Question-1: With the advent of quantum computing, traditional public-key cryptosystems such as RSA and Ecc are potentially vulnerable to shor's algorithm.

Solution:

The implications of quantum computing on the security of cryptographic protocols:-

1. RSA Vulnerability: RSA security based on the difficulty of factoring large prime numbers.

2. Ecc Vulnerability: Ecc relies on the hardness of the Elliptive Curve Discrete Logarithm Problem (ECDLP which shor's algorithm can also solve efficiency.

3. Impact on Digital signature's and key Exchange:

Protocols such as TLS, SSH and PGP rely on RSA and Ecc for secure communication.

Post-Quantum Cryptographic Algorithms:

1. Latice-Based Cryptography:

→ Example Algorithms: Kyber (key encapsulation),

Dilithium (Digital signature), NTRU Encrypt and NTRUSign.

→ Security Basis: These algorithms rely on the hardness of problems such as the Learning with Errors (LWE) problem and the Shortest Vector Problem (SVP).

→ Why Quantum Resistant? :- Shor's algorithm does not provide on efficient way to solve lattice problems. The best-known quantum attacks require exponential time.

2. Code-Based Cryptography:

→ Example Algorithm: McEliece Cryptosystem, BIKE (Bit-flipping key Encapsulation).

→ Security Basis: These schemes rely on the hardness and decoding general linear error-correcting code.

→ Why Quantum-Resistant: No efficient quantum algorithm exists to solve the decoding problem in random linear codes.

3] Hash-Based Cryptography:

→ Example Algorithms: SPHINCS+ (Stateless Hash-Based Signature

→ Security Basis: Use cryptographic hash functions.

→ Why Quantum-Resistant? :- Hash functions can be mad quantum-resistant by using larger output sizes to counter Grover's speed up.

4] Multivariate Polynomial Cryptography:

→ Example Algorithms: Rainbow (multivariate quadratic signature scheme).

→ Security Basis: Relies on solving systems of multivariate polynomial equations.

→ Why Quantum Resistant?: No known quantum algorithm efficiently solves the general multivariate polynomial syste

5] Isogeny-Based Cryptography:

→ Example Algorithms: SIKE (Supersingular Isogeny key Encapsulation)

→ Security Basis: Use the difficulty of finding isogenies.

→ Why Quantum Resistant?: Unlike RSA and ECC the structure of isogeny problems makes them resistant to Shor's Algorithm.

Question2:

Solution: Here's a Python implementation of a novel Pseudo-Random Number Generator (PRNG) that utilizes
→ Current timestamp (time.time())
→ Process ID (os.getpid())
→ Modulus operation to constrain the output within a given range.

Implementation of a custom PRNG in Python:

```
import time
import os
class CustomPRNG:
def __init__(self, seed=None):
    if seed is None:
    seed = int(time.time()*1000000) ^ os.getpid()
        self.state = seed
    def next(self, min_value=0, max_value=100):
        a=1664525
        b=1013904223
        m = 2**32
        self.state = (a*self.state+c) % m
        return min_value + (self.state % (max_value-
                                min_value+1))
        prng = CustomPRNG()
        Print("Random Number:", prng.next(1,100))
```

How this PRNG Works:

→ Uses the current timestamp and process ID for randomness.

→ Applies a linear congruential Generation (LCG) with secure parameters.

→ Uses a modulus operation to constrain output within a specific range.

Question-3:

Sol^{n}: Comparison between traditional ciphers (such as the Caesar cipher, Vigenere cipher, playfair cipher) with modern symmetric ciphers like AES and DES:

Aspect	Traditional cipher (Caesar, vigenere, playfair)	Modern cipher (AES and DES)
Key length	Very short (≤ 25 for Caesar, ~100 for vigenere)	Long (AES: 128-256 bits
Speed	Extremely fast	Fast but require more computation.
Security	Weak	Strong
Cryptanalysis Resistance	Vulnerable to brute force frequency analysis.	Resistant to known attacks.

Question-4:

Solution:

Definition of the Action:

The symmetric group S_4 acts on the set of 2-element subsets of $X = \{1, 2, 3, 4\}$ denoted as:

$Y = \{\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\}\}$.

For $\sigma \in S_4$ and $A = \{a, b\} \in Y$, the action is defined as:

$$\sigma \cdot \{a, b\} = \{\sigma(a), \sigma(b)\}$$

Well-Definedness:

Since S_4 consists of bijections any σ maps A to another 2-element subset of X, ensuring the action remains within Y.

Orbit size of $\{1,2\}$:

The total number of 2-element subsets of X is:

$$\binom{4}{2} = \frac{4!}{2!\,(4-2)!} = 6$$

Since S_4 acts transitively, the orbit of $\{1,2\}$ has 6 elements.

Stabilizer of $\{1,2\}$:

The stabilizer consists of permutations that keep $\{1,2\}$ fixed, These are:

$$\{e, (1,2), (3,4), (1,2)(3,4)\}$$

forming a subgroup isomorphic to the klain four group V_4 with order 4.

Verification via orbit-stabilizer Theorem:

$$|S_4| = |\text{orbit}(\{1,2\}) \times |\text{stab}(\{1,2\})|$$

$$= 6 \times 4$$

$$= 24$$

Thus the results are correct.

Question-5:

Soln

Construction of $GF(2^2)$:

The finite feild $GF(2^2)$ consists of four elements:

$$GF(2^2) = \{0, 1, \alpha, \alpha+1\}$$

Where α is a root of the irreducible polynomial x^2+x+1 over $GF(2)$. These means:

$$\alpha^2+\alpha+1 = 0$$

$$\Rightarrow \alpha^2 = \alpha+1.$$

i) Proving that $GF(2^2)$ forms a multiplicative group:

The nonzero elements of $GF(2^2)$ are:

$$G = \{1, \alpha, \alpha+1\}$$

To show that G forms a group under multiplication, we check the group property.

1. Closure:

Multiplying any two elements in G_F results in another element in G_F.

→ $1 \cdot \alpha = \alpha$, $1 \cdot (\alpha+1) = \alpha+1$

→ $\alpha \cdot (\alpha+1) = \alpha^2 + \alpha = (\alpha+1) + \alpha = 1$

→ $\alpha \cdot \alpha = \alpha^2 = \alpha+1$

2. Associativity:

Multiplication in feilds is always associative.

3. Identity Element:

The identity element is 1 as $1 \cdot \alpha = \alpha$ for all $\alpha \in G_F$.

4. Inverse:

→ $1^{-1} = 1$

→ $\alpha^{-1} = \alpha+1$, since $\alpha \cdot (\alpha+1) = 1$

→ $(\alpha+1)^{-1} = \alpha$, since $(\alpha+1) \cdot \alpha = 1$

Since all group properties hold, G_F forms a multiplicative group.

ii. Checking whether G_F is cyclic:

A group is cyclic if there exists an element $g \in G_F$ such that every other element can be written

as g^n for some n. We check for a generator:

1. check α:
   → $\alpha^1 = \alpha$
   → $\alpha^2 = \alpha+1$
   → $\alpha^3 = (\alpha+1).\alpha = \alpha^2+\alpha = (\alpha+1)+\alpha = 1$

→ Since $\alpha^3 = 1$, it cycles through all elements of G_F, proving α is a generator.

Since there exists a generator (α), G_F is cyclic.

Question-6:

Sol^n:

Step1: Definition of $G_F L(2,R)$

The general linear group $GL(2,R)$ consists of all invertible 2x2 matrices with real entries:

$$GL(2,R) = \{A \in M_{2x2}(R) \mid \det(A) \neq 0\}$$

Step2: Scalar matrices as a subgroup

A scalar matrix is of the form:

$$S = \lambda I = \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

where $\lambda \neq 0$ (to ensure invertibility). The set o all such matrices is:

$$S = \{\lambda I \mid \lambda \in R^*\} \cong R^*$$

This forms a subgroup because:

→ Closure: If $S_1 = \lambda I$ and $S_2 = \mu I$, then $S_1 S_2 = (\lambda\mu)I$ which is still a scalar matrix.

→ Identity: The identity matrix I belongs to S.

→ Inverses: The inverse of λI is $(\lambda^{-1})I$, which is also in S.

Thus, S is a subgroup of GL(2,R)

step3: Normality of S in GL(2,R)

To show that S is normal, we check whether it is invariant under conjugation.

For any $A \in GL(2,R)$ and $S = \lambda I$:

$$ASA^{-1} = A(\lambda I)A^{-1} = \lambda(AIA^{-1}) = \lambda(AA^{-1}) = \lambda I = S$$

Since conjugation by any matrix in GL(2,R) keeps the scalar matrices unchanged, S is a normal subgroup.

step4: Constructing the factor Group GL(2,R)/S.

The factor group GL(2,R)/S consists of cosets of S in GL(2,R). Each coset is of the form:

$$AS = \{A \cdot \lambda I \mid \lambda \in R^*\}$$

Each coset represents the equivalence class of A up to scalar multiples.

Steps: Interpretation of the factor Group Structure

→ The factor group $GL(2,R)/S$ represents the group of all invertible 2×2 matrices modulo scalar matrices.

→ This group is called the projective General linear Group $PGL(2,R)$, which describes transformation in projective geometry.

→ It captures the structure of linear transforming ignoring uniform scaling.

Question-7:

Soln:

Diffie-Hellman key Exchange Protocol:

Definition:

The Diffie-Hellman key exchange is a cryptographic protocol that allows two parties to securely establish a shared secret key over an insecure communication channel.

Steps of the protocol:

1. Public parameters:

→ Choose a large prime number p and a primitive

root g (also called a generator) modulo p.

⇒ These values are publicly known.

2. key exchange process:

→ Alice chooses a private key a (random integer) and computes her public key:

$$A = g^a \bmod p$$

She sends A to Bob

→ Bob chooses a private key b and computes his public key:

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

He sends B to Alice.

3. Shared secret computation:

→ Alice computes:

$$s = B^a \bmod p = (g^b \bmod p)^a \bmod p = g^{ab} \bmod p$$

→ Bob computes:

$$s = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p$$

→ Since both compute $g^{ab} \bmod p$, they now share a common secret key.

Applications in secure communication:

→ Used in SSL/TLS for secure web browsing (HTTP

→ Enaryption of message in communication protocols (eg. signal, whatsapp)

→ secure key agreement in VPNs and cryptographic protocols.

Security of Diffie-Hellman Protocol:

① Role of Discrete Logarithm Problem (DLP)

→ The security relies on the DLP, which makes it infeasible to compute the secret key from public values.

→ This prevents passive eavesdropping.

② Common Attacks and migation:

→ Man-in the-middle (MITM) Attack: Attacker intercepts key exchange and establishes separate keys with both parties.

Solution: Use authentication (digital signatures certificate

→ Eavesdropping Attack: Attackers listens but cannot compute the shared key.

Solution: Use a large prime p (2048 bit or higher)

→ Pre-computation Attack: Reusing prime p allows precomputed attacks.

Solution: Use Elliptic Curve Diffie-Hellman (ECDH) for stronger security.

DLP ensures security but MITM attacks require authentication measures for protection.

Impact of using a small Prime Modulus in Diffie-Hellman Protocol:

1. Weakened security: A small prime p makes the discrete logarithm problem (DLP) solvable in polynomial time.

2. Vulnerability to Brute-force Attacks: If p is small, an attacker can perform an exhaustive search to find the private keys quickly.

Question-8:

Soln: Let G be a group, and let H and K be two subgroups of G. We need to show that $H \cap K$ is also a subgroup of G.

1. Non-emptiness:

Since H and K are subgroups of G, they both contain the identity element e. Thus, $e \in H \cap K$, so $H \cap K \neq \emptyset$.

2. Closure under the group operation:

If $a, b \in H \cap K$, then a, b are in both H and K. Since H and K are subgroups, their product ab is in both H and K, so $ab \in H \cap K$.

3. Closure under inverses:

If $a \in b$ $H \cap K$, then a is in both H and K. Since H and K are subgroups, a^{-1} is in both H and K, so $a^{-1} \in H \cap K$.

Since $H \cap K$ satisfies the subgroup properties, it is a subgroup of G.

Example:

Consider the group $\mathbb{Z}$ under addition:

→ Let $H = 2\mathbb{Z} = \{\ldots, -4, -2, 0, 2, 4, \ldots\}$ (even integers)

→ Let $K = 3\mathbb{Z} = \{\ldots, -6, -3, 0, 3, 6, \ldots\}$ (multiple of 3)

Their intersection is:

$H \cap K = 6\mathbb{Z} = \{\ldots, -12, -6, 0, 6, 12, \ldots\}$

which is also a subgroup of $\mathbb{Z}$.

Thus the intersection of two subgroups is always a subgroup.

Question-9:

Soln:

Proof that Z_n is commutative:

The ring Z_n consists of the equivalence classes of integer modulo n, with, addition and multiplication defined as:

→ Addition: $[a]+[b]=[a+b]$

→ Multiplication: $[a].[b]=[a.b]$

Since integers addition and multiplication are commutative we have;

$$[a]+[b]=[b]+[a],$$

$$[a].[b]=[b].[a]$$

Thus, Z_n is a commutative ring.

Zero divisors in Z_n:

An element $[a] \in Z_n$ is a zero division if there exists a nonzero element $[b] \in Z_n$ such that;

$$[a].[b]=[0]$$

This means $a.b=0 \pmod n$, but neither a nor b is congruent to 0 modulo n.

→ If n is composite, then Z_n has zero divisors. For example, in Z_6 we have:

$$[2].[3]=[6]=[0] \pmod 6,$$

so [2] and [3] are zero divisors.

If n is prime, there are no zero divisors because that only solutions to $a \cdot b = 0 \pmod{n}$ require either $a=0$ or $b=0$.

Conditions for Z_n to be a field:

A ring Z_n is a field is every nonzero element has a multiplicative inverse. This happens if and only if Z_n has no zero divisors, which is true if and only if n is prime.

→ If n is prime - every nonzero element $[a]$ has a multiplicative inverse $[b]$ ~~out~~ such that:

$$[a] \cdot [b] = [1].$$

Thus, Z_n forms a field.

→ If n is composite, Z_n contains zero divisors, so it cannot be a field.

Question. 10

Sol^n: Vulnerabilities of the DES (Data Encryption Standard) cipher:

→ Small key size (56-bit key):

DES uses a 56-bit key, which is too short by modern security standards.

→ Brute-force Attacks:

Modern hardware can perform exhaustive key searches rapidly.

→ Susceptibility to cryptanalysis:

DES is vulnerable to differential cryptanalysis and linear cryptanalysis.

→ Weak and semi-weak keys:

There are four weak keys and twelve semi-weak key pairs that reduce security.

→ Meet-in-the-middle Attack:

It is vulnerable to the meet-in-the-middle attack

Why DES is considered Insecure for modern use:

→ Easily broken using modern computing powers.

→ Brute-force attacks can be performed in hours or minutes.

→ Advanced cryptanalysis techniques make DES ineffective again sophisticated attacks.

→ Stronger alternatives like AES (Advanced Encryption Standard) and Triple DES (3DES) provide much better security.

Role of Brute-force Attacks in Breaking DES:

A brute-force attack is a method of breaking an encryption algorithm by trying all possible keys until the correct one is found.

→ DES uses a 56-bit key:

The key space consists of 2^{56} possible keys.

→ Real-World Brute-force Attacks on DES:

In 2006, advanced computes could break DES in just a few minutes.

→ Distributed and cloud computing Threats:

Today, cloud-based computing and FPGAs/GPU, can

Perform parallel brute-force attacks.

Impact of key length on security:

1. Shorter keys are Easier to Break:

→ A 56-bit DES key can be brute-forced quickly with modern computing powers.

→ Increase key length exponentially increases security.

2. Comparision of key lengths:

→ DES (56-bit): Insecure, breakable in hours or ~~mit~~ minutes.

→ Triple DES: more secure but slower.

→ AES-128: Secure; requires 2^{128} attempts, 128-bit key making brute-force in feasible.

→ AES-256: Extremely secure; even with the most powerful computes.

How AES Addressed the shortcomings of DES:

AES overcomes the shortcomings of DES in the following ways:

1. Increased key size:

→ DES uses a 56-bit key, which is vulnerable to brute force attacks.

→ AES supports three key sizes:

- AES-128 (128 bit key)
- AES-192 (192-bit key)
- AES-256 (256-bit key)

2. Improved Security Against Cryptanalysis:

→ AES uses a more complex structure with substitution-permutation network instead of the feistel structure in DES,

→ AES has more rounds (10, 12, 14) compared to DES's 16 rounds, increasing resistance to attacks.

3. Resistance to Brute-forced Attacks

→ AES-128 requires 2^{128} operations

→ AES-256 is even more secure, requiring 2^{256} operations.

4. No weak or semi-weak keys
→ AES does not have such key weakness, ensuring consistent security.
5. Faster and more efficient than DES.

Question-11:

Soln:

i) How the feistel structure of DES Handles Differential Cryptanalysis:

→ DES uses 16 rounds, making it harder to track input differences.
→ S-Boxes ~~inst~~ introduce non-linearity, resisting differential attacks.
→ The avalanche effect ensures small input changes cause major output changes.
→ However, DES's 56-bit key and improved cryptanalysis techniques make it vulnerable.

ii) Why AES is more Resistant to Differential Cryptanalysis than DES:

→ SubBytes: Strong S-boxes provide better non-linearity.

→ ShiftRows and MixColumns: Spread differences across the block, increasing diffusion.

→ AddRoundKey: combines key at each round, preventing attacks.

→ Larger Block and key size:
More secure than DES.

Question-12:

Soln:

Finding the Modular inverse Using the Extended Euclidean Algorithm:

The modular inverse of a modulo n exists if and only if $\gcd(a, n) = 1$. If is the integer x such that:

$$ax \equiv 1 \pmod{n}$$

We use the Extended Euclidean Algorithm to find this inverse:

Steps:

1. Apply the Euclidean Algorithm to find $\gcd(a, n)$
2. Use the Extended Euclidean Algorithm to express $\gcd(a, n) = 1$ in the form:

$$ax + ny = 1$$

for some integers x and y.

3. The value of $x \pmod{n}$ is the modular inverse of a modulo n.

Example: Find the modular inverse of 3 modulo 26.

1. Apply Euclidean Algorithm:

$$26 = 3(8)+2$$
$$3 = 2(1)+1$$
$$2 = 1(2)+0$$

Since, gcd $(3,26) = 1$, an inverse exists.

2. Back-substituting to express 1 as a linear combinations:

$$1 = 3-1(2)$$
$$1 = 3-1(26-8(3))$$
$$1 = 3(9)-26(1)$$

So, $x=9$ is the modular inverse.

3. Final answer:

$$3^{-1} \equiv 9 \pmod{26}$$

The modular inverse of 3 modulo 26 is 9.

Use of the Extended Euclidean Algorithm in RSA key Generation:

1. RSA key generation requires finding the modular inverse of e modulo $\phi(n)$.

2. The public extent e is chosen such that gcd $(e, \phi(n)) = 1$.

3. The extended Euclidean Algorithm is used to compute the modular inverse of e, which gives the private key d, s.t satisfying:

$$ed \equiv 1 \pmod{\phi(n)}$$

Importance of Efficiency in cryptographic systems:

→ RSA involves large prime numbers (1024-bit, 2048 bit

→ The extended Eualidian Algorithm runs in polynomia time $O(\log n)$ ensuring fast key generation.

→ Efficient modulerc inversion is crucial for decryption and signing operation in RSA

Question-13:

(i) ECB mode Insecurity for Highly Redundant data:

→ In Electronic Code Book (ECB) mode, each plaintext block P_i is encrypted independently:

$$C_i = E_K(P_i)$$

→ If two plaintext blocks are identical ($P_i = P_j$) then their ciphertexts are also identical ($C_i = C_j$)

⇒ This lack of diffusion makes ECB insecure for highly redundant data, as patterns in the plaintext appear in the ciphertext, allowing attackers to recognize structure.

(ii) Recurrence Relation for CBC Mode and Error Propagation in Decryption:

→ Encryption in CBC mode:

Each plaintext block P_i is XORed with the previous ciphertext block before encryption:

$$C_i = E_K(P_i \oplus C_{i-1})$$

where $C_0 = IV$ (Initialization vector)

→ Decryption in CBC Mode:

$$P_i = D_K(C_i) \oplus C_{i-1}$$

• A single-bit error in ciphertext C_i affects only P_i after decryption.

• However, an error in C_{i-1} affects both P_i and P_{i+1}, but not further blocks.

Question-14

Soln.

Vulnerability of LFSRs to known-Plaintext Attacks:

→ Linear feedback shift Registers (LFSRs) generate keystreams based on linear recurrence relations:

$$S_n = c_1 S_{n-1} \oplus c_2 S_{n-2} \oplus \cdots \oplus c_m S_{n-m}$$

→ Since LFSRs are linear, attackers can collect enough keystream bits and solve a system of linear equations to recover the feed back coefficients, breaking the cipher.

→ Berlekamp-massey algorithm can efficiently reconstruct the LFSR sequence from a known a plaintext attack.

Mathematical Method to mitigate this vulnerability:

→ Use nonlinear combination functions: Instead of single LFSR, use multiple LFSRs combined through a nonlinear function $f(s_1, s_2 \ldots, s_n)$

→ Geffe Generator: combines three LFSRs using Boolean functions to introduce non-linearity, making attacks more difficult.

→ Clock-controlled, LFSRs: use irregular clocking mechanisms to disrupt predictability.

Question-15:

Soln:

① Shannon's Definition of Perfect Secrecy:

A cryptographic system achives perfect secrecy if the ciphertext C reveals no information about the plaintext M, ie, the probability distribution of m remains unchanged even after observing C:

$$P(m/c) = P(M)$$

for all mem and $c \in C$

Equivalently, in terms of conditional probability:

$$P(c|M) = P(c)$$

for all plaintext and ciphertext.

(ii) Proof that the one-Time-Pad achieves perfect secrecy:

Given conditions:

→ key k is uniformly random

→ $|k| \geq |M|$

→ Encryption $c = m \oplus k$

→ Decryption $M = c \oplus k$

Question - 16

Soln:

Computation of the first 5 Numbers in an LCG sequence:

Given the LCG recurrence relation:

$$x_{n+1} = (a x_n + c) \bmod m$$

Lets choose specific values:

→ Multiplier $a = 5$

→ Increment $c = 3$

→ Modulus $m = 16$

→ Seed $x_0 = 7$

Computing the first 5 numbers:

1. $x_1 = (5 \times 7 + 3) \bmod 16 = (35 + 3) \bmod 16$
$= 38 \bmod 16 = 6$

2. $x_2 = (5 \times 6 + 3) \bmod 16 = (30 + 3) \bmod 16$
$= 33 \bmod 16 = 1$

3. $x_3 = (5 \times 1 + 3) \bmod 16 = 8 \bmod 16 = 8$

4. $x_4 = (5 \times 8 + 3) \bmod 16 = 43 \bmod 16 = 11$

5. $x_5 = (5 \times 11 + 3) \bmod 16 = 58 \bmod 16 = 10$

final LCG sequence: 7, 6, 1, 8, 11, 10

Question: 17

Soln: Definition of a Ring in Abstract Algebra:

A ring $(R, +, \cdot)$ is an algebraic structure consisting of a set R equipped with two binary operations: addition (+) and multiplication (·), satisfying the following properties:

Key properties of a Ring:

1. Additive closure: If $a, b \in R$, then $a+b \in R$
2. Associativity of Addition: $(a+b)+c = a+(b+c)$ for all $a, b, c \in R$
3. Additive Identity: There exists an element $0 \in R$ such that $a+0 = a$ for all $a \in R$
4. Additive Inverse: For each $a \in R$, there exists $a \in R$ such that $a+(-a)=0$
5. Commutativity of Addition: $a+b = b+a$ for all $a, b \in R$.
6. Multiplicative closure: If $a, b, \in R$;then $a \cdot b \in R$
7. Associativity of Multiplication: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for all $a, b, c \in R$.

8. Distributive laws:

→ $a \cdot (b+c) = a \cdot b + a \cdot c$

→ $(a+b) \cdot c = a \cdot c + b \cdot c$

Examples of Rings:

1. Commutative Ring

→ Example: The set of integers $\mathbb{Z}$ with standard addition and multiplication.

→ Reason: For all $a, b \in \mathbb{Z}$, $a \cdot b = b \cdot a$ satisfying commutativity.

2. Non-Commutative Ring:

→ Example: The set of 2×2 matrices over $\mathbb{R}$ (denoted $m_2(\mathbb{R})$)

→ Reason: matrix multiplication is not commutative i.e. there exist matrices A, B such that $AB \neq BA$.

Question:18

Sol":

Given values:

1. $p = 5, q = 11$

2. Compute n:

$$n = p \times q = 5 \times 11 = 55$$

3. Compute Euler's totient function:

$$\phi(n) = (p-1)(q-1) = (5-1)(11-1) = 4 \times 10 = 40$$

4. Choose public exponent $e = 3$

5. Compute private key d using $d = e^{-1} \bmod \phi(n)$:

→ Using Extended Euclidian Algorithm,

$d = 27$ since $3 \times 27 \equiv 1 \bmod 40$

Encryption:

$$c = m^e \bmod n = 2^3 \bmod 55 = 8$$

ciphertext : $c = 8$

Decryption:

$$m = c^d \bmod n = 8^{27} \bmod 55$$

Using modular exponentiation,

$$m = 2$$

Recovered message : 2

Digital signature Example:

Given values:

1. $p=7, q=3$

2. Compute n:

$n=7\times3=21$

3. Compute $\phi(n)$:

$\phi(n)=(7-1)(3-1)=12$

4. Choose $e=5$, compute d using Extended Euklidian Algorit

→ $d=5^{-1} \bmod 12 = 5$ (since $5\times5\equiv1 \bmod 12$).

Signing the Hash:

$S=H(m)^d \bmod n = 3^5 \bmod 21 = 243 \bmod 21 = 12$

Signature: $S=15$

Verification:

$V=s^2 \bmod n = 12^5 \bmod 21 = 248832$

$\bmod 21$

$=3$

$V=H(m)$. the signature is valid.

Question: 19

Elliptive curve over Z_p with $p=23$, $a=1$, $b=1$

Equation:

$$y^2 \equiv x^3+ax+b \bmod 23$$

i.e, $y^2 = x^3+x+1 \bmod 23$

i. Verify if $P=(3,10)$ lies on the curve

Substuting $x=3, y=10$ into the equation:

$$10^2 = 3^3+3+1 \bmod 23$$

$$100 \equiv 27+3+1 \equiv 31 \equiv 100 \bmod 23$$

Since both sides are equal, $P=(3,10)$ lies on the curve.

Question-20

Elliptive curve Digital signature Algorithm calculation:

Given elliptic curve:

$$y^2 \equiv x^3 + 7x + 10 \bmod 37$$

Base point : $G = (2,5)$, order $n = 19$

Private key : $d = 9$

Random nonce : $k = 3$

Hash of message : $H(m) = 8$

(i) Compute public key Q

$$Q = d G = 9G$$

$$Q = (24, 29)$$

Public key : $Q = (24, 29)$

(ii) Compute signature pair (r, s)

1. Compute kG :

$$3G = (35, 17)$$

2. Compute r : $r = x \bmod n = 35 \bmod 19 = 16$

3. Compute s : $s = k^{-1}(H(m) + dr) \bmod n$

Question-21

Soln: Key Properties of Cryptographic Hash Functions (SHA Family):

i) Essential Characteristics of a secure Hashfunction:

1. Pre-image Resistance - Given a hash output H(m), it is computationally infeasible to find the original message m.

2. Second Pre-Image Resistance - Given a message M_1 it is hard to find another message M_2 such that $H(M_1) = H(M_2)$.

3. Collision Resistance - It is infeasible to find two different message M_1 and M_2 such that

$$H(M_1) = H(M_2)$$

4. Avalanche Effect - A small change in the input causes a significantly different hash output.

5. Deterministic - The same input always produces the same output.

ii) Impact of Hash length on security:

→ The output length determines the resistance to brute-force and collision attacks.

→ A longer hash reduces the probability of collisions due to the Birthday Paradox.

→ SHA-256 provides 2^{128} collision resistance, making it secure against current computational capabilities

(iii) Real-World Applications of SHA:

1. Digital signatures: Hash functions ensure message integrity before signing with a private key.
2. Blockchain systems: SHA-256 secures transsmission and links block via hash pointers,
3. Password Hashing: Ensures secure password storage using salted hashes.
4. Integrity Verification: Used in checksums and digital forensics to detect modifications.

Question-22

Soln: Concept of Galois fields (Finite Feilds) GF(p) and GF(2^n) Definition:

1. GF(p) (Prime field) - A finite field with p elements, Where p is a prime number. Arithmatic operations are performed modulo p.

2. GF(2^n) (Binary field) - A field with 2^n elements commonly used in cryptography. Arithmetic is does using polynomials over GF(2) modulo an irreducible polynomial.

Use in crytographic Primitives

1. Elliptic curve cryptography (ECC)
   - Points on eliptic curves are defined over GF(2^n)
   - Provides security with smaller key sizes compared to RSA.

2. AES (Advanced Encryption standard)
   - AES operates in GF(2^8) using field arithmetic for mix columns and subBytes transformations.

- Ensures efficient and secure encryption.

Importance of Field Arithmetic:

- Ensures security - Provides a structured and secures way to perform operations in cryptographic algorithm.
- Efficiency: Enables fast computations, especia in hardware implementations.
- Error Resistance: Used in error detection/ correction codes like Reed-solomon codes.

Question-23

Soln: Lattice-Based cryptography and post-Quantum security -

i) Shortest vector Problem (SVP) and security Role

- The shortest vector Problem (SVP) involves finding the shortest nonzero vector in a given lattice.
- It is computationally hard, even for quantum computers, making it a strong security foundation for lattice-based cryptography.
- Learning with Errors (LWE) and Ring-LWE rely on SVP for security.

ii) Comparison with RSA and Ecc in shor's Algorithm context.

- RSA and Ecc rely on integer factorization and discrete logarithm problems, both efficiently solvable by shor's Algorithm on a quantum.

• Lattice-based cryptography, based on SVP and LWE remains hard for both classical and quantum computers, making it post-quantum secure.

iii) Quantum cryptography Vs Lattice-Based Cryptography:

Aspect	Lattice-Based cryptography	Quantum Cryptography
Goal	Secure against quantum attacks	Uses quantum mechanics for security.
Principle	Based on hard mathematical problems (SVP, LWE)	Uses quantum principles like entanglements and superposition.
Example	NTRU, kyber (Post-Quantum Encryption)	Quantum key Distribution (QKD) like BB84.
Implementation	Works on classical computers	Requires quantum communication channels.

Question - 21

Soln: Proof of maximum Period of an LFSR key stream:

LFSR Recurrence Relation:

A linear Feedback shift Register (LFSR) generates the keystream using:

$$k_t = c_1 k_{t-1} \oplus c_2 k_{t-2} \oplus \cdots \oplus c_m k_{t-m}$$

over $GF(2)$, where $\oplus$ represents XOR.

key Idea: characteristic Polynomial and Maximum Period:

→ The characteristic polynomial of the LFSR

$$P(x) = x^m - c_1 x^{m-1} \cdots c_m.$$

→ If $P(x)$ is primitive over $GF(2)$, then the sequence generated by the LFSR is maximal length, meaning it cycles through all possible nonzero states before repeating.

Derivation of Maximum Period:

1. An LFSR with m-bit state has 2^m possible states
2. The all-zero state is not allowed.
3. A primitive polynomial ensures that the LFSR cycles through all $2^m - 1$ possible nonzero states before it repeating.

Question-25:

Lattice-Based cryptography and [WE-Based signature scheme]

i) Process of signing a message using

1. Key generation

- Generate a public key PK and Private key SK.
- Private key: s_k is a secret vector chosen fromala ve.
- Public key: PK is derived from s_k usually by creati

2. Signing the Message

- Message hashing: Hash the message M to obtain a fixed-sized value h(M).
- Signature generation.

3. Verification

- Signature Verification
- The verifier uses PK to check that the signature o' is valid for the message M and the corres.
- If the signature matches the expected value (based on LWE) the message is authentic.

ii) Signing a Message Using LWE-Based Scheme:

1. Private key and public key setup

→ Let the private key sk be a vector is a lattice and the public key Pk be a noisy lattice vector derived using the LWE problem.

2. Message signing process

→ Given Message M, connects its has $h(M)$.

→ Uses the private key sk and solve the LWR problem to generate a signature and for $h(M)$

→ This insists solving a system of linear equations where the noise (errors) is controlled, : ensuring the difficulty of solving the problem for an advergency.

3. Role of LWE problem is security

→ The LWE problem ensures security by making it computationally hard for an adversary to forge signatures without knowledge of the private key sk, even if they know the public key pk.

→ It provides semantic security because solving the LWE problem is believed to be hard for both classical and quantum computers.