Q1: How does Shor's algorithm threaten the security of RSA and Elliptic Curve Cryptography (ECC), and what are the potential consequences for current digital infrastructure?

Soln:

Shor's algorithm, developed by mathematician Peter Shor in 1994, poses a significant threat to widely used cryptographic systems, particularly RSA and Elliptic Curve Cyptography (ECC).

RSA is based on the difficulty of ~~factoring~~ large integers, while ECC depends on the difficulty of solving the elliptic curve discrete logarithm problem. Classical ~~pred~~ computers are unable to solve these problems efficiently.

However, Shor's algorithm, when run on a sufficiently

powerful quantum computer can solve both of these problems in polynomial time. Specifically, it can factor large integers and compute discrete logarithms exponentially faster than the best-known classical algorithms. This renders both RSA and ECC vulnerable to being broken, meaning that encrypted messages could be decrypted, digital signatures could be forged and secure communications could be compromised.

The implications for digital infrastructure are profound. RSA and ECC are foundational to securing internet traffic, banking systems, emails, digital certificates and more. If large scaled quantum computers become practical

much of the world's digital security would be obsolete. Data encrypted today could be stored and decrypted in the future - a threat known as "harvest now, decrypt later." This is spurred a global race to develop quantum-resistant (post-quantum) cryptographic algorithms that can withstand quantum attacks.

In summary, Shor's algorithm undermines the core assumptions that make RSA and ECC secure. Its potential realization threatens the integrity, confidentiality, and trust of the entire digital ecosystem, necessitating a swift transition to quantum-safe cryptography.

Q2: Discuss the role of quantum key distribution (QKD) in future cryptographic systems. How does it differ from classical public key encryption?

Soln:

→ Quantum key Distribution (QKD) is a revolutionary approach to secure communication that uses principles of quantum mechanics to distribute encryption keys between parties in a provably secure manner. Unlike classical public-key encryption, which relies

→ QKD is important for protecting communication against future quantum computer attacks.

Difference from classical public key encryption is given bellow:

Quantum key Distribution (QKD)	Classical Public key Encryption
i. Security based on physical laws of quantum mechanics, providing provable unconditional security.	i. Security based on computational hardness assumptions
ii. Keys are securely distributed over quantum channels using quantum states like Photons.	ii. Keys are exchanged or encrypted mathematically over classical communication channel.
iii. Any attempts to eavesdrop disturb the quantum state which can be immediately detected.	iii. Eavesdropping can not be detected directly, Security depends on difficulty of decryption with the private key.
iv. Produces symmetric keys to be used for encryption/ decryption after secure exchange.	iv. Uses public, private key-pairs for encryption/decryption and digital signature.
v. Unconditional secure against quantum computer attacks.	v. Vulnerable to quantum algorithms like Shor's which can break many

	classical crypto systems.
vi. Require specilized quantum hardware.	vi. Implementation with software and classical hardware widely deployable today

Q3: What is the main differences between lattice based cryptography and traditional number theoretic approaches like RSA, particularly in the context of quantum resistance?

Ans:

Lattice Based Cryptography	Traditional Cryptography
1. Based on math problems about point in space called lattice	1. Based on hard math problems like factoring large number.

Lattice Based Cryptography	Traditional Cryptography
2. Considered safe against quantum computers, no known quantum attack can easily break it.	2. Vulnerable to quantum computers, using shor's algorithm which can break it fast
3. Keys and ciphertext are usually large in size.	3. Keys are smaller and faster to use
4. Algorithms often use simple math operations like addition and multiplication on vectors	4. Uses complex operations like modular exponentiation.
5. Supports advanced functions like homomorphic encryption and identity based encryption.	5. Mostly supports basic encryption and digital signatures.

Question 4:

Develop a python based PRNG that uses the current system time and a customer seed value. Write a complete program and corresponding output.

Ans: PRNG means Pseudo - Random Number Generator.

```
import time
#Simple PRNG using linear congruence
def prng(custom_seed):
    m = 2**48
    a = 25214903917
    c = 11
#mix time and seed
    state = (custom_seed ^ time.time_ns()) % m
```

```
#generate 10 random integers
for _ in range(10):
    state = (a * state + c) % m
    print(state)
#Example run
custom_seed = 2025
prng(custom_seed)
```

Example Output:

```
1 8 7 5 8 5 9 5 0 6 8 8 4 1 7
1 1 3 7 3 9 0 2 7 9 0 4 3 9 8
1 0 5 2 8 1 3 5 5 0 6 8 2 9 3
2 9 4 3 3 4 9 0 1 9 6 6 1 2 4 6
. . . . .
```

Ques 5: Explain the sieve of Eratosthenes algorithm and use it to find all Prime numbers less than 50. How does its time complexity compare to trial division?

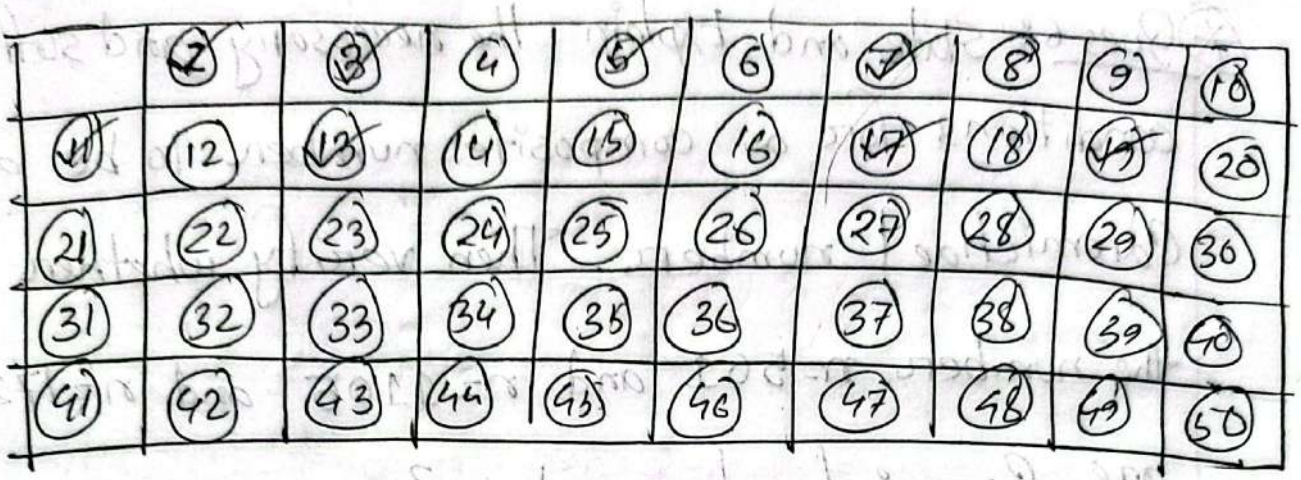
Ans: Explain the Sieve of Eratosthenes algorithm and use it to find all Prime numbers less than 50

Ans:

→ Sieve Eratosthenes Algorithm is an effective way to find all prime numbers up to a given number n. It works by iteratively marking the multiple of each prime number starting from 2.

How it works:

1. Create a list of numbers from 2 to n.
2. Start with the first prime number.
3. Mark all multiples of 2 (4, 6, 8, ...) as not prime.
4. Move to the next unmarked number (3) and mark all multiples of 3 (6, 9, 12, ...)
5. Repeat this for the next unmarked numbers (5, 7, 11, ...) up to n.
6. Remaining unmarked numbers are prime.



Prime less than 56: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47

Time Complexity:

Sieve of Eratosthenes:

Time complexity is approximately $O(n \log \log n)$ which is very efficient for generating all primes up to n.

Trial division:

Time complexity is $O(n\sqrt{n})$ - much slower for large n.

→ Sieve of is significantly faster than trial division when generating a prime number.

→ Trial division is better for checking if a single number is prime, not generating many.

(*) Ques 6: State and Explain the necessary and sufficient conditions for a composite number to be a Carmichael number. Then verify whether the number $n=561$ and $n=1105$ and $n=1729$ are Carmichael number?

Ans: A ~~Char~~ Carmichael number is a special type of composite number ~~nnnn~~ n that fools Fermat's primality test.

A composite number n is called a Carmichael number if $a^{n-1} \equiv 1 \bmod n$ for all integers such that $\gcd(a, n) = 1$.

Although, ~~nnn~~ n is not prime, it still behaves like a prime number under Fermat's test

~~A compo~~ Necessary and Sufficient Condition's (Korselt's Criterion)

Let n be a composite number. Then

n is a Carmichael number if and only if:

1. n is square-free (no prime factor is repeated)
2. For every divisor p of n, it holds that $p-1 \mid n-1$

Verify the three numbers :-

We apply Korselt's criteria to each n

1. $n = 561$

→ factorization $561 = 3 \times 11 \times 17$ (composite)

→ Square free : yes (distinct prime)

→ $n-1 = 560$

For $P=3 : P-1=2$, $560/2 = 280$ (divides)

for $P=11 : P-1=10$, $560/10 = 56$ (divides)

for $P=17 : P-1=16$, $560/16 = 35$ (divides)

All condition satisfied. So 561 is a Carmichael number.

2. $n = 1105$

→ Factorization $1105 = 5 \times 13 \times 17$ (composite)

→ Square free : Yes.

→ $n-1 = 1104$

$P=5$; $P-1=4$, $1104/4 = 276$ (divides)

$P=13$; $P-1=12$, $1104/12 = 92$ (divides)

$P=17$; $P-1=16$, $1104/16 = 69$ (divides)

All condition satisfied. So 1105 is a Carmichael number.

3. $n = 1729$

→ Factorization : $1729 = 7 \times 13 \times 19$ (Composite)

→ Square free = Yes

→ $n-1 = 1728$

$P=7$; $P-1=6$; $1728/6 = 288$ (divides)

$P=13$; $P-1=12$; $1728/12 = 144$ (divides)

$P=19$; $P-1=18$; $1728/18 = 96$ (divides)

All condition is satisfied, So, 1729 is a Carmichael number.

So, All these three numbers are Carmichael number.

Ques 7: Determine whether the following are valid algebric structures and justify your answer:-

i. Is the set Z_{11} with operations $(+, .)$ a ring?

ii. Are the sets $(Z_{37}, +)$ and $(Z_{35}, \times)$ Abelian groups?

Ans:

(i) Yes, Z_{11} is a ring.

→ Z_{11} is the set of integers modulo 11.

$Z_{11} = \{0, 1, 2, \ldots, 10\}$

→ Addition and multiplication are defined modulo 11.

→ A ring requires:-

1. $(Z_{11}, +)$ is an Abelian groups (for every a, then is a $-a \bmod 11$)

closure, associativity, identity(0), inverse and commutativity are all satisfied.

2. $(Z_{11}, \cdot)$ is closed and associative.

3. Distributive property holds.

$$a.(b+c) \equiv a.b + a.c \pmod{11}$$

All these properties of ring is satisfied, So, Z_{11} is a ring.

(ii) a) $(\mathbb{Z}_{37}, +)$ is an ~~a+b~~ Abelian group:

→ It consists of integers modulo 37:

$$\mathbb{Z}_{37} = \{0, 1, \ldots, 36\}$$

→ Operation: Addition modulo 37.

→ Closure, associativity, identity 0, inverse ($\forall a \in \mathbb{Z}_{37}, \exists -a \bmod 37$), Commutativity

So, $(\mathbb{Z}_{37}, +)$ is an Abelian group.

(b) $(\mathbb{Z}_{35}, \times)$

This is the set of units modulo 35 under multiplication:

$\mathbb{Z}_{35}$,

(*) Ques 8: What is the remainder when -52 is reduced modulo 31?

Step 1: Write the congruence idea

We want r such that $-52 \equiv r \pmod{31}$

and $0 \leq r < 31$

Step 2: Add multiples of 31 until we get a non-negative remainder

$-52 + 31 = -21$ (Still negative)

$-21 + 31 = 10$ (non negative and less than 31)

Step 3: $-52 \equiv 10 \pmod{31}$

(*) Ques 9: Determine the multiplicative inverse of 7 mod 26 if it is exists. (Use extended Euclidean Algorithm).

Ans:

We want to find the multiplicative inverse of 7 mod 26 - meaning we want an integer x such that:

$7x \equiv 1 \pmod{26}$

Step 1: Checks if inverse exists. (Apply Euclidean Algo.)

An inverse exists if gcd (7, 26) = 1

$$26 = 3 \times 7 + 5$$
$$7 = 1 \times 5 + 2$$
$$5 = 2 \times 2 + 1$$
$$2 = 2 \times 1 + 0$$

GCD(7, 26) = 1 means inverse exists.

Step 2: Extended Euclidean Algorithm (Back substitute to find x)

$$1 = 5 - 2 \times 2$$
$$1 = 5 - 2 \times (7 - 1 \times 5)$$
$$\therefore 1 = 5 - 2.7 + 2.5$$
$$1 = 3.5 - 2.7$$
$$1 = 3.(26 - 3.7) - 2.7 \quad \text{[Replace 5]}$$
$$1 = 3.26 - 9.7 - 2.7$$
$$\therefore 1 = 3.26 - 11.7$$

Step 3: Identify inverse

$$1 = 3.26 - 11.7$$
$$\therefore -11.7 = 1 \bmod (26)$$
$$\therefore x = -11$$

Now, reduce $-11 \bmod 26$:

$$-11 \equiv 15 \pmod{26}$$

So, the multiplicative inverse of 7 mod 26 is 15.

Ques 10:

Evaluate $(-8 * 5) \bmod 17$, and explain how to simplify negative modular multiplication.

Ans: Step 1: Multiply $-8 \times 5 = -40$

Step 2: Reduce modulo 17

$$-40 \bmod 17$$

To get a positive remainder, we can add 17 repeatedly until the result is in the range $0 \leq r < 17$.

$-40 + 17 = -23$ (still negative)

$-23 + 17 = -6$ (still negative)

$-6 + 17 = 11$ (positive)

$\therefore (-8 \times 5) \bmod 17 = 11$

(Ans)

Ques 11: State and prove Bézout's Theorem. Use it to find the multiplicative inverse of 97 modulo 385.

Ans: Bézout's Theorem also called Bézout's Identity.

Let a and b be integers, not both zero. Then there exist integers x and y such that:

$$ax + by = \gcd(a, b)$$

Moreover, the gcd of ~~a, b~~ a and b is the smallest positive integer that can be expressed as a linear combination of a and b.

If $\gcd(a, m) = 1$, then Bézout theorem tells us there exists an x such that:

$$ax + ~~by~~ = my = 1$$

$$\Rightarrow ax \equiv 1 \mod m$$

that x is the multiplicative inverse of a mod m.

Use Bézout theorem to find the inverse of 97 mod 385

we have to solve, $97x \equiv 1 \mod 385$

$$\Rightarrow 97x + 385y = 1$$

We solve this by using Extended Euclidean Algorithm

Step 1. Apply Euclidean Algorithm

gcd(385, 97): $385 = 3.97 + 94$

$97 = 1.94 + 3$

$94 = 31.3 + 1$

$3 = 3.1 + 0$

$\therefore \gcd(385, 97) = 1 \rightarrow$ inverse exists.

Step 2: Extended Euclidean Algorithm (Back-Substitution)

$$1 = 94 - 31.3$$

$$\Rightarrow 1 = 94 - 31.(97 - 1.94)$$

$$\Rightarrow 1 = 94 - 31.97 + 31.94$$

$$\Rightarrow 1 = 32.94 - 31.97$$

$$\Rightarrow 1 = 32.(385 - 3.97) - 31.97$$

$$= 32.385 - 96.97 - 31.97$$

$$1 = 32.385 - 127.97$$

$\therefore 97.(-127) + 385.32 = 1$

$\therefore -127$ is the reverse of 97 mod 385

We want positive inverse, so reduce

$$-127 \bmod 385 = 385 - 127 = 258$$

(Ans)

Ques 12:

Using Bézout's identity, prove that the equation $ax + by = \gcd(a, b)$ has integer solutions. Find x such that $43x = 1 \bmod 240$.

Ans:

Let a, b be integers, not both zero, and let $d = \gcd(a, b)$. Use Euclidian algorithm to compute d.

$$a = q_1 b + r_1, \quad 0 \le r_1 < b$$

$$b = q_2 r_1 + r_2$$

$$r_1 = q_3 r_2 + r_3$$

!

$r_{k-2} = q_k r_{k-1} + r_k$

$r_{k-1} = q_{k+1} r_k + 0$

So, $r_k = d$. Each remainder r_i is an integer linear combination of a and b because $r_1 = a - q_1 b$ and therefore $r_{i+1} = r_{i-1} - q_{i+1} r_i$. By back substitution we express r_k as $r_k = xa + yb$ for some integers x, y. Hence d is an integer linear combination of a and b. Since any common divisor of a and b divides every linear combination, d is the smallest positive such combination and we have Bézout's identity:

$$ax + by = \gcd(a, b)$$

⇒ Find x with $43x \equiv 1 \pmod{240}$

We need an integer x with $43x + 240y = 1$. Use the Euclidean algorithm –

$240 = 5.43 + 25$

$43 = 1.25 + 18$

$25 = 1.18 + 7$

$18 = 2.7 + 4$

$7 = 1.4 + 3$

$4 = 1.3 + 1$

$3 = 3.1 + 0$

Back-substitute to express 1 as a combination of 43 and 240:

$1 = 4 - 1.3$

$= 4 - 1.(7 - 1.4)$

$= 4 - 1.7 + 1.4 \quad = 2.4 - 1.7$

$= \cancel{2.4}\ 2.(18 - 2.7) - 1.7$

$= 2.18 - 4.7 - 1.7$

$= 2.18 - 5.7$

$= 2.18 - 5.(25 - 1.18)$

$= 2.18 - 5.25 + 5.18$

$= 7.18 - 5.25$

$= 7.(43 - 1.25) - 5.25$

$= 7.43 - 12.25$

$= 7.43 - 12.(240 - 5.43)$

$\therefore 1 = \cancel{67.25} - 12.240$

∴ So, $67 \cdot 25 + (-12) \cdot 240 = 1$, Thus $x = 67$ is a solution, it gives the multiplicative inverse of 43 modulo 240.

Reducing modulo 240, the inverse is,

$$43^{-1} \equiv 67 \pmod{240}.$$

Ques 13: Prove Fermat's Little Theorem and explain how it is used to test for primality. Is 561 a prime number based on this test? Evaluate 5^123 mod 175 using Fermat's Little Theorem. Show all steps.

Ans:

If p is prime and a is an integer with gcd(a,p)=1 then

$$a^{p-1} \equiv 1 \pmod{p}$$

Proof (standard multiplicative-residue proof):- Consider the nonzero residue classes modulo p: $\{1, 2, \dots, p-1\}$. Multiplying each element by a (with $\gcd(a, p) = 1$) permutes this set modulo p. So

$$a.1, a.2, \dots, a.(p-1)$$

is congruent modulo p to some permutation of $1, 2, \dots, p-1$. Taking the product of the terms in both lists gives

$$a^{p-1}.(1.2 \dots (p-1)) \equiv 1.2 \dots (p-1) \pmod{p}$$

Since $1.2 \dots (p-1)$ is not divisible by p, we may cancel it in the field $\mathbb{Z}/p\mathbb{Z}$, yielding

$$a^{p-1} \equiv 1 \pmod{p}.$$

Using Fermat's theorem as a primality test

→ Pick an integer a with $1 < a < n$.

→ compute $a^{n-1} \bmod n$.

- If $a^{n-1} \not\equiv 1 \pmod{n}$, then n is definitely composite.
- If $a^{n-1} \equiv 1 \pmod{n}$, then n is probably prime.

→ Is 561 prime based on this test?

$$561 = 3 \times 11 \times 17$$

For any a with $\gcd(a, 561) = 1$, by Fermat's theorem.

$$a^2 \equiv 1 \pmod{3}, \quad a^{10} \equiv 1 \pmod{11}, \quad a^{16} \equiv 1 \pmod{17}$$

$\text{lcm}(2, 10, 16) = 80$ and 560 is a multiple of each of 2, 10, 16, we get $a^{560} \equiv 1$ modulo each of 3, 11, 17.

By Chinese Remainder Theorem this implies

$$a^{560} \equiv 1 \text{ (modulo } 561)$$

for every a with $\gcd(a, 561) = 1$, so the fermat test will say 'probably prime' for 561, but 561 is actual composite. So, 561 is a Carmichael number.

→ $175 = 25 \cdot 7$. Because $\gcd(5, 175) = 5 \neq 1$, Fermat's theorem does not apply directly to modulo 175.

1. Modulo 25

For $n \geq 2$, $5^2 = 25 \equiv 0 \pmod{25}$. Hence for any exponent ≥ 2,

$$5^{123} \equiv 0 \pmod{25}$$

2. Modulo 7:

$\gcd(5, 7) = 1$, so we can use Fermat: $5^6 \equiv 1 \pmod{7}$

Compute exponent reduction: $123 \equiv 3 \pmod{6}$. Thus

$$5^{123} \equiv 5^3 \pmod{7}$$

$5^2 = 25 \equiv 4 \pmod{7}$ so $5^3 \equiv 5 \cdot 4 = 20 \equiv 6 \pmod{7}$

$$5^{123} \equiv 6 \pmod{7}$$

3. Combine by CRT

$$x \equiv 0 \pmod{25}, \quad x \equiv 6 \pmod{7}$$

$x = 25k$. Then $25k \equiv 6 \pmod 7$. Since $25 \equiv 4 \pmod 7$.

$$4k \equiv 6 \pmod 7$$

Multiply both sides by the inverse of 4 modulo 7.

~~Since 25 = 4~~ The inverse is $4^{-1} \equiv 2 \pmod 7$

$k = 5 + 7t$ and $x = 25k = 25.5 + 25.7t = 125 + 175t$.

The last residue modulo 175 is

$$x \equiv 125 \pmod{175}$$

Ans: $5^{123} \equiv 125 \pmod{175}$

Ques 14. State and prove the Chinese Remainder Theorem Then solve the following system of conguence.

$$x \equiv 2 \pmod 3,\ x \equiv 3 \pmod 5,\ x \equiv 2 \pmod 7$$

Ans:

State: Let $m_1, m_2, \ldots, m_k$ be pairwise coprime positive integers $\gcd(m_i, m_j) = 1$ for $i \neq j$

for any integers $a_1, \ldots a_k$ the system

$$x = a_i \pmod{m_i} \quad i = 1, \ldots, k$$

has a unique solution modulo $m = m_1, m_2 \ldots m_k$

Proof:

Let $M = \prod_{i=1}^{k} m_i$ and for each i put $M_i = \frac{M}{m}$

Since gcd $(M_i, m_i) = 1$, there exists an inverse y_i with

$$M_i y_i \equiv 1 \pmod{m_i}$$

Then $x = \sum_{i=1}^{k} a_i M_i y_i$

Satisfies $x \equiv a_i \pmod{M_i}$ for every i (because all terms with $j \neq i$ vanish modulo m_i and $M_i y_i \equiv 1 \pmod{m_i}$

So, solution exists.

$$\begin{cases} x \equiv 2 \pmod 3 \\ x \equiv 3 \pmod 5 \\ x \equiv 2 \pmod 7 \end{cases}$$

Here $m_1 = 3, m_2 = 5, m_3 = 7$ are pairwise coprime and $M = 3 \times 5 \times 7 = 105$

$M_i \bmod m_i$

i	m_i	a_i	$M_i = \frac{M}{m_i}$	$y_i = M_i^{-1} \pmod{m_i}$	$a_i M_i y_i$
1	3	2	35	$35 \equiv 2 \bmod 3$, $y_1 = 2$	$2 \times 35 \times 2 = 140$
2	5	3	21	$y_2 = 1$	$3 \times 21 \times 1 = 63$
3	7	2	15	$y_3 = 1$	30

Now, $140 + 63 + 30 = 233$

$\Rightarrow 233 \bmod 105$

$= 23$

$\therefore x \equiv 23 \pmod{105}$ (Ans)

Ques 15:

Briefly explain the CIA triad in information security. How does each component contribute to building a secure system?

Ans:

The CIA triad is a fundamental model for designing and evaluating security, policies consist of :-

Confidentiality:

(i) Defination: Ensures that information is accessable only to authorized individuals

(ii) Goal: Prevent unauthorized access or disclosure

(iii) Example: Using encryption and access control.

(iv) Contribution is security: Protects sensitive data from hackers, eaves droppers and unauthorized access (users).

Integrity:

(i) Defination: Ensures data is accurate complete and unaltered except by authorized person.

(ii) Goal: Prevent unauthorized modification

(iii) Example: Digital signature, checksum and hashing

(iv) Contribution to security: Maintain trust in data by ensuring it's not tampered with

or corrupted

Availability:

(i) Definition: Ensures authorized users have reliable and timely access to information and resources.

(ii) Goal: Prevent disruption of services.

(iii) Example: Redundant systems, backups and DDOS protection

(iv) Contribution to security: Ensures that data and systems are accessible when needed, even during failure or attacks.

Ques 16: How does steganography differ from cryptography in the context of information security, and what are common techniques used for hiding data in digital media?

Ans: **Cryptography:** It is the process of converting information into an unreadable format (ciphertext) to protect its content from unauthorized access. The presence of the message is visible but its meaning is hidden.

Steganography: It is the practice of hiding the very existance of information by embedding it inside another medium.

Cryptography	Steganography
① Protect the content of a message	① Hide the existance of a message.
② Message is visible but unreadable	② Message is invisible to easual obsenvers
③ Message is known to exist	③ Message is hard to detect

Common steganography techniques in digital media:-

1. Least Significant bit (LSB) insertion: Change the smallest bits in image pixels to hide data.
2. Masking and filtering: Hide data in important parts of an image (like watermark). So, it's harder to remove.
3. Transform domain techniques: Hide data in frequency parts of media instead of directly in pixels.
4. Meta data Hiding: Put secret data into unused file information fields (meta data) without changing the visible content.

Ques 17: what are the key difference between phishing malware and denial of service (DoS) attacks in terms of their method and impact of system security?

Ans: Here is the difference between phishing, Malware an denial-of-service (DoS) attack :-

Attack type	Method	Impact on Security
Phishing	Tricking users (usually) via fake emails, websites or messages) into revealing sensitive information like password, bank details, or credit card numbers	Compromise confidentiality by stealing personal or log in data
Malware	Installing malicious software (virus, worm, Trojan, ransomware, spyware) on a system often via downloads, email attachments, or infected sites	Can harm confidentiality integrity and availability by stealing data, modifying files or disabling systems.
Denial of Service (DoS)	Flooding a network, server or website with excessive traffic to overload and make it unavailable to users.	Targets availability by disrupting normal access to services.

Ques-18 : Explain how legal frameworks such as General Data Protection Regulation (GDPR) help mitigate cyber attacks and protect users privacy :

Ans:

~~The explanation~~ The GDPR is a legal framework in the European Union designed to strengthen data privacy, data security and user rights. It helps mitigate cyber attacks and protect user privacy in several ways:

1. Data Protection by Design and by Default.
   → Security must be built into systems from the start.
2. Data Minimization and Purpose Limitation:
   Collect and store only the necessary data.
3. Security and Breach Notification: Use strong safeguards and report breaches within 72 hours.

4. Accountability & Penalties: Show compliance or face heavy fines.

5. Users Rights and Transparency:

Users can access, delete or transfer their data anytime.

How GDPR helps against Cyber attacks.

Cybersecurity Threat	GDPR Protection Mechanism
Data breaches	Encryption, access control and breach notification requirements
Phishing/identity theft	Data minimization reduces exposed personal ~~threat~~ data
Insider threats	Strict accountability and audit trails
Ransomware	Backup and recovery plans enforced through security obligations
Unauthorized tracking	User consent and clear data usage policies.

Ques-19: Explain the basic working of the DES Algorithm using a single 64 bit paintext block and a 56 bit key show how the initial permutation, round function and final permutation contribute to the encryption process.

Ans: DES basic working.

→ Using 64 bit plaintex and 56 bit key.

Step 1: Input block and key

→ plain text 64 bits

→ key 64 bits input but only 56 bit are used.
(8 bit for parity)

Step-2: Initial permutation (IP)

→ The 64 bit plain text undergoes initial permutation (IP) using a fix table

→ The bits are re-arranged according to the IP table.

Output: Two 32 bit halves. Lo and Ro

Step 3: 16 Round of feistel structure

Each Round has:

→ Expansion (E)

→ Key mixing (XOR)

→ Substitution (S-box)

→ Permutation (P)

Steps Per Round:

a. Expansion (E-box)

R(n-1) 32 bits → expand to 48 bits using expansion table

b. key mixing

→ The 48 bit expanded R(n-1) is XORed with 48 bit Subkey K(n)

→ Subkeys are generated from the 56 bit main key one for each round

c. Substitution (S-box)

→ The XOR Result is divided 8 blocks of 6 bits

→ Each block goes through an S-box (Substitution box), outputing 4 bit each.

→ Final result : 32 bits.

d. Permutation (P-box)

→ The 32-bit output from S-box is permuted again using a fixed table.

→ Increases confusion and diffusion

e. Feistel Swap

→ New values: $L(n) = R(n-1)$

$R(n) = L(n-1) \text{ XOR } f(R(n-1), k(n))$

4. Repeat for 16 Rounds

→ Each round uses a different 48 bit subkeys generated from the original key. using PC-1 shifts and PC-2 tables.

5. Final Permutation (FP):

After Round 16, R_{16} and L_{16} are swapped and then the final permutation (inverse of IP) is applied.

Qus, 20:

Given that,

$R_0 = 0x f0\ f0\ F0\ F0$

$K_1 = 0X0F0F0F\ 0F$

$L_0 = 0X = AAAAAAAA$

Step-1: First Round function (only XOR)

$f(R_0, K_1) = R_0 \oplus K_1$

$= 0X\ F0F0\ F0\ F0 \oplus 0X0F0F0F0F$

$= 0X\ FFFFFFFF$

Step-2: Compute L_1 and R_1

$L_1 = R_0 = 0XF0F0F0F0$

$R_1 = L_0 \oplus F(R_0, K_1) = 0XAAAAAAAA \oplus 0XFFFFFFFF$

$= 0X55555555$

$L_1 = 0XF0F0F0F0$ and $R_1 = 0X55555555$

Qn 21:

Ans: $0X23 \rightarrow Row = 2, Col = 3 \longrightarrow D_4$

$0XA7 \rightarrow Row = A, Col = 7 \longrightarrow 63$

$0X4E \rightarrow Row = 4, Col = E \longrightarrow 2F$

$0X19 \rightarrow Row = 1, Col = 9 \longrightarrow C_6$

Output word = $[D_4, 63, 2F, C_6]$

When no value found given, N/A (not available)

Ans: 23. Given that,

Col = [0x01, 0x02, 0x03, 0x04]

= [00000001, 00000010, 00000011, 00000100]

MixColumns Matrix.

$$M_2 = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

Step 1: First Row of output

= 0x02 · 0x01 ⊕ 0x03 · 0x02 ⊕ 0x01 · 0x03

⊕ 0x01 · 0x04

= 0x02 ⊕ 0x06 ⊕ 0x03 ⊕ 0x04

= 00000010 ⊕ 00000110 ⊕ 00000011 ⊕

00000100

= 00000011

= 0x03

Step-2: Second Row Output

$0x01 \cdot 0x01 \oplus 0x02 \cdot 0x02 \oplus 0x03 \cdot 0x03 \oplus 0x01 * 0x04$

$= 0x01 \oplus 0x04 \oplus (0x03 \cdot 0x02 \oplus 0x03 \cdot 0x01) \oplus 0x01 * 0x04$

$= 0x01 \oplus 0x04 \oplus (0x06 \oplus 0x03) \oplus 0x04$

$= 0x01 \oplus 0x04 \oplus 0x05 \oplus 0x04$

$= 00000001 \oplus 00000100 \oplus 00000101 \oplus 00000100$

$= 00000100 = 0x04$

Step 3: Third Row output

$= 0x01 \cdot 0x01 \oplus 0x01 \cdot 0x02 \oplus 0x02 \cdot 0x03 \oplus 0x03 \cdot 0x04$

$= 0x01 \oplus 0x02 \oplus 0x03 \oplus ((0x02 + 0x01) \cdot 0x04)$

$= 0x01 \oplus 0x02 \oplus 0x03 + (0x02 \cdot 0x04 \oplus 0x01 \cdot 0x04)$

$= 0x01 \oplus 0x02 \oplus 0x03 \oplus 0x08 \oplus 0x04$

$= 00000001 \oplus 00000010 \oplus 00000011 \oplus 00001006 + 00000100$

$= 00001001 = 0x09$

Step 4: Fourth Row output.

$= (0x03 \cdot 0x01) \oplus (0x01 \cdot 0x02) \oplus (0x01 \cdot 0x03) \oplus (0x02 \cdot 0x04)$

$= 0x03 \oplus 0x02 \oplus 0x03 \oplus 0x08$

$= 0x0011 \oplus 0x0010 \oplus 0x0011 \oplus 0x1000$

$= 0x1010$

$= 0x0A$

Output = [0x03, 0x04, 0x09, 0x0A]

24

AES-OFB:-

is a stream cipher mode of AES encryption.

→ It turns the block cipher (AES) into a synchronous stream cipher using feedback

Working Procedure:

① Initialization Vector (IV) is choosen randomly and shared with the receiver.

② IV is encrypted usin AES and a security key → This generate the first output block.

③ This output block is XORed with the plaintext block to produce the ciphertext.

④ The same output block is used as the input for AES in the next round (not the ciphertext)

⑤. This repeats for ~~ear~~ every blocks.

Next_input = AES(Previous_output)

⑥ Description uses the same output stream to XOR with the ciphertext → gives the original plaintext

Synchronization between encryption and decryption:-

→ Both sender and Receiver uses the same IV and key and perform the same AES encryption

→ Since the same sequence of output blocks is generated at both ends, they remain synchronized.

→ Even if plaintext/ciphertext is lost, they will still stay in sync as long as IV is the same.

Ans 25:

CBC (Cipher Block Chaining) and CFB (Cipher Feedback) modes both causes error propagation.

→ A single bit error in the cipher text will affect multiple blocks in decryption.

In CBC Mode:

→ Each cipher text block is used to decrypt the next plaintext.

→ So, if one cipher text block is corrupted

① The current plaintext block become garbled

② The next block is affected due to XOR with corrupted block

Effect: 1 error affects 2 blocks.

In CFB Mode:

→ Each cipher text block is feedback into the encryption process.

→ If a cipher text block is corrupted

① The corresponding plaintext block become wrong.

→ The next one is also affected.

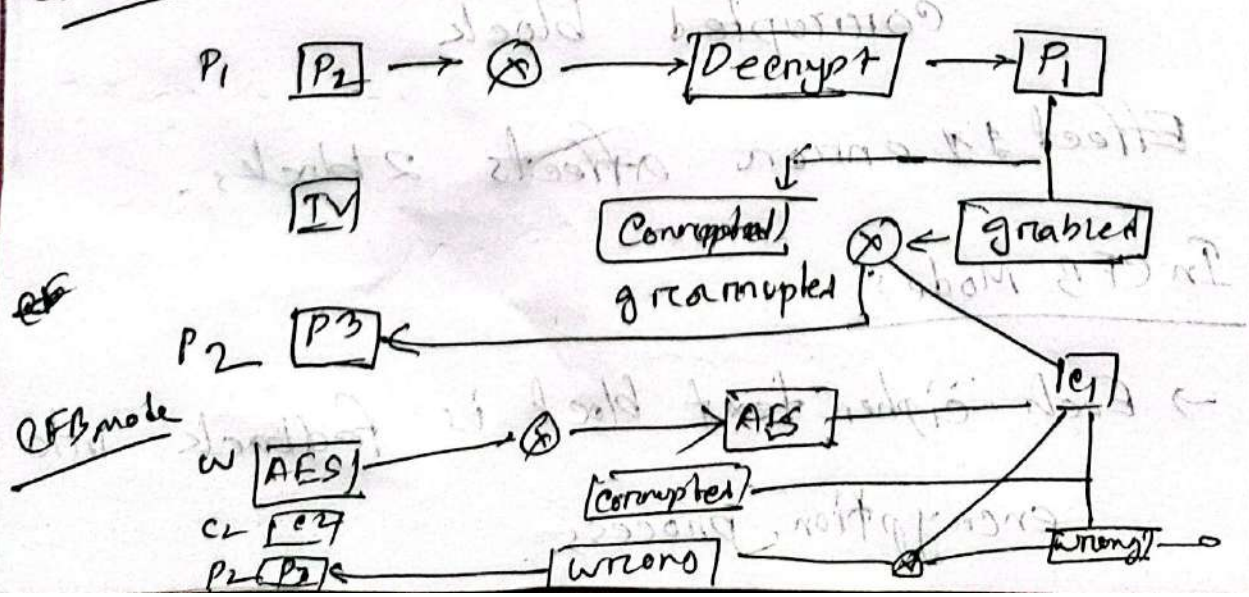
Impact on Integrity:

→ Loss of integrity means wrong plaintext is recovered.

→ Need of Message Authentication Code (MAC) or Authentication Encryption to ensure connection

Error Propagation in AES

CBC Mode



Impact:

Single bit errors in ciphertext causes multiple bit errors in decrypted outputs (CBC/OFB)

Ans 26

Recomanded Mode → AES – CTR (counter Mode)

Justification:

(i) Support Parallel Processing:
Each block is encrypted independently using a counter value; allowing simultanous encryption/decryption

(ii) No block dependency: CTR does not ~~relay~~ rely on previous ciphertext blocks like CBC, so its faster and more efficient.

(iii) Maintain data confidentiality: CTR hides repeating patterns in plaintext, unlike ECB which

leaks structures.

(iv) Ideal for large files: Works best for big files where speed and efficiency matter.

(v) Errors does not propagate: A single bit error affects only the corresponding block, making recovery easier

(vi) Flexible counter based design

Ans 27:

Using the RSA : encryption formula.

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

$$C = 1^5 \bmod 14$$

$$C = 1 \bmod 14$$

Ciphertext C=1

Decryption:

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

$$= 1^{11} \bmod 14$$

$$= 1 \bmod 14$$

$$= 1.$$

Ans:28 Given that, $H(m) = 5$

$d = 3$

$n = 33$

Digital Signature Formula: $S = (H(m))^d \bmod n$

$= 5^3 \bmod 33$

$= 125 \bmod 33$

$S = 26$

∴ Digital signature = 26

Ans 29: Given that,

Prime module $p = 17$

Base (Generator) $g = 3$

Aleya's private key = 4

Babol's private key = 5

Step 1: Aleya's public key =

$A = g^a \bmod p$

$A = 3^4 \bmod 17$

$\therefore A = 13$ ∴ Aleya's public key = 13

Step 2: Badol's public key.

$$B = g^b \bmod p$$

$$= 3^5 \bmod 17$$

$$= 243 \bmod 17$$

$$= 5$$

∴ Badol's public key = 5

Ans 30: Given,

H(m) = (Sum of ASCII values of characters in m) mod 100.

Message "AB"

ASCII of A = 65

ASCII of B = 66

Sum = 65 + 66 = 131

Hash : 131 mod 100 = 31

message "BA"

Sum = 66 + 65 = 131

Hash : 131 mod 100 = 31

This is a collision : two different input produce the same hash output. It shows that weak hash functions are not collision resistant and can easily lead to security vulnerabilities.

Ans 31

Given, Message $M = 15$

Secret key $k = 7$

Formula, $MAC = (M + k) \bmod 17$

Original MAC :

$$MAC = (15 + 7) \bmod 17$$
$$= 22 \bmod 17$$
$$= 5$$

∴ original MAC = 5

if attacker change message to 10. then the mac will be,

$$MAC = (10 + 7) \bmod 17$$
$$= 0.$$

If they guess, probability of connect guess = $\frac{1}{12}$

which is very low.

If attacker changes the message then the formula of $MAC_{new} = (10+k) \bmod 17$.

where k is known,

Now, $10 + k \neq 5 \bmod 17$

$k \neq -5 \bmod 17$

$\Rightarrow k \neq 12$

which is not connect secret key.

Ans '32: TLS = Transport Layer Security used in HTTPS and secure communication.

Main idea: The TLS handshake is how two devices (like browser and server) agree on encryption key and start a secure session. It uses assymetric cryptography (public/private keys) first, then switches to symmetric cryptography for speed.

Step-1: Client Hello

→ The client sends:

i. Supported TLS version
ii. Supported cipher suites (encryption algorithm)
iii. Random numbers (Client Random)
iv. Optional extensions.

Step 2: Server Hello.

→ The server responds with.

i. Choosen TLS version
ii. Selected cipher suites suite
iii. Random number (Server Random)
iv. Digital certificates.

Step 3: Server Authentication.

→ The client verifies:

i. Certificate's validity (via CA)
ii. Domain name matches
iii. Certificate hasn't expired or been revoked

Step 4: key exchange

two common methods:

(1) RSA key exchange (older)

→ Client generates a Pre-master Secret (random number)

→ Encrypts it with the server's public key from the certificate.

- Sends it to the server
- Only the server can decrypt it with its private key.

(b) Diffie-Hellman:

- Both client and server exchange ephemeral public keys
- Using asymmetric math, both sides compute the same

Step 5: Session key Derivation

- Both parties now have
  i. Client Random
  ii. Server Random
  iii. Pre-Master Secret
- They feed these into a key derivation function (KDF)
  i. Symmetric encryption key
  ii. Message authentication keys.

Step 6: Finished Messages.

→ Both client and server send finished messages.

Ans 33

SSH (Secure Shell) has three main layers

1. Transport Layer Protocol
   → Handles encryption, server authentication and integrity
   → Negotiates algorithms and establishes a secure channel.
2. User Authentication Layer
   → Varifies two clients identity (Passwords, public key etc).
3. Connection Layer
   → Manages multiple logical channels over the secure connection.

Ans 34:

Step 1: Client Hello.

→ The client sends

① Supported TLS version

② Supported encryption algorithm.

Step 2: Server Hello

→ The server response with

① Choose TLS version

② Selected ciphen suite.

Step 3: Server Certificate

→ The server sends digital certificates

→ check CA signatures

Step 4: Server key exchange

→ If needed the server sends extra key exchange parameters

Step 5: Client certificate

→ If mutual authentication, the client also

sends its certificates.

Step 6: Key exchange and pre-master secret:

Step 7: Change cipher key

Step 8: Finished message.

Ans 35.

The general form of an elliptic curve equation over a finite field is

$$y^2 \equiv x^3 + ax + b \pmod{p}$$

where,

→ a and b are constants in the field F_p.

→ p is a prime number (for prime fields)

→ The discriminant condition $4a^3 + 27b^2 \neq 0 \pmod{p}$ ensures the curve has no

singularities (no cusps or self-interactions)

Why it is used in Cryptography?

① Strong security per key size: Elliptic curve cryptography (ECC) provides the same security as traditional Algorithms (like RSA) but with a smaller keys, making it faster and more efficient.

② Hard mathematical Problems: The security relies on the Elliptic curve Discrete algorithms problem's (ECDLP) which is extremely hard to solve.

③ Efficient Computation: Suitable for devices with slow-processing power, like IOT devices and mobile systems.

④ Widely Adopted: Use in TLS, digital Signatures (ECDSA) and key exchange (ECDH)

Ans 36:

Elliptic curve cryptography (ECC) achieves the same level security as RSA with much smaller key sizes because it is based on a harder underlying mathematical problem.

Key Reason:

→ RSA security is based on the difficulty of integer factorization.

→ ECC security is based on the Elliptic curve Discrete logarithmic Problem (ECDLP) given two points P and Q = kP on an elliptic curve, it is extremely hard to find k.

Security level Comparison

RSA key Size	ECC key size	Approx Security
1024 bits	160 bits	~ 80 bit Security
2048 bits	224 bits	~ 112 bit Security
3072 bits	256 bits	~ 128 bit security

Ans 37 Give, elliptic curve

$$y^2 \equiv x^3 + 2x + 3 \pmod{97}$$

point P (3, 6)

step 1: check Left Hand Side (LHS)

$$y^2 = 6^2$$
$$= 36 \pmod{97}$$
$$= 36$$

Step 2: Checks Right Hand Side (RHS)

$$= x^3 + 2x + 3$$
$$= 3^3 + 2.3 + 3$$
$$= 27 + 6 + 3$$
$$= 36 \pmod{97} = 36$$

∴ LHS = RHS. So, point (3, 6) lies on the curve.

Ans 36: Given that

$P = 23, g = 5, h = 8$

message $m = 10$

Random $k = 6$

Formula, $C_1 = g^k \bmod p$

$C_2 = m \cdot h^k \bmod p$

Step 1:

$C_1 = 5^6 \bmod 23$

$= (5^2)^3 \bmod 23$ — (1)

$\therefore 5^2 = 25 \bmod 23$

$= 2$

from (1) $C_1 = 2^3 \bmod 23$

$= 8 \bmod 23$

$e_1 = 8$

Step 2: $C_2 = m \cdot k^k \bmod p = 10 \cdot 8^6 \bmod 23$

$= 10 \times (8)^2 \bmod 23 = 10 \times 36 \bmod 23$

$= 360 \bmod 23 = 15$

$\therefore C_2 = 15$

$\therefore (C_1, C_2) = (8, 15)$

Ans-39

Light weight cryptography is designed to use low memory, low processing power and minimal energy, making it suitable for resource-constrained IOT devices.

Importance:

→ IOT devices have limited CPU and battery can not handle heavy algorithm like RSA easily.

→ Ensures data confidentiality, integrity and authentication without overloading the device.

Example:

Present: A block cipher with 64 bit block size and 80/128 bit key, optimized for low hardware area and low energy usage

Ans 40:

Common IOT specific attacks and mitigation.

1. Firmware attack:

→ Attackers replaces legitimate firmware with malicious code

Impact: Can control the device remotely or steal data

Mitigation: Secure boot, firmware, signing and version verification before updates.

2. Physical Tampering

→ Direct manipulation of IoT hardware (opening devices, probing chips).

Impact: Can extract encryption keys or modify circu

Mitigation: Tamper-evident seals, hardware, encryption, protective casing.

3. IoT Botnets (Mirai)

→ Malware infects IoT devices and uses them in large scale DDos attacks

Impact: Disrupts services, cause internet outag