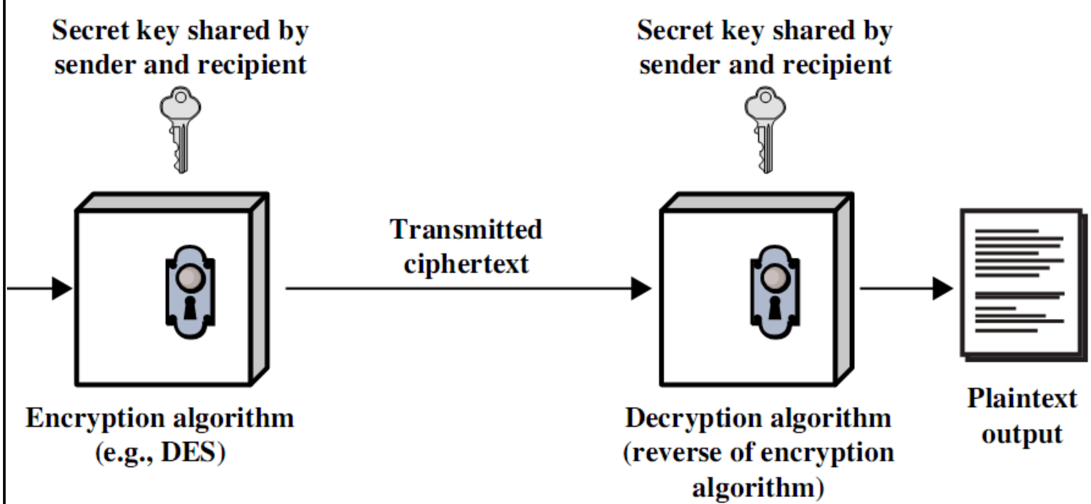
USN | | | | | | | | | |

Internal Assessment Test 1 solution– September 2025

Sub:	Cryptography and Network Security				Sub Code:	BCS703	Branch:	CSE
Date:	29.09.2025	Duration:	90 mins	Max Marks: 50	Sem / Sec:	VII (A, B & C)		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	Explain the Symmetric Cryptosystem with a neat diagram.	10	CO1	L2



A symmetric encryption scheme has five ingredients:

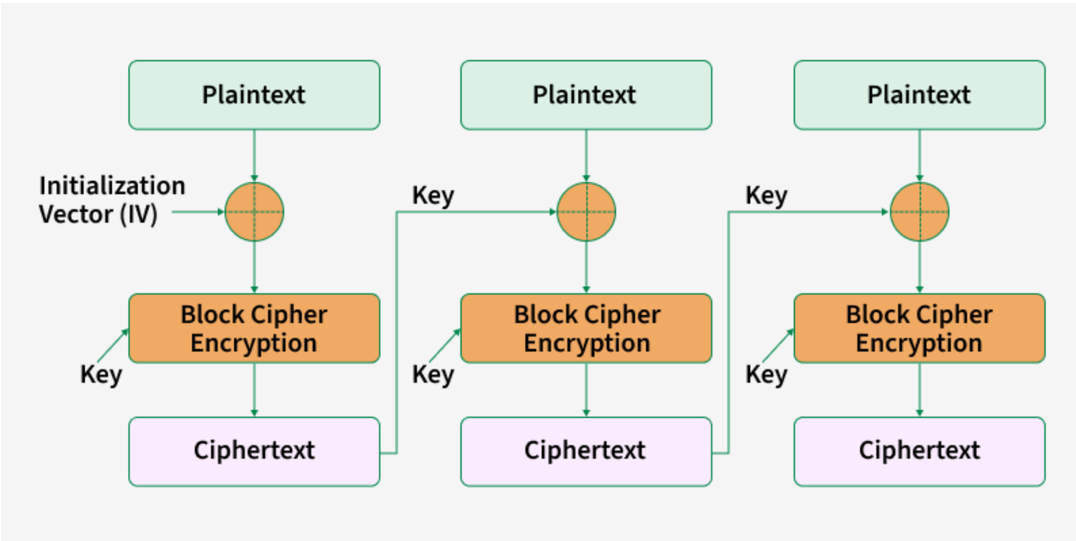
- **Plaintext:** This is the original intelligible message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key:** The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time.
- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key.
- **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

Define Block Cipher and Stream Cipher

What is Block Cipher?

A block cipher encrypts data in fixed-size blocks usually 64 or 128 bits at a time. The encryption algorithm processes each block of data separately using the cryptographic key to transform the plaintext into the ciphertext. Block ciphers function on complex mathematical computation and permutation to ensure that the data encrypted is safe. The choice of block size does not directly affect the strength of the encryption scheme.

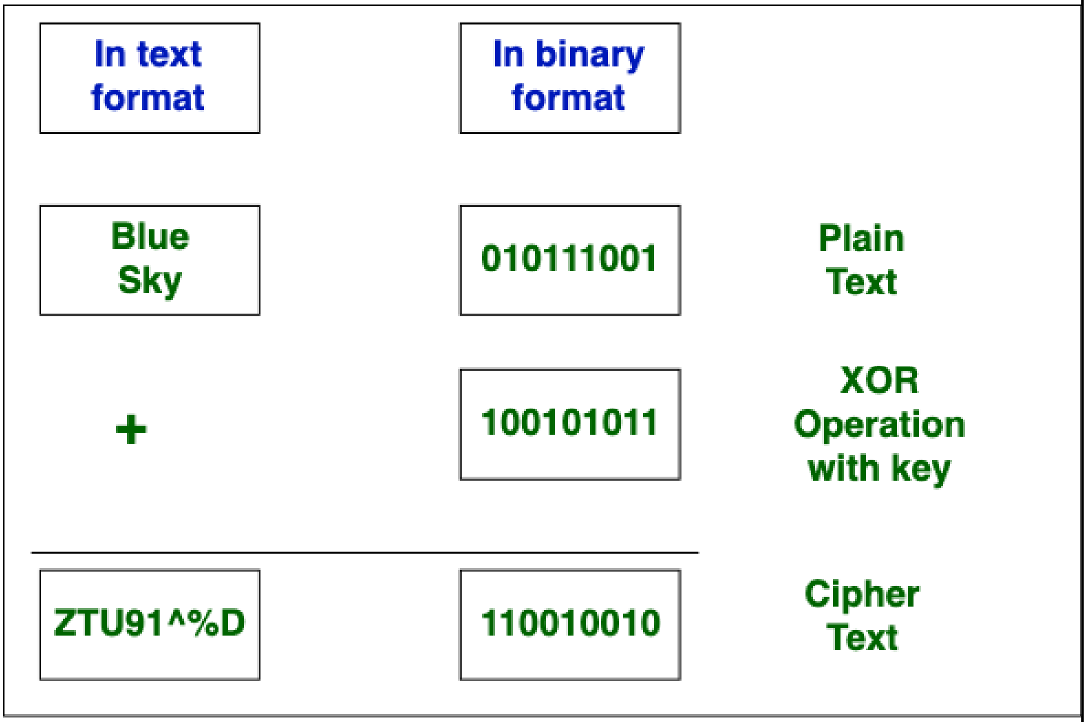
The strength of the cipher depends upon the key length. However, any size of the block is acceptable. The following aspects can be kept in mind while selecting the size of a block: Avoid very small block sizes, Do not have very large block sizes, and in multiples of 8-bit.



What is Stream Cipher?

A stream cipher encrypts data one bit or one byte at a time rather than in fixed-size blocks in block cipher. It generates a keystream that is combined with the plaintext to the produce ciphertext. Stream ciphers are made for the scenarios where data needs to be encrypted in the continuous stream making them suitable for the real-time applications.

It can be categorized into the synchronous, self-synchronizing and one-time pad types. The Synchronous encryption requires independently generated keystream from both the plaintext and the ciphertext. They have to be in the same state, with the same key, in order to decode the data properly.



Stream Cipher

2	Differentiate between Confusion and diffusion.	10	CO1	L3

No	Confusion	Diffusion
1	Cipher text cannot give the clue about plain text.	Increase the redundancy of plain text and generate cipher text.
2	Confusion technique is used in both block and stream cipher.	Diffusion technique is only used in block cipher.
3	Confusion hides the relation between the ciphertext and key.	Diffusion hides the relation between the ciphertext and the plaintext.
4	This technique is possible through substitution algorithm.	While it is possible through transportation algorithm.
5	If a single bit in the key is changed, all the bits in the ciphertext will also have to be changed.	In case a symbol in the plaintext is changed, several or all symbols in the cipher text will also have to be changed.

Explain Fiestel's encryption and decryption algorithm with a neat diagram

Feistel Cipher Structure

- The inputs to the encryption algorithm are a plaintext block of length 2w bits and akey K. The plaintext block is divided into two halves, L_0 and R_0.
- The two halves of the data pass through n rounds of processing and then combine toproduce the ciphertext block.
- Each round i has as inputs L_{i-1} and R_{i-1}, derived from the previous round, as well as asubkey K_i, derived from the overall K.
- In general, the subkeys K_i are different from K and from each other.

A **substitution** is performed on the left half of the data. This is done by applying a*round function* F to the right half of the data and then taking the exclusive-OR of the output of that function and the left half of the data. Following this substitution, a **permutation** is performed that consists of the interchange of the two halves of the data.

The exact realization of a Feistel network depends on the choice of the following parameters and design features:

Block size: Larger block sizes mean greater security, but reduced encryption/decryption speed for a given algorithm.

Key size: Larger key size means greater security but may decrease encryption/decryptionspeed. The greater security is achieved by greater resistance to brute-force attacks and greater confusion.

Number of rounds: The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.

Subkey generation algorithm: Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.

Round function: Again, greater complexity generally means greater resistance to cryptanalysis.

There are two other considerations in the design of a Feistel cipher: •

- **Fast software encryption/decryption:** The speed of execution of the algorithmbecomes a concern.
- **Ease of analysis:** if the algorithm can be concisely and clearly explained, it iseasier to analyze that algorithm for cryptanalytic vulnerabilities

Feistel Decryption Algorithm

The process of decryption with a Feistel cipher is essentially the same as the encryption process. The rule is as follows: Use the ciphertext as input to the algorithm,but use the subkeys K_i in reverse order. That is, use K_n in the first round, K_{n-1} in the second round, and so on until K_1 is used in the last round.

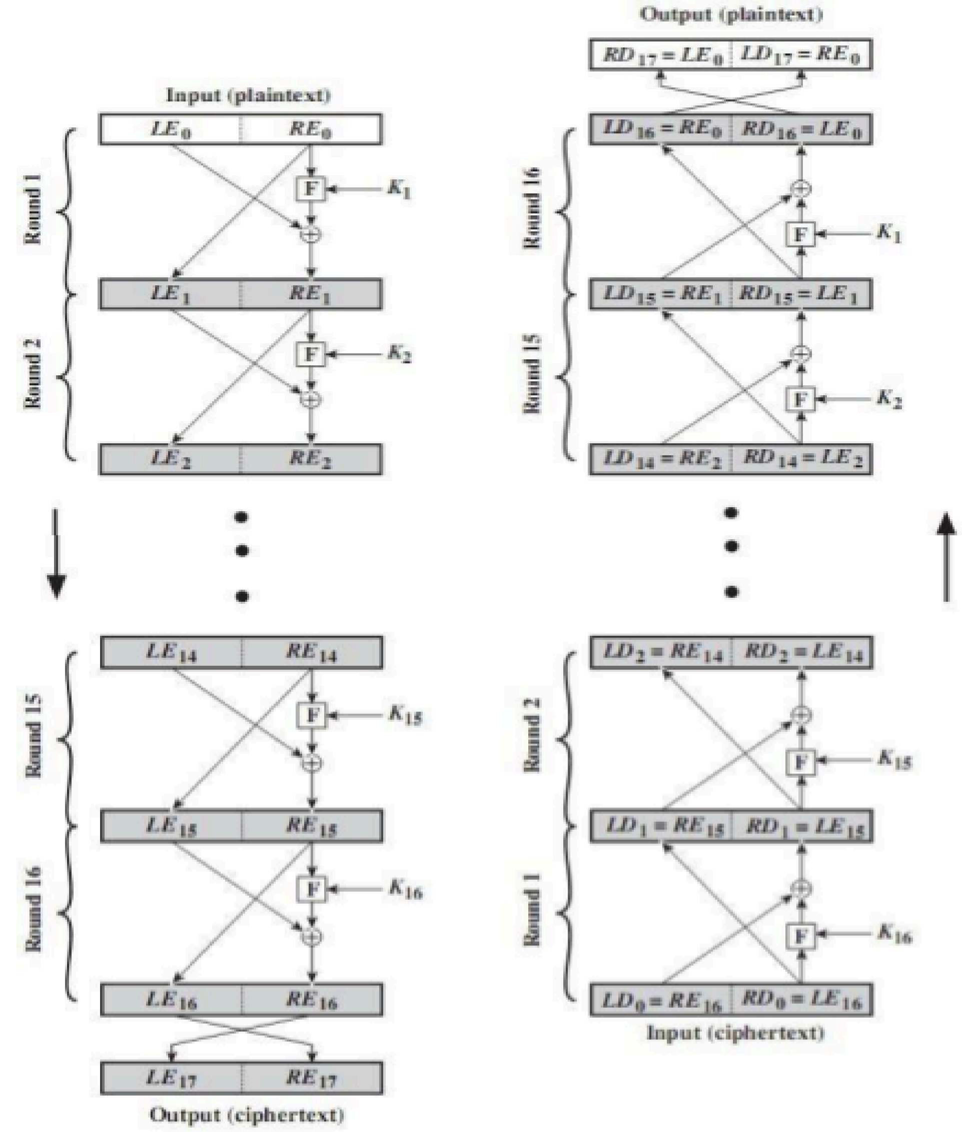
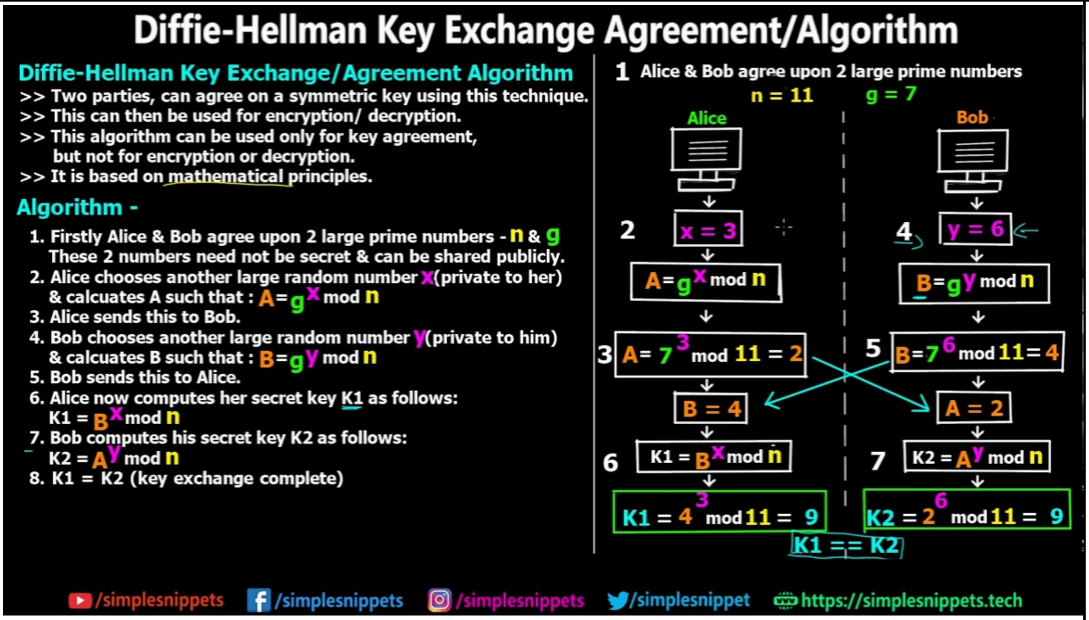
<table>
<tr><td></td><td>

Figure 3.3 Feistel Encryption and Decryption (16 rounds)</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>Outline the Diffie-Hellman Key Exchange algorithm.
Diffie Hellman Key Exchange

Diffie Hellman Key Exchange

The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel (over Internet).This key can then be used to encrypt subsequent communications using a symmetric key cipher.
Working of Algorithm:
Consider two parties, ‘A’ and ‘B’ that need to agree upon a single shared key for the duration of their current session. Both ‘A’ and ‘B’ will be knowing about a common modulus ‘p’ and the generator ‘g’ of the selected modulus.
In order to exchange the shared secret, both will participate in the following sequence of steps</td><td>10</td><td>CO1</td><td>L2</td></tr>
</table>



Consider a Diffie-Hellman Key exchange with a common prime q = 11 and the primitive root α = 2.

(i) If the user A has a public key $Y_A = 9$. What is A's Private key X_A.

(ii)If the user B has a public key $Y_B = 3$. What is the secret key K shared with A

We are given:

- Prime modulus (q = 11)
- Primitive root (\alpha = 2)
- User A's public key (Y_A = 9)
- User B's public key (Y_B = 3)

The public key in Diffie-Hellman is calculated as:

Y=αX mod q

Where:

- (Y) is the public key,
- (X) is the private key,
- (\alpha) is the primitive root,
- (q) is the prime modulus.

(i) Find A's private key (X_A)

We are given:
We try values of (X_A) from 1 upwards:

YA=2XAmod11=9

[
\begin{align*}
2^1 & = 2 \mod 11 = 2 \
2^2 & = 4 \mod 11 = 4 \
2^3 & = 8 \mod 11 = 8 \
2^4 & = 16 \mod 11 = 5 \
2^5 & = 32 \mod 11 = 10 \
2^6 & = 64 \mod 11 = 9 \quad \Rightarrow \text{Match!}
\end{align*}
]

✅ So, **A's private key (X_A = 6)**

(ii) Find the shared secret key (K)

The shared key is computed as:

[
K = Y_B^{X_A} \mod q
]

We are given:

- (Y_B = 3)
- (X_A = 6)
- (q = 11)

So:

[
K = 3^6 \mod 11
]

Calculate (3^6) first:

[
3^6 = 729
]

Now take (729 \mod 11):

	[729 \div 11 = 66 \text{ remainder } 3] So, **Shared secret key (K = 3)** --- Final Answers: **(i)** A's private key: 6 **(ii)** Shared secret key KEY: 3			
4	Explain Hill Cipher algorithm. Apply the same to perform encryption and decryption for plaintext=”paymoremoney” using key $k = \begin{matrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{matrix}$ **1. <u>Hill Cipher</u>** This encryption algorithm takes *m* successive plaintext letters and substitutes for them *m* ciphertext letters. The substitution is determined by *m* linear equations in which each character is assigned a numerical value (a = 0, b = 1, c, z = 25). For *m* = 3, the system can be described as $c1 = (k11p1 + 21p2 + k31p3) \bmod 26$ $c2 = (k12p1 + k22p2 + k32p3) \bmod 26$ $c3 = (k13p1 + k23p2 + k33p3) \bmod 26$ This can be expressed in terms of row vectors and matrices $$(c_1\ c_2\ c_3) = (p_1\ p_2\ p_3)\begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \bmod 26$$ or $$\mathbf{C} = \mathbf{PK} \bmod 26$$	10	CO2	L3

where C and P are row vectors of length 3 representing the plaintext and ciphertext, and K is a 3 * 3 matrix representing the encryption key. Operations are performed mod 26.

consider the plaintext “paymoremoney” and use the encryption key

$$\mathbf{K} = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

The first three letters of the plaintext “pay” are represented by the vector (15 0 24). C = PK mod 26

Hill Algorithm

$$C = E(K, P) = P \times K \mod 26$$

$$P = D(C, K) = CK^{-1} \mod 26$$

$$= P \times K \times K^{-1} \mod 26$$

Encryption

$$\begin{pmatrix} C_1 & C_2 & C_3 \end{pmatrix} = \begin{pmatrix} P_1 & P_2 & P_3 \end{pmatrix} \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix} \bmod 2$$

$$C_1 = (P_1 K_{11} + P_2 K_{21} + P_3 K_{31}) \bmod 26$$

$$C_2 = (P_1 K_{12} + P_2 K_{22} + P_3 K_{32}) \bmod 26$$

$$C_3 = (P_1 K_{13} + P_2 K_{23} + P_3 K_{33}) \bmod 26$$

(1) Encrypt "Pay more money" using Hill cipher with Key

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \Rightarrow 3 \times 3 \text{ matrix}$$

Pay more money

P	a	y	m	o	r	e	m	o	n	e	y
15	0	24	12	14	17	4	12	14	13	4	24

Key = 3×3

$\boxed{PT = \underline{Pay}\ \ \underline{mor}\ \ \underline{emo}\ \ \underline{ney}}$

Pay:

$$(C_1\ C_2\ C_3) = (P_1\ P_2\ P_3)\begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix}$$

$$= (15\ \ 0\ \ 24)\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

$$C_1 = 15 \times 17 + 0 \times 21 + 24 \times 2$$

$$C_2 = 15 \times 17 + 0 \times 18 + 24 \times 2$$

$$C_3 = 15 \times 5 + 0 \times 21 + 24 \times 19$$

$$= (303\ \ 303\ \ 531) \bmod 26$$

$$= (17\ \ 17\ \ 11)$$

$$= (R\ \ R\ \ L)$$

$$\boxed{Pay = RRL}$$

Decryption

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

Key inverse matrix

* K^{-1} matrix is required

$$K^{-1} = \frac{1}{\text{Det } K} \times \text{Adj } K$$

Det K :
$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

$$\text{Det}\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \bmod 26$$

focus on the first row & column

$= 17(18 \times 19 - 2 \times 21)$

$= 17(21 \times 19 - 2 \times 21)$

$= 5(21 \times 2 - 2 \times 18)$

	mor: $(C_1 C_2 C_3) = (12,14,17)\begin{pmatrix}17 & 17 & 5\\ 21 & 18 & 21\\ 2 & 2 & 19\end{pmatrix} \bmod 26$ $\begin{matrix} C_1 = \\ C_2 = \\ C_3 = \end{matrix}\begin{pmatrix}12\times17+14\times21+17\times2\\ 12\times17+14\times18+17\times2\\ 17\times5+17\times21+17\times19\end{pmatrix} \bmod 26$ $= (532\ \ 490\ \ 677)$ $= (12\ \ 22\ \ 1)$ mor $= (M\ W\ B)$ $\boxed{\text{mor} = \text{MWB}}$			
5	The RSA algorithm is a public-key cryptosystem used for secure data transmission. It relies on the difficulty of factoring large numbers into their prime factors. The process involves key generation, encryption, and decryption. Key Generation: • Choose two distinct large prime numbers, p and q. • Calculate `n = p * q`. n is part of both the public and private keys. • • Calculate Euler's totient function `φ(n) = (p - 1) * (q - 1)`. This represents the number of positive integers less than n that are relatively prime to n. • Choose an integer e (public exponent) such that `1 < e < φ(n)` and `gcd(e, φ(n)) = 1`. e is part of the public key. • Calculate d (private exponent) such that `(d * e) mod φ(n) = 1`. d is part of the private key. This d is the modular multiplicative inverse of e modulo `φ(n)`. • **Public Key:** `(e, n)` • **Private Key:** `(d, n)`	10	CO2	L3

<table>
<tr><td></td><td>Encryption:

To encrypt a message M using the public key (e, n):

$C = M^e \bmod n$, where C is the ciphertext.

Decryption:

To decrypt a ciphertext C using the private key (d, n):

$M = C^d \bmod n$, where M is the original plaintext.

RSA Example with $p=17, q=11, e=7, M=88$:

• Key Generation:
  • $p = 17$, $q = 11$
  • $n = p * q = 17 * 11 = 187$
  • $\varphi(n) = (p - 1) * (q - 1) = (17 - 1) * (11 - 1) = 16 * 10 = 160$
  • $e = 7$ (given, and $\gcd(7, 160) = 1$)
  • Calculate d: We need $(7 * d) \bmod 160 = 1$.
  •

Using the extended Euclidean algorithm or by trial and error, we find $d = 23$ because $7 * 23 = 161$, and $161 \bmod 160 = 1$.

• Public Key: $(7, 187)$
• Private Key: $(23, 187)$
• Encryption:Calculating $88^7 \bmod 187$:
• Decryption:Calculating $11^{23} \bmod 187$:</td><td></td><td></td><td></td></tr>
<tr><td>6</td><td>Explain in detail about Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is a public-key encryption method that uses the mathematics of elliptic curves to secure data, offering equivalent security to RSA but with significantly smaller key sizes. This makes ECC more efficient, especially for resource-constrained devices like mobile phones and IoT devices, by requiring less computing power, memory, and bandwidth. Key applications include secure web browsing (SSL/TLS), digital signatures, and blockchain technologies like Bitcoin.

How it works

• ECC is a public-key cryptosystem, meaning it uses a pair of keys: a public key for encrypting messages and a private key for decrypting them.</td><td>10</td><td>CO2</td><td>L2</td></tr>
</table>

	● It is based on the algebraic structure of elliptic curves, which are defined by a specific mathematical equation, such as ● y2=x3+ax+by squared equals x cubed plus a x plus b ● *y*2=*x*3+*ax*+*b* . ● ECC relies on the difficulty of the elliptic curve discrete logarithm problem. For a given starting point on the curve and a private key (a number, n), it is easy to find the resulting point (public key) by repeatedly "adding" the point to itself on the			

curve.

- However, it is computationally infeasible to determine the private key (n) if you only know the public key, making it a secure one-way function for encryption.

Advantages over RSA

- **Smaller key sizes:** For the same level of security, ECC requires much smaller keys. For example, a 256-bit ECC key is comparable in security to a 3072-bit RSA key.

- **Higher efficiency:** Smaller keys lead to faster computations and lower resource usage, which is critical for mobile and IoT devices with limited battery life and processing power.

- **Better scalability:** ECC scales more efficiently as computing power increases, requiring smaller key size increases to maintain security over time compared to RSA.

Applications

- **Secure web communication:** Widely used in protocols like [Transport Layer Security (TLS)] to establish secure connections for websites.

- **Digital signatures:** Provides a way to verify the authenticity and integrity of digital documents and transactions.

- **Blockchain and cryptocurrency:** Used in systems like Bitcoin to create secure and efficient public/private key pairs for transactions.

- **Secure messaging:** Used in various messaging apps to encrypt conversations.