20CB603/20CS603/20DS603/20IT603

III/IVB. Tech (Regular) DEGREE EXAMINATION

Scheme of Evaluation

May,2026 **Common to CB,CS, DS &IT**

Sixth Semester **Cryptography & Network Security**

Time: Three Hours **Maximum:70 Marks**

Answer question 1 compulsorily. **(14X1 = 14 Marks)**

Answer one question from each unit. **(4X14=56 Marks)**

			CO	BL	M
1.	a)	Define a cryptographic attack A cryptographic attack is a malicious attempt to undermine, bypass, or expose weaknesses in an encryption system.	CO1	L1	1M
	b)	What is a security service? Security service refer to capabilities and measures designed to protect people, property, data, and assets from threats like unauthorized access, theft, or cyberattacks.	CO1	L1	1M
	c)	What is a security mechanism? Security mechanisms are the technical tools, processes, and safeguards designed to protect systems, networks, and data from attacks.	CO1	L1	1M
	d)	What is a substitution cipher? A substitution cipher is a method of encryption where units of plaintext (usually single letters) are systematically replaced with other letters, numbers, or symbols.	CO1	L1	1M
	e)	What is AES? **AES** (Advanced Encryption Standard) is the global standard for encrypting electronic data. Published by the US National Institute of Standards and Technology (NIST) in 2001, it is a highly secure, symmetric-key algorithm used to scramble data so it cannot be read without the correct secret key.	CO2	L1	1M
	f)	Name any one transformation used in AES. The Advanced Encryption Standard (AES) operates on a 128-bit block arranged as a 4x4 grid of bytes, known as the "State". Each encryption round consists of four distinct transformations: • SubBytes • ShiftRows • MixColumns • AddRoundKey A Student can write any one of these transformations.	CO2	L1	1M
	g)	What is asymmetric key cryptography? Asymmetric key cryptography, or public-key cryptography, uses a mathematically linked pair of keys: a public key for encryption and a private key for decryption.	CO2	L1	1M
	h)	What is RSA algorithm? The RSA (Rivest-Shamir-Adleman) algorithm is a foundational public-key cryptography system used to securely transmit data over the internet.	CO2	L1	1M
	i)	What is a digital signature? A digital signature is a cryptographic mathematical scheme used to verify the authenticity, integrity, and non-repudiation of a digital document, message, or software.	CO3	L1	1M
	j)	What is the purpose of digital signature? Digital signatures are cryptographic tools used to authenticate the identity of a sender and ensure that digital documents or messages remain secure and unaltered.	CO3	L1	1M
	k)	Define KDC. A **Key Distribution Center (KDC)** is a central cryptographic server in a network (typically within the Kerberos protocol) that securely issues session keys and access tickets.	CO3	L1	1M
	l)	What is SSL? SSL (Secure Sockets Layer) is the standard security technology for establishing an encrypted link between a web server and a browser.	CO4	L1	1M

	m)	What is TLS? TLS (Transport Layer Security) is the standard cryptographic protocol used to secure communications over the internet	CO4	L1	1M
	n)	Differentiate two modes of IPSec.	CO4	L1	1M

Transport Mode	**Tunnel Mode**
IP Payload is encrypted	IP Payload is encrypted
IP header is not encrypted	IP header is encrypted
Original IP header is used for routing decisions	New IP packet encapsulates the encrypted one with a new header that is used for routing decisions.

Unit –I

2.	a)	Explain the security goals, services, and mechanisms in cryptography with examples.	CO1	L2	7M

1. Security Goals (The Objectives)

Security goals are the fundamental objectives of any secure information system.

- **Confidentiality:** Ensures that sensitive data is kept private and accessible only to authorized individuals.
- **Integrity:** Guarantees that data has not been altered, tampered with, or destroyed during transit or storage.
- **Authentication:** Validates the identity of the communicating parties to ensure they are who they claim to be.
- **Non-repudiation:** Prevents a sender or receiver from denying their involvement in a communication or transaction.
- **Availability:** Ensures that information and systems are accessible and usable by authorized parties upon demand.

2. Security Services (The Capabilities)

Security services are the system-level capabilities that implement the security goals.

- **Data Confidentiality Service:** Protects data against unauthorized disclosure.
- **Data Integrity Service:** Assures that messages are received exactly as sent, with no duplication, insertion, or modification.
- **Authentication Service:** Ensures the origin of a message or the identity of the two communicating parties is genuine.
- **Non-repudiation Service:** Provides proof of origin and proof of delivery, such that neither party can deny the message.
- **Access Control Service:** Dictates and enforces exactly who or what is allowed to access specific system resources.

3. Security Mechanisms (The Tools)

Security mechanisms are the actual mathematical and cryptographic processes designed to detect, prevent, or recover from security attacks.

- **Encipherment (Encryption):**
  - *What it is:* The mathematical transformation of readable plaintext into unreadable ciphertext to provide confidentiality.
  - *Example:* Converting a plaintext message like \("Hello World"\) into an encrypted string using an AES Encryption Tool.
- **Digital Signatures:**
  - *What it is:* An electronic signature appended to data that binds the sender's identity to the message. It achieves both authentication and non-repudiation.
  - *Example:* Using a private key to "sign" a digital contract so the receiver can mathematically verify the sender using a public key.
- **Data Integrity Mechanisms:**
  - *What it is:* Cryptographic hash functions that compute a fixed-size check value from the data.
  - *Example:* Running a file through a SHA-256 Hash Calculator to produce a 64-character hash; if the file is modified by even a single byte, the resulting hash changes completely.
- **Authentication Exchange:**
  - *What it is:* Mechanisms used to transfer information to establish identity, such as handshaking protocols.
  - *Example:* The TCP 3-Way Handshake or sending one-time passwords (OTP) during a login attempt

	b)	Discuss different types of cryptographic attacks and their impact on security.	CO1	L2	7M

Types of Cryptographic Attacks:

Ciphertext-Only Attacks

In this scenario, the attacker has access only to the encrypted message. While seemingly limited, skilled adversaries can sometimes deduce patterns or weaknesses, especially in poorly implemented or outdated cryptographic systems. This type of attack relies heavily on statistical analysis & pattern recognition.

Example: The breaking of the Enigma machine during World War II was initially based on ciphertext-only attacks, combined with educated guesses about message contents.

Known-Plaintext Attacks

Here, the attacker possesses both the plaintext & its corresponding ciphertext. This additional information can be leveraged to reverse-engineer the encryption key or algorithm. Known-plaintext attacks are more powerful than ciphertext-only attacks as they provide a clear relationship between the original message & its encrypted form.

Example: Linear cryptanalysis, a technique used to break block ciphers, is a form of known-plaintext attack.

Chosen-Plaintext Attacks

A more potent variant, where the attacker can choose specific plaintext to be encrypted &analyze the resulting ciphertext. This method can reveal vulnerabilities in the encryption process. Chosen-plaintext attacks are particularly dangerous because they allow the attacker to select inputs that might expose weaknesses in the cryptographic system.

Example: The padding oracle attack, which can be used against some modes of operation for block ciphers, is a type of chosen-plaintext attack.

Man-in-the-Middle Attacks

These attacks involve intercepting communication between two parties, potentially allowing the attacker to eavesdrop, modify or inject false information. Man-in-the-middle attacks are particularly insidious because they can bypass even strong encryption if the initial key exchange is compromised.

Example: SSL stripping, where an attacker downgrades an HTTPS connection to HTTP, is a form of man-in-the-middle attack.

Side-Channel Attacks

Rather than directly attacking the cryptographic algorithm, these attacks exploit information leaked during the encryption process, such as power consumption or electromagnetic emissions. Side-channel attacks are particularly concerning because they can bypass mathematically secure algorithms by exploiting their physical implementation.

Example: Timing attacks, which analyze the time taken to perform cryptographic operations, can sometimes reveal secret keys.

Birthday Attacks

Exploiting the mathematics of probability, these attacks aim to find collisions in hash functions, potentially compromising the integrity of digital signatures or password systems. The name comes from the birthday paradox in probability theory.

Example: Finding two different inputs that produce the same MD5 hash, which has been practically demonstrated, is a form of birthday attack.

Brute-Force Attacks

The most straightforward yet computationally intensive approach, involving trying every possible key until the correct one is found. While conceptually simple, brute-force attacks can be highly effective against systems with short key lengths or weak key generation.

Example: Password cracking tools often use a combination of dictionary attacks & brute-force methods to guess passwords.

Advanced Attack Techniques

- **Differential Cryptanalysis:** This powerful technique analyzes how differences in plaintext pairs affect the resulting ciphertext differences. It's particularly effective against block ciphers & has led to the breaking of several once-thought-secure algorithms.
- **Algebraic Attacks:** These attacks exploit the algebraic structure of cryptographic algorithms, attempting to solve systems of equations to recover the key. They're particularly relevant to certain stream ciphers & some public-key systems.
- **Quantum Attacks:** While still theoretical, quantum attacks leverage the principles of quantum computing to break cryptographic systems. Shor's algorithm, for instance, could potentially break widely used public-key cryptography systems in polynomial time on a quantum computer.

		The primary security impacts of successful cryptographic attacks fall into three main categories: **1. Data Breaches & Loss of Confidentiality** When attackers break encryption, they gain unauthorized access to plaintext data that was supposed to be securely hidden. [1] • **Massive Information Theft:** Attackers can steal massive databases of personally identifiable information (PII), intellectual property, or classified documents. • **Credential Exposure:** Leaked passwords, API keys, and cryptographic keys can be used by hackers to infiltrate internal networks or launch further attacks. • **Real-World Damage:** Companies face customer churn, loss of consumer trust, and long-term brand damage. **2. Data Manipulation & Loss of Integrity** Attackers do not always need to decrypt the data entirely; if they can alter the encrypted communication or files. • **Unauthorized Modifications:** Attackers can tamper with financial transactions, alter code updates, or manipulate system instructions without detection. • **Fraud:** Attackers can conduct identity theft, redirect wire transfers, or forge digital signatures to authorize malicious activities. **3. Business & Legal Consequences** The fallout from inadequate cryptographic security extends far beyond the technical compromise. • **Regulatory Fines:** Organizations frequently face hefty penalties for failing to protect user data in accordance with strict privacy laws. • **Operational Downtime:** Systems may be forcibly taken offline for emergency incident response, forensic investigations, and security patches. • **Legal Liability:** Class-action lawsuits and legal liabilities often follow large-scale compromises of user data.			
		(OR)			
3.	a)	Explain monoalphabetic substitution ciphers with suitable examples. **Monoalphabetic Cipher** is a part of the substitution technique in which a single cipher alphabet is used per message (mapping is done from plain alphabet to cipher alphabet). Monoalphabetic cipher converts plain text into cipher text and re-convert a cipher text to plain text. Monoalphabetic Cipher eliminates the brute-force techniques for cryptanalysis. Moreover, the cipher line can be a permutation of the 26 alphabetic characters. In **Cryptography**, various encryption techniques are used to provide data security. The classical Encryption technique is categorized into two divisions: **1. Substitution Cipher Technique** • Caesar Cipher • **Monoalphabetic Cipher** • Polyalphabetic Cipher • Playfair Cipher • Hill Cipher • One-time Pad **Types of Monoalphabetic Substitution Ciphers** **Additive Cipher** It is also Known as Shift Cipher which shifts plain text to form Cipher-text. • **Mathematical Expression**: ○ **For Encryption:** C=(P + K) mod 26 where 'P' is the character in plain text, 'K' is the key, and 'C' is the Cipher. ○ **For Decryption:** P=(C-K) mod 26. • **Example :Input : P=** GEEKS ,Key= 4 .Output : C=KIIOW **Caesar Cipher** A type of Addictive Cipher but the value of key is always '3' here. • **Mathematical Expression:** ○ **For Encryption:** C=(P + K) mod 26 where 'P' is the character in plain text, 'K' is the key, and 'C' is the Cipher. ○ **For Decryption**: P=(C-K) mod 26 • **Example:** Input: P=GEEKS ,Output : C=JHHNV **Multiplicative Cipher** Letters are changed here using a multiplication key.	CO1	L2	7M

- **Mathematical Expression:**
  - **For Encryption**: C=(P * K) mod 26 where, 'P' is the character in plain text, 'K' is the key, and 'C' is the Cipher.
- **Example:** Input : P=VMH , Key= 3 .Output : C=HEL

Affine Cipher

A mathematical function is used to convert plain text into cipher text.

- **Mathematical Expression:**
  - **For Encryption** : C=(P * K1+K2) mod 26 where, 'P' is the character in plain text, 'K1' is the multiplicative key, 'K2' is the additive key and 'C' is Cipher.
  - **For Decryption :** P=((C-K2)/ K1) mod 26.
- **Example :** Input : P=ARM , Key1=3,Key2=5 .Output : C=HEL

How Does Monoalphabetic Cipher Work?

Let us understand how this cipher technique works with an example, let us suppose the mapping of plaintext using the table below.

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
A	B	C	D	E	F	G	H	I	J	K	L	M

13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Example Input (Plain-text)- GFG

- **Explanation:** In Monoalphabetic cipher, the mapping is done randomly and the difference between the letters is not uniform. Here, the word is mapped to S (G->S), F is mapped to R(F->R) and G was already mapped to S so we cannot change it (G->S).
- **Example Output (Cipher-text)**- SRS

Advantages of Monoalphabetic Cipher

- Better Security than Caesar Cipher.
- Provides Encryption and Decryption to data.
- Monoalphabetic Cipher maintains a frequency of letters.

Disadvantages of Monoalphabetic Cipher

- Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet.
- Prone to guessing attack using the English letters frequency of occurrence of letters.
- The English Language is used so the nature of plain text is known.
- Less secure than a polyalphabetic cipher.

Student can take any example to explain the working of **Monoalphabetic Cipher**.

b) Describe transposition ciphers and compare them with substitution ciphers. CO1 L2 7M

A **transposition cipher** is an encryption method where the original letters or units of the plaintext are systematically rearranged to form the ciphertext, without changing the identity of the letters themselves.

How Transposition Ciphers Work

Instead of replacing letters, a transposition cipher scrambles their order based on a specific algorithm or geometric design. For example, in a **Columnar Transposition**, you write the plaintext horizontally into a grid of a fixed width, then read the letters vertically by column to create the encrypted message. [1, 2]

- **Plaintext:** HELLO WORLD
- **Ciphertext:** H L O L D E O W R L (written in columns of 4)

Common examples of transposition ciphers include:

- **Rail Fence Cipher:** The plaintext is written diagonally across a number of "rails" and then read in rows.
- **Route Cipher:** Text is written into a grid and the path used to read it out (e.g., spiral, boustrophedon) acts as the key.

Substitution vs. Transposition

The foundational difference comes down to **identity** versus **position**

Feature	Substitution Cipher	Transposition Cipher
Mechanism	Changes the *identity* of the letters (replaces them with other letters, numbers, or symbols).	Rearranges the *position* of the letters; the letters themselves remain exactly the same.
Core Principle	Provides **Confusion**: conceals the relationship between the plaintext and ciphertext.	Provides **Diffusion**: spreads the redundancy of the plaintext across the entire message.
Vulnerability	Vulnerable to frequency analysis because common letters in the language maintain their high frequency.	Resistant to simple frequency analysis since letter frequencies remain the exact same as the original language.
Common Examples	Caesar Cipher, Monoalphabetic / Polyalphabetic Ciphers, Vigenère.	Columnar Transposition, Rail Fence, Route Cipher.

Unit –II

4.	a)	Explain the Advanced Encryption Standard (AES) algorithm with its transformations.	CO2	L2	7M

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

The features of AES are as follows −

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

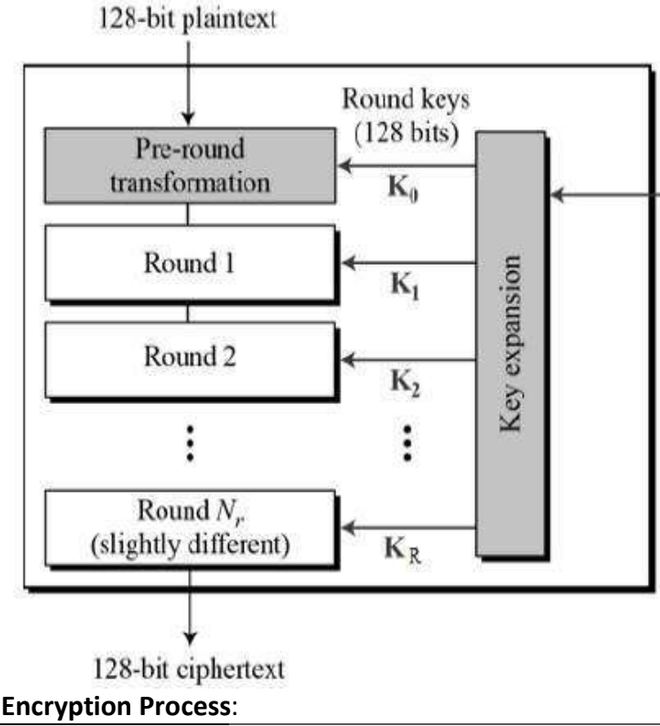
Operation of AES

AES is an iterative rather than Feistel cipher. It is based on substitutionpermutation network. 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).

Interestingly, 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 −

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

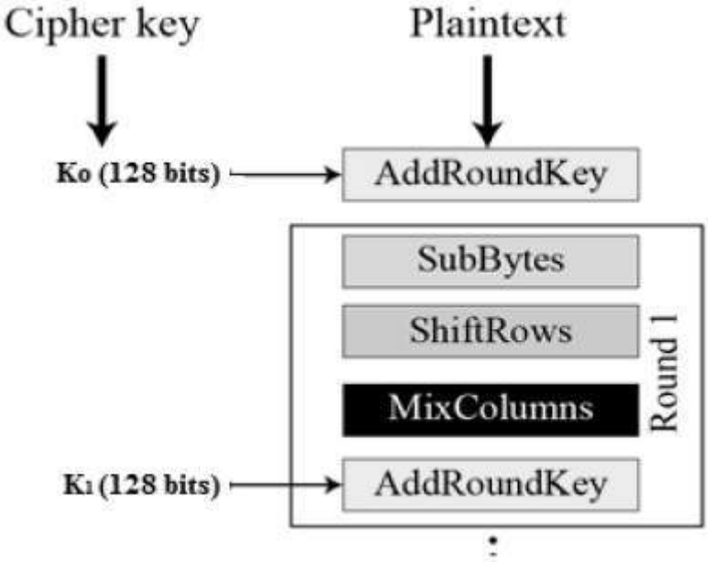
The schematic of AES structure is given in the following illustration −



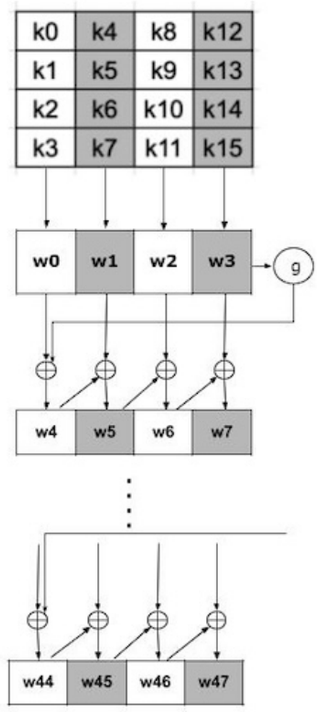
R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Encryption Process:

		Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below −  Byte Substitution (SubBytes) 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. Shiftrows 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. Shift is carried out as follows − • First row is not shifted. • Second row is shifted one (byte) position to the left. • Third row is shifted two positions to the left. • Fourth row is shifted three positions to the left. • The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other. MixColumns 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. It should be noted that this step is not performed in the last round. Addroundkey 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. Each round consists of the four processes conducted in the reverse order − • Add round key • Mix columns • Shift rows • Byte substitution Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms needs to be separately implemented, although they are very closely related. AES Analysis In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered. Additionally, 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.			
	b)	Describe the key expansion process in AES. How are round keys generated? For use in AES encryption, a single initial key can be expanded into a series of round keys using the AES (Advanced Encryption Standard) key expansion technique. These round keys are needed for each round of AES encryption and decryption.	CO2	L2	7M

The AES key expansion method receives a four-word (16-byte) key and returns a linear array of 44 words (176 bytes). This is sufficient to provide both the initial AddRoundKey step and a four-word round key for each of the cipher's ten rounds.



How algorithm works?

In simple terms, the AES key expansion algorithm functions as follows −

- **Initial Key** − The algorithm makes use of an initial key. Depending on the level of security needed, this key's length can range from 128 to 256 bits.
- **Round Constants** − The approach makes use of a set of round constants, which are predefined values used in the key expansion process.
- **Word Size** − The key divides words into individual blocks. A word typically has 32 bits in it. For example, four 32-bit words are created from a 128-bit key.
- **Key Schedule** − The term "key schedule" refers to a set of round keys generated using the key expansion process. The initial round key and the extra round keys that were derived from it are both included in this schedule.
- **Expansion Rounds** − The algorithm performs several tasks in each expansion round, such as −
  - **RotWord** − This function rotates the bytes in a word.
  - **SubWord** − Applyes a substitution operation using a predetermined S-box.
  - **Rcon** − XORs the word using a round constant.
- **Round Keys** − The order of round keys that are still in place following all expansion rounds defines the key schedule. Each round key is used in the corresponding round of AES encryption or decryption.

Overall, the key expansion method increases security and prevents cryptographic attacks by ensuring that each AES encryption and decryption round has a unique round key.

Mathematical Representation

Let us denote −

- K is the first key, having N bits in its length.
- The word count (N_k) of the key is its total word count (e.g., 4 for a 128-bit key, 6 for a 192-bit key, and 8 for a 256-bit key).
- N_r, the AES round identifier, is 10 for AES-128, 12 for AES-192, and 14 for AES-256.

Key Expansion Process

- Step 1: Make the round keys first. Establish the first word K in a word array W. Iterate to generate (N_r + 1) round keys −

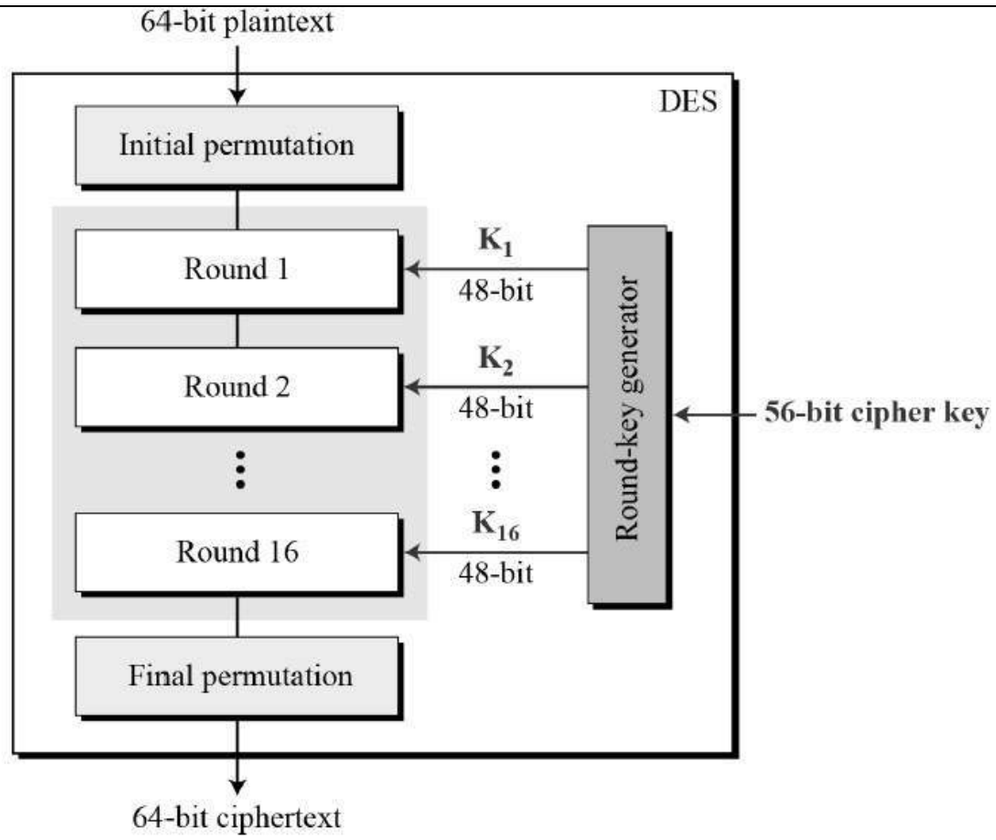
```
for i = N_k to (4 * (N_r + 1) - 1):
  if i mod N_k == 0:
    temp = RotWord(W[i-1]) ^ SubWord(W[i-1]) ^ Rcon(i / N_k)
  else:
    temp = W[i-1] ^ W[i-N_k]
  W[i] = temp
```

- Step 2: Finalize Round Keys: After the loop, the array W has all round keys.

Notations

- **RotWord(w)** − Rotate the word w's bytes in a manner that is cyclic.

		• **SubWord(w)** − Use the AES S-box to swap out each byte in the word w. • **Rcon(i)** − Produce the current round's i 's round constant, or Rcon. Compute the Round Constant Rcon The round constant for the i-th round can be found by computing **Rcon(i)=(RC[i],0,0,0)**, where RC[i] is the i-th element of the round constant array. This representation provides a mathematical description of how the original key is transformed into a set of round keys used in AES encryption and decoding. Example of AES Key Expansion Now we are going to discuss the AES (Advanced Encryption Standard) key expansion algorithm with an example. For this example, we will be using AES-128, so our initial key will be 128 bits (16 bytes). The first key can be considered as a set of bytes − 2b 7e 15 16 28 ae d2 a6 ab f7 97 66 76 15 13 1 This key is 128 bits long. Let's look at every step of the key expansion process − Initial Key 2b 7e 15 16 28 ae d2 a6 ab f7 97 66 76 15 13 1 Expansion Rounds We start by adding the first key to our list of round keys. Following that, we iteratively generate more round keys until we reach a total of 11. Round 1 In the RotWord, SubWord, and XOR operations, we use the round constant − RotWord: 7e 15 16 2b SubWord: 63 cb e7 8c Rcon: 01 00 00 00 Round 1 Key: 63 cb e7 8c 09 cf 4f 3c 3b a9 82 fb 11 13 d8 2c Round 2 The word from the previous round key is XORed four positions back − Round 2 Key: a0 fa fe 17 88 54 2c b1 23 a3 39 39 2a 6c 76 05 Subsequent Rounds Round keys are generated in this way until a total of 11 keys have been generated. Final Round Keys Round 0 (Initial Key): 2b 7e 15 16 28 ae d2 a6 ab f7 97 66 76 15 13 1 Round 1: 63 cb e7 8c 09 cf 4f 3c 3b a9 82 fb 11 13 d8 2c Round 2: a0 fa fe 17 88 54 2c b1 23 a3 39 39 2a 6c 76 05 ... Round 10: 3d 47 0e 52 77 37 2e 10 1f 7e 0e 20 6a 51 7f a7 In this example, an original key was expanded into a number of round keys using the AES key expansion technique. Every round key is created from the one before it using round constants and operations like XOR, RotWord, and SubWord. These round keys are then used in each round of AES encryption to increase security and prevent cryptographic attacks.			
		(OR)			
5.	a)	Explain in detail about DES function. The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST). DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure of DES is depicted in the following illustration −	CO2	L2	7M

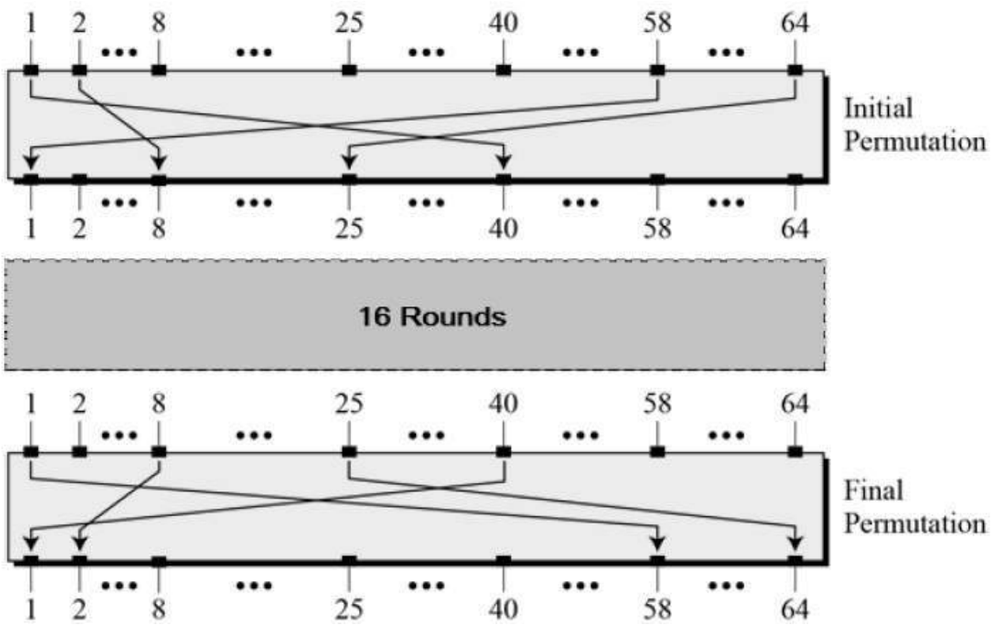


Since DES is based on the Feistel Cipher, all that is required to specify DES is −

- Round function
- Key schedule
- Any additional processing − Initial and final permutation

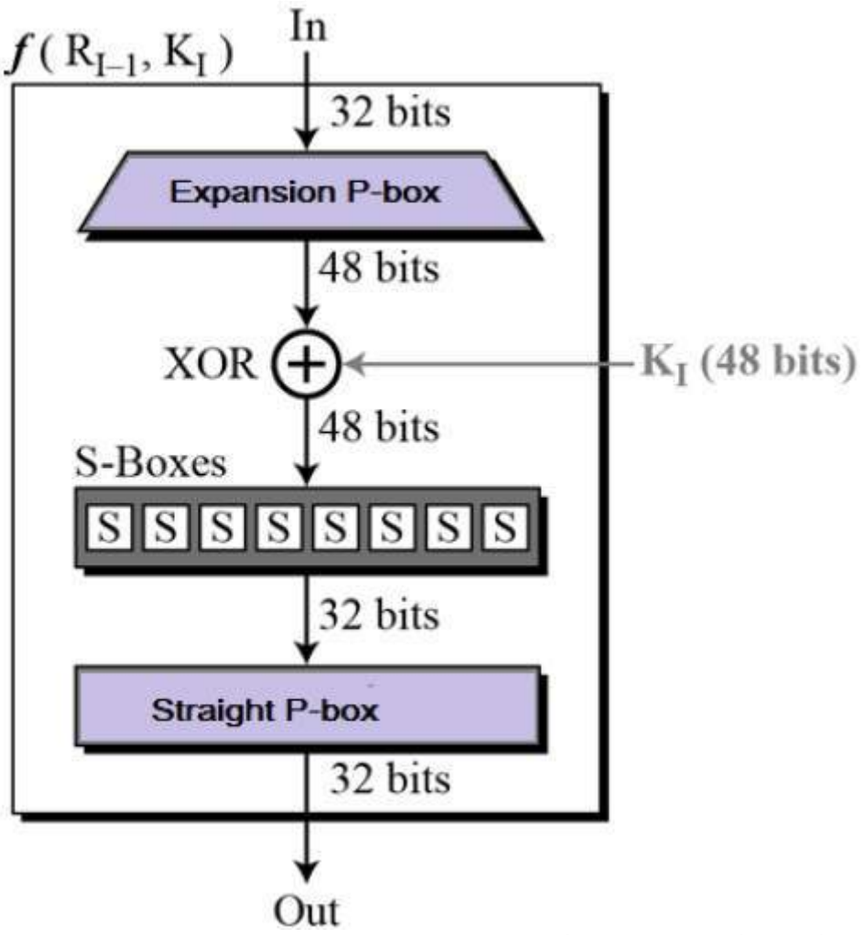
Initial and Final Permutation

The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other. They have no cryptography significance in DES. The initial and final permutations are shown as follows −

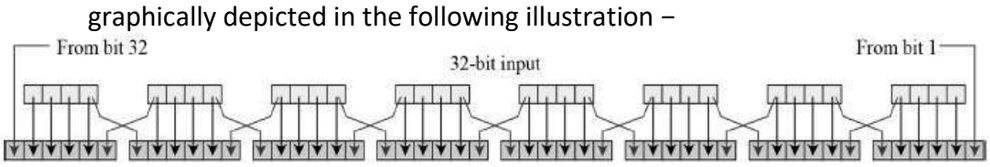


Round Function

The heart of this cipher is the DES function, *f*. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



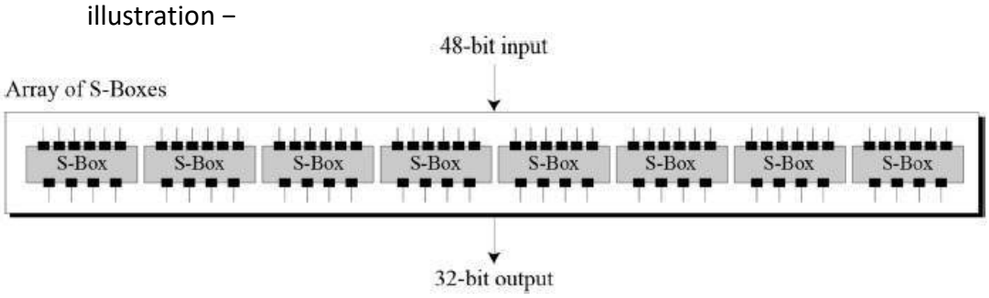
- **Expansion Permutation Box** − Since right input is 32-bit and round key is a 48-bit, we first need to expand right input to 48 bits. Permutation logic is graphically depicted in the following illustration −



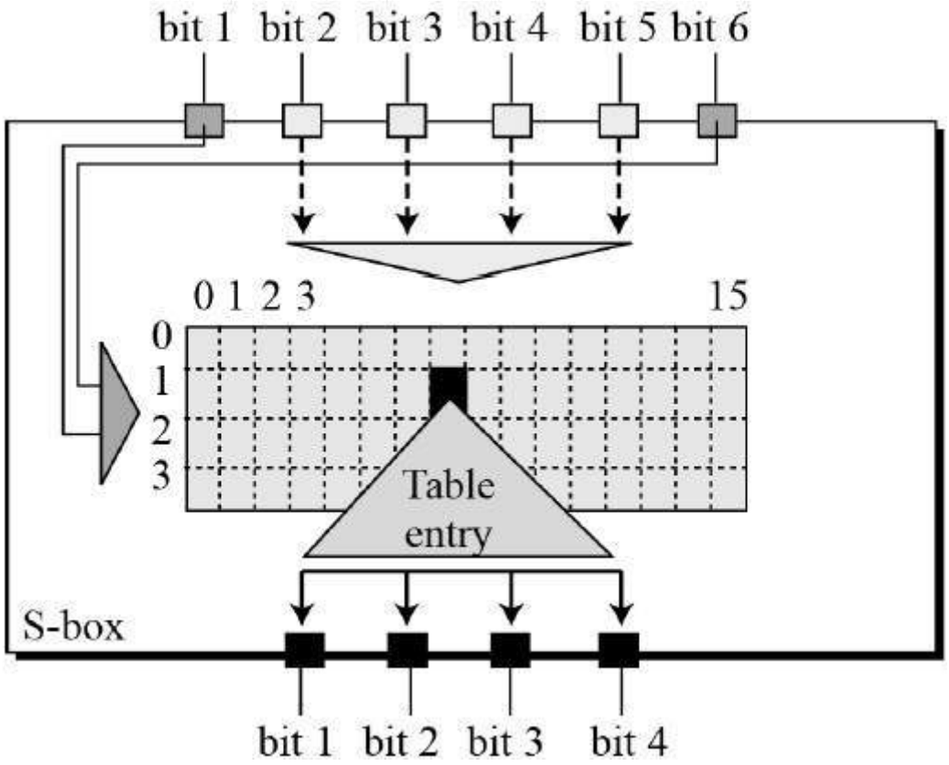
- The graphically depicted permutation logic is generally described as table in DES specification illustrated as shown −

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

- **XOR (Whitener).** − After the expansion permutation, DES does XOR operation on the expanded right section and the round key. 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 following illustration −



- The S-box rule is illustrated below −

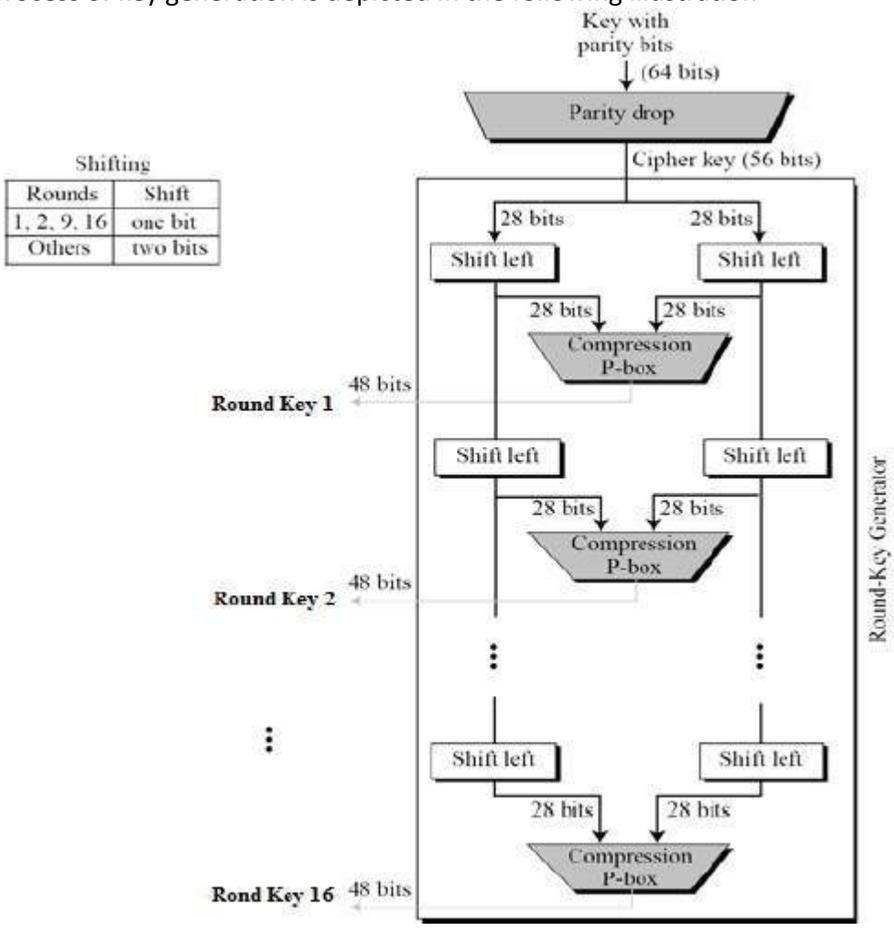


- There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.
- **Straight Permutation** − The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:

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

Key Generation

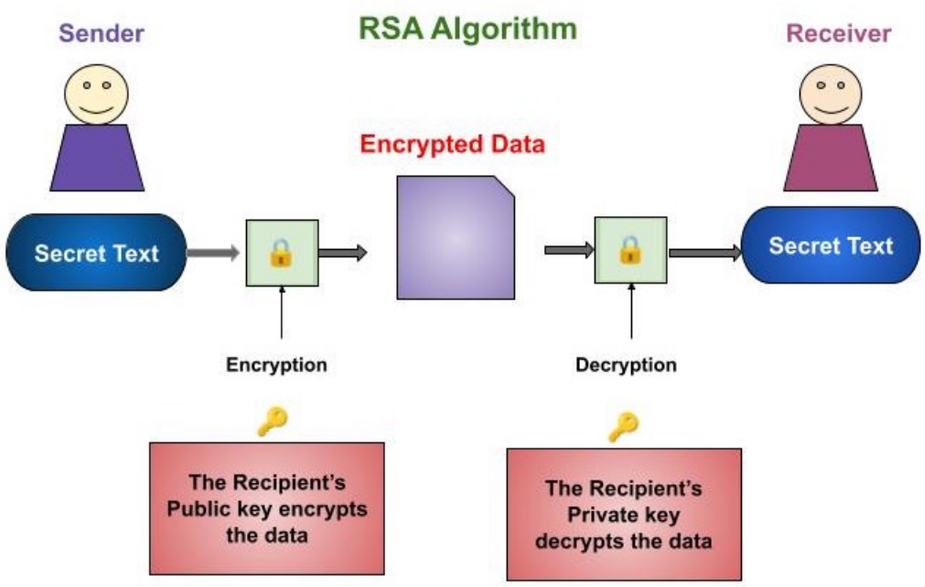
The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. The process of key generation is depicted in the following illustration −

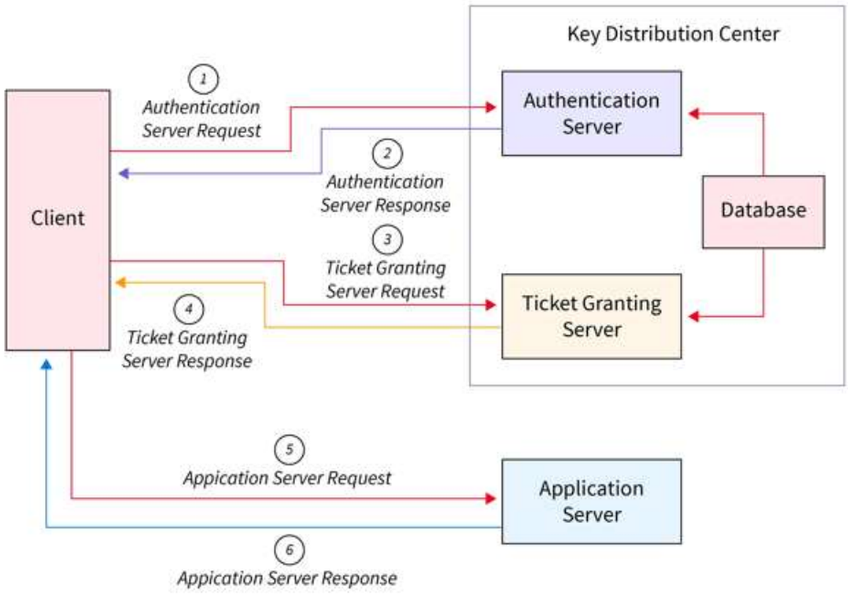


The logic for Parity drop, shifting, and Compression P-box is given in the DES description.

DES Analysis

The DES satisfies both the desired properties of block cipher. These two properties

		make cipher very strong. • **Avalanche effect** – A small change in plaintext results in the very great change in the ciphertext. • **Completeness** – Each bit of ciphertext depends on many bits of plaintext. During the last few years, cryptanalysis have found some weaknesses in DES when key selected are weak keys. These keys shall be avoided. DES has proved to be a very well designed block cipher. There have been no significant cryptanalytic attacks on DES other than exhaustive key search.			
	b)	Discuss asymmetric key cryptography. Explain the RSA cryptosystem with an example. In simple terms, **Asymmetric Key Cryptography** (also known as Public Key Cryptography) is like having a mailbox with two different keys. Anyone can walk up and drop a letter through the slot (**Public Key**), but only the owner of the mailbox has the specific key to open it and read the mail (**Private Key**). [1, 2, 3, 4, 5] Unlike symmetric encryption, where you use the same password to lock and unlock a file, asymmetric encryption uses a mathematically linked pair of keys. If you encrypt data with the Public Key, only the corresponding Private Key can decrypt it. [1, 2, 3, 4] --- **The RSA Cryptosystem** RSA (Rivest-Shamir-Adleman) is a famous encryption scheme that makes use of a combination of public and private keys. This means you have a non-public key and one that can be shared publicly. Each key can be used to encrypt data, but only the opposite can be decrypted. RSA was evolved in 1977 via MIT researchers Ron Rivest, Adi Shamir and Leonard Adleman, whose name bears the set of rules call.  Despite its popularity as a robust encryption method, RSA's high computing demands makes it inefficient and resource-intensive. As a result, it may not be the best choice for encrypting large messages or files because it may drain the system's resources. How does RSA work? RSA is based on the problem of breaking down large numbers into their top factors. To create an RSA key pair, you need to pick very big prime numbers, p and q. It is crucial to pick those primes randomly and ensure they are simply unique from each different. The product of p and q, represented as n, becomes the modulus for the public and private keys. While n is publicly known, the values of p and q remain confidential. The carmecheals' totient function of the product of two primes, p and q, is computed. An integer, e, is chosen as the public exponent. The final step involves calculating d, which serves as the private exponent. Example Let's say we choose p = 61 and q = 53. n = p * q = 61 * 53 = 3233 (n) = (p-1)(q-1) = 60 * 52 = 3120 Let's choose e = 17 (a common choice). d = 2753 (computed such that (17 * d) mod 3120 = 1) So, the public key is (3233, 17), and the private key is (3233, 2753). Encryption	CO2	L2	7M

		To encrypt a message, the sender Uses the public key (n, e) provided by the recipient. Converts the plaintext message (m) into a ciphertext (c) using the formula – c = m^e mod n. Suppose we want to encrypt the message "HELLO". We convert it to ASCII: H(72) E(69) L(76) L(76) O(79). We will encrypt each ASCII value separately – • H: C = 72^17 mod 3233 = 2103 • E: C = 69^17 mod 3233 = 2464 • L: C = 76^17 mod 3233 = 2190 • L: C = 76^17 mod 3233 = 2190 • O: C = 79^17 mod 3233 = 875 Decryption Once the ciphertext (c) is received, the recipient uses their private key (n, d) to decrypt the message. Computes the plaintext message (m) using the formula – m = c^d mod n. The recipient receives the ciphertext (2103, 2464, 2190, 2190, 875). They use their private key to decrypt each value – • M = 2103^2753 mod 3233 = 72 (H) • M = 2464^2753 mod 3233 = 69 (E) • M = 2190^2753 mod 3233 = 76 (L) • M = 2190^2753 mod 3233 = 76 (L) • M = 875^2753 mod 3233 = 79 (O) So, the decrypted message is "HELLO".			
Unit –III					
6.	a)	Explain in detail about kerboros. Kerberos is a security or authentication protocol for computer network security. Developed in the 1980s by MIT for Project Athena, Kerberos has since found application in many operating systems and networks. Kerberos functions with 3 actors and uses a Key Distribution Centre (KDC) as a third-party authorizer/ intermediary between the client and the internet. This router prevents privatenetwork access and is strengthened further by Kerberos' symmetric-key cryptography. Since most domain controllers (DCs)/network servers on the Active Directory Domain Services (ADDS) require and allow authentication for host access, Kerberos has found extensive use. Kerberos Authentication Before we can understand the authentication process, each of the 3 actors' functions must be understood. **The Kerberos Authentication Process**  • The Client/User - The client-server requests access on the user's behalf, which is then transferred to the Application Server (AS). • Application Server - Hosts the service or resources the user wants access to. • Key Distribution Centre (KDC) - Consists of 3 sub-parts that handle the triple as, i.e., Authentication, Authorization, and Accounting, namely • Authentication Server (AS)	CO3	L3	7M

- Kerberos Databased (KDB) - stores details of tested users, e.g., ID and password.
- Ticket Granting Server (TGS) - the connecting link between a user and the target service.

Kerberos Authentication Workflow
These are broadly the stages by which Kerberos verifies a connection.

Stage 1 - The client-server requests the AS in the KDC to allocate a Ticket Granting Ticket (TGT).

Stage 2 - The AS runs users' credentials, e.g., user ID and network address. Verified users' details are found as values by the AS in the centralized Kerberos Database. Once the values are verified, the AS generates a secret client key using the client's password hash, which is encrypted by a session key (SK1) and the TGT (which has information like the client ID, domain address, timestamp, SK1 and the lifetime of the ticket) are included. The AS also simultaneously generates a secret TGS key.

Stage 3 - The client receives the secret user key, decrypts the TGT and SK1, and sends the authenticators to TGS.

Stage 4 - The KDC uses the TGS secret key to decrypt and get the TGT and SK1. TGS matches the authenticator to user credentials, and the validity of the timestamp is verified. If everything is in order, the KDC creates an encrypted service ticket that contains the next session key (SK2). SK2 is also sent to the network service to validate the user's request.

Stage 5 - When the client decrypts the service ticket, SK2 and authenticator are received. These are then sent to the target server.

Stage 6 - Finally, the target server uses SK2 to verify all client details and the service ticket authenticator. The target server relays a message to the client to confirm mutual authentication, granting a secure connection.

How does the Kerberos Encryption happen?
When creating a new account, you need to give different usernames and passwords. The username and Kerberos version number are combined to create a unique string called "salt" to maximize randomness. The secret key needs to share between the client and the target service, which, when matched, gives access.
Pros and Cons of Kerberos
Below are some strengths and weaknesses of this protocol:
Pros

- **Single Log-in and Access Control** ? Kerberos only needs a single login, after which renewal doesn't need re-iterating information. Single-point login is easier for user access and administrators, e.g., in business, to enforce security policies.
- **Reciprocal/Shared and Reusable Authentication** ? The user and the service authenticate each other, so both parties are assured that the other has been verified.

Plus, once a user has been verified, the authentication is durable for the ticket lifespan, and personal information doesn't have to be given again for this period.

- **Straightforward Transparency** ? Simple logs of the client's name and timestamp are maintained for effortless auditing and transparency.
- **Versatile Security System** ? Kerberos uses several safeguard features in its security arsenal. These include cryptography, encrypted secret keys for each of the three entities, and third-party authentication (Multi-Factor Authentication). Passwords are never stored or sent across the network, and only the password hash is used, which makes impersonation harder.
- **Restricted Ticket Lifetime** ? The ticket issued by TGS only valid for a certain period, after which the administrator nullifies any attempts to access the target server.

Cons

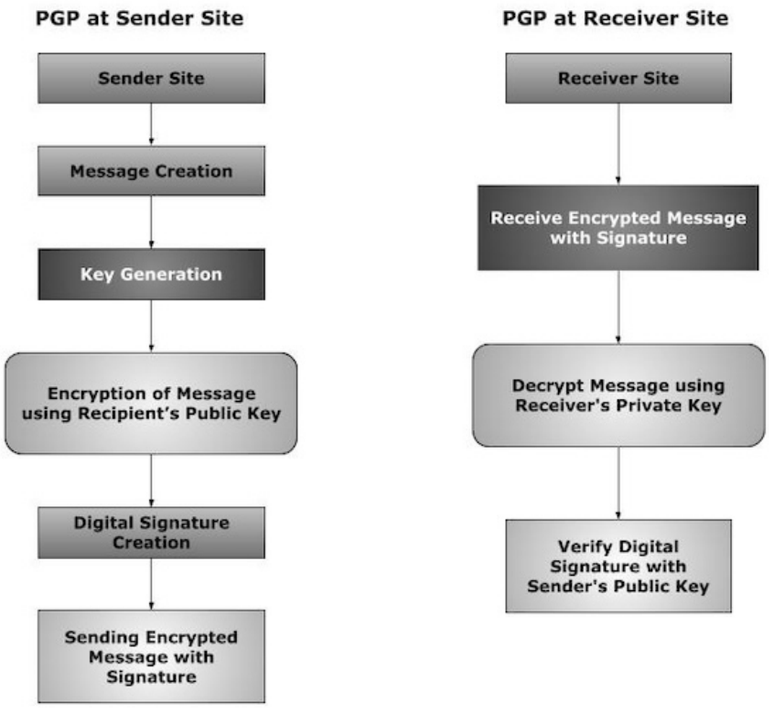
- **Strict Synchronization** ? The client, KDC, and host site must be available at and configured to the same times and dates to prevent the ticket lifespan from expiring. Kerberos does give time leeway of about 5 minutes, after which the process has to be repeated due to relay attacks.

		• **Cluster Hosting** ? Accessing multiple domains together requires simultaneous replication of the servers, or else all the access requests fail. Each client/device and network server need its own hostname and secret key as identifiers. Overlaps or duplications can cause login issues. • **Centralized System** ? Kerberos is a centralized protocol, so single point attack can shut down the entire system, which makes a secondary/backup authentication mechanism necessary • **Node Compliance** ? All devices, network services, and operating systems must be Kerberos -compatible. Both sides must be able to read and resolve NetBIOS and DNS addresses, as these are the most used Kerberos SPNs, and they need access to TCP and UDP 88 ports that handle traffic for Kerberos. **Conclusion** Kerberos isn't going anywhere for the foreseeable future. It is still reliable and continues to effectively shield users from unauthorized attacks. Kerberos has lasted this long despite technology and cyber-crime advances. Many large companies like Microsoft and Apple have Kerberos built into their operating systems. Plus, each operating system can use Kerberos as its foundational authentication protocol and customize its software security.			
	b)	Explain about PGP. An encryption program called Pretty Good Privacy (PGP) offers cryptographic privacy and authentication for data transmission. PGP is used to improve the security of email conversations as well as to sign, encrypt, and decrypt files, directories, messages, emails, even complete disc partitions. In 1991, Phil Zimmermann created PGP. The OpenPGP standard (RFC 4880), an open standard for data encryption and decryption, is adhered to by PGP and related programs. Current versions of PGP can be used with GnuPG and other OpenPGP-compatible programmes. Cybercriminals are able to duplicate communications using a victim's name or identity, making email a prime attack method. By encrypting the data to make email communication more private, PGP is designed to solve this issue and enhance email security. One of the first free public-key cryptography software was PGP. On computer servers used for bulletin board systems, it was initially used to allow individual users to communicate. Later on, other programs like email supported it and helped standardise it. It is now often used to secure both individuals and companies and has grown into a fundamental standard for email security. For data used in online communication, the data encryption program offers cryptographic authentication and privacy. This makes it possible to encrypt and decrypt files, emails, and text messages using PGP. **How PGP Work?** PGP uses a combination of hashing, data compression, and cryptography techniques to function. It is comparable to other widely used encryption techniques like Secure File Transfer Protocol (SFTP), which secures data in transit, Secure Sockets Layer (SSL), which authenticates network users, and Kerberos, which secures websites. Using the public key method, PGP allows users to have a private key that is known only to them and a unique encryption key that is known to the public. When a user uses their public key to transmit a message to someone, it is encrypted; the recipient uses their private key to decrypt it. To encrypt data as it moves across networks, it combines symmetric and asymmetric key technologies with private-key and public-key encryption. **PGP works in some below stages** − • PGP creates a huge, uncrackable, one-time public encryption technique that serves as the random session key. • The message is then encrypted while it is being transferred using the recipient's public key and the session key. The recipient gives that key to anyone from which they want to receive messages. • Once the sender of the message gives their session key, the recipient can use their private key to decrypt the message. While PGP uses a faster algorithm, encrypting a complete message can take some time. PGP improves cryptographic security and reduces disc space and transmission time by compressing plaintext data. The minimised version of the message is encrypted using the public key, exactly like the entire message was. The recipient takes both, decrypts the entire message using their private key to unlock the shorter	CO3	L3	7M

key.

PGP at Sender and Receiver Site

PGP at the sender and receiver's site can help visualize the process. Below is a simple chart diagram showing the steps involved in using PGP encryption at the sender's site −



At Sender's Site

The steps involved in using PGP encryption at the sender's site −

- **Message Creation** − The sender creates a message they intend to send it safely.
- **Key Generation** − A sender makes a pair of encryption keys where one is shared publicly and the other stays private.
- **Encryption of Message** − Encrypting the message text with the receiver's public key ensures that only the receiver would decrypt it through the private key.
- **Digital Signature Creation** − A sender can create a digital signature on the communication using his or her private key. It serves to prove the sender's identity and guarantee that the message has not been altered.
- **Encryption Message Sending with Signatures** − The sender forwards the encrypted message to the recipient and sends along a digital signature. Upon receiving this message, the receiver will use his private key for decrypting while using the sender's public key to verify whether the message is authentic or not.

At Receiver's Site

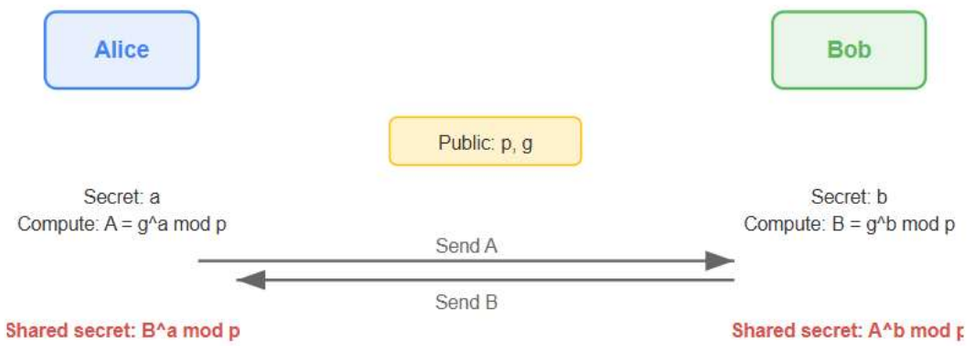
he steps to be followed in using PGP encryption at the receiver's site −

- **Receive Encrypted Message with Signature** − When the sender sends an encrypted message and its digital signature, the receiver gets it.
- **Decrypt Message** − The receiver decrypts the encrypted message using a private key that matches the one they have made public.
- **Digital Signature Verification** − To achieve this, the receiver utilizes the sender's public key to verify the digital signature of the attached message. This way, it can be confirmed that the message was truly sent by the alleged sender without changing anything along the way.

Usage of PGP

PGP encryption is a popular tool to protect messages and data −

- **Confidential communication** − It enciphers emails, files, text messages as well as disk partitions by ensuring that they can only be decrypted and read by the person for whom they were meant.
- **Authentication and Integrity Checking** − Digitally signing with PGP helps in verifying identities of senders and finding out if messages are changed whatsoever.
- **Ensuring Message Delivery** − Public keys come along with identity certificates which contain a specific recipient information and alert against any interference attempts.
- **Email Encryption** − In order to keep your data secure, it is normal that you use PGP to encrypt email messages.

		• **Digital Signature Verification** – By using PGP one is able to verify whom the message was sent from by using their digital signature. It will be in most cases combined with threat detection tools for increased security. • **File Encryption** – PGP's robust RSA encryption makes it suitable for securiing files. **Advantages of PGP** Ther are some advantages of PGP – • The main advantage of PGP encryption is its unbreakable algorithm. It has been acknowledged as the best approach for improving cloud security and has been used by users who want to encrypt their private conversations. This is because PGP prevents hackers, governments, and nation-states from accessing PGP-encrypted files or emails.			
		(OR)			
7.	a)	Explain Diffie-Hellman key exchange algorithm with an example. The **Diffie-Hellman Key Exchange** (also known as exponential key exchange) is a method for securely exchanging cryptographic keys over an insecure channel. It is a fundamental building block of many secure communication protocols, including **SSL**/**TLS** and SSH. This protocol allows two parties to establish a shared secret key without directly transmitting it, making it impossible for eavesdroppers to intercept the actual key. The security relies on the mathematical difficulty of the discrete logarithm problem. How Diffie-Hellman Key Exchange Works The Diffie-Hellman key exchange works by allowing two parties (Alice and Bob) to agree on a shared secret key over an insecure channel. The process involves the following steps: • **Public Parameters** Alice and Bob agree on two large prime numbers, p and g, which are publicly known. • **Private Key Generation** Alice chooses a secret integer a, and Bob chooses a secret integer b. • **Public Key Exchange** Alice computes A = g^a mod p and sends it to Bob. Bob computes B = g^b mod p and sends it to Alice. • **Shared Secret Computation** Alice computes s = B^a mod p, and Bob computes s = A^b mod p. • **Result** Both Alice and Bob now have the same shared secret key s, which they can use for secure communication. **Diffie-Hellman Key Exchange Process**  **Common Applications** • **SSL/TLS Protocols** Used in HTTPS connections to establish secure communication between web browsers and servers. • **SSH (Secure Shell)** Enables secure remote login and command execution over insecure networks. • **VPN Connections** Protocols like IPSec and OpenVPN use Diffie-Hellman to establish secure tunnels. • **Secure File Transfer** SFTP and other secure file transfer protocols rely on this key exchange method.	CO3	L2	7M

Vulnerabilities and Countermeasures

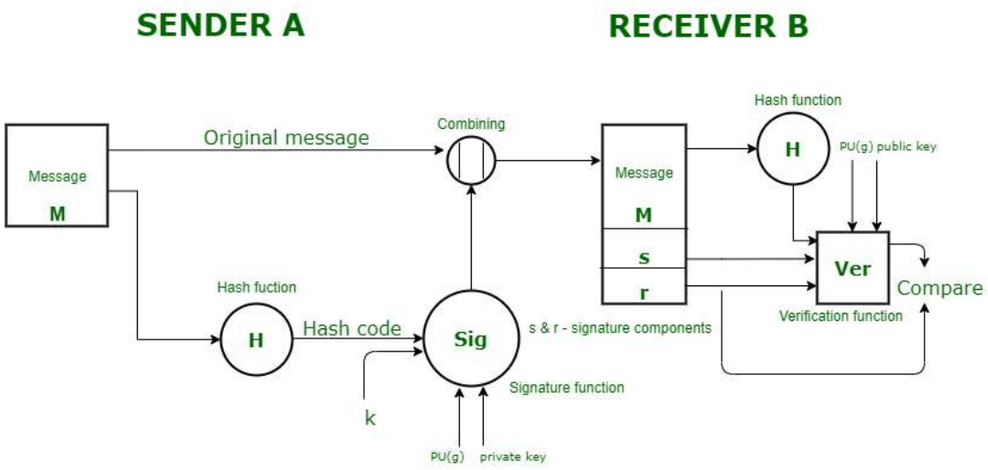
Vulnerability	Description	Countermeasure
Man-in-the-Middle Attack	Attacker intercepts and modifies exchanged messages	Use certificate-based authentication and message authentication codes
Small Subgroup Attack	Weak prime numbers with small subgroups can be exploited	Use large prime numbers with no known small subgroups
Weak Random Numbers	Predictable secret exponents compromise security	Use cryptographically secure random number generators

Conclusion

The Diffie-Hellman Key Exchange is a cornerstone of modern cryptography that enables secure communication over insecure channels. While it has some vulnerabilities, proper implementation with strong parameters and authentication mechanisms makes it highly secure and widely trusted in real-world applications.

b) Explain the Digital Signature Standard (DSS) with its working. CO3 L3 7M

Digital Signature Standard (DSS) is a Federal Information Processing Standard(FIPS) which defines algorithms that are used to generate digital signatures with the help of Secure Hash Algorithm(SHA) for the authentication of electronic documents. DSS only provides us with the digital signature function and not with any encryption or key exchanging strategies.



Sender Side: In DSS Approach, a hash code is generated out of the message and following inputs are given to the signature function -

1. The hash code.
2. The random number 'k' generated for that particular signature.
3. The private key of the sender i.e., PR(a).
4. A global public key(which is a set of parameters for the communicating principles) i.e., PU(g).

These input to the function will provide us with the output signature containing two components - 's' and 'r'. Therefore, the original message concatenated with the signature is sent to the receiver.

Receiver Side : At the receiver end, verification of the sender is done. The hash code of the sent message is generated. There is a verification function which takes the following inputs -

1. The hash code generated by the receiver.
2. Signature components 's' and 'r'.
3. Public key of the sender.
4. Global public key.

The output of the verification function is compared with the signature component 'r'. Both the values will match if the sent signature is valid because only the sender with the help of it private key can generate a valid signature.

		Benefits of advanced signature: 1. A computerized signature gives better security in the exchange. Any unapproved individual can't do fakeness in exchanges. 2. You can undoubtedly follow the situation with the archives on which the computerized mark is applied. 3. High velocity up record conveyance. 4. It is 100 percent lawful it is given by the public authority approved ensuring authority. 5. In the event that you have marked a report carefully, you can't deny it. 6. In this mark, When a record is get marked, date and time are consequently stepped on it. 7. It is preposterous to expect to duplicate or change the report marked carefully. 8. ID of the individual that signs. 9. Disposal of the chance of committing misrepresentation by a sham. **Burdens of computerized signature:** 1. You really want to investigate all the similarity issues. In there are a great deal of similarity settings like a refreshed rendition of driver and programming. 2. Programming is one of the main pressing concerns while utilizing a computerized signature testament. 3. On the off chance that you are having a place with the corporate world and running a commodity import association, you want to create a computerized signature for E-tagging. 4. In this signature, Lost or burglary of keys and the utilization of weak storage spaces. 5. There is a more grounded need for a norm through which these various strategies can cooperate. In this period of quick mechanical headway, a considerable lot of these tech items have a short timeframe of realistic usability. 6. To successfully utilize a computerized signature, the two shippers and beneficiaries might need to purchase computerized endorsements. 7. To work with computerized endorsements, the shipper and beneficiaries need to purchase check programming at an expense. 8. A computerized signature includes the essential road for any business is cash.			
		Unit –IV			
8.		Explain in detail about two security protocols of IPSec. A set of protocols called IPsec is used to secure connections between devices. Secure data transmission over public networks is made possible by IPsec. It functions by authenticating the source from which IP packets arrive and encrypting them. It is mainly used to set up VPNs. "IP" stands for "Internet Protocol," and "sec" for "secure" in the name "IPsec." The primary Internet routing protocol, or Internet Protocol, uses IP addresses to identify the destination of data. Because IPsec improves this process with authentication and encryption. The Internet Engineering Task Force developed IPSec in the 1990s to ensure data confidentiality, authenticity, and integrity when connecting to open networks. For example, to access company files remotely users link to the internet using an IPSec virtual private network (VPN). With the help of IPSec protocol encryption we can avoid unauthorised monitoring, sensitive data. The server can also check that the data packets it receives are authorised. How it works? The steps below are used by computers and the IPSec protocol to exchange data − • By comparing the data transmission against its security policy, the sender device examines whether IPSec protection is necessary. In that case, the recipient device and the computer starting the secure IPSec transfer are connected. • The requirements needed to create a secure connection are determined by both machines. The encryption, authentication, and other security association (SA) parameters should be agreed upon by both parties. • Encrypted data is sent and received by the computer, verifying that the data came from reliable sources. To make sure the underlying content is trustworthy, it performs tests. • The computer terminates the IPSec connection after the transmission is	CO4	L2	14 M

finished or the session has expired.

IPSec encryption

Software that scrambles data to protect its content from unauthorised persons is called IPSec encryption. A decryption key is needed to decrypt data that has been encrypted using an encryption key. Many encryption methods, like as AES, Blowfish, Triple DES, ChaCha, and DES-CBC, are supported by IPSec.

Asymmetric and symmetric encryption are used by IPSec for both security and speed during data transit. Asymmetric encryption maintains the privacy of the decryption key while making the encryption key public. The same public key is used by symmetric encryption to both encrypt and decrypt data. To speed up data transfer, IPSec uses asymmetric encryption to create a secure connection before switching to symmetric encryption.

IPSec Protocols

IPSec protocols securely transmit data packets. A data packet is a predefined structure that formats and prepares information for network transmission. It consists of three parts: the header, the payload, and the trailer.

- A header is a portion that starts the data packet and contains instructions for routing it to the correct destination.
- Payload refers to the actual information included within a data packet.
- The trailer is extra data added to the tail of the payload to signify the end of the data packet.

So below are some IPSec protocols listed −

- **AH (Authentication Header)** − The Authentication Header (AH) protocol assures that data packets come from an authentic source and have not been tampered with, similar to a tamper-proof seal on a consumer product. These headers do not enable encryption or help to conceal data from attackers.
- **ESP (Encapsulating Security Protocol)** − The Encapsulating Security Protocol (ESP) encrypts both the IP header and the payload for each packet, unless transport mode is enabled, in which case just the payload is encrypted. Each data packet is accompanied by an ESP-specific header and trailer.
- **SA (Security Association)** − SA refers to a set of protocols for negotiating encryption keys and algorithms. Internet Key Exchange (IKE) is one of the most often used SA protocols.
- **IKE (Internet Key Exchange)** − The Internet key exchange (IKE) protocol provides a secure connection between two internet-connected devices. Both devices create a security association (SA), which requires settling up encryption keys and algorithms before transmitting and receiving subsequent data packets.

Finally, while the Internet Protocol (IP) is not a component of the IPsec suite, IPsec operates directly on top of IP.

IPSec Algorithms and Protocols

A group of security protocols and cryptography-based services known as IPSec are designed to secure communication between devices sending data over an untrusted network. It is possible to setup an IPSec VPN between your Firebox and many other equipment or cloud-based endpoints that support these common protocols, as IPSec is based on a set of generally accepted protocols and algorithms.

Encryption Algorithms

- AES (Advanced Encryption Standard)
- 3DES (Triple-DES)
- DES (Data Encryption Standard)

Authentication Algorithms

- HMAC-MD5 (Hash Message Authentication Code - Message Digest Algorithm 5)
- HMAC-SHA1 (Hash Message Authentication Code - Secure Hash Algorithm 1)
- HMAC-SHA2 (Hash Message Authentication Code - Secure Hash Algorithm 2):
    - SHA2-256: produces a 265-bit (32 byte) message digest
    - SHA2-384: produces a 384-bit (48 byte) message digest
    - SHA2-512: produces a 512-bit (64 byte) message digest

Usage of IPSec

IPsec is useful for the following tasks −

- Sending data over the open internet needs router security.
- Application data should be encrypted.
- In the case that the data comes from a recognised sender, authenticate it immediately.
- By creating encrypted circuits, or IPsec tunnels, that encrypt all data transferred between two endpoints, network data can be secured.

IPSec is used by organisations as a security against replay attacks. The act of intercepting and altering a live transmission by sending data to a computer working as a mediator is known as a replay attack or man-in-the-middle attack. Each data packet is given a sequential number by the IPSec protocol, which also checks for duplicate packets.

Modes of IPSec

IPSec operates in two modes having different levels of protection. So see the modes below –

- **Tunnel** – The IPSec tunnel mode works well for sending data over public networks because it improves data security against unauthorised parties. The computer encrypts all data, both the payload and header, and adds a new header to it.
- **Transport** – IPSec transport mode encrypts only the data packet's payload while leaving the IP header as it is. The unencrypted packet header helps routers to figure out the destination address of each data packet. As a result, IPSec transport is used in a closed and trusted network, like to protect a direct link between two computers.

Benifits of IPSec

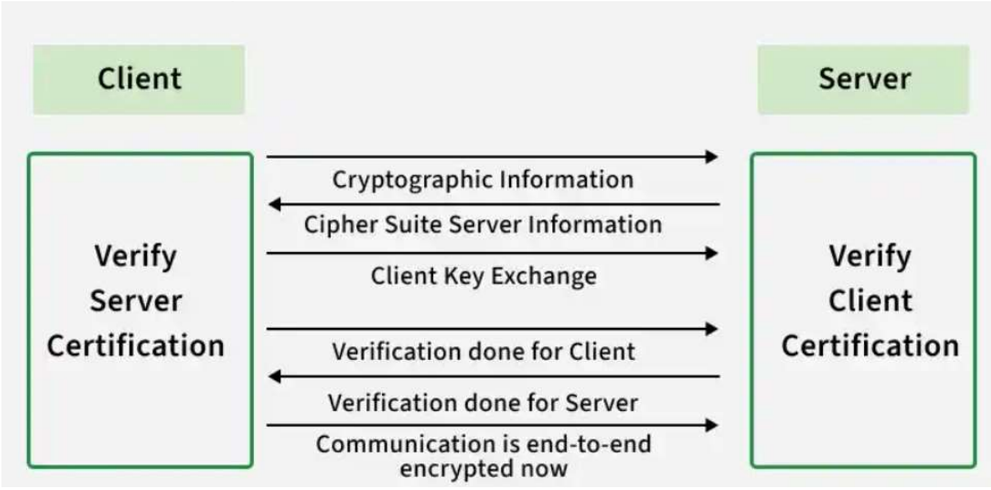
- IPSec is an open standard protocol that is widely supported by vendors and may be utilised in a variety of contexts.
- It offers powerful cryptographic security features to secure sensitive data while also ensuring network privacy and integrity.
- IPSec can be used to secure huge networks and can be scaled up or down as required.
- It can be configured to secure a variety of network topologies, like point-to-point, site-to-site, and remote access links.
- IPSec can helo reduce network congestion and increase network efficiency.

Drawbacks of IPSec

- A drawback of IPSec is its advanced configuration, which demands professional knowledge and abilities.
- IPSec can have an influence on network performance because of the overhead associated with IP packet encryption and decryption.
- IPSec only protects IP traffic, leaving other protocols like ICMP, DNS, and routing open to attacks.
- IPSec relies on proper key management to guarantee the security of the cryptographic keys that are used for encryption and authentication.
- IPSec may not be compatible with some network devices and software, resulting in problems with interoperability.

(OR)

9		Explain the working of SSL handshake protocol. The SSL and TLS handshake establishes a system for SSL/TLS clients and servers to start communication between them. • This protocol is used to establish sessions. • It allows the client and server to verify each other by transferring a series of messages. • It uses four phases to finalize its circle.	CO4	L2	14 M



SSL/TLS Handshake

- **Phase-1:** Deciding which version of the Protocol to use. The system decides which protocol to use. Client and Server exchange hello-packets with each other to confirm. In this IP session, cipher suite, and Agree on which version of the protocol to use.

- **Phase-2:** Server sends his certificate and Server-key-exchange. The server end phase-2 by exchanging the hello packet.
- **Phase-3:** Verification, in this phase, the Client replies to the server by sending his certificate and Client-exchange-key.
- **Phase-4:** In this phase, the Change Cipher suite is passed and all the verifications and security checks are done after this Handshake Protocol ends.

Importance of TLS/SSL Handshake

- The server proves its identity to the client using a digital certificate issued by a trusted Certificate Authority (CA).
- It allows the client and server to securely exchange cryptographic information without exposing it to attackers.
- Both sides agree on the encryption algorithms and security parameters that will be used during communication.
- A shared session key is generated, which is later used to encrypt and decrypt the data exchanged during the session.

Components of TLS Handshake

The TLS handshake involves several key components that work together to establish a secure and encrypted communication between the client and the server.

- **Client:** The client is the user's device that initiates the connection and sends a request to the server.
- **Server:** The server is a remote system that hosts the website or service and responds to the client's request during the handshake process.
- **Digital Certificate:** Issued by a trusted Certificate Authority (CA), it is used by the server to prove its identity to the client and establish trust.
- **Cipher Suite:** A set of cryptographic algorithms that define how encryption, key exchange, and message authentication will be performed during communication.
- **Session Key:** A symmetric key generated during the handshake, used to encrypt and decrypt data exchanged between the client and server.

Steps in the TLS/SSL Handshake

A secure connection is established by the client and server during the TLS handshake. Here are the main steps :

1. ClientHello: This is where the client begins its handshake with a "hello" message sent to a server, which includes supported encryption methods as well as other parameters.

2. ServerHello: In response, the server also sends its own hello message, selecting an encryption method before sending over its digital certificate for authentication.

3. Authentication: To prove its identity, the server presents a digital certificate. Then it is up to the client to verify if such a certificate is from a trusted source.

4. Key Exchange: In this step, both client and server agree on a shared symmetric key that will be used for client data encryption/decryption during their session. It may involve Diffie-Hellman key exchange or other ways.

5. Session Key Generation: Both sides can use these exchanged keys to generate session keys that will be used for encryption/decryption of data while they are in session.

6. Secure Communication Begins: Once the handshake is completed, the client and server start exchanging encrypted data.

SSL (Secure Sockets Layer)	TLS (Transport Layer Security)
SSL is the older cryptographic protocol used to secure communication over the Internet.	TLS is the successor to SSL and provides improved security.
Developed by Netscape in the 1990s.	Developed by the Internet Engineering Task Force (IETF).
SSL versions include SSL 2.0 and SSL 3.0, which are now deprecated.	TLS versions include TLS 1.0, TLS 1.1, TLS 1.2, and TLS 1.3.
Provides weaker encryption and security mechanisms compared to TLS.	Provides stronger encryption and enhanced security features.
SSL is no longer considered secure and is not used in modern systems.	TLS is widely used today for securing web communication such as HTTPS.

Cipher Suite:

A Cipher suite is a set of encryption rules that decides how the TLS handshake works. TLS/SSL protocols use some algorithms from a cipher suite to generate keys and

encrypt information so that the communication is end-to-end encrypted. A cipher Suite specifies one algorithm for each of the following tasks

- **Key Exchange Algorithms:** This algorithm protects the information required to create shared keys.
- **Bulk Encryption Algorithms:** Bulk encryption algorithms are those algorithms that encrypt the messages exchanged between clients and servers.
- **Message Authentication Algorithms:** Message authentication algorithms generate messages and signatures that ensure the Combination of a message.

Key Exchange Mechanism

The key exchange mechanism in TLS establishes a shared secret between the client and server for secure communication.

1. RSA (Rivest–Shamir–Adleman)

It is a key exchange method where the client encrypts a premaster secret using the server's public key.

- The server decrypts the premaster secret using its private key
- Does not provide forward secrecy
- Simpler but less secure compared to modern methods

2. Diffie-Hellman (DH)

This is a method that allows the client and server to generate a shared secret without directly transmitting it.

- Uses mathematical computation to derive a common key
- Can use static keys, which reduces security
- More secure than RSA but less efficient than ECDHE

3. Elliptic Curve Diffie-Hellman (ECDHE)

It is an advanced key exchange method that uses elliptic curve cryptography to generate a shared secret.

- Provides forward secrecy using temporary (ephemeral) keys
- More efficient with smaller key sizes
- Widely used in modern TLS versions (e.g., TLS 1.3)

Applications of TLS Handshake

- **HTTPS (Secure Web Browsing):** Used in web browsers to secure communication between users and websites, ensuring data confidentiality and integrity.
- **Secure APIs:** Protects data exchange between applications and services by encrypting API requests and responses.
- **Email Protocols:** Secures email communication in protocols such as SMTPS, IMAPS, and POP3S to protect sensitive information.

TLS Handshake Failures

These failures occur when a secure connection cannot be established between the client and server due to configuration or compatibility issues.

- **Invalid or Expired Certificate:** The server's certificate is not trusted, expired, or incorrectly configured, leading to authentication failure.
- **Protocol Version Mismatch:** The client and server do not support a common TLS version, preventing successful negotiation.
- **Cipher Suite Mismatch:** No compatible encryption algorithms are found between client and server during the hand shake.

Prepared by **HOD-IT**

P.Surekha(Information Technology)

1) Signature of Faculty (Data Science)

2) Signature of Faculty (Cyber Security)

3) Signature of Faculty (CSE)