USN | | | | | | | | | | |

Internal Assessment Answer Script – Sep 2025

Sub:	**Data Security and Privacy**					Sub Code:	**BAD703**	Branch :	AIML/CSE AIML
Date:	**29/09/25**	Duration:	90 min	Max Marks:	50	Sem/Sec :	VII /A, B		OBE

	Answer any FIVE FULL Questions	**M**	**CO**	**RBT**
1	**Describe the methods of secure key storage and their challenges (5M+ 5M)** **Solution:** **Key storage** is a critical aspect of cryptography. Even strong algorithms become useless if secret keys are exposed. Secure methods of key storage are therefore essential. Good key management includes secure storage, backup, recovery, using hardware, when possible, etc. **Methods of Secure Key Storage** 1. **Avoiding Key Storage** o Keys are generated whenever needed and not stored permanently. o *Example:* Session keys in secure communication. 2. **Software Storage** o Keys are stored in files, databases, or applications, usually encrypted with passphrases. o Relies on operating system protections and access control. 3. **Hardware Storage** o Keys are stored inside secure hardware such as: ▪ Hardware Security Modules (HSMs) ▪ Trusted Platform Modules (TPMs) ▪ Smartcards / USB tokens ▪ Secure enclaves in mobile devices o Keys never leave the secure hardware boundary; operations are done inside the device Challenges in Key Storage 1. **Avoiding Key Storage** o Not practical for long-term keys. o Requires high-quality random number generation. 2. **Software Storage** o Vulnerable to malware and insider attacks. o Weak passphrases or insecure backups can lead to leakage. o Hard to ensure all copies are deleted. 3. **Hardware Storage** o Expensive and requires physical protection. o Hardware failure or loss may cause permanent loss of keys. o Side-channel attacks and firmware bugs are possible. o Backup and recovery of non-exportable keys is difficult.	10M	CO3	L2

Key Storage Risk Factors

Security depends on both device type and environment:

Zone	Device	Environment	Security level
1	General-purpose	Uncontrolled	Lowest security (e.g., PC in public place)
2	General-purpose	Controlled	Improved by physical/network security
3	Specialized device	Uncontrolled	Exposed to attacks (e.g., ATM machines)
4	Specialized device	Controlled	Highest security (e.g., HSM in data centers)

2 a **What are the principles of security in cryptography? Explain with examples. (3M+2M)** 5M CO1 L2

Cryptography serves as the cornerstone of information security, ensuring that data remains confidential, unaltered, and accessible only to authorized entities. The fundamental principles of cryptography.

1. Confidentiality
Confidentiality ensures that information is accessible only to those authorized to access it. This is achieved through encryption, which transforms readable data (plaintext) into an unreadable format (ciphertext) using an encryption algorithm and a key. Only individuals possessing the corresponding decryption key can revert the ciphertext back to its original plaintext form.

Example: In secure email communication, the content of the email is encrypted before transmission. Only the intended recipient, who possesses the decryption key, can decrypt and read the email content.

2. Integrity
Integrity ensures that information remains accurate and unaltered during storage or transmission. Cryptographic hash functions are commonly used to verify data integrity. A hash value (checksum) is generated from the original data; any alteration in the data will result in a different hash value, indicating potential tampering.

Example: When downloading software, the website may provide a hash value. After downloading, the user can compute the hash of the downloaded file and compare it with the provided hash to ensure the file has not been altered.

3. Authentication
Authentication verifies the identity of entities involved in communication. This ensures that the parties are who they claim to be. Techniques such as digital signatures and certificates are employed to authenticate users and devices.

Example: In online banking, users authenticate themselves using usernames and passwords, and transactions may require additional authentication methods like OTPs (One-Time Passwords) sent to registered mobile numbers.

4. Non-repudiation
Non-repudiation ensures that an entity cannot deny the authenticity of their signature on a document or the sending of a message itself. Digital signatures provide proof of the origin and integrity of data, preventing the sender from denying their actions.

Example: In legal contracts, digital signatures are used to ensure that the signatory cannot later deny having signed the document.

5. Access Control
Access control mechanisms restrict access to resources to authorized users only. This principle ensures that sensitive information and systems are protected from

	unauthorized access and potential misuse. **Example:** In a corporate environment, access to confidential files is restricted based on user roles; only authorized personnel can access certain documents.			
2 b	**Describe the working of the Playfair cipher and encrypt the message "KEEP DATA SAFE" using the key "NETWORK" (1M+4M)** **What is Playfair Cipher?** • It's a **cipher that encrypts letters in pairs (digrams)** instead of one letter at a time. • Makes it **harder to break** than a simple substitution cipher because there are 676 possible diagrams. • Uses a **5×5 matrix** of letters built from a keyword. • **I and J share one cell** in the matrix. **Rules for Encryption** For each pair of letters in the message: 1. **Same row:** replace each letter with the one to its **right** (wrap around). 2. **Same column:** replace each letter with the one **below** (wrap around). 3. **Different row & column:** form a rectangle and replace each letter with the one in **its row and the column of the other letter**. 4. **Repeating letters:** separate them with **X**. 5. **Odd letters:** add **X** at the end. **3. Encrypting "KEEP DATA SAFE" with key "NETWORK"** **Step 1: Build 5×5 matrix using the keyword "NETWORK"** 1. Write the keyword first (remove duplicates): N E T W O R K 2. Fill the remaining letters (I/J together): A B C D F G H I/J L M P Q S U V X Y Z **Step 2: Prepare plaintext** • Remove spaces → **KEEPDATASAFE** • Split into digraphs → KE | EP | DA | TA | SA | FE **Step 3: Encrypt each digraph using rules** 1. **KE** → K and E are in the **same column** → take letters **below**: o K → F, E → K → Cipher: **FK** 2. **EP** → E and P are in **different rows and columns** → rectangle rule: o E → T, P → M → Cipher: **TM** 3. **DA** → D and A → rectangle → D→G, A→D → Cipher: **GD** 4. **TA** → T and A are in **same column** → T→A, A→G → Cipher: **AG** 5. **SA** → S and A → rectangle → S→Q, A→C → Cipher: **QC** 6. **FE** → F and E are in **same column** → F→K, E→E → Cipher: **KE** **Step 4: Combine ciphertext** FK TM GD AG QC KE	5M	CO1	L3
3	**Describe the RSA algorithm and solve with an example (choose p = 11, q = 17, e = 7, message = 88) (4M+6M)** ➢ RSA (Rivest–Shamir–Adleman) is a **public-key cryptosystem** used for secure data transmission. ➢ Introduced in **1977 by Ron Rivest, Adi Shamir, and Len Adleman**.	10M	CO2	L3

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

- Based on the **difficulty of factoring large prime numbers**.
- Uses a **public key** for encryption and a **private key** for decryption.

Steps of RSA Algorithm:

1. Select two prime numbers p and q, where $p \neq q$.
2. Compute $n = p \times q$.
3. Compute Euler's totient: $\phi(n) = (p-1)(q-1)$.
4. Choose e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$.
5. Compute $d = e^{-1} \mod \phi(n)$, i.e., $d \cdot e \equiv 1 \pmod{\phi(n)}$.
6. Public key: $\{e, n\}$, Private key: $\{d, n\}$.
7. **Encryption:**

$$C = M^e \mod n$$

where M = plaintext, C = ciphertext.

8. **Decryption:**

$$M = C^d \mod n$$

Step 1: Key Generation

- $p = 11, q = 17$
- $n = p \cdot q = 11 \cdot 17 = 187$
- $\phi(n) = (p-1)(q-1) = 10 \cdot 16 = 160$
- Choose $e = 7$, since $\gcd(7, 160) = 1$
- Compute d:

$$7 \cdot d \equiv 1 \pmod{160} \implies d = 23$$

Keys:

- Public key = $\{e, n\} = \{7, 187\}$
- Private key = $\{d, n\} = \{23, 187\}$

Step 2: Encryption

- Plaintext = $M = 88$
- Ciphertext:

$$C = 88^7 \mod 187$$

Using modular exponentiation:

- $88^2 \mod 187 = 77$
- $88^4 \mod 187 = 132$
- $88^7 = 88 \cdot 77 \cdot 132 \mod 187 = 11$

$$C = 11$$

Step 3: Decryption

- Ciphertext $C = 11$
- Plaintext:

$$M = 11^{23} \mod 187$$

Using modular exponentiation:

- $11^2 \mod 187 = 121$
- $11^4 \mod 187 = 55$
- $11^8 \mod 187 = 33$
- $11^{16} \mod 187 = 154$
- $11^{23} = 11 \cdot 121 \cdot 55 \cdot 33 \cdot 154 \mod 187 = 88$

$$M = 88$$

Ciphertext after encryption: C=11

Decrypted plaintext: M=88

4	**Explain the structure of the Feistel cipher with a neat diagram. Illustrate its working with a suitable example (7M+3M)**	10M	CO2	L3

- Feistel cipher structure was introduced by **IBM (Lucifer cipher)** and later used in DES.
- It is a **block cipher structure** that is still widely used, even in **Format-Preserving Encryption (FPE)**.
- The key property: **encryption and decryption use the same structure** (only the order of round keys is reversed).

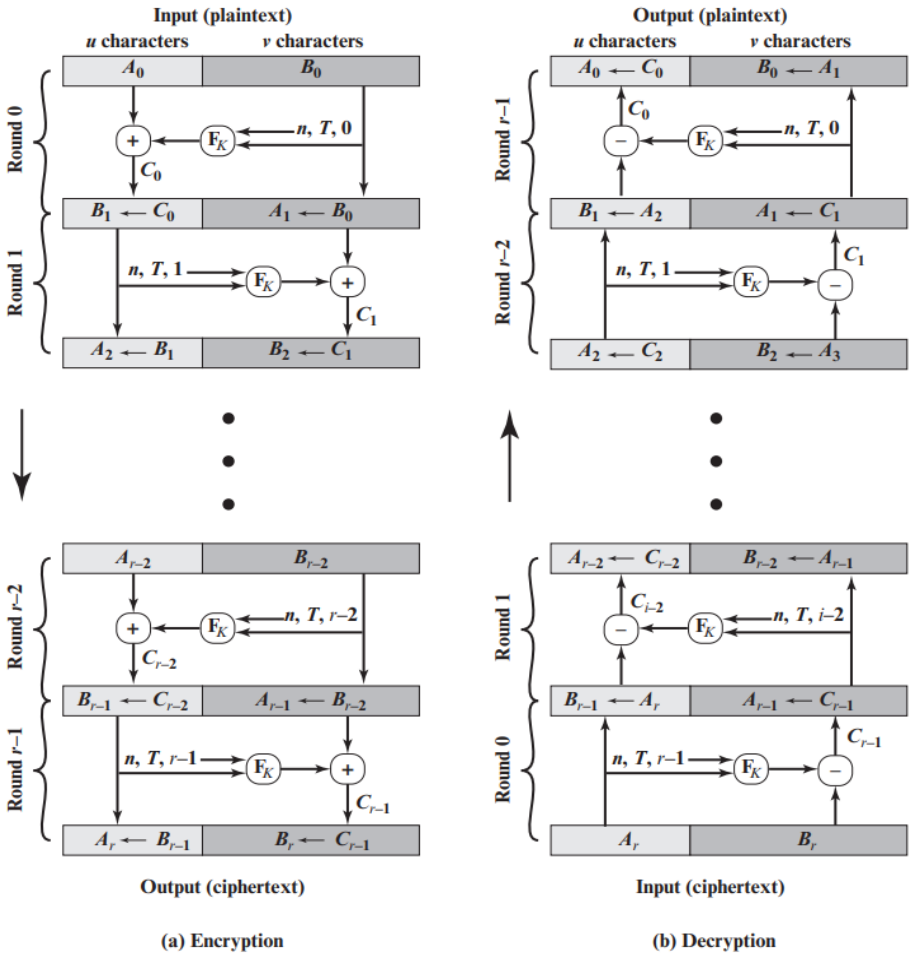


Figure 7.12 Feistel Structure for Format-Preserving Encryption

	• The plaintext block of size n is divided into two halves:
	  • A_0 (left half, length u)
	  • B_0 (right half, length v)
	• The block passes through **r rounds**.
	• Each round applies a **round function FK** with a subkey.
	
	Encryption Process (per round):
	For round i:
	
	$$A_i = B_{i-1}$$
	$$B_i = A_{i-1} + FK(B_{i-1}, K_i) \pmod{radix^m}$$
	
	• On **even rounds**, substitution is applied on the **left half (A)**.
	• On **odd rounds**, substitution is applied on the **right half (B)**.
	
	Decryption Process:
	• Similar to encryption, but **addition is replaced by subtraction**.
	• Subkeys are applied in **reverse order**.
	
	• **Plaintext (8 bits):** `10101100`
	• Split: $A_0 = 1010$, $B_0 = 1100$
	• Round keys: $K_1 = 0110$, $K_2 = 1001$
	• Round function: $F(R, K) = R \oplus K$
	
	Round 1:
	• $A_1 = B_0 = 1100$
	• $B_1 = A_0 \oplus (B_0 \oplus K_1)$
	• $= 1010 \oplus (1100 \oplus 0110)$
	• $= 1010 \oplus 1010 = 0000$
	
	Round 2:
	• $A_2 = B_1 = 0000$
	• $B_2 = A_1 \oplus (B_1 \oplus K_2)$
	• $= 1100 \oplus (0000 \oplus 1001)$
	• $= 1100 \oplus 1001 = 0101$
	
	Ciphertext = (A_2,B_2) = (0000,0101) = 00000101			

5a	**Explain the Caesar cipher. Encrypt the message "SECURITY" using a shift key of 3. Also, show the decryption process (2M+3M)**	5M	CO1	L3

- The **Caesar cipher** is one of the earliest and simplest substitution ciphers.
- Each letter in the plaintext is replaced by a letter **shifted by a fixed number of positions** in the alphabet.
- If the shift key = *k*, then the encryption rule is:

$$C = (P + k) \mod 26$$

- The decryption rule is:

$$P = (C - k) \mod 26$$

where,

- P = position of plaintext letter (0–25)
- C = position of ciphertext letter (0–25)
- k = shift key

Plaintext message: SECURITY
Shift key (k): 3

Plaintext	**S**	**E**	**C**	**U**	**R**	**I**	**T**	**Y**
Position	18	4	2	20	17	8	19	24
+ Key (3)	21	7	5	23	20	11	22	1
Cipher	V	H	F	X	U	L	W	B

Ciphertext: VHFXULWB

Decryption Process

Apply formula $P = (C - k) \mod 26$

Ciphertext	**V**	**H**	**F**	**X**	**U**	**L**	**W**	**B**
Position	21	7	5	23	20	11	22	1
– Key (3)	18	4	2	20	17	8	19	24
Plaintext	S	E	C	U	R	I	T	Y

Recovered Plaintext: SECURITY

5b	**Explain the Blum Blum Shub (BBS) generator. Generate the first four pseudorandom numbers using the parameters p=7, q=11, X_0 = 3.(2M+3M)** ➢ The **Blum Blum Shub (BBS)** generator is a cryptographically secure pseudorandom bit generator (CSPRBG). ➢ It was proposed by **Blum, Blum, and Shub (1986)** and is considered one of the strongest pseudorandom generators because its security is based on the difficulty of factoring large numbers. The working principle is as follows:	5M	CO2	L3

1. Choose two large prime numbers p and q such that:

$$p \equiv q \equiv 3 \pmod 4$$

Example: $p = 7, q = 11$.

2. Compute

$$n = p \times q$$

3. Choose a random seed s, such that s is relatively prime to n.
4. Initialize:

$$X_0 = s^2 \pmod n$$

5. Generate the sequence using the recurrence relation:

$$X_i = (X_{i-1})^2 \pmod n, \quad i \geq 1$$

6. The output bit at each stage is:

$$B_i = X_i \pmod 2$$

- The sequence of least significant bits forms the pseudorandom sequence.
- The BBS generator passes the **next-bit test**, meaning that given k bits of output, the $(k+1)$th bit cannot be predicted with probability significantly greater than 0.5.
- Security depends on the hardness of factoring n.

$$p = 7, \quad q = 11, \quad X_0 = 3$$

1. Compute modulus:

$$n = p \times q = 7 \times 11 = 77$$

2. Initialize (already given as $X_0 = 3$).
3. Iterations:

- **Step 1**:

$$X_1 = (X_0)^2 \pmod n = (3^2) \pmod{77} = 9$$

$$B_1 = X_1 \pmod 2 = 9 \pmod 2 = 1$$

- **Step 2**:

$$X_2 = (X_1)^2 \pmod{77} = (9^2) \pmod{77} = 81 \pmod{77} = 4$$

$$B_2 = X_2 \pmod 2 = 4 \pmod 2 = 0$$

- **Step 3**:

$$X_3 = (X_2)^2 \pmod{77} = (4^2) \pmod{77} = 16$$

$$B_3 = X_3 \pmod 2 = 16 \pmod 2 = 0$$

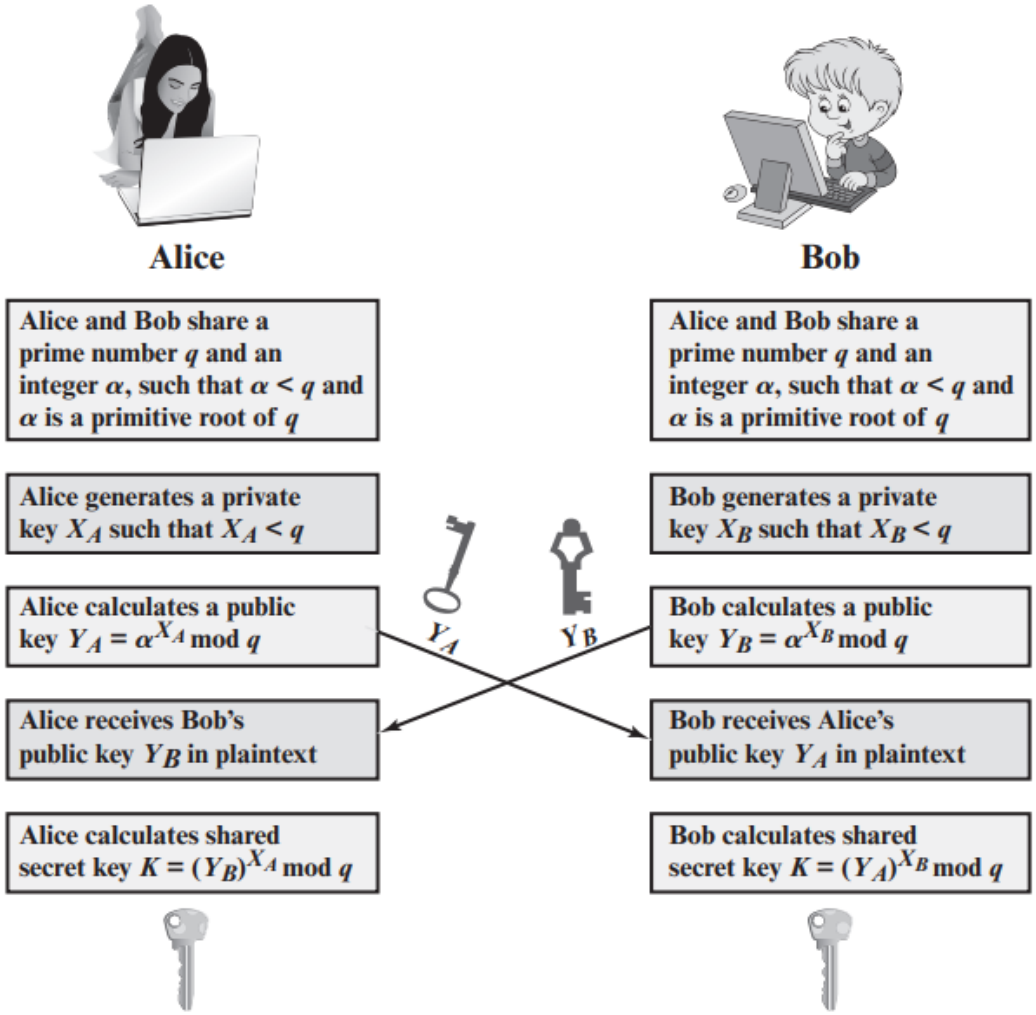
- **Step 4**:

$$X_4 = (X_3)^2 \pmod{77} = (16^2) \pmod{77} = 256 \pmod{77} = 25$$

$$B_4 = X_4 \pmod 2 = 25 \pmod 2 = 1$$

The first four pseudorandom numbers (bits) are:

$$B_1 = 1, \quad B_2 = 0, \quad B_3 = 0, \quad B_4 = 1$$

6	**Analyze the Diffie-Hellman Key exchange protocol. Perform the key exchange using the following parameters q=29, α=2. User A selects X_A=5, User B selects X_B=11 (5M+5M)** • The Diffie–Hellman algorithm allows **two parties (A and B)** to securely agree on a **shared secret key** over an insecure channel. • This shared key can then be used with **symmetric encryption** to secure communication  Figure 10.1 The Diffie–Hellman Key Exchange $K = (Y_B)^{X_A} \bmod q$ $= (\alpha^{X_B} \bmod q)^{X_A} \bmod q$ $= (\alpha^{X_B})^{X_A} \bmod q$ by the rules of modular arithmetic $= \alpha^{X_B X_A} \bmod q$ $= (\alpha^{X_A})^{X_B} \bmod q$ $= (\alpha^{X_A} \bmod q)^{X_B} \bmod q$ $= (Y_A)^{X_B} \bmod q$	10M	CO2	L3

$$
\begin{aligned}
K &= (Y_B)^{X_A} \bmod q \\
&= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\
&= (\alpha^{X_B})^{X_A} \bmod q && \text{by the rules of modular arithmetic} \\
&= \alpha^{X_B X_A} \bmod q \\
&= (\alpha^{X_A})^{X_B} \bmod q \\
&= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\
&= (Y_A)^{X_B} \bmod q
\end{aligned}
$$

1) Public parameters

- Prime $q = 29$
- Base (primitive root) $\alpha = 2$

  These are public.

2) Each user's public value

Compute $Y_A = \alpha^{X_A} \bmod q$ and $Y_B = \alpha^{X_B} \bmod q$.

- $Y_A = 2^5 \bmod 29 = 32 \bmod 29 = 3$.
- $Y_B = 2^{11} \bmod 29$.

  Compute quickly: $2^5 \equiv 3$ so $2^{10} \equiv 3^2 = 9$. Then $2^{11} \equiv 9 \cdot 2 = 18$.

  So $Y_B = 18$.

Public keys: $Y_A = 3,\ Y_B = 18$.

3) Shared secret (computed by both sides)

- A computes $K = Y_B^{X_A} \bmod 29 = 18^5 \bmod 29$.

  Work: $18^2 = 324 \equiv 5$.

  $18^4 \equiv 5^2 = 25$.

  $18^5 \equiv 18^4 \cdot 18 \equiv 25 \cdot 18 = 450 \equiv 450 - 15 \cdot 29 = 450 - 435 = 15$.

  So A gets $K = 15$.
- B computes $K = Y_A^{X_B} \bmod 29 = 3^{11} \bmod 29$.

  Work: $3^2 = 9,\ 3^4 = 9^2 = 81 \equiv 23,\ 3^8 \equiv 23^2 = 529 \equiv 7$.

  $3^{11} = 3^8 \cdot 3^2 \cdot 3 \equiv 7 \cdot 9 \cdot 3 = 7 \cdot 27 = 189 \equiv 189 - 6 \cdot 29 = 189 - 174 = 15$.

  So B also gets $K = 15$.

Shared secret key = 15.