 = \begin{pmatrix} 16 \bmod 26 \\ 208 \bmod 26 \end{pmatrix} = \begin{pmatrix} 16 \\ 00 \end{pmatrix} = \begin{matrix} Q \\ A \end{matrix}$$

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 17 \\ 11 \end{pmatrix} \bmod 26 = \begin{pmatrix} 73 \bmod 26 \\ 162 \bmod 26 \end{pmatrix} = \begin{pmatrix} 21 \\ 6 \end{pmatrix} = \begin{matrix} V \\ G \end{matrix}$$

$$\begin{pmatrix} 03 & 02 \\ 05 & 07 \end{pmatrix} \times \begin{pmatrix} 03 \\ 25 \end{pmatrix} \bmod 26 = \begin{pmatrix} 59 \bmod 26 \\ 190 \bmod 26 \end{pmatrix} = \begin{pmatrix} 7 \\ 8 \end{pmatrix} = \begin{matrix} H \\ I \end{matrix}$$

The result is "WIXHTDYBANYBKOUUHJQAVGHI".

Ex. 2.3.16 : Use a Hill cipher to encipher the message "ATTACK AT DAWN". Use the following key

$$K = \begin{pmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 17 & 7 \end{pmatrix}$$

☑ **Soln. :** The key matrix consists of size 3 × 3, where 3 is the number of rows in the plaintext. Hence, we divide the given plaintext in matrix of size 1 × 3 as below.

$$\begin{pmatrix} A \\ T \\ T \end{pmatrix}, \begin{pmatrix} A \\ C \\ K \end{pmatrix}, \begin{pmatrix} A \\ T \\ D \end{pmatrix}, \begin{pmatrix} A \\ W \\ N \end{pmatrix}$$

Now organize the plaintext message as a matrix of numbers.

$$\begin{pmatrix} 00 \\ 19 \\ 19 \end{pmatrix}, \begin{pmatrix} 00 \\ 02 \\ 10 \end{pmatrix}, \begin{pmatrix} 00 \\ 19 \\ 03 \end{pmatrix}, \begin{pmatrix} 00 \\ 22 \\ 13 \end{pmatrix}$$

Now, multiply each plaintext matrix with the key matrix and perform modulo 26 operations on the product.

$$\begin{pmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 17 & 7 \end{pmatrix} \times \begin{pmatrix} 00 \\ 19 \\ 19 \end{pmatrix} \bmod 26 = \begin{pmatrix} 171 \bmod 26 \\ 57 \bmod 26 \\ 456 \bmod 26 \end{pmatrix} = \begin{pmatrix} 15 \\ 05 \\ 14 \end{pmatrix} = \begin{matrix} P \\ F \\ O \end{matrix}$$

$$\begin{pmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 17 & 7 \end{pmatrix} \times \begin{pmatrix} 00 \\ 02 \\ 10 \end{pmatrix} \bmod 26 = \begin{pmatrix} 58 \bmod 26 \\ 14 \bmod 26 \\ 104 \bmod 26 \end{pmatrix} = \begin{pmatrix} 06 \\ 14 \\ 00 \end{pmatrix} = \begin{matrix} G \\ O \\ A \end{matrix}$$

$$\begin{pmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 17 & 7 \end{pmatrix} \times \begin{pmatrix} 00 \\ 19 \\ 03 \end{pmatrix} \bmod 26 = \begin{pmatrix} 91 \bmod 26 \\ 41 \bmod 26 \\ 344 \bmod 26 \end{pmatrix} = \begin{pmatrix} 13 \\ 15 \\ 06 \end{pmatrix} = \begin{matrix} N \\ P \\ G \end{matrix}$$

$$\begin{pmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 17 & 7 \end{pmatrix} \times \begin{pmatrix} 00 \\ 22 \\ 13 \end{pmatrix} \bmod 26 = \begin{pmatrix} 153 \bmod 26 \\ 57 \bmod 26 \\ 465 \bmod 26 \end{pmatrix} = \begin{pmatrix} 23 \\ 5 \\ 23 \end{pmatrix} = \begin{matrix} X \\ F \\ X \end{matrix}$$

The result is "PFOGOANPGXFX".

$$\begin{pmatrix}03 & 02\\05 & 07\end{pmatrix}\times\begin{pmatrix}17\\04\end{pmatrix}\bmod 26=\begin{pmatrix}59\bmod 26\\113\bmod 26\end{pmatrix}=\begin{pmatrix}7\\9\end{pmatrix}=\begin{matrix}H\\J\end{matrix}$$

$$\begin{pmatrix}03 & 02\\05 & 07\end{pmatrix}\times\begin{pmatrix}22\\14\end{pmatrix}\bmod 26=\begin{pmatrix}16\bmod 26\\208\bmod 26\end{pmatrix}=\begin{pmatrix}16\\00\end{pmatrix}=\begin{matrix}Q\\A\end{matrix}$$

$$\begin{pmatrix}03 & 02\\05 & 07\end{pmatrix}\times\begin{pmatrix}17\\11\end{pmatrix}\bmod 26=\begin{pmatrix}73\bmod 26\\162\bmod 26\end{pmatrix}=\begin{pmatrix}21\\6\end{pmatrix}=\begin{matrix}V\\G\end{matrix}$$

$$\begin{pmatrix}03 & 02\\05 & 07\end{pmatrix}\times\begin{pmatrix}03\\25\end{pmatrix}\bmod 26=\begin{pmatrix}59\bmod 26\\190\bmod 26\end{pmatrix}=\begin{pmatrix}7\\8\end{pmatrix}=\begin{matrix}H\\I\end{matrix}$$

The result is "WIXHTDYBANYBKOUUHJQAVGHI".

Ex. 2.3.16 : Use a Hill cipher to encipher the message "ATTACK AT DAWN". Use the following key

$$K=\begin{pmatrix}2 & 4 & 5\\9 & 2 & 1\\3 & 17 & 7\end{pmatrix}$$

☑ **Soln. :** The key matrix consists of size 3×3, where 3 is the number of rows in the plaintext. Hence, we divide the given plaintext in matrix of size 1×3 as below.

$$\begin{pmatrix}A\\T\\T\end{pmatrix},\begin{pmatrix}A\\C\\K\end{pmatrix},\begin{pmatrix}A\\T\\D\end{pmatrix},\begin{pmatrix}A\\W\\N\end{pmatrix}$$

Now organize the plaintext message as a matrix of numbers.

$$\begin{pmatrix}00\\19\\19\end{pmatrix},\begin{pmatrix}00\\02\\10\end{pmatrix},\begin{pmatrix}00\\19\\03\end{pmatrix},\begin{pmatrix}00\\22\\13\end{pmatrix}$$

Now, multiply each plaintext matrix with the key matrix and perform modulo 26 operations on the product.

$$\begin{pmatrix}2 & 4 & 5\\9 & 2 & 1\\3 & 17 & 7\end{pmatrix}\times\begin{pmatrix}00\\19\\19\end{pmatrix}\bmod 26=\begin{pmatrix}171\bmod 26\\57\bmod 26\\456\bmod 26\end{pmatrix}=\begin{pmatrix}15\\05\\14\end{pmatrix}=\begin{matrix}P\\F\\O\end{matrix}$$

$$\begin{pmatrix}2 & 4 & 5\\9 & 2 & 1\\3 & 17 & 7\end{pmatrix}\times\begin{pmatrix}00\\02\\10\end{pmatrix}\bmod 26=\begin{pmatrix}58\bmod 26\\14\bmod 26\\104\bmod 26\end{pmatrix}=\begin{pmatrix}06\\14\\00\end{pmatrix}=\begin{matrix}G\\O\\A\end{matrix}$$

$$\begin{pmatrix}2 & 4 & 5\\9 & 2 & 1\\3 & 17 & 7\end{pmatrix}\times\begin{pmatrix}00\\19\\03\end{pmatrix}\bmod 26=\begin{pmatrix}91\bmod 26\\41\bmod 26\\344\bmod 26\end{pmatrix}=\begin{pmatrix}13\\15\\06\end{pmatrix}=\begin{matrix}N\\P\\G\end{matrix}$$

$$\begin{pmatrix}2 & 4 & 5\\9 & 2 & 1\\3 & 17 & 7\end{pmatrix}\times\begin{pmatrix}00\\22\\13\end{pmatrix}\bmod 26=\begin{pmatrix}153\bmod 26\\57\bmod 26\\465\bmod 26\end{pmatrix}=\begin{pmatrix}23\\5\\23\end{pmatrix}=\begin{matrix}X\\F\\X\end{matrix}$$

The result is "PFOGOANPGXFX".

Ex. 2.3.17 : If the plaintext "FRIDAY" is encrypted using a 2×2 Hill cipher to yield the cipher text "PQCFKU", determine the key used for encryption and decryption.

☑ **Soln. :** Given plaintext is "FRIDAY" where numeric equivalent of each character is as follows:

F = 05, R = 17, I = 08, D = 03, A = 00, Y = 24

Given ciphertext = "PQCFKU" where numeric equivalent of each character is as follows:

P = 15, Q = 16, C = 02, F = 05, K = 10, U = 20

Given 2×2 Hill Cipher for encryption.

We have, $[C] = [K]_{2\times2} [P] \bmod 26$

i.e. $[K]_{2\times2} = [C] [P]^{-1} \bmod 26$

Let's take 2-characters of plaintext and its corresponding ciphertext.

$$\begin{bmatrix} 15 \\ 16 \end{bmatrix} = [K]_{2\times2} \begin{bmatrix} 05 \\ 17 \end{bmatrix} \bmod 26$$

Similarly, take another 2-characters of plaintext and its corresponding ciphertext.

$$\begin{bmatrix} 02 \\ 05 \end{bmatrix} = [K]_{2\times2} \begin{bmatrix} 08 \\ 03 \end{bmatrix} \bmod 26$$

Now, combine the above to form 2×2 matrix.

$$\begin{bmatrix} 15 & 02 \\ 16 & 05 \end{bmatrix} = [K]_{2\times2} \begin{bmatrix} 05 & 08 \\ 17 & 03 \end{bmatrix} \bmod 26$$

$$\therefore [K]_{2\times2} = \begin{bmatrix} 15 & 02 \\ 16 & 05 \end{bmatrix} \times \begin{bmatrix} 05 & 08 \\ 17 & 03 \end{bmatrix}^{-1} \bmod 26$$

To find $\begin{bmatrix} 05 & 08 \\ 17 & 03 \end{bmatrix}^{-1} \bmod 26$:

Find determinant:

$$\begin{vmatrix} 05 & 08 \\ 17 & 03 \end{vmatrix} \bmod 26 = -121 \bmod 26 = 9$$

Now find $\frac{1}{9}\begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} \bmod 26$

$$9^{-1}\begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} \bmod 26 \equiv 1 \bmod 26$$

Multiplicative inverse of 9 modulo 26 = 3.

Also, 3 mod 26 = 3, $-8 \bmod 26 = 18$, $-17 \bmod 26 = 9$ and 5 mod 26 = 5

$$\therefore \begin{bmatrix} 05 & 08 \\ 17 & 03 \end{bmatrix}^{-1} \bmod 26 = 3\begin{bmatrix} 3 & 18 \\ 9 & 5 \end{bmatrix} \bmod 26 = \begin{bmatrix} 9 & 2 \\ 1 & 15 \end{bmatrix}$$

$$\therefore [K]_{2\times2} = \begin{bmatrix} 15 & 2 \\ 16 & 5 \end{bmatrix} \times \begin{bmatrix} 9 & 2 \\ 1 & 15 \end{bmatrix} \bmod 26 = \begin{bmatrix} 137 & 60 \\ 149 & 107 \end{bmatrix} \bmod 26 = \begin{bmatrix} 7 & 8 \\ 19 & 3 \end{bmatrix}$$

Similarly, results can be verified by taking other pairs of plaintext and ciphertext.

UEx. 2.3.18 (SPPU - Q. 1(b), May 2019, 5 Marks)

Using Hill Cipher encrypt the message 'ESSENTIAL'. The key for encryption is 'ANOTHERBZ'

☑ **Soln. :** We assume the key matrix consists of size 3 × 3, where 3 is the number of rows in the plaintext. Hence, we divide the given plaintext in matrix of size 1 × 3 as below.

$$\begin{pmatrix} E \\ S \\ S \end{pmatrix}, \begin{pmatrix} E \\ N \\ T \end{pmatrix}, \begin{pmatrix} I \\ A \\ L \end{pmatrix}$$

Now organize the plaintext message as a matrix of numbers.

$$\begin{pmatrix} 4 \\ 18 \\ 18 \end{pmatrix} \begin{pmatrix} 4 \\ 13 \\ 19 \end{pmatrix} \begin{pmatrix} 8 \\ 0 \\ 4 \end{pmatrix}$$

We organize the key as 3*3 matrix as follows:

$$\begin{pmatrix} A & N & O \\ T & H & E \\ R & B & Z \end{pmatrix} = \begin{pmatrix} 0 & 13 & 14 \\ 19 & 7 & 4 \\ 17 & 1 & 25 \end{pmatrix}$$

Now, multiply each plaintext matrix with the key matrix and perform modulo 26 operations on the product.

$$\begin{pmatrix} 0 & 13 & 14 \\ 19 & 7 & 4 \\ 17 & 1 & 25 \end{pmatrix} \times \begin{pmatrix} 4 \\ 18 \\ 18 \end{pmatrix} \bmod 26 = \begin{pmatrix} 486 \bmod 26 \\ 274 \bmod 26 \\ 536 \bmod 26 \end{pmatrix} = \begin{pmatrix} 18 \\ 14 \\ 16 \end{pmatrix} = \begin{pmatrix} S \\ O \\ Q \end{pmatrix}$$

$$\begin{pmatrix} 0 & 13 & 14 \\ 19 & 7 & 4 \\ 17 & 1 & 25 \end{pmatrix} \times \begin{pmatrix} 4 \\ 13 \\ 19 \end{pmatrix} \bmod 26 = \begin{pmatrix} 4356 \bmod 26 \\ 243 \bmod 26 \\ 556 \bmod 26 \end{pmatrix} = \begin{pmatrix} 19 \\ 9 \\ 10 \end{pmatrix} = \begin{pmatrix} T \\ J \\ K \end{pmatrix}$$

$$\begin{pmatrix} 0 & 13 & 14 \\ 19 & 7 & 4 \\ 17 & 1 & 25 \end{pmatrix} \times \begin{pmatrix} 8 \\ 0 \\ 4 \end{pmatrix} \bmod 26 = \begin{pmatrix} 56 \bmod 26 \\ 168 \bmod 26 \\ 236 \bmod 26 \end{pmatrix} = \begin{pmatrix} 4 \\ 12 \\ 2 \end{pmatrix} = \begin{pmatrix} E \\ M \\ C \end{pmatrix}$$

The result is "SOQTJKEMC"

2.4 TRANSPOSITION CIPHERS

GQ. Explain Transposition Ciphers with illustrative examples.

UQ. What is transposition scheme of cryptography and Explain any one method of it with suitable example. (SPPU - Q. 2(a), May 2017, 10 Marks)

UQ. Use Transposition Cipher to encrypt plain text 'I love my India' and use the key 'HEAVEN'. (SPPU - Q. 2(b), May 2019, 5 Marks)

- A transposition cipher does not substitute one symbol for another, instead it changes the location of the symbols i.e. it performs some permutation over the plaintext.
- In other words, a transposition cipher reorders (transposes) the symbols.
- A symbol in the first position of the plaintext may appear in the fifth position of the ciphertext. A symbol in the sixth position of the plaintext may appear in the second position of the ciphertext.
- Transposition ciphers are vulnerable to several kinds of ciphertext-only attacks.

2.4.1 Keyless Transposition Ciphers

- These are simple transposition ciphers are used in past and are keyless.
- There are two methods for permutation of characters.
- In the first method, the text is written into a table, column by column and then transmitted row by row. It is also called Rail-Fence cipher wherein the plaintext is

arranged in two lines in a zigzag pattern and the ciphertext is created reading the pattern row by row.

- In the second method, the text is written into the table row by row and then transmitted column by column. The number of columns will be given.

Ex. 2.4.1 : Use the Rail-Fence cipher to encrypt the message "HAPPY BIRTHDAY TO YOU".

☑ **Soln. :**

Plaintext : HAPPYBIRTHDAYTOYOU

In Rail-Fence cipher, the plaintext is arranged in two lines in a zigzag pattern.

H		P		Y		I		T		D		Y		O		O	
	A		P		B		R		H		A		T		Y		U

The ciphertext is created reading the pattern row by row.

Ciphertext: "HPYITDYOOAPBRHATYU".

Ex. 2.4.2 : Use the keyless transposition cipher to encrypt the message "WE ARE DISCOVERED SAVE YOURSELF" in a table of five columns.

☑ **Soln. :**

Plaintext :

WEAREDISCOVEREDSAVEYOURSELF

In this method, the text is written into the table row by row and then transmitted column by column. The number of columns given is 5.

W	E	A	R	E
D	I	S	C	O
V	E	R	E	D
S	A	V	E	Y
O	U	R	S	E
L	F			

The ciphertext is

"WDVSOLEIEAUFASRVRRCEESEODYE".

2.4.2 Keyed Transposition Ciphers

- In keyless transposition ciphers, the permutation on characters is done using writing plaintext in one way (row by row, for example) and reading it in another way (column by column, for example). The permutation is done on the whole plaintext to create the whole ciphertext.
- In keyed transposition cipher, we divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately.
- If in a grouping, a block falls short of characters, then add bogus character 'Z' at the end to make the last group the same size as the others.
- The key used for encryption and decryption is a permutation key, which shows how the characters are permuted.

Ex. 2.4.3 : Encrypt the message "ENEMY ATTACKS TONIGHT" using a block size of 5 and the key 31452.

☑ **Soln. : Plaintext :** ENEMYATTACKSTONIGHT

Divide the plaintext into groups of block size = 5 as follows : ENEMY, ATTAC, KSTON, IGHTZ

Now, arrange the characters in each block as per the given key 31452.

This permutation yields: EEMYN, TAACT, TKONS, HITZG

Thus, the ciphertext is :

EEMYNTAACTTKONSHITZG

UEx. 2.4.4 (SPPU - Q. 2(b), May 2019, 5 Marks)

Use Transposition Cipher to encrypt plain text 'I love my India' and use the key 'HEAVEN'.

☑ **Soln. : Plaintext :** ILOVEMYINDIA

Key = HEAVEN

Divide the plaintext into groups of block size = 6 as follows : ILOVEM, YINDIA

Now, arrange the characters in each block as per the given key HEAVEN.

This permutation yields: ON, LI, EI, IY, MA, VD

Thus, the ciphertext is :

ONLIEIIYMAVD

2.4.3 Keyed Columnar Transposition Ciphers

- In this transposition cipher, better scrambling is achieved by combining keyless and keyed transposition ciphers.
- Encryption or decryption is done in three steps.
- First, the text is written row by row into a table.
- Second, the permutation is done by reordering the columns.
- Third, the new table is read column by column.
- The first and third steps provide a keyless global reordering and the second step provides a blockwise keyed reordering.

Ex. 2.4.5 : Encrypt and decrypt the message "ENEMY ATTACKS TONIGHT" with keyed columnar transposition cipher with encryption key 31452 and decryption key 25134.

☑ **Soln. :**

▶ **Encryption**

Plaintext : ENEMYATTACKSTONIGHT

Encryption key = 31452

Since key size is 5, we write the plaintext row by row into 5 columns.

1	2	3	4	5
E	N	E	M	Y
A	T	T	A	C
K	S	T	O	N
I	G	H	T	Z

Given encryption key is 31452. So arrange the columns in key order.

3	1	4	5	2
E	E	M	Y	N
T	A	A	C	T
T	K	O	N	S
H	I	T	Z	G

Now, read column by column to get ciphertext.

Ciphertext: "ETTHEAKIMAOTYCNZNTSG".

▶ **Decryption**

Ciphertext : ETTHEAKIMAOTYCNZNTSG

Decryption key = 25134

Since key size is 5, we write the ciphertext column by column into 5 columns.

1	2	3	4	5
E	E	M	Y	N
T	A	A	C	T
T	K	O	N	S
H	I	T	Z	G

Given decryption key is 25134. So arrange the columns in key order.

2	5	1	3	4
E	N	E	M	Y
A	T	T	A	C
K	S	T	O	N
I	G	H	T	Z

Now, read row by row to get plaintext.

Plaintext: "ENEMYATTACKSTONIGHTZ".

2.4.4 Double Transposition Ciphers

- Double transposition ciphers were used by the Germans in World War I, as well as by the Allied and Axis Powers during World War II.
- During this time, multiple anagramming was discovered as a way to find the key of a double transposition cipher if the same key was used more than once.
- A double transposition is a columnar transposition that is applied twice. This can be done with one or two keys, but typically two keys are used to increase the security of the cipher.

Ex. 2.4.6 : Encrypt the following message with the following keys (in order) through a double transposition cipher.

Plaintext : GIVE HIM MONEY ; **Keys :** HAT, RED

☑ **Soln. :**

▶ **Step 1 :** Arrange the plaintext into as many columns as there are letters in the first key.

▶ **Step 2 :** Rearrange the columns based on the key. The letters of the key will be placed in alphabetical order and appropriate columns will be moved with the key letters.

H	A	T
1	2	3
G	I	V
E	H	I
M	M	O
N	E	Y

A	H	T
2	1	3
I	G	V
H	E	I
M	M	O
E	N	Y

▶ **Step 3 :** The ciphertext is now written out by writing the letters starting at the top left and going down each column.

Ciphertext after first transposition: IHMEGEMNVIOY

▶ **Step 4 :** Use the ciphertext from the previous transposition for the plaintext in the second transposition with the second key.

R	E	D
1	2	3
I	H	M
E	G	E
M	N	V
I	O	Y

▶ **Step 5 :** Place the key letters into alphabetical order and move the corresponding columns with each letter.

D	E	R
3	2	1
M	H	I
E	G	E
V	N	M
Y	O	I

▶ **Step 6 :** Write out the letters starting at the top left and going down each column to obtain the final ciphertext.

Final Ciphertext : MEVYHGNOIEMI

2.4.5 Vernam Cipher (One-Time Pad)

- The Vernam Cipher is an algorithm invented in 1917 to encrypt teletype (TTY) messages.
- Vernam Cipher is a method of encrypting alphabetic text.
- It is one of the transposition techniques for converting a plain text into a cipher text. In this mechanism we assign a number to each character of the Plain-Text, like (A = 00, B = 01, C = 02, …, Z = 25).
- In Vernam cipher algorithm, we take a key to encrypt the plain text whose length should be equal to the length of the plain text and is chosen completely in random.
- It is also called one-time pad (OTP) because the key must be newly generated each time the sender wants to send the message to the receiver.
- For encryption, first assign a number to each character of the plain-text and the key according to alphabetical order.
- Then, add both the number (Corresponding plain-text character number and Key character number) and perform modulo 26. Subtract the number 26 if the sum is greater than 26, if it isn't then leave it.
- For decryption apply just the reverse process of encryption.

Ex. 2.4.7 : Encrypt the message "MEET ME OUTSIDE" with Vernam cipher using a random key "BDUFGHWEIUFGW".

☑ **Soln. :**

Plaintext	M	E	E	T	M	E	O	U	T	S	I	D	E
P's Values	12	04	04	19	12	04	14	20	19	18	08	03	04
OTP	B	D	U	F	G	H	W	E	I	U	F	G	W
OTP's Values	01	03	20	05	06	07	22	04	08	20	05	06	22
C's Values	13	07	24	24	18	11	10	24	01	12	13	09	00
Ciphertext	N	H	Y	Y	S	L	K	Y	B	M	N	J	A

The ciphertext is: "NHYYSLKYBMNJA".

NOTES

2.5 DIFFERENCE BETWEEN SUBSTITUTION CIPHER AND TRANSPOSITION CIPHER

The Table 2.5.1 distinguishes both substitution and transposition ciphers.

Table 2.5.1 : Difference between Substitution Cipher and Transposition Cipher

Sr. No.	Parameter	Substitution Cipher Technique	Transposition Cipher Technique
1.	Algorithm	Each character is replaced with other character/number/symbol.	Each character is positioned differently from its original position.
2.	Forms	Mono Alphabetic Substitution Cipher and Poly Alphabetic Substitution Cipher are its two forms.	Key-less Transposition Cipher and Keyed Transposition cipher are its two forms.
3.	Change	Character identity is changed but position remains same.	Character position is changed but identity remains same.
4.	Detection	A letter less frequently used can be easily traced.	A letter near to original position get traced easily.
5.	Example	Caesar Cipher is an example of Substitution Cipher.	Rail-Fence Cipher is an example of Transposition Cipher.

2.6 BLOCK AND STREAM CIPHERS

Block and stream ciphers are two ways that you can encrypt data. Also known as bulk ciphers, they are two categories of symmetric encryption algorithms.

2.6.1 Block Cipher

- A block cipher breaks down plaintext messages into fixed-size blocks before converting them into ciphertext using a key.
- Block ciphers mix chunks of plaintext bits together with key bits to produce chunks of ciphertext of the same size, usually 64 or 128 bits.

2.6.2 Stream Cipher

- A stream cipher breaks a plaintext message down into single bits, which then are converted individually into ciphertext using key bits.
- Stream ciphers don't mix plaintext and key bits; instead, they generate pseudorandom bits from the key and encrypt the plaintext by XORing it with the pseudorandom bits.

2.6.3 Difference between Block Cipher and Stream Cipher

Table 2.6.1 shows differentiates between block cipher and stream cipher.

Table 2.6.1 : Block Cipher Vs Stream Cipher

Sr. No.	Block Ciphers	Stream Ciphers
1.	Symmetric key ciphers that encrypt and decrypt data in fixed-size blocks.	Symmetric key ciphers that encrypt and decrypt data bit-by-bit.
2.	Slower processing.	Faster processing.
3.	Require more resources.	Require fewer resources.
4.	Can take on stream cipher properties through certain modes of operation.	Cannot take on block cipher properties.
5.	Rely on stateless and statefull modes of operation, which include Electronic Code Book (ECB), Cipher Block Chaining (CBC)	Can be synchronous or asynchronous.

Sr. No.	Block Ciphers	Stream Ciphers
6.	Used nearly everywhere in cyber security.	Used for some data in-transit encryption, including in some SSL/TLS cipher suites.
7.	Examples: AES, DES, Blowfish, RC5 algorithms	Examples: RC4, A5 (used in GSM) algorithms

2.7 BLOCK CIPHER MODES OF OPERATION

UQ. Explain block cipher modes of operation.
(SPPU - Q. 3(B), May 2013, 8 Marks)

UQ. Enlist block ciphering modes of operation. Explain CBC mode in detail.
(SPPU - Q. 4(a), May 2015, 8 Marks)

UQ. Explain block ciphering modes operations with suitable diagram.
(SPPU - Q. 4(b), May 2016, 8 Marks)

UQ. What is block Cipher? Explain counter mode of block Cipher. **(SPPU - Q. 3(b), May 2019, 5 Marks)**

There are five types of operations in block cipher modes :

1. Electronic Code Block (ECB) Mode - block cipher
2. Cipher Block Chaining (CBC) Mode - block cipher
3. Cipher Feedback (CFB) Mode - block ciphers acting as stream ciphers
4. Output Feedback (OFB) Mode - block ciphers acting as stream ciphers
5. Counter (CTR) mode - block cipher

Very soon, we will see that ECB is used for transmitting a single value in secure manner, CBC is used for encrypting blocks of text authentication, CFB is used for transmitting an encrypted stream of data authentication, OFB is used for transmitting an encrypted stream of data, CTR is used for transmitting block-oriented applications.

2.7.1 Electronic Code Book (ECB) Mode

- Electronic code book is the simplest mode of operation of block cipher. It works on processing a series of sequentially listed message blocks but 64-bit block at a time. Each block is separately encrypted.
- Generally, a message is larger than 64 bits in size, it can be broken down into series of blocks and the encryption procedure is repeated. Each block is encrypted using the same key and makes the block of ciphertext.
- At the receiver side, the data is divided into blocks, each of 64 bits. The key used for encryption; the same key is used for decryption. It takes the 64-bit ciphertext as input and converts the ciphertext into plain text using the same key.
- The ECB mode is **deterministic** as the same key is used for all blocks' encryption. If the block of plain text is repeated in the original message, then its corresponding ciphertext block will also be repeated.
- Because, the same key used for all the blocks, ECB mode is used for an only small message where the repetition of the plain text block is lesser.

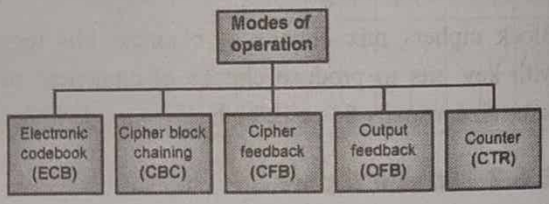
  Encryption : $C_i = E_K (P_i)$

  Decryption : $P_i = D_K (C_i)$
- Technically for a given key, a codebook of ciphertexts for all possible plaintext blocks can be created. Encryption would be then only looking up for required plaintext and select the corresponding ciphertext. Hence the name is given as Electronic Codebook mode of operation (ECB).

E : Encryption D : Decryption

P_i : Plaintext block i C_i : Ciphertext block i

K : Secret key



(1B1)**Fig. 2.7.1 : Modes of operation for Block ciphers**

Advantages of ECB Mode

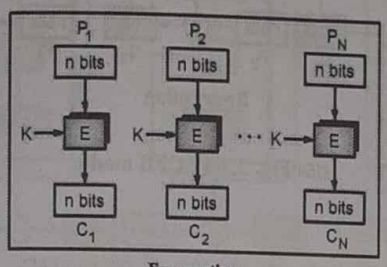
(1) Simplest way of block cipher

(2) Faster way of encryption as parallel encryption of blocks of bits is possible.

☞ Disadvantages of ECB Mode

(1) A cipher text from ECB can allow an attacker to guess the plaintext by trial-and-error since there is a direct relationship between plaintext and ciphertext i.e., it is prone to cryptanalysis.

(2) Hence, the ECB mode is not used in most of the applications.



(1B2)Fig. 2.7.2 : ECB mode

2.7.2 Cipher Block Chaining (CBC) Mode

- CBC can be called as the advancement on ECB. Here, at the sender side, the plain text is divided into blocks.
- In this mode, **Initialization Vector (IV)** is used, which can be a random block of text. IV is used to make the ciphertext of each block unique since the key used is same for encryption as we use for ECB.
- For encryption, the first block of plain text and IV is combined using the XOR operation and then the resultant message is encrypted using the key and thus forms the first block of ciphertext.
- The previous block of ciphertext is used as IV for the next block of plain text. The same procedure is followed for all blocks of plain text.
- That indicates the key used in CBC mode is the same; only the IV is different.
- For decryption, at the receiver side, the ciphertext is divided into blocks. The first block ciphertext is decrypted using the same key, which is used for encryption. The resultant message is XORed with the IV to get the first block of plain text.
- The second block of ciphertext is also decrypted using the same key, and the result of the decryption will be XORed with the first block of ciphertext to get the second block of plain text.
- The same procedure is repeated for all the blocks.

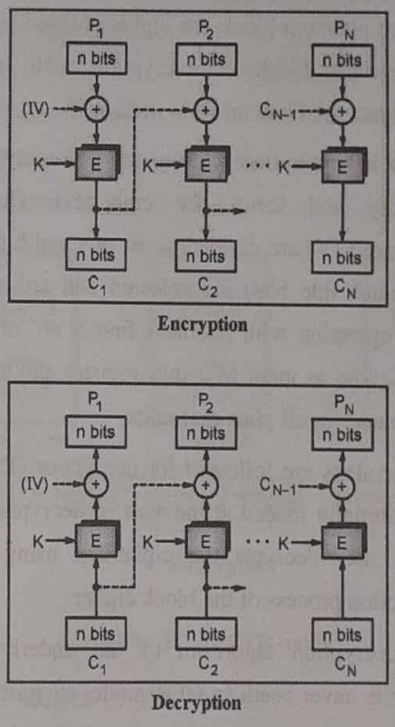
Encryption	Decryption
$C_0 = IV$	$C_0 = IV$
$C_i = E_K (P_i \oplus C_{i-1})$	$P_i = D_K (C_i) \oplus C_{i-1}$

- CBC is **non-deterministic** since even if the block of plain text is repeated in the original message, it will produce a different ciphertext for corresponding blocks.

E : Encryption D : Decryption

P_i : Plaintext block i C_i : Ciphertext block i

K : Secret key IV : Initial vector (C_0)



(1B3)Fig. 2.7.3 : CBC mode

☞ Advantages of CBC Mode

(1) Better resistive nature towards cryptanalysis than ECB due to changing IV.

(2) CBC works well for greater inputs.

(3) CBC forms the basis for a well-known data origin authentication mechanism. Thus, it is used for those applications that require both symmetric encryption and data origin authentication.

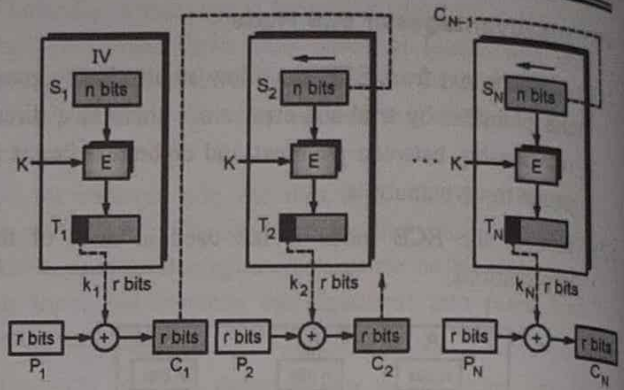
☞ Disadvantages of CBC Mode

(1) The error in transmission gets propagated to few further blocks during decryption due to chaining effect.

(2) Parallel encryption is not possible since every encryption requires previous cipher.

2.7.3 Cipher Feedback (CFB) Mode

- In this mode, the data is encrypted in the form of units where each unit is of 8 bits. Here, in order to encrypt the next plaintext block, the cipher is given as feedback to the next block of encryption with some new specifications. First, an IV is initialized.
- The IV is kept in the shift register. It is encrypted using the key and forms the ciphertext. Output bits (encrypted IV) are divided as set of s and b bits. S bits (left hand side bits) are selected and are applied an XOR operation with plaintext first S no. of bits. The result given as input to a shift register and the process is repeated for all plain text units.
- Similar steps are followed for decryption. Pre-decided IV is initially loaded at the start of decryption. In this mode, user decrypts the ciphertext using only the encryption process of the block cipher.
- The decryption algorithm of the underlying block cipher is never used. In CFB mode, encipherment and decipherment use the encryption function of the underlying block cipher.



Encryption

(1B4)**Fig. 2.7.4 : CFB mode**

Encryption :

$C_i = P_i \oplus \text{SelectLeft}_r \{E_K [\text{ShiftLeft}_r (S_{i-1}) \mid (C_{i-1})]\}$

Decryption :

$P_i = C_i \oplus \text{SelectLeft}_r \{E_K [\text{ShiftLeft}_r (S_{i-1}) \mid (C_{i-1})]\}$

E : Encryption D : Decryption

S_i : Shift register P_i : Plaintext block i

C_i : Ciphertext block i T_i : Temporary register

K : Secret key IV : Initial vector (S_1)

☞ Advantages of CFB

(1) It is difficult for applying cryptanalysis since there is some data loss due to use of shift register.

(2) By converting a block cipher into a stream cipher, CFB mode provides some of the advantageous properties of a stream cipher while retaining the advantageous properties of a block cipher too.

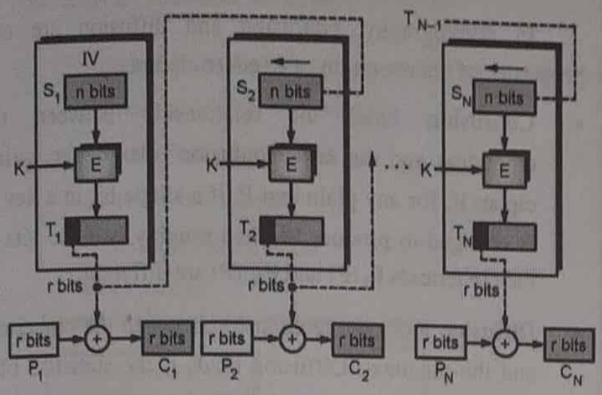
☞ Disadvantage of CFB

(1) The error of transmission gets propagated due to changing of blocks.

2.7.4 Output Feedback (OFB) Mode

- The OFB mode follows nearly same process as the Cipher Feedback mode except that it sends the encrypted output (output of the IV encryption) as **feedback for the next stage** of the encryption process instead of the actual cipher which is XOR output.

- Plain text and leftmost 8 bits of encrypted IV are combined using XOR to produce the ciphertext. For the next stage, the ciphertext, which is the form in the previous stage, is used as an IV for the next iteration.



Decryption

(1B5)**Fig. 2.7.5 : OFB mode**

- In this mode, instead of sending selected s bits, all bits of the block are sent. The same procedure is repeated for all blocks. It involves feeding the successive output blocks from the underlying block cipher back to it.

E : Encryption	D : Decryption
S_i : Shift register	P_i : Plaintext block i
C_i : Ciphertext block i	T_i : Temporary register
K : Secret key	IV : Initial vector (S_1)

☞ Advantages of OFB Mode

(1) Hold great resistance towards bit transmission errors.

(2) It also decreases dependency or relationship of cipher on plaintext.

☞ Disadvantages of OFB Mode

(1) Repeatedly encrypting the initialization vector may produce the same state that has occurred before.

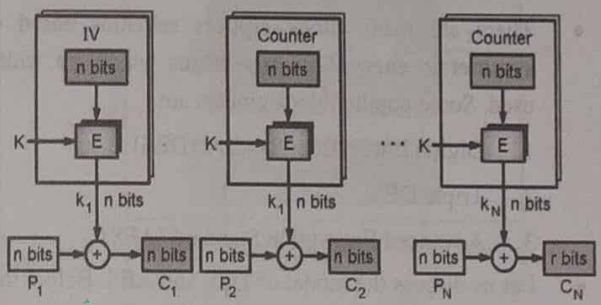
(2) This is an unlikely situation, but in such a case, the plaintext will start to be encrypted by the same data as it was previously.

2.7.5 Counter (CTR) Mode

- The CTR is a simple **counter-based block cipher** implementation. It uses the sequence of numbers as an input for the algorithm.
- Every time a counter initiated value is encrypted and given as input to XOR with plaintext which results in ciphertext block.
- When the block is encrypted, to fill the next register next counter value is used. Every time, the counter value is incremented by 1 for next stage. This process is continued until the last plaintext block has been encrypted.
- For decryption, the ciphertext block is XORed with the output of encrypted contents of counter value.
- After decryption of each ciphertext block counter is updated as we do for encryption. In other words, CTR mode also converts a block cipher to a stream cipher.
- In this mode, both the sender and receiver need to access to a reliable counter.
- This shared counter is not necessarily a secret value, but challenge is that both sides must keep the counter synchronized.
- The CTR mode can be considered as a counter-based version of CFB mode without the feedback and can be implemented in parallel.

E : Encryption	IV : Initialization vector
P_i : Plaintext block i	C_i : Ciphertext block i
K : Secret key	K_i : Encryption key i

The counter is incremented for each block.



Encryption

(1B6)**Fig. 2.7.6 : CTR mode**

☞ Advantages of CTR Mode

1. It does not have message dependency.
2. It does not propagate error of transmission at all.
3. Parallel encryption is possible.

☞ Disadvantages of CTR Mode

1. It requires a synchronous counter at sender and receiver.
2. Loss of synchronization leads to incorrect recovery of plaintext.

- As seen in previous chapter, **Symmetric key cryptography (private/secret key cryptography)** uses single key for encryption and decryption whereas in **Asymmetric key cryptography (public key cryptography)**, public key and private keys are used for encryption and decryption.
- Some examples of symmetric key cryptography are AES, DES, Triple DES and RC5 while ECC, El Gamal, Diffie-Hellman, DSA and RSA are based on asymmetric key cryptography.

⏭ 2.8 SYMMETRIC KEY CRYPTOGRAPHY

- **Symmetric key cryptography** is a type of encryption where only one key called as a **secret key** is used to both encryption and decryption.
- The entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process. This encryption method differs from asymmetric/public key encryption where a pair of keys, one public and one private, is used to encrypt and decrypt messages.
- There are many **block ciphers schemes based on symmetric encryption algorithms** which are widely used. Some popular block ciphers are :
  1. Digital Encryption Standard (DES)
  2. Triple DES
  3. Advanced Encryption Standard (AES)
- Let us discuss the model of DES and AES. Before this, we need to understand **Feistel cipher**. Feistel refers to as the ideal block cipher, because it allows for the maximum number of possible encryption mappings from the block. DES is just one example of a Feistel Cipher.
- Any cryptographic system based on Feistel cipher structure uses the same algorithm for both encryption and decryption. A Feistel cipher can have three types of components: self-invertible, invertible, and noninvertible whereas a non-Feistel cipher uses only invertible components.
- In non-Feistel cipher (e.g., AES), a component in the encryption cipher has the corresponding component in the decryption cipher.

☞ Confusion and Diffusion

In cryptography, confusion and diffusion are two properties of the operation of a secure cipher.

- **Confusion** hides the relationship between the ciphertext and the key. Confusion rules better with a cipher if, for any plain text P, if a single bit in a key K Is changed to produce k', then roughly half the bits in the ciphertexts $E_k(P)$ and $E_{K'}(P)$ are different.
- **Diffusion** hides the relationship between the ciphertext and the plaintext. Diffusion holds if the statistics of a block of the plain text is irretrievably dissipated across the block of its ciphertext. Thus, changing a single bit in a block of a plain text will have the effect of changing each bit of the block of ciphertext.
- Diffusion and confusion can be achieved using iterated product ciphers where each iteration is a combination of S-boxes, P-boxes, and other components.

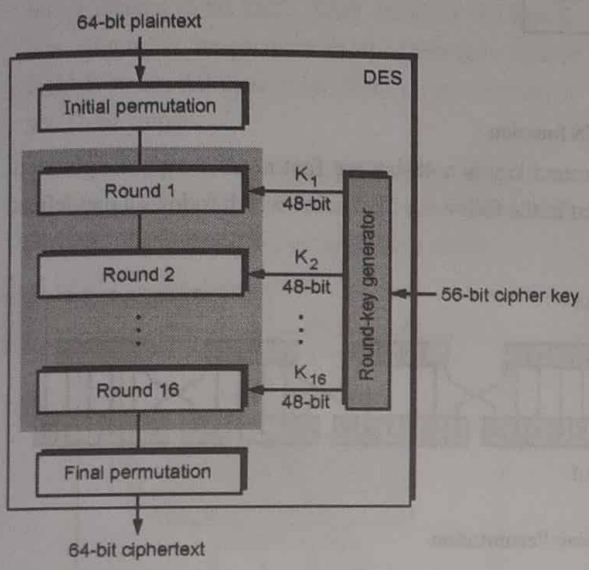
2.8.1 Data Encryption Standard (DES)

- The DES (Data Encryption Standard) algorithm is a **symmetric key** block cipher created in the early 1970s by an IBM team which is successor to a Feistel block cipher called Lucifer.
- It was published in March 1975 and adopted by the National Institute of Standards and Technology (NIST).
- The Advanced Encryption Standard (AES) replaced the DES encryption algorithm as the accepted standard in 2002, following a public competition to find a replacement. Triple DES (3DES) remains approved for sensitive government information through 2030.
- Triple DES is a symmetric key block cipher which applies the DES cipher in triplicate. It encrypts with the first key (k1), decrypts using the second key (k2), then encrypts with the third key (k3). There is also a two-key variant, where k1 and k3 are the same keys.
- At the encryption site, the algorithm takes the plain text in 64-bit blocks and converts them into 64-bit ciphertext using 56-bit keys. DES uses 16 rounds of the Feistel structure, using a different key for each round.

- Since it is asymmetric, at the decryption site, it takes 64-bit ciphertext and converts it into 64-bit plain text using the same 56-bit cipher key.

2.8.2 General Structure of DES

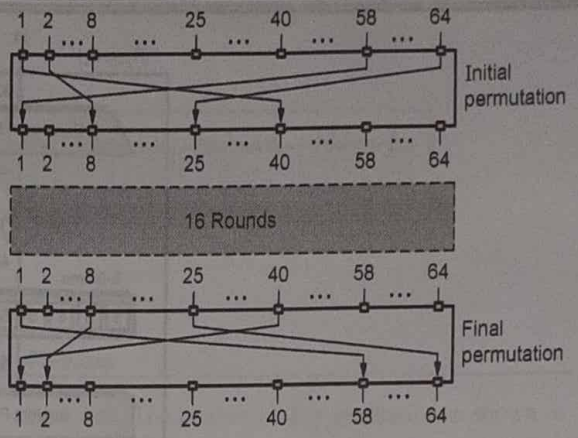
- A single block of plain text is transformed into 64-bit ciphertext after following stages:
  1. An initial permutation
  2. 16 Feistel rounds
  3. A Final Permutation



(1B7)**Fig. 2.8.1 : DES structure**

1. Initial and Final Permutation

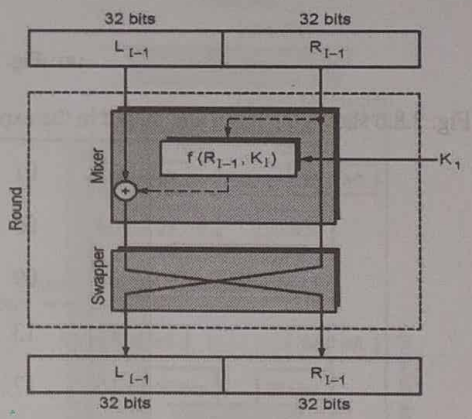
- The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other. There are pre-defined permutation rules for P-boxes.
- Each of these permutations takes **64-bit input** and permutes them according to predefined rule. For example, in initial permutation, the 64th bit in the input becomes the 25th bit in the output.
- Similarly, in the final permutation, 25th bit becomes 64th bit in the output.
- The initial and final permutations are shown in Fig. 2.8.2 :



(1B8)**Fig. 2.8.2 : Initial and permutation steps**

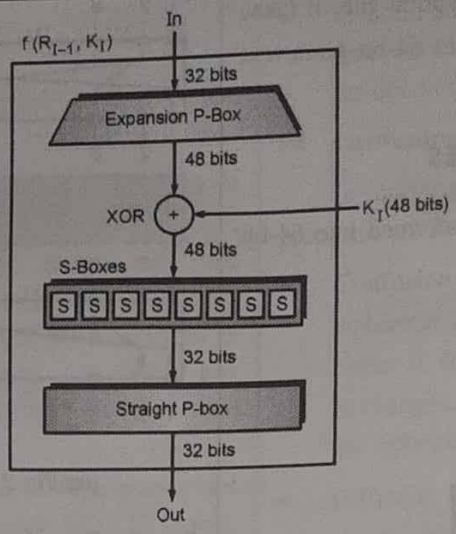
2. Rounds and DES Function

- DES uses **16 Feistel rounds**. Every round takes Left 32-bits (L_{I-1}) and Right 32-bits (R_{I-1}) from previous round (from initial permutation box for the very first time) to produce L_I and R_I which go to the next round (to final permutation box). Each round uses two cipher elements: mixer and swapper.
- The swapper is invertible, the mixer is invertible because of XOR operation. Each round is diagrammatically shown in Fig. 2.8.3



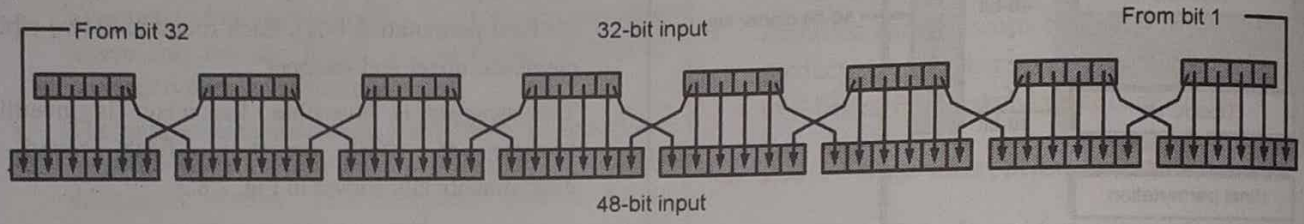
(1B9)**Fig. 2.8.3 : A round in DES for encryption**

- DES function is the heart of DES. The DES function applies a 48-bit key to the rightmost 32 bits (R_{I-1}) to produce a 32-bit output. This function consists of four sections:
  (1) Expansion P-box (2) A whitener (XOR)
  (3) A group of S-boxes (4) A straight P-box



(1B10)**Fig. 2.8.4 : DES function**

- **Expansion P-box :** Since right input (R_{I-1}) is 32-bit and round key is a 48-bit, we first need to expand right input (R_{I-1}) to 48 bits. Logic of permutation is graphically depicted in the following illustration which follows a pre-defined rule:



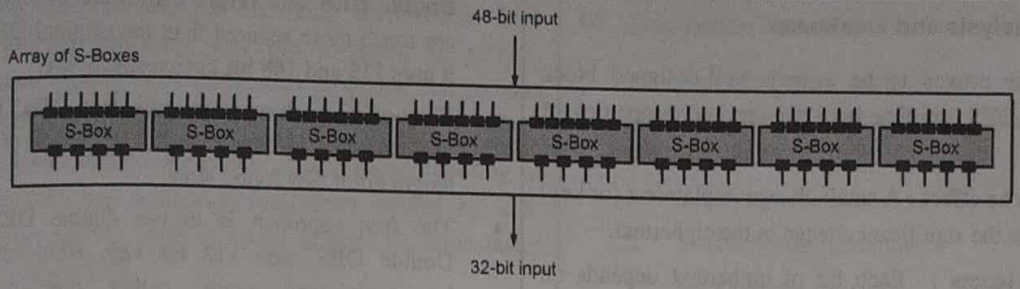
(1B11)**Fig. 2.8.5 : Expansion Permutation**

- Fig. 2.8.6 shows the input and output in the expansion permutation.

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

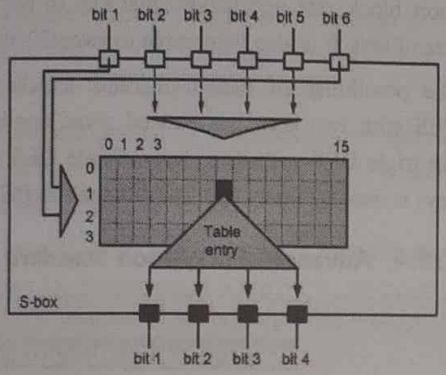
(1B12)**Fig. 2.8.6 : Expansion P-box table**

- ***A whitener* (XOR) :** After the expansion permutation, DES does XOR operation on the expanded right section (48-bit) and the round key (48-bit). The round key is used only in this operation.
- **Substitution Boxes :** The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. Refer the Fig. 2.8.7.



(1B13)**Fig. 2.8.7 : A group of S-boxes**

- The 48-bit input from the second operation is divided into 8 chunks (6-bit each). Each chunk is fed into S-box. As a result, we get 4-bit chunk. There are a total of eight S-boxes. All these 4-bit chunks are combined to get 32-bit output.
- The substitution in each S-box follows pre-defined rule as shown in Fig. 2.8.8. Each S-box has its own table (4 rows, 16 columns). So, we need eight tables



(1B14)**Fig. 2.8.8 : S-box rule**

- **Straight Permutation :** The 32-bit output of S-boxes is then subjected to the straight permutation to get 32-bit output with rule shown in Fig.2.8.9:

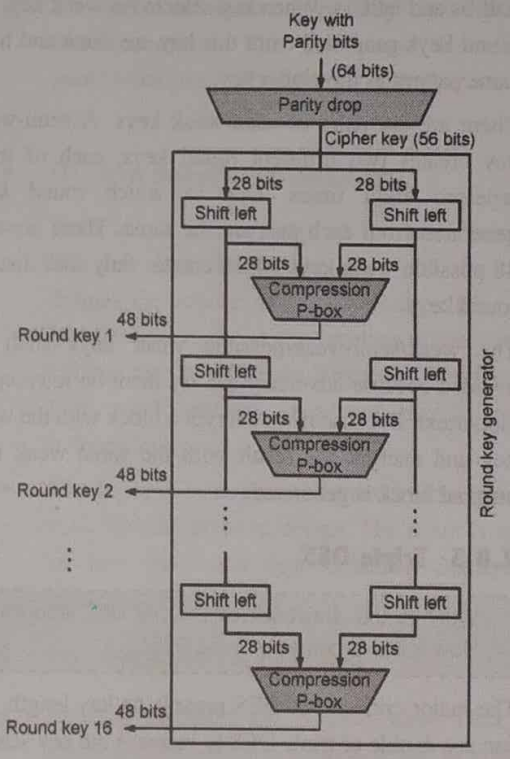
16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

(1B15)**Fig. 2.8.9 : Straight Permutation table**

Key generation in DES

- The cipher key is given as a 64-bit key out of which 8 extra bits (parity bits) are dropped. The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. The process of key generation is shown in the Fig. 2.8.10.
- A Parity drop drops parity bits (bits 8, 16, 24, ..., 64) from 64-bit key and permutes the rest 56 bits according to pre-defined table. Then these 56 bits are divided into two 28-bit parts. In every round, each part is then shifted left one or two bits. The compression P-box then changes the 56 bits to 48 bits which is the key for the round.

Shifting	
Rounds	**Shift**
1, 2, 9, 16	one bit
Others	two bits



(1B16)**Fig. 2.8.10 : Key generation**

☞ DES Analysis and weakness

DES has proved to be a very well-designed block cipher. The DES satisfies both the desired properties of block cipher.

- **Avalanche effect :** A small change in plaintext (or key) results in the significant change in the ciphertext.
- **Completeness :** Each bit of ciphertext depends on many bits of plaintext.
- Cryptanalysis have found some weaknesses in DES in cipher design as well as in the cipher key. Among the attempted attacks on DES, three are of interest: brute-force, differential cryptanalysis, and linear cryptanalysis.
- There are existing weaknesses in the design of S-boxes (due to which differential cryptanalysis and linear cryptanalysis are possible) and P-boxes. As well as the most serious weakness of DES is key size (56 bits). So, for exhaustive key search (Brute force attack) adversary needs to check 2^{56} keys.
- Four out of 2^{56} keys are weak keys. These keys after parity drop operation, consists of either all 0s, all 1s or half 0s and half 1s. When key selected is weak key, the round keys generated from this key are same and have same pattern as the cipher key.
- There are six pairs of semi-weak keys. A semi-weak key creates two different round keys, each of them repeated eight times. Due to which round keys generated from each pair are the same. There are also 48 possible weak keys which creates only four distinct round keys.
- The weak/semi-weak/possible weak keys shall be avoided because adversary can try them on intercepted ciphertext. Because if we encrypt a block with the weak key and encrypt the result with the same weak key, original block is generated.

2.8.3 Triple DES

GQ. What is the drawback of Double DES algorithm? How is it overcome by Triple DES?

- The major criticism of DES regards its key length. We can use double or triple DES to increase the key size.
- **Double DES** and **Triple DES** were introduced which are much more secured than the original DES because it uses **112** and **168 bit** keys respectively.
- They offer much more security than DES. Triple DES was designed to overcome the drawback of small key length but it was found slow.
- The first approach is to use double DES (2DES). Double DES uses 112 bit key. However, using a known-plaintext attack called meet-in-the-middle attack proves that double
- DES improves this vulnerability slightly (to 2^{57} tests), but not tremendously (to 2^{112}).
- There are two variants of Triple DES known as 3-key Triple DES (3TDES) and 2-key Triple DES (2TDES).
- Triple DES is also vulnerable to meet-in-the middle attack because of which it gives total security level of 2^{112} instead of using 168 bit key.
- The block collision attack can also be done because of short block size and using same key to encrypt large size of text. It is also vulnerable to sweet32 attack.
- The possibility of known-plaintext attacks on triple DES with two keys has enticed some applications to use triple DES with three keys. Triple DES with three keys is used by many applications such as PGP.

2.8.4 Advanced Encryption Standard (AES)

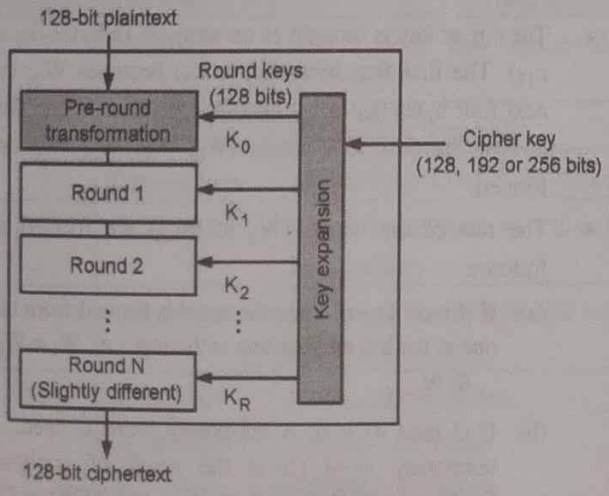
UQ. Explain AES algorithm with example.
(SPPU - Q. 4(B), May 2013, 8 Marks)

UQ. Explain AES algorithm in detail.
(SPPU - Q. 4(a), Dec. 2016, 8 Marks
Q. 3(a), May 2018, 8 Marks

UQ. Explain working of AES in detail.
(SPPU - Q. 4(b), May 2019, 5 Marks)

- The **AES** is a **symmetric-key block cipher** published by the **National Institute of Standards and Technology** (NIST) in December 2001 and chosen by the U.S. government to protect classified information.
- The NIST started development of AES in 1997 when it announced the need for an alternative to DES, which was starting to become vulnerable to brute-force attacks. It was intended to be easy to implement in hardware and software, as well as in restricted environments such as a smart card and offer decent defences against various attack techniques.

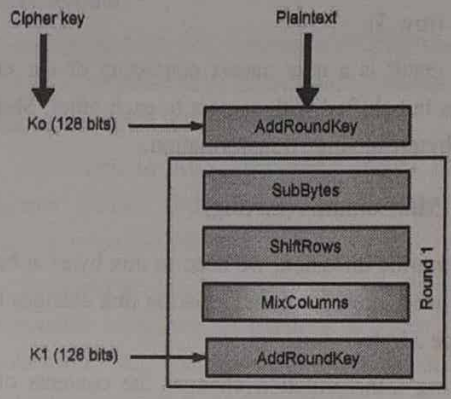
- AES was created for the U.S. government with additional voluntary, free use in public or private, commercial or non-commercial programs that provide encryption services.
- However, nongovernmental organizations choosing to use AES are subject to limitations created by U.S. export control. AES is found at least six time faster than triple DES.
- AES is a **non-Feistel cipher** that encrypts and decrypts a data block of 128 bits. Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES has defined three versions with 10, 12, and 14 rounds.
- Every version uses different key size, which can be 128, 192, or 256 bits, depending on the number of rounds but the round keys are always 128 bits. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. It can be observed from the following Fig. that the number of round keys generated by the key-expansion algorithm Is always one more than the number of rounds.
- AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes.
- These 16 bytes are arranged in four columns and four rows for processing as a matrix. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).



(1B17)**Fig. 2.8.11 : Design of AES encryption cipher**

☞ Encryption Process

- At encryption site, each round comprises of four transformations that are invertible. The transformations are:
  1. SubBytes (substitution)
  2. ShiftRows (permutation)
  3. MixColumns (mixing)
  4. AddRoundKey (key adding)



(1B18)**Fig. 2.8.12 : Structure of each round**

- The pre-round section uses only one transformation (AddRoundKey). The last round has only three transformations, MixColumns transformation is not used.

(1) Byte Substitution (SubBytes)

- Here, a byte is interpreted as two hexadecimal digits. The left digit defines the row, and the right digit defines the column of the substitution table. The two hexadecimal digits at the junction of the row and the column are the new byte. There are 16 distinct byte to byte transformations since each byte is transformed independently.
- The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns. SubBytes is an intrabyte transformation.

(2) ShiftRows (permutation)

- Shifting permutes the bytes. Unlike DES, in AES shifting transformation is done at the byte level. Each of the four rows of the matrix is shifted to the left.

- Any entries that 'fall off' are re-inserted on the right side of row. The number of shifts depends upon row number. Shift is carried out as follows:
  (1) First row is not shifted. (row 0)
  (2) Second row is shifted one (byte) position to the left. (row 1)
  (3) Third row is shifted two positions to the left. (row 2)
  (4) Fourth row is shifted three positions to the left. (row 3)
- The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other. ShiftRows is a byte-exchange transformation.

(2) MixColumns (mixing)

- To provide diffusion, we need to mix bytes at bit level. We need interbyte transformation that changes the bits inside a byte.
- Mixing transformation changes the contents of every byte by taking four bytes at a time and combining them to create four new bytes. Each column of four bytes is now transformed using a special mathematical function.
- This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. This step is not performed in the last round.

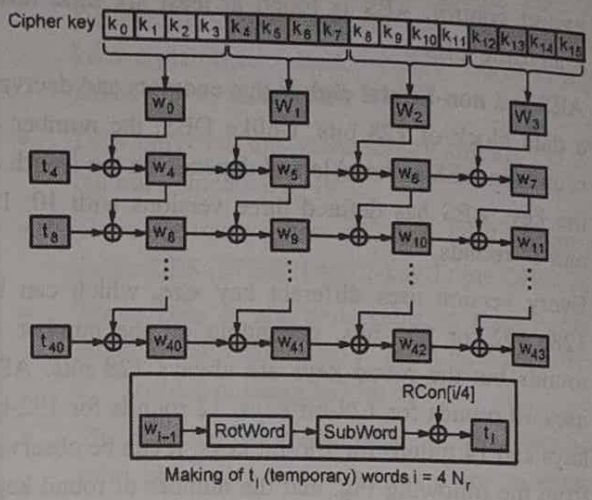
(4) AddRoundkey (key adding)

- AddRoundkey transformation proceeds one column at a time. The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round, then the output is the ciphertext.
- Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

Decryption Process

- The process of decryption of an AES ciphertext is similar to the encryption process in the reverse order. At the decryption site, each round consists of the four processes conducted in the reverse order :
  1. Add round key (self-invertible)
  2. InvMixColumns
  3. InvShiftRows
  4. InvSubByte

Key Expansion



(1B19)Fig. 2.8.13 : Key expansion in AES

- AES uses a key-expansion process to create round keys for each round. If the number of rounds is N_r, the key-expansion routine creates $N_r + 1$ 128-bit round keys from one single 128-bit cipher key.
- Fig. 2.8.13 depicts creation of words for AES-128 version. It shows the process of creating 44 words from the original key.

Steps for key generation in AES-128

- The cipher key is thought as an array of 16 bytes (k_0 to k_{15}). The first four bytes (k_0 to k_3) becomes W_0; the next four bytes (k_4 to k_7) becomes W_1 and so on. This is how the first four words (W_0, W_1, W_2, W_3) are formed.
- The rest of the words (W_4 to W_{43}) are formed as follows:
  (a) If (I mod 4) $\neq$ 0; Then the word is formed from the one at the left and the one at the top. i.e., $W_i = W_{i-1} \oplus W_{i-4}$.
  (b) If (I mod 4) = 0; A temporary word is used. A temporary word (t) is the result of applying SubWord and RotWord on W_{i-1} and XORing the result with some round constants, R_{Con}. i.e., $W_i = t \oplus W_{i-4}$.

- **RotWord :** It takes a word as an array of four bytes and shifts each byte to the left with wrapping. It is applied to only one row.
- **SubWord :** It takes each byte in the word and substitutes another byte for it. It is applied only to four bytes.
- **Round Constants (R_{Con}) :** Each round constant is a 4-byte value in which the rightmost three bytes are always zero.

The process for key expansion for other two versions is same except slight changes. The differences are :

- In AES-192, instead of four, six words would be generated (W_0, W_1, W_2, W_3, W_4, W_5). For the remaining words, if (I mod 6) ≠ 0; $W_i = W_{i-1} + W_{i-6}$; otherwise, $W_i = t + W_{i-6}$.
- In AES-256, instead of four, eight words would be generated (W_0, W_1, W_2, ..., W_7). For the remaining words, if (I mod 8) ≠ 0; $W_i = W_{i-1} + W_{i-8}$; otherwise, $W_i = t + W_{i-8}$. If (I mod 4) = 0, but (I mod 8) ≠ 0, then W_i = SubWord $(W_{i-1}) + W_{i-8}$.

☞ AES Analysis

- AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered. Brute-Force attack, Statistical attacks, Differential and linear attacks are not able to break the security of AES.
- AES has built-in flexibility of key length, which allows a degree of 'future-proofing' against progress in the ability to perform exhaustive key searches. However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed.
- In 2009, there was a known-key attack against AES-128. A known key was used to discern the structure of the encryption.
- However, the hack only targeted an eight-round version of AES-128, rather than the standard 10-round version, making the threat relatively minor.
- A major risk to AES encryption comes from side-channel attacks. Side-channel attacks are aimed at picking up leaked information from the system.
- Side-channel attacks, however, may reduce the number of possible combinations required to attack AES with brute force. Side-channel attacks can be mitigated by preventing possible ways data can leak.
- Additionally, using randomization techniques can help eliminate any relationship between data protected by the cipher and any leaked data that could be collected using a side-channel attack.

2.8.5 Comparison Between AES vs. DES

UQ. Differentiate AES and DES algorithms. **(SPPU - Q. 8(b), May 2015, 8 Marks, Q. 3(b), May 2018, 8 Marks)**

Sr. No.	Key	AES	DES
1	Definition	AES stands for Advanced Encryption Standard.	DES stands for Data Encryption Standard.
2	Creation	The creation is in 1999.	The creation is in 1976.
3	Derived from	AES derives from Square cipher.	DES derives from Lucifer cipher.
4	Designed By	AES was designed by Vincent Rijmen and Joan Daemen.	DES was designed by IBM.
5	Key Length	Key length varies from 128 bits, 192 bits to 256 bits.	Key length is of 56 bits.
6	Rounds of Operations	Rounds per key length: 128 bits – 10 rounds 192 bits – 12 rounds 256 bits – 14 rounds	16 rounds of identical operations.

Sr. No.	Key	AES	DES
7	Structure	AES structure is based on substitution-permutation.	DES structure is based on feistel structure.
8	Security	AES is de-facto world standard and is more secure than DES.	DES is weak and Triple DES is more secure than DES.
9	Rounds	Byte substitution, Shift Row, Mix Column and Key Addition.	Expansion, XOR operation with round key, Substitution and Permutation.
10	Size	AES can encrypt 128 bits of plain text.	DES can encrypt 64 bits of plain text.
11	Known attacks	Side channel attacks are possible.	Brute-force, Statistical analysis, Linear cryptanalysis and Differential cryptanalysis.

Descriptive Questions

Q. 1 Distinguish between symmetric key and asymmetric key cryptography.

Q. 2 Explain Substitution Ciphers with illustrative examples.

Q. 3 Use additive cipher with key = 4 to encrypt the message "the quick brown fox jumped over the lazy dog".

Q. 4 Use the multiplicative cipher to encrypt the message "Knowledge is power" with a key of 3.

Q. 5 Using Affine cipher, encrypt the Plaintext "MONEY" with a key pair (11,4).

Q. 6 Using Playfair cipher, encrypt the plaintext "hide the gold in the tree stump" using the keyphrase "*playfair example*".

Q. 7 Using Vigenere cipher, encrypt the plaintext "a simple example" using the keyword "*battista*".

Q. 8 Using Hill cipher, encrypt the plaintext message "retreat now" using the keyphrase "*back up*" and a 3 x 3 matrix.

Q. 9 Using Hill cipher, encrypt the plaintext message "SHORTER EXAMPLE" using the keyphrase "HILL" and a 2 x 2 matrix.

Q. 10 Encrypt the message "BUY SOME MILK AND EGGS" using a transposition cipher with key word "MONEY".

Q. 11 Distinguish between Block Cipher and Stream Cipher.

...Chapter Ends

❑❑❑

UNIT III

CHAPTER 3

Asymmetric Key Cryptography

University Prescribed Syllabus

Number theory : Prime number, Fermat and Euler theorems, Testing for primality, Chinese reminder theorem, discrete logarithm, Public Key Cryptography and RSA, Key Management, Diffie-Hellman key exchange, El Gamal algorithm, Elliptic Curve Cryptography.

3.1 NUMBER THEORY

3.1.1 Prime Number

UQ. Explain in general terms an efficient procedure for picking a prime number.
(SPPU - Q. 5(b), Dec. 2012, 8 Marks)

- A prime number is a natural number greater than 1 and has no positive division other than itself.
- Prime number cannot be divided by any number other then 1 and itself.
- Test for a number to be prime :

  Let p be a given number and let n be the smallest counting number less than $n^2 \geq p$. Now, test whether p is divisible by any of the prime numbers less than or equal to n. If yes, the p is not prime otherwise p is prime.

  Example : 2, 3, 5, 7, 11 : are prime numbers

  Test for 137 :

- We know that, $(12)^2 > 137$. Prime number less than 12 are 2, 3, 5, 7, 11 clearly none of them divides

3.1.2 Modular Arithmetic

3.1.2(A) Euclidean Algorithm

- The Euclidean method, which is a straightforward procedure for computing the greatest common divisor of two positive integers, is one of the fundamental techniques of number theory.
- The greatest common divisor of a and b is the largest integer that divides both a and b. We also define gcd (0, 0) = 0.
- More formally, the positive integer c is said to be the greatest common divisor of a and b if

  1. c is a divisor of a and of b.
  2. Any divisor of a and b is a divisor of c.

- Because we require that the greatest common divisor be positive,

  gcd (a, b) = gcd (a, – b) = gcd (–a, b) = gcd (–a, –b).

  In general, gcd (a, b) = gcd (a - b, b) = gcd (a, b-a).

- We'll start with a simple definition: If the sole common positive integer factor of two integers is 1, they are considered relatively prime.
- 8 and 15 are relatively prime because the positive divisors of 8 are 1, 2, 4, and 8, and the positive divisors of 15 are 1, 3, 5, and 15. So 1 is the only integer on both lists.

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 i.e. (A = B·Q + R).
- If we subtract a smaller number from a larger (we reduce a larger number), GCD doesn't change. So if we keep subtracting repeatedly the larger of two, we end up with GCD.
- Now instead of subtraction, if we divide the smaller number, the algorithm stops when we find remainder 0.
- Here, GCD (A, B) = GCD (B, R).

Ex. 3.1.1 : Find the GCD of 78 and 36 using the Euclidean Algorithm.

☑ **Soln. :**

A = 78, B = 36

A ≠0, B ≠0

Use long division to find that 78/36 = 2 with a remainder of 6.

We can write this as: 78 = 36 * 2 + 6

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

Ex. 3.1.2 : Find the GCD of 270 and 192 using the Euclidean Algorithm.

☑ **Soln. :**

A = 270, B = 192

A ≠0, B ≠0

Use long division to find that 270/192 = 1 with a remainder of 78.

We can write this as: 270 = 192 * 1 + 78

Find GCD (192, 78), since GCD (270, 192) = GCD (192, 78) = 6.

3.1.3 Testing for Primality

- Prime numbers are of immense importance in cryptography, computational number theory, information science and computer science. There are several algorithms to test if a number is prime.
- A primality test is deterministic if it outputs True when the number is a prime and False when the input is composite with probability 1. Otherwise, the primality test is probabilistic. A probabilistic primality test is often called a pseudoprimality test.

3.1.4 Fermat's Theorem

- Fermat's little theorem is the basis for the Fermat primality test and is one of the fundamental results of elementary number theory.
- Fermat's theorem states :

  If p is prime and a is a positive integer not divisible by p, then

  $$a^{p-1} \equiv 1 \pmod{p}$$
- In an alternative form, we can write it as:

  $a^p \equiv a \pmod{p}$
- For example, if a = 2 and p = 7, then $2^7 = 128$, and $128 - 2 = 126 = 7 \times 18$ is an integer multiple of 7.

3.1.5 Euler's Theorem

- Before presenting Euler's theorem, it's necessary to establish a key concept in number theory : **Euler's totient function**, abbreviated Ø (n), which is defined as the number of positive integers less than n that are relatively prime to n.

  Ø (1) = 1.
- Since a number less than or equal to and relatively prime to a given number is called a **totative**, the totient function Ø (n) can be simply defined as the number of totatives of n.
- For example, for 24, there are eight totatives {1, 5, 7, 11, 13, 17, 19, and 23}, so Ø (24) = 8.
- Now suppose that we have two prime numbers p and q with $p \neq q$. Then we can show that,

for n = pq, Ø (n) = Ø (pq) = Ø (p) * Ø (q) = (p – 1) * (q – 1)

- For example, for 21, Ø (21) = Ø (3) * Ø (7) = (3 – 1) * (7 – 1) = 2 * 6 = 12. Here the 12 totatives are {1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20}.
- **Euler's theorem** states that for every a and n that are relatively prime that if gcd (a, n) = 1, then

  $a^{Ø(n)} \equiv 1 \pmod{n}$
- For example, a = 3; n = 10; Ø (10) = 4, $a^{Ø(n)} = 3^4 = 81 = 1 \pmod{10} = 1 \pmod{n}$
- Or, a = 2; n = 11; Ø (11) = 10, $a^{Ø(n)} = 2^{10} = 1024 = 1 \pmod{11} = 1 \pmod{n}$

3.1.6 Chinese Remainder Theorem

- One of the most useful results of number theory is the Chinese remainder theorem (CRT). In essence, the CRT says it is possible to reconstruct integers in a certain range from their residues modulo a set of pairwise relatively prime moduli.
- **Theorem**: Let p, q be co-prime. Then the system of equations

  $$X \equiv a \pmod{p}$$
  $$X \equiv b \pmod{q}$$

  has a unique solution for **x modulo pq**.
- The 10 integers in Z_{10}, that is the integers 0 through 9, can be reconstructed from their two residues modulo 2 and 5 (the relatively prime factors of 10).
- Say the known residues of a decimal digit X are $r_2 = 0$ and $r_5 = 3$; that is, X mod 2 = 0 and X mod 5 = 3. Therefore, x is an even integer in Z_{10} whose remainder, on division by 5, is 3. The unique solution is X = 8.

Ex. 3.1.3 : Solve using chinese remainder theorem

$X \equiv 2 \bmod 3$

$X \equiv 2 \bmod 7$

$X \equiv 1 \bmod 4$

☑ **Soln. :**

$x \equiv 2 \bmod 3$

$x \equiv 2 \bmod 7$

$x \equiv 1 \bmod 4$

$a_1 = 2, a_2 = 2, a_3 = 1; m_1 = 3, m_2 = 7, m_3 = 4$

$m = m_1 * m_2 * m_3 = 3*7*4 = 84$

$z_1 = m/m_1 = 84/3 = 28$

$z_2 = m/m_2 = 84/7 = 12$

$z_3 = m/m_3 = 84/4 = 21$

$z_i y_i \bmod m_i = 1$

$z_1 y_1 \bmod m_1 = 1$

$28y_1 \bmod 3 = 1$

$y_1 = 1$

$z_2 y_2 \bmod m_2 = 1$

$12y_2 \bmod 7 = 1$

$y_2 = 3$

$z_3 y_3 \bmod m_3 = 1$

$21y_3 \bmod 4 = 1$

$y_3 = 1$

$x = (a_1 z_1 y_1 + a_2 z_2 y_2 + a_3 z_3 y_3) \bmod m$

$= (2*28*1 + 2*12*3 + 1*21*1) \bmod m$

$= (56 + 72 + 21) \bmod 84 = 149 \bmod 84$

$= 65 \bmod 84 = 65$

Ex. 3.1.4 : Solve using chinese remainder theorem

$X \equiv 4 \bmod 3$

$X \equiv 1 \bmod 5$

$X \equiv 2 \bmod 3$

☑ Soln. :

$X \equiv 4 \bmod 3$

$X \equiv 1 \bmod 5$

$X \equiv 2 \bmod 3$

$a_1 = 2, a_2 = 1, a_3 = 2; m_1 = 3, m_2 = 5, m_3 = 3$

$m = m_1 * m_2 * m_3 = 4*5*3 = 60$

$z_1 = m/m_1 = 60/4 = 15$

$z_2 = m/m_2 = 60/5 = 12$

$z_3 = m/m_3 = 60/3 = 20$

$z_i y_i \bmod m_i = 1$

$z_1 y_1 \bmod m_1 = 1$

$15y_1 \bmod 4 = 1$

$y_1 = 3$

$z_2 y_2 \bmod m_2 = 1$

$12y_2 \bmod 5 = 1$

$y_2 = 3$

$z_3 y_3 \bmod m_3 = 1$

$20y_3 \bmod 3 = 1$

$y_3 = 2$

$x = (a_1 z_1 y_1 + a_2 z_2 y_2 + a_3 z_3 y_3) \bmod m$

$= (3*15*3 + 1*12*3 + 2*20*2) \bmod m$

$= (135 + 36 + 80) \bmod 60$

$= 251 \bmod 60$

$= 11 \bmod 60 = 11$

3.1.7 Discrete Logarithm

- Euler's Theorem says that the order of any element modulo m divides Ø (m).
- If u is a unit modulo m and the order of u is Ø(n), we say that u is a primitive root modulo m.
- **Example :** The powers of 2 modulo 5 are 2, 4, 3, and 1, so 2 is a primitive root mod 5 (since it has order 4). Similarly, we can check that 3 is also a primitive root mod 5.
- If there is a primitive root modulo m, then every unit can be written in terms of that primitive root.
- If b is a unit modulo m and a is another unit with $a \equiv b^d \pmod m$, we say that d is the discrete logarithm of a modulo m to the base b, and write $d = \log_b(a)$.
- The discrete logarithm obeys the standard rules of logarithms.
  1. Specifically, suppose that k is the order of b modulo m.
  2. Then $\log_b(ac) \equiv \log_b(a) + \log_b(c) \pmod k$ and $\log_b(a^r) \equiv r \log_b(a) \pmod k$ for any integer r and any residue classes a and c whose discrete logarithms to the base b are defined.
- **Example :** Find the discrete logarithms of each unit modulo 11 to the base 2.
- Since 2 is a primitive root modulo 11, we can write each unit as a power of 2. The simplest way to do this is simply to compute each of the values $2^0, 2^1, \ldots, 2^{10}$ modulo 11; here is a table of the results:

N	1	2	3	4	5	6	7	8	9	10
$\log_2 n$	0	1	8	2	4	8	7	3	6	5

- Observe, for example, that $3 \cdot 6 \equiv 7 \pmod{11}$, and $\log_2(3) + \log_2(6) \equiv \log_2(7) \pmod{10}$, since 10 is the order of 2 modulo 11.
- Likewise, $3^3 \equiv 5 \pmod{11}$, and $3 \log_2(3) \equiv \log_2(5) \pmod{10}$.

3.2 PUBLIC KEY CRYPTOGRAPHY

- Unlike symmetric encryption scheme, in Public Key cryptography or Asymmetric Cryptosystem encryption and decryption are performed using set of keys.
- One key is known to everyone called as Public key and another is Secret key called as Private key.
- Here, each receiver possesses a unique decryption key, referred to as his private key. Receiver needs to publish an encryption key, referred to as his public key. Either of the two keys (Public and Private key) can be used for encryption with other key used for decryption.

3.2.1 RSA Algorithm

UQ. Explain RSA algorithm.
(SPPU - Q. 6(a), Dec. 2012, 8 Marks, Q. 5(b), May 2017, 8 Marks)

UQ. Explain RSA algorithm and its application.
(SPPU - Q. 5(B), May 2013, 8 Marks)

UQ. In a public-key system using RSA, you intercept the ciphertext C = 10 sent to a user whose public key is e = 5, n =35. What is the plaintext M?
(SPPU - Q. 5(b), Dec. 2013, 8 Marks)

UQ. Perform encryption and decryption using RSA algorithm for following value of keys message. Discuss each step in detail. (i)p = 3, q = 11, e = 7, m = 5 (ii) p = 17, q = 31, e = 7, m = 2
(SPPU - Q. 5(b), Dec. 2014, 8 Marks)

UQ. What is RSA ? If RSA prime No. p = 3, q = 11, e = 3 and m = 00111011 (m-message), then calculate private key d and cipher text.
(SPPU - Q. 5(a), May 2015, 8 Marks)

- The RSA algorithm is an **asymmetric cryptography** algorithm; this means that it uses a public key and a private key. The RSA algorithm is named after Ron Rivest, Adi Shamir, and Leonard Adleman, those who invented it in 1978.
- As their names suggest, a public key is shared publicly, while a private key is secret and must not be shared with anyone. This public key will be used for the encryption and corresponding private key will be used for the decryption.
- RSA keys can be typically 1024 or 2048 bits long so that keys could not be broken easily. The RSA scheme is based on the fact that it is difficult to factorize a large integer.
- The integers used by this method are sufficiently large making it difficult to solve. The public key generally consists of two numbers where one number is multiplication of two large prime numbers. And private key is also derived from the same two prime numbers.
- So, if factorization of this large number is done, the private key can be compromised. Therefore, strength of encryption totally lies on the key size and if the key size is doubled or tripled, the strength of encryption increases exponentially.

Algorithm

The RSA algorithm ensures that the keys are as secure as possible. It has three basic operations:

I. Key Generation

It's a one-time operation to generate a public key/private key pair. However, if the key is compromised, for security reasons to get fresh pair of keys, this operation can be used again.

- **Step 1 :** Select two large prime numbers, p and q. The prime numbers need to be large so that they will be difficult for someone to figure out.

Calculate n = p * q [Here, n is the specified large number]

Calculate the totient function; $\phi(n) = (p - 1)(q - 1)$. [$(n) = \phi(n - 1)$, if n is a prime number]

- **Step 2 :** Select an encryption key, integer e, such that e is co-prime to $\phi(n)$ and $1 < e < \phi(n)$. That is, number e as a derived number which should be greater than 1 and less than $(p - 1) * (q - 1)$. The pair of numbers (n,e) makes up the public key and it is made public.

Note : Two integers are co-prime if the only positive integer that divides them is 1 i.e., there should be no common factor of (p – 1) and (q – 1) except 1.

Step 3 : Calculate the decryption key, d such that $d = e^{-1} \bmod \phi(n)$, in other words, $e*d = 1 \bmod \phi(n)$.

d can be found using the extended Euclidean algorithm, which takes p and q as the input parameters. The pair (n, d) makes up the private key.

The number of the bits, b, used to represent n is referred to as key size. i.e., $b = [\log_2 n]$.

II. Encryption

Consider a sender who sends the plain text message to someone whose public key is (n,e). Calculate corresponding ciphertext C as

$$C = p^e \bmod n$$

(III) Decryption

Considering receiver has the private key d, given a block of ciphertext C, the corresponding plain text will be calculated as

$$\text{Plaintext} = C^d \bmod n$$

3.2.1(A) Advantages of RSA

(1) RSA has overcome the weakness of symmetric algorithm i.e. authenticity and confidentiality.

(2) Asymmetric encryption algorithms are mainly used for the encryption key of the symmetric encryption algorithms.

3.2.1(B) Disadvantages of RSA

(1) RSA has too much computation.

(2) RSA algorithm can be very slow in cases where large data needs to be encrypted by the same computer.

(3) It requires a third party to verify the reliability of public keys.

(4) Data transferred through RSA algorithm could be compromised through middlemen who might temper with the public key system.

(5) Brute Force, Mathematical attacks, Timing attacks and Chosen ciphertext attacks are some possible approaches to attack RSA algorithm.

Ex. 3.2.1 : Suppose the two prime numbers are p = 53 and q = 59.

☑ **Soln. :** So, $n = p * q = 3127$ and

$\phi(n) = (p - 1)(q - 1) = 52 * 58 = 3016$.

Choose public key e, such that e is co-prime to ϕ (n) and $1 < e < \Phi$ (n).

Let us consider e = 3.

Now calculate Private Key, d :

To get d, apply, $d = (k*\Phi(n) + 1) / e$ for some integer k.

For k = 1, 3017 is not divisible by 3.

For k = 2, 6033 is divisible by 3 and the value of d is 2011.

Now we are ready with our public Key (n = 3127 and e = 3) and Private Key (d = 2011 and n = 3127).

Let us encrypt "HI" :

Convert letters to numbers: H = 8 and I = 9

Thus, ciphertext $C = 89^e \bmod n = 1394$.

Now we will decrypt 1394 :

Plaintext = $C^d \bmod n$.

Thus, our Encrypted Data comes out to be 89. 8 = H and I = 9 i.e., "HI".

Ex. 3.2.2 : Suppose the two prime numbers are p = 3 and q = 11.

☑ **Soln. :** So, $n = p * q = 3 * 11 = 33$ and

$\phi(n) = (p - 1)(q - 1) = 2 * 10 = 20$.

Choose public key e, such that e is co-prime to $\phi(n)$ and $1 < e < \Phi(n)$. Co-prime means it should not multiply by factors of $\phi(n)$ and also not divide by $\phi(n)$.

Factors of ϕ (n) are, $20 = 5 * 4 = 5 * 2 * 2$ so, e should not multiply by 5 and 2 and should not divide by 20.

So, primes are 3, 7, 11, 17, 19..., as 3 and 11 are taken, choose e as 7.

Therefore, e = 7

Now calculate Private Key, d :

To get apply, $d = (k*\Phi(n) + 1) / e$ for some integer k.

For k = 1, 21 is divisible by 7 and the value of d is 3.

Now we are ready with our public Key (n = 33 and e = 7) and Private Key (d = 3 and n = 33).

If m = 2; the encryption of m = 2 is : $C = 2^7 \bmod 33 = 29$.

The decryption of C = 29 is: $m = 29^3 \bmod 33 = 2$.

Ex. 3.2.3 : In an RSA cryptosystem, a particular A uses two prime numbers p = 13 and q =17 to generate her public and private keys. If the public key of A is 5. Then the private key of A is?

☑ **Soln. :** Here, p = 13 and q = 17

Compute n = 13 * 17 = 221 and $\phi(n) = (p - 1)(q - 1)$ $= (13 - 1) * (17 - 1) = 12 * 16 = 192$.

Choose public key e, such that e is co-prime to $\phi(n)$ and $1 < e < \Phi(n)$. Co-prime means it should not multiply by factors of $\phi(n)$ and also not divide by $\phi(n)$.

Factors of $\phi(n)$ are, $192 = 96 * 2 = 48 * 2 * 2 = 24 * 2 * 2 * 2 = 12 * 2 * 2 * 2 * 2 = 6 * 2 * 2 * 2 * 2 * 2 = 3 * 2 * 2 * 2 * 2 * 2 * 2$ so, e should not multiply by 3 and 2 and should not divide by 192.

So, primes are 3, 5, 7, 11, 13, 17, 19..., as 13 and 17 are taken, choose e as 5.

Therefore, e = 5.

Now calculate Private Key, d :

To get apply, $d = (k*\Phi(n) + 1) / e$ for some integer k.

For k = 1, 193 is not divisible by 5.

For k = 2, 385 is divisible by 5 and the value of d is 77.

Therefore, d = 11.

Ex. 3.2.4 : Given modulus n = 91 and public key e = 5, find the values of p, q and phi(n) and d using RSA. Encrypt M = 25. Also perform Decryption.

☑ **Soln. :**

Given : n = 91, e = 5, find p, q, $\phi(n)$.

$n = p * q$

p and q are the prime numbers. If we consider p as 3 the q will be 7.

Therefore, p = 3 and q = 7.

Now, $\phi n) = (p - 1)(q - 1) = 2 * 6 = 12$.

Given: e = 5.

Private key, d = ?

To get apply, $d = (k*\phi(n) + 1) / e$ for some integer k.

For k = 1, 13 is not divisible by 5.

For k = 2, 25 is divisible by 5 and the value of d is 5.

Therefore, d = 5.

Let us encrypt M = 25:

Thus, ciphertext $C = 25^e \bmod n = 25^5 \bmod 91 = 51$.

Now we will decrypt 51:

Plaintext = $C^d \bmod n$.

Thus, our Encrypted Data comes out to be 25.

Ex. 3.2.5 : Given modulus n = 221 and public key, e = 7, find the values of p, q, phi(n), and d using RSA. Encrypt M = 5.

☑ **Soln. :**

Given : n = 221, e = 7, find p, q, ϕ (n).

$n = p * q$

p and q are the prime numbers. If we consider p as 13 the q will be 17.

Therefore, p = 13 and q = 17.

Now, $\phi(n) = (p-1)(q-1) = 12 * 16 = 192$.

Given : e = 7.

Private key, d = ?

To get apply, $d = (k*\Phi(n) + 1) / e$ for some integer k.

For k = 1, 193 is not divisible by 7.

For k = 2, 385 is divisible by 7 and the value of d is 55.

Therefore, d = 55.

Let us encrypt M = 5:

Thus, ciphertext $C = 5^e \bmod n = 5^7 \bmod 221 = 112$.

Ex. 3.2.6 : Perform encryption and decryption using RSA algorithm with p = 7, q = 11, e = 17 and M = 8.

☑ **Soln. :**

Given : p = 7, q = 11, e = 17, find n, ϕ (n), d.

$n = p * q = 7 * 11 = 77$.

Now, $\phi(n) = (p-1)(q-1) = 6 * 10 = 60$.

Given : e = 17.

Private key, d = ?

To get apply, $d = (k*\Phi(n) + 1) / e$ for some integer k.

For k = 1, 61 is not divisible by 17.

For k = 2, 121 is not divisible by 17.

For k = 3, 181 is not divisible by 17.

For k = 15, 901 is divisible by 17 and the value of d is 53.

Therefore, d = 53.

Let us encrypt M = 8:

Thus, ciphertext $C = 8^e \bmod n = 8^{17} \bmod 77 = 57$.

Now we will decrypt 57:

Plaintext = $C^d \bmod n = 57^{53} \bmod 77 = 8$.

Thus, our Encrypted Data comes out to be 8.

UEx. 3.2.7 (SPPU - Q. 5(b), Dec. 2013, 8 Marks)

In a public-key system using RSA, you intercept the ciphertext C = 10 sent to a user whose public key is e = 5, n =35. What is the plaintext M?

☑ **Soln. :**

We know that the ciphertext C = 10, and the public key PU = {e, n} = {5, 35}.

Based on Euler's Totient function, $\phi(n)$ is defined as the number of positive integers less than n and relatively prime to n. We could find that $\phi(n) = 24$. Now, we guess two prime numbers p and q. Let p be 5 and q be 7. All the following conditions will be satisfied based on the guess:

(1) $n = p*q = 5*7=35$

(2) $\phi(n) = (p-1)(q-1)=(5-1)(7-1)=4*6=24$

(3) $\gcd(\phi(n), e) = \gcd(24, 5) =1$, $1< e < \phi(n)$

We calculate d in the next step.

Based on RSA key generation algorithm,

$d \equiv e^{-1} \bmod \phi(n)$ which is equivalent to $ed \equiv 1 \bmod \phi(n)$ or $ed \bmod \phi(n) =1$.

We have e = 5, $\phi(n) = 24$. So, $5d \bmod 24 =1$, and d = 5. Now, we find the private key PR = {d, n} = {5, 35}. Based on RSA decryption algorithm, $M = C^d \bmod n = 10^5 \bmod 35 = 5$.

We also can verify the correctness by the RSA encryption algorithm as the following: $C = M^e \bmod n = 5^5 \bmod 35 = 10$.

Therefore, we conclude that the plaintext M is 5.

3.2.2 Diffie-Hellman Key Exchange

UQ. Explain Diffie-Hellman key exchange algorithm and man-in-the-middle attack. (SPPU - Q. 6(b), Dec. 2012, 10 Marks)

UQ. Explain Diffie-Hellman key exchange. (SPPU - Q. 6(B), May 2013, 8 Marks)

UQ. Users A and B use the Diffie-Hellman key exchange technique with a common prime q = 71 and a primitive root a = 7.
(i) If user A has private key X_A = 5, what is A's public key Y_A ?
(ii) If user B has private key X_B = 12, what is B's public key Y_B ?
What is the shared secret key? (SPPU - Q. 6(a), Dec. 2013, 8 Marks)

UQ. Consider Diffie-Hellman scheme with common prime numbe q = 1 and a primitive root α = 2. If user A has private key Y_A = 9, what is private key X_A. (SPPU - Q. 6(b), Dec. 2014, 8 Marks)

- The Diffie Hellman (DH) Algorithm is a key-exchange protocol based on the **discrete logarithm problem,** which is used to exchange the secret key between the sender and the receiver. This algorithm facilitates the exchange of secret key without transmitting it over the Internet.
- It is named after their inventors **Whitfield Diffie and Martin Hellman**. Here, Keys are not actually exchanged but they are jointly derived. Diffie-Hellman key exchange algorithm is used for securely exchanging cryptographic keys over a public communications channel.
- To understand the algorithm, as shown in Fig. 2.3.2, consider two sample parties, Alice and Bob, initiating a dialogue. Consider Alice as sender and Bob as receiver.
- Each has a piece of information they want to share, while preserving its secrecy.
- The key is exchanged in the following precise steps :

Consider, Private key of the sender = X_A

Public key of the sender = Y_A

Private key of the receiver = X_B

Public key of the receiver = Y_B

▶ **Step 1 :** One of the parties choose two numbers 'g' and 'p' and exchange with the other party.

'g' is the primitive root of prime number 'p'.

After this exchange, both the parties know the value of 'a' and 'n'.

▶ **Step 2 :** Both the parties already know their own private key.

Both the parties calculate the value of their public key and exchange with each other.

Sender calculates its public key as- $Y_A = g^{X_A} \bmod n$

Receiver calculates its public key as- $Y_B = g^{X_B} \bmod n$

▶ **Step 3 :** Both the parties receive public key of each other.

Now, both the parties calculate the value of secret key.

Sender calculates secret key as:

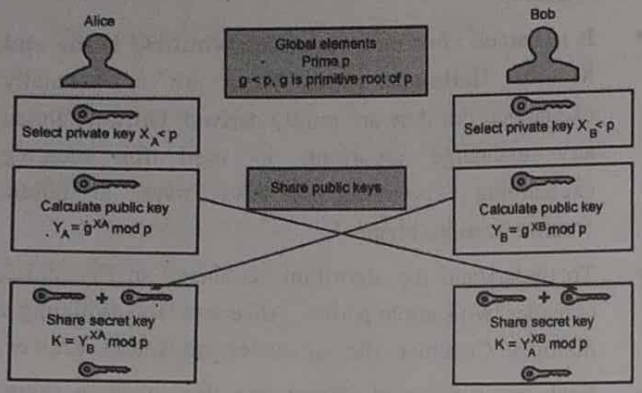
$$\text{Secret key} = (Y_B)^{X_A} \bmod p$$

Receiver calculates secret key as:

$$\text{Secret key} = (Y_A)^{X_B} \bmod p$$

Finally, both the parties obtain the same value of secret key.

As they decipher the other's message, they can extract the public information and with knowledge of their own secret, deduce the new information that was carried along.



(1B21)**Fig. 3.2.1 : Diffie-Hellman Key Exchange**

3.2.2(A) Advantages of Diffie-Hellman Key Exchange

(1) The sender and receiver does not need any prior knowledge of each other.

(2) Once the keys are exchanged, the communication of data can be done through an insecure channel.

(3) The sharing of the secret key is safe.

3.2.2(B) Disadvantages of Diffie-Hellman Key Exchange

(1) The algorithm can be used only for symmetric key exchange.

(2) It cannot be used for signing digital signatures.

(3) It is expensive in terms of resources and CPU performance time.

(4) It does not authenticate any party in the transmission, the Diffie Hellman key exchange is vulnerable to a man-in-the-middle attack.

Ex. 3.2.8 : Suppose that two parties A and B wish to set up a common secret key between themselves using the Diffie-Hellman key exchange technique. They agree on 7 as the modulus and 3 as the primitive root. Party A chooses 2 and party B chooses 5 as their respective secrets. Find their shared secret key.

☑ **Soln. : Given :** $p = 7$, $g = 3$

Private key of A $(X_A) = 2$;

Private key of B $(X_B) = 5$

▶ **Step 1 :** Both the parties calculate the value of their public key and exchange with each other.

Public key of A $(Y_A) = g^{X_A} \bmod n = 3^2 \bmod 7 = 2$

Public key of B $(Y_B) = g^{X_B} \bmod n = 3^5 \bmod 7 = 5$

▶ **Step 2 :** Both the parties calculate the value of secret key at their respective side.

Secret key obtained by A $= (Y_B)^{X_A} \bmod p = 5^2 \bmod 7 = 4$

Secret key obtained by B $= (Y_A)^{X_B} \bmod p = 2^5 \bmod 7 = 4$

Finally, both the parties obtain the same value of secret key.

The value of common secret key = 4.

Ex. 3.2.9 : In a Diffie-Hellman Key Exchange, Alice and Bob have chosen prime value q = 17 and primitive root = 5. If Alice's secret key is 4 and Bob's secret key is 6, what is the secret key they exchanged?

☑ **Soln. : Given :** $p = 17$, $g = 5$

Private key of Alice $(X_A) = 4$;

Private key of Bob $(X_B) = 6$

▶ **Step 1 :** Both the parties calculate the value of their public key and exchange with each other.

Public key of Alice $(Y_A) = g^{X_A} \bmod n = 5^4 \bmod 17 = 13$

Public key of Bob $(Y_B) = g^{X_B} \bmod n = 5^6 \bmod 17 = 2$

▶ **Step 2 :** Both the parties calculate the value of secret key at their respective side.

Secret key obtained by Alice $= (Y_B)^{X_A} \bmod p = 2^4 \bmod 17 = 16$

Secret key obtained by Bob $= (Y_A)^{X_B} \bmod p = 13^6 \bmod 17 = 16$

Finally, both the parties obtain the same value of secret key.

The value of common secret key = 16.

Ex. 3.2.10 : Given generator g = 2 and n = 11, Using the Diffie-Hellman algorithm, Solve the following:

Show that 2 is primitive root of 11.

If A's public key is 9, what is A's private key?

If B's public key is 3, what is B's private key?

Calculate the shared secret key.

☑ **Soln. :**

1. 2 is a primitive root of 11.

   Let p be a prime. Then a is a primitive root for p if the powers of a, 1, a, a^2, a^3, ... Include all of the residue classes mod p (except 0).

   $2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16 = 5, 2^5 = 5 * 2 = 10 = -1, 2^6 = 2 * (-1) = -2, 2^7 = 2 * (-2) = -4, 2^8 = 2 * (-4) = -8 = 3, \ 2^9 = 2 * (3) = 6, 2^{10} = 2 * 6 = 12 = 1.$

   Therefore, 2 is a primitive root of 11.

2. For A's private key

   A's public key = x = 9

   Private key of A = $g^x \bmod n = 2^9 \bmod 11 = 6$.

3. For B's private key

   B's public key = y = 3

   Private key of B = $g^y \bmod n = 2^3 \bmod 11 = 8$.

4. Shared secret key

   K1 = $B^x \bmod n = 8^9 \bmod 11 = 7$.

   K2 = $A^y \bmod n = 6^3 \bmod 11 = 7$.

   Therefore, the shared secret key is 7.

UEx. 3.2.11 (SPPU - Q. 6(a), Dec. 2013, 8 Marks) Users A and B use the Diffie-Hellman key exchange technique with a common prime 71 and primitive root 7. Show that 7 is a primitive root of 71.

If user A has private key x = 5, what is A's public key R1?

If user B has private key y=15, what is B's public key R2?

Calculate the shared secret key.

☑ **Soln. :**

▶ **Step 1 :** 7 is a primitive root of 71.

Let p be a prime. Then a is a primitive root for p if the powers of a, 1, a, a^2, a^3, ... Include all of the residue classes mod p (except 0).

$7^1 = 7, 7^2 = 49, 7^4 = 49^2 = 2401, 7^8 = 2401^2 = 2938, 7^{16} = 2938^2 = 1652, 7^{32} = 1652^2 = 1923, 7^{64} = 1923^2 = 2876, 7^{70} = 7^{64} * 7^4 * 7^2 = 2876 * 2401 * 49 = 1563$ not congruent to 1.

Therefore, 7 is a primitive root of 71.

▶ **Step 2 :** For A's public key

Public key of A (Y_A) = $g^{X_A} \bmod n = 7^5 \bmod 71 = 51$.

Therefore, A's public key = 51.

▶ **Step 3 : For B's public key**

Public key of (Y_B) = $g^{X_B} \bmod n = 7^{12} \bmod 71 = 4$

Therefore, B's public key = 4.

▶ **Step 4 : Shared secret key**

K1 = $B^x \bmod n = 4^5 \bmod 71 = 30$.

K2 = $A^y \bmod n = 30^{12} \bmod 71$

= $[(30^5 \bmod 71) * (30^5 \bmod 71) * (30^3 \bmod 71)] \bmod 71$

= [37*37*48] mod 71

= 65712 mod 71

= 37.

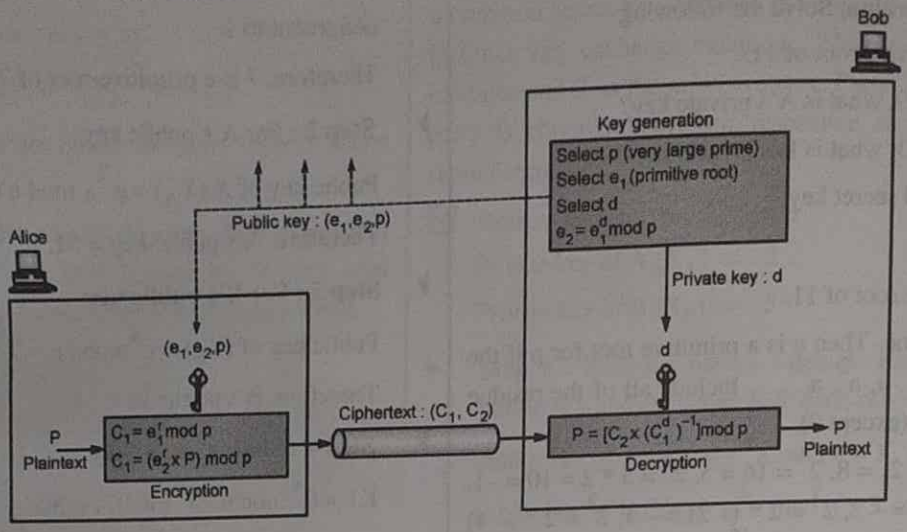
Therefore, the shared secret key is 37.

3.2.3 El Gamal Encryption

GQ. Write short note on : El Gamal Algorithm.

- The El Gamal cryptosystem is a **public key encryption** algorithm first described by Taher Elgamal in 1985 and is closely related to the Diffie-Hellman key exchange.
- The Diffie-Hellman key exchange provides a method of sharing a secret key between Alice and Bob, but does not allow Alice and Bob to otherwise communicate securely.
- ElGamal is a public key cryptosystem based on the discrete logarithm problem for a cyclic group G, i.e. every person has a key pair, the secret key and the public key. Given only the public key, one has to find the discrete logarithm to get the secret key.
- The cryptosystem is both an **encryption scheme** which helps sender and receiver with the problem of exchanging sensitive information over an insecure channel eavesdropped by their adversary and a **digital signature scheme** which helps them with creating digital signatures.
- The signature scheme is slightly different from the encryption scheme and various digital signature schemes such as the Schnorr signature scheme and the Digital Signature Algorithm (DSA) are based on ElGamal's signature scheme but with shorter keys.

☞ Working of ElGamal cryptosystem



(1B22)**Fig. 3.2.2 : ElGamal cryptosystem**

Suppose Alice wants to communicate to Bob, the following steps would be followed :

▶ **Step 1 : Key generation**

Bob generates public and private key.

Bob chooses a very large number p and a cyclic group G.

He selects d (private key) be the member of group G such that $1 < d < p - 2$.

He also selects e_1 to be primitive root in the group G.

Then he computes, $e_2 = e_1^d \bmod p$

Bob publishes Public key = (e_1, e_2, p) and retains Private key = d.

▶ **Step 2 : Encryption**

Alice encrypts data using Bob's public key.

Alice selects a random integer r from cyclic group G and let the plaintext be P.

Then she computes ciphertext.

$$C_1 = e_1^r \bmod p$$

$$C_2 = (e_2^r * P) \bmod p$$

Then she sends C_1 and C_2 over the channel.

▶ **Step 3 : Decryption**

Bob decrypts the message using his private key d.

Plaintext $P = [c_2 * (C_1^d)^{-1} \bmod p]$

3.2.3(A) Advantages of ElGamal Encryption

(1) Keys are generated using discrete logarithms.

(2) Encryption results are twice the size of the original size.

3.2.3(B) Disadvantages of ElGamal Encryption

(1) ElGamal is slow and uncanny.

(2) A known-plain text attack is possible in ElGamal if the same r is used twice during encryption.

(3) The ciphertext is twice as long as the plaintext due to message expansion by a factor of two takes place during encryption.

(4) A safe prime number makes generation of large-enough keys.

(5) One property of ElGamal is, that it is (semi-) homomorphic w.r.t. the group operation i.e, malleable. You can see that as an unwelcome property. If you consider this property useful or a security risk, depends on your actual goal.

☞ **Example :**

Let p = 131 and e1 = 2.

Let Bob's private key, d = 97.

So, his public key is $e_2 = e_1^d \bmod p = 2^{97}$

mod 131 ≡ 14.

Let the message to be sent (P) = 75.

Alice selects a random integer r = 33.

Alice then computes,

$C_1 = e_1^r \bmod p = 2^{33} \bmod 131 \equiv 103.$

$C_2 = (e_2^r * P) \bmod p = (14^{33} * 75) \bmod 131 \equiv 51.$

Bob decrypts the message using,

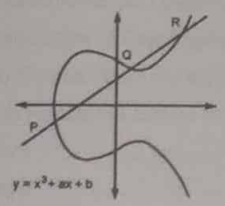
Plaintext $P = [c_2 * (C_1^d)^{-1} \bmod p] = 51 * (103^{-97}) \bmod 131 = 75.$

3.2.4 Elliptical Curve Cryptography (ECC)

UQ. Explain Elliptic Curve Cryptography in details.

(SPPU - Q. 6(b), Dec. 2013, 8 Marks, Q. 6(b), May 2015, 8 Marks, Q. 5(b), May 2018, 8 Marks, Q. 3(a), May 2019, 5 Marks, Q. 6(b), Dec. 2013, 8 Marks, Q. 6(b), Dec. 2014, 8 Marks)

- RSA and Diffie-Hellman cryptographic methods are based on the creation of keys by using very large prime numbers. Hence, key creation requires great computational power.
- **Elliptic curve cryptography** makes use of **elliptic curve equation** instead of the traditional method of generation as the product of very large prime numbers to create faster, smaller, and more efficient cryptographic keys.



(1B23)**Fig. 3.2.3 : Elliptic Curve**

- It is a **public key encryption** technique, focuses on pairs of public and private keys for decryption and encryption.
- With ECC, you can use smaller keys to get the same levels of security. In a world where mobile devices must do more and more cryptography with less computational power, ECC offers high security with faster, shorter keys compared to RSA.
- ECC has gained popularity recently due to its smaller key size and ability to maintain security. In contrast to RSA, ECC bases its approach on how elliptic curves are structured mathematically over finite fields.
- Therefore, ECC generates keys that are more difficult to crack. Due to this, ECC is the next generation implementation of public key cryptography and more secure than RSA.
- An elliptic curve for ECC is a plane curve over a finite field which is made up of the points satisfying the equation: $y = x^3 + ax + b$.
- As shown in Fig. 2.3.4, elliptic curve cryptography example, any point on the curve can be mirrored over the x-axis and the curve will stay the same.
- Any non-vertical line will intersect the curve in three places or fewer. Equations based on elliptic curves are relatively easy to perform, and extremely difficult to reverse. This characteristic is very valuable for cryptography purposes.

3.2.4(A) Advantages of ECC

(1) Size is a serious advantage of elliptic curve cryptography, because it translates into more power for smaller, mobile devices.

(2) You can achieve security using smaller keys using ECC.

(3) RSA's factoring encryption is more vulnerable, ECC is far simpler and requires less energy to factor.

3.2.4(B) Disadvantages of ECC

(1) It increases the size of the encrypted message significantly more than RSA encryption.

(2) ECC algorithm is more complex and more difficult to implement than RSA.

Descriptive Questions

Q. 1 Given modulus n = 91 and public key e = 5, find the values of p, q and phi(n) and d using RSA. Encrypt M = 25. Also perform Decryption.

Q. 2 Given generator g = 2 and n = 11, Using the Diffie-Hellman algorithm, Solve the following:

a. Show that 2 is primitive root of 11.

b. If A's public key is 9, what is A's private key?

c. If B's public key is 3, what is B's private key?

d. Calculate the shared secret key.

Q. 3 Given modulus n = 221 and public key, e = 7, find the values of p, q, phi(n), and d using RSA. Encrypt M = 5.

Q. 4 Explain Diffie-Hellman algorithm in detail with suitable example.

Q. 5 Use Chinese Remainder Theorem and find x such that

x = 2 mod 3

x = 3 mod 5

x = 2 mod 7

Q. 6 State and explain Fermat's Little theorem with suitable example and use.

Q. 7 What is the importance of prime numbers in cryptography?

Q. 8 Explain El-Gammal algorithm in brief.

Q. 9 Explain Elliptical Curve Cryptography in detail.

Q. 10 Use Fermat's Little Theorem to compute:

5^{40} mod 7 (b) 13^{29} mod 23

...Chapter Ends

❑❑❑