

Question:

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

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

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

② ECC vulnerability : ECC relies on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP), which Shor's algorithm can also solve efficiently.

③ Impact on Digital Signatures and Key Exchange: protocols such as TLS, SSH and PGP rely on RSA and ECC for secure communication.

post-Quantum Cryptographic Algorithms :-

① Lattice-Based Cryptography :-

- Example Algorithms : Kyber (key encapsulation), Dilithium (digital signatures), NTRU Encrypt and NTRU Sign.
- 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 an efficient way to solve lattice problems. The best-known quantum attacks require exponential time.

② Code-Based Cryptography :-

- Example Algorithms: McEliece (cryptosystem), BIKE (Bit-flipping key Encapsulation).
- Security Basis: These schemes rely on the hardness of decoding general linear error-correcting codes.
- 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 Signatures)
- Security Basis:- Use cryptographic hash functions.
- Why Quantum-Resistant? :- Hash functions can be made quantum-resistant by using larger output sizes to counter Grover's speedup.

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 system.

⑤ Isogeny-Based Cryptography:

- Example Algorithms: SIKE (Supersingular Isogeny Key Encapsulation)
- Security Basis: Uses the difficulty of finding isogenies.
- Why Quantum-Resistant? : Unlike RSA and ECC, the structure of isogeny problems makes them resistant to Shor's algorithm.

Question:

② 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 + b) % m
        return min_value + (self.state % (max_value - min_value))
```

```
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 Comparison between traditional ciphers (such as the Caesar cipher, Vigenere cipher, playfair cipher) with modern symmetric ciphers like AES and DES:-

Aspect	Traditional Ciphers (Caesar, vigenere, playfair)	Modern Ciphers (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:

①

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:

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

For $\sigma \in S_4$ and $A = \{a, b\} \in \mathcal{Y}$ - 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 \mathcal{Y} .

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

The total number of 2-element subsets of X is: $\binom{4}{2} = \frac{4!}{2!(4-2)!}$

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 Klein 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: 5Construction of $GF(2^n)$:-

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

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

where α is a root of the irreducible polynomial

$x^2 + x + 1$ over $GF(2)$. These means:-

$$\begin{aligned}x^2 + x + 1 &= 0 \\ \Rightarrow \alpha^2 &= \alpha + 1.\end{aligned}$$

i) proving that $GF(2^n)$ forms a multiplicative group :-

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

$$G_1 = \{1, \alpha, \alpha + 1\}$$

To show that G_1 forms a group under multiplication, we check the group properties.

① Closure:-

Multiplying any two elements in G_1 results in another element in G_1 .

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

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

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

② Associativity:

multiplication in fields is always associative.

③ Identity Element:

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

④ Inverse:

$$\rightarrow 1^{-1} = 1$$

$$\rightarrow \bar{\alpha}^{-1} = \alpha + 1, \text{ since } \alpha \cdot (\alpha + 1) = 1$$

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

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

ii) Checking whether G is cyclic:-

A group is cyclic if there exists an element $g \in G$ such that every other element can be written as g^n for some n .

We check for a generator:

1. Check α :

$$\rightarrow \alpha^1 = \alpha$$

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

$$\rightarrow \alpha^3 = (\alpha+1) \cdot \alpha = \alpha^2 + \alpha = (\alpha+1) + \alpha = 1$$

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

Since there exists a generator (α), $\frac{G}{\alpha}$ is cyclic.

Question: 6.

Step 1: Definition of $GL(2, R)$

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

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

Step 2: 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 of all such matrices is:

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

This forms a subgroup because:

\rightarrow 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.

\rightarrow Identity: The identity matrix I belongs to S .

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

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

Step 3: 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 $S = \lambda I$:-

$$ASA^{-1} = A(\lambda I)A^{-1} = \lambda(AI\bar{A}) = \lambda(A\bar{A}') = \lambda I = S$$

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

Step 4: 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:

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

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

Step 5: 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 transformations ignoring uniform scaling.

Question: 7:

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 (HTTPS)
- Encryption of messages 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 & mitigation

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

Solution: Use authentication (digital signatures, certificates)

- Eavesdropping Attack: Attacker listens but cannot compute the shared key.

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

- Pre-Computation Attack: Using 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, allowing attackers to compute the secret key easily.
2. Vulnerability to Brute-force Attacks: If p is small - an attacker can perform an exhaustive search to find the private keys quickly.
3. Susceptibility to pre-Computation Attacks: When the same small prime p is reused, attackers can create lookup tables to break the key exchange faster.
4. Quantum Computing Threat: With Shor's algorithm - a small p allows a quantum computer to efficiently solve the DLP, making the protocol completely insecure.

Question: 8

Let G_1 be a group, and let H and K be two subgroups of G_1 . We need to show that $H \cap K$ is also a subgroup of G_1 .

1. Non-emptiness:

Since H and K are subgroups of G_1 , 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 $a \in H \cap K$, 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_1 .

Example:

Consider the group \mathbb{Z} under addition:

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

\rightarrow Let $K = 3\mathbb{Z} = \{\dots, -6, -3, 0, 3, 6, \dots\}$ (multiples of 3)

Their intersection is:

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

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

Thus the intersection of two subgroups is always a subgroup.

Question - 9:

Proof that \mathbb{Z}_n is commutative:

The ring \mathbb{Z}_n consists of the equivalence classes of integers modulo n , with addition and multiplication defined as :-

$$\rightarrow \text{Addition} : [a] + [b] = [a+b]$$

$$\rightarrow \text{multiplication} : [a] \cdot [b] = [a \cdot b]$$

Since integer addition and multiplication are commutative, we have;

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

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

Thus, \mathbb{Z}_n is a commutative ring.

Zero divisors in \mathbb{Z}_n :

An element $[a] \in \mathbb{Z}_n$ is a zero divisor if there exists a nonzero element $[b] \in \mathbb{Z}_n$ such that:

$$[a] \cdot [b] = [0]$$

This means $a \cdot b \equiv 0 \pmod{n}$, but neither a nor b is congruent to 0 modulo n .

\rightarrow If n is composite, then \mathbb{Z}_n has zero divisors.

For example, in \mathbb{Z}_6 we have:

$$[2] \cdot [3] = [6] = [0] \pmod{6},$$

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

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

Conditions for \mathbb{Z}_n to be a field:

A ring \mathbb{Z}_n is a field if every nonzero element has a multiplicative inverse. This happens if and only if \mathbb{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 $[b]$ such that:
 $[a] \cdot [b] = [1]$.

Thus, \mathbb{Z}_n forms a field.

→ If n is composite, \mathbb{Z}_n contains zero divisors, so it cannot be a field.

Question - 10 :-

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 power.
- Brute-force attacks can be performed in hours or minutes.
- Advanced cryptanalysis techniques make DES ineffective against sophisticated attackers.
- 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 computers could break DES in just a few minutes.

→ Distributed and Cloud Computing Threats :-

Today, cloud-based computing and FPGAs/GPUs 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 power.

→ Increase key length exponentially increases security

② Comparison of key lengths:

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

→ Triple DES : more secure but slower

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

→ AES - 256 : Extremely secure ; even with the 256-bit key most powerful computers?

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-force 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:

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

- DES uses 16 rounds, making it harder to track input differences.
- S-Boxes 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 & mixcolumns : Spread differences across the block, increasing diffusion.
- AddRoundkey : Combines key at each round, preventing attacks.
- Larger Block & Key size : more secure than DES.

Question - 12:

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 it 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:

$$an + ny = 1$$

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 combination:

$$1 = 3 - 1(2)$$

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

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

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 $\phi(n)$ (Euler's totient function)
2. The public exponent 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 , satisfying:

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

Importance of Efficiency in cryptographic Systems:

- RSA involves large prime numbers (1024-bit, 2048-bit)
- The Extended Euclidean Algorithm runs in polynomial time ($O(\log n)$) ensuring fast key generation.
- Efficient modular inversion is crucial for decryption and signing operations in RSA

Suggestion - 13:

i) ECB mode Insecurity for Highly Redundant data:-

→ In Electronic Codebook (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 $C_0 = I.V$ (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:Vulnerability of LFSRs to Known-Plaintext Attacks

→ linear Feedback Shift Registers (LFSRs) generate keystreams based on linear recurrence relations :

$$S_n = C_1 S_{n-1} \oplus C_2 S_{n-2} \oplus \dots \oplus C_m S_{n-m}$$

→ Since LFSRs are linear, attackers can collect enough keystream bits and solve a system of linear equations to recover the feedback 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 $f(s_1, s_2, \dots, s_n)$

→ Grieffe Generator: Combines three LFSRs using Boolean functions to introduce non-linearity, making attackers more difficult.

→ Clock-Controlled LFSRs: Use irregular clocking mechanisms to disrupt predictability.

Question - 15 :

① Shannon's Definition of Perfect Secrecy:

A cryptographic system achieves perfect secrecy if the ciphertext C reveals no information about the plaintext m , i.e., the probability distribution of m remains unchanged even after observing c :

$$P(m|c) = P(m)$$

for all $m \in M$ and $c \in C$.

Equivalently, in terms of conditional probability:

$$P(c|m) = P(c)$$

for all plaintext and ciphertext.

② 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 $m = c \oplus K$

proof :-

For any plaintext m and ciphertext c ,
we analyze :-

$$P(c|m) = P(k=c \oplus m)$$

Since k is uniformly random, every key
is equally likely, so :-

$$P(k=c \oplus m) = \frac{1}{|K|}$$

Since, $|k| > |m|$, the probability distribution
remains uniform over all possible plaintext.

(iii) Impracticality of perfect secrecy for large-scale
Communication Systems:-

① Key Length Requirements:- perfect secrecy,
like in the One-Time Pad, requires the key
length to be at least as long as the message.

② Key distribution and management:-

Distributing unique keys for each message
or communication session is highly inefficient
and difficult to manage at scale.

③ Key Reuse Prohibition:- The key must be used
only once and reusing keys compromises security.

④ Computational vs. Information-theoretic Security:-
modern cryptography focuses on computational security

Question - 16:

Computation of the first 5 numbers in an LCG sequence:-

Given the LCG recurrence relation:

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

Let's choose specific values:

$$\rightarrow \text{multiplier } a = 5$$

$$\rightarrow \text{Increment } c = 3$$

$$\rightarrow \text{modulus } m = 16$$

$$\rightarrow \text{seed } x_0 = 7$$

Computing the first 5 numbers:-

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

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

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

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

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

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

Question: 17 :

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 (\cdot), 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 :

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

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

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 R (denoted $M_2(R)$).

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

Relation of Rings to Finite Fields & Cryptographic Algorithms:-

① Rings and Finite Fields:

→ A finite field (or Galois field $GF(p^n)$) is a special type of ring where every nonzero element has a multiplicative inverse.

→ Finite Fields are constructed from rings by ensuring no zero divisors and

requiring a prime number order.

② Role of Rings in RSA Cryptography :-
→ RSA relies on the ring of integers modulo n , \mathbb{Z}_n , where $n = p \cdot q$ (product of two large primes).

→ The multiplicative group of units \mathbb{Z}_n^* (elements with inverse) is crucial for RSA encryption and decryption.

→ The extended Euclidean Algorithm helps compute the modular inverse for key generation.

Question 18:

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 $e = 3$ (such that $\text{gcd}(e, \phi(n)) = 1$).

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

→ Using Extended Euclidean Algorithm,

$$d = 27 \text{ since } 3 \times 27 \equiv 1 \pmod{40}$$

Encryption:

$$C = m^e \pmod{n} = 2^3 \pmod{55} = 8$$

Ciphertext: $C = 8$

Decryption: $m = C^d \pmod{n} = 8^{27} \pmod{55}$

Using modular exponentiation,
 $m = 2$

Recovered message: 2

Digital Signature Examples:-

Given values:

$$1. P=7, q=3$$

2. Compute n :

$$n = 7 \times 3 + 1 = 21$$

3. Compute $\phi(n)$:

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

④ Choose $e=5$, compute d using Extended Euclidean Algorithm:

$$\rightarrow d = 5^{-1} \pmod{12} = 5 \text{ (since } 5 \times 5 \equiv 1 \pmod{12}).$$

Signing the Hash:

$$S = H(m)^d \pmod{n} = 5^5 \pmod{21} = 243 \pmod{21} = 12$$

signature: $S=12$

Verification:

$$V = S^e \pmod{n} = 12^5 \pmod{21} = 248832 \pmod{21}$$

$$= 3$$

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

Ensuring Integrity & Authenticity:-

- Integrity :- If the message is altered, the computed hash won't match the verified signature.
- Authenticity :- Only the private key holder (sender) can sign the message, and anyone can verify using the public key.

Question 19 :

Elliptic curve over \mathbb{Z}_p with $p = 23, a = 1, b = 1$

Equation:

$$y^2 \equiv x^3 + ax + b \pmod{23}$$

i.e., $y^2 \equiv x^3 + x + 1 \pmod{23}$

① Verify if $P = (3, 10)$ lies on the curve

Substituting $x = 3, y = 10$ into the equations:

$$10^2 \equiv 3^3 + 3 + 1 \pmod{23}$$

$$100 \equiv 27 + 3 + 1 \equiv 31 \equiv 100 \pmod{23}$$

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

ii) Compute $2P$ using the doubling formula formula for point doubling:

$$\lambda = \frac{3x_1^2 + a}{2y_1} \pmod{p}$$

$$x_2 = \lambda^2 - 2x_1 \pmod{p}$$

$$y_2 = \lambda(x_1 - x_2) - y_1 \pmod{p}$$

Substituting $P = (3, 10)$, $a = 1$ and $p = 23$:

$$\begin{aligned}\lambda &= \frac{3(3^2) + 1}{2(10)} \pmod{23} = \frac{28 + 1}{20} \pmod{23} \\ &= \frac{28}{20} \pmod{23}\end{aligned}$$

Finding the modular inverse of 20 modulo 23:

$$20^{-1} \equiv 7 \pmod{23}$$

$$\lambda = 28 \times 7 \pmod{23} = 196 \pmod{23} = 12$$

Now, computing x_2 :

$$\begin{aligned}x_2 &= 12^2 - 2(3) \pmod{23} = (144 - 6) \pmod{23} \\ &\equiv 138 \equiv 1 \pmod{23}\end{aligned}$$

Now computing y_2 :

$$y_2 = 12(3 - 1) - 10 \pmod{23} = 12(2) - 10 = 24 - 10$$

$$2P = (x_2, y_2) = (1, 14) \pmod{23}$$

iii) Compute $P+Q$ for $P=(3,10)$ and $Q=(9,7)$
formula for addition:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \bmod P$$

$$x_3 = \lambda^2 - x_1 - x_2 \bmod P$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \bmod P$$

Substituting $P = (3,10)$ and $Q = (9,7)$:

$$\lambda = \frac{7-10}{9-3} \bmod 23 = \frac{-3}{6} \bmod 23$$

Finding the modular inverse of 6 modulo 23:

$$6^{-1} \equiv 4 \bmod 23$$

$$\lambda = (-3) \times 4 \bmod 23 = -12 \equiv 11 \bmod 23$$

Now, computing x_3 :

$$x_3 = 11^2 - 3 - 9 \bmod 23$$

$$= 121 - 12 \equiv 109 \equiv 18 \bmod 23$$

Now, computing y_3 :

$$y_3 = 11(3-18) - 10 \bmod 23 = 11(-15) - 10 = -165 - 10$$

$$\equiv -164 \equiv 9 \bmod 23$$

Result: $P+Q = (18, 9)$

Quesn 20 :

Elliptic Curve Digital Signature Algorithm (ECDSA) Calculations:

Given elliptic curve:

$$y^2 \equiv x^3 + 7x + 10 \pmod{37}$$

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 :

$$Q = dG = 9G$$

Using elliptic curve scalar multiplication, compute $9G$, which gives:

$$Q = (24, 29)$$

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

ii) Compute Signature Pair (r, s)

1. Compute $3G$:

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

2. Compute r : $r = x \pmod{n} = 35 \pmod{19} = 16$

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

Finding s^1 modulo 19:

$$3^1 \equiv 13 \pmod{19}$$

now,

$$s = 13(8 + 9 \times 16) \pmod{19}$$

$$s = 13(8 + 144) \pmod{19}$$

$$s = 13(152) \pmod{19}$$

$$s = 1976 \pmod{19} = 14$$

Signature Pair: $(n, s) = (16, 14)$

(iii) Verify Signature

1. Compute $w = s^1 \pmod{n}$:

$$14^1 \equiv 11 \pmod{19}$$

2. Compute u_1 and u_2 :

$$u_1 = n \cdot w \pmod{n} = 16 \cdot 11 \pmod{19} = 6$$

3. Compute $u_1 g_1 + u_2 g_2$:

$$6g_1 + 5g_2 = (35, 17) + (25, 17)$$

$$= (35, 17)$$

4. Verify $x \pmod{n} = r$

$$35 \pmod{19} = 16 = r$$

Signature is valid.

Question: 21:

key Properties of Cryptographic Hash Functions (SHA family) :-

- i) Essential Characteristics of a Secure Hash Function:
 - 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 messages, 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 transactions and links blocks via hash pointers.
3. Password Hashing: Ensures secure password storage using salted hashes.
4. Data Integrity Verification: Used in checksums and digital forensics to detect modifications.

Question - 22:

Concept of Galois Fields (Finite Fields)

$\text{GF}(P)$ and $\text{GF}(2^n)$ Definition.

1. $\text{GF}(P)$ (Prime Field) - A finite field with p elements, where P is a prime number. Arithmetic operations are performed modulo P .

2. $\text{GF}(2^n)$ (Binary Field) - A field with 2^n elements, commonly used in cryptography. Arithmetic is done using polynomials over $\text{GF}(2)$ modulo an irreducible polynomial.

Use in cryptographic primitives

1. Elliptic Curve Cryptography (ECC)

- Points on elliptic curves are defined over $\text{GF}(2^n)$
- Provides security with smaller key sizes compared

2. AES (Advanced Encryption Standard) → RSA.

- AES operates in $\text{GF}(2^8)$, using field arithmetic for Mix columns and subbytes transformations.
- Ensures efficient and secure encryption.

Importance of Field Arithmetic:

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

Question) 23

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 computer.

- 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 entanglement & superposition
Example	NTRU, Kyber (Post-Quantum Encryption)	Quantum Key Distribution (QKD) like BB84
Implementation	Works on classical computers	Requires quantum communication channels.

Question: 24:

Proof of maximum Period of an LFSR keystream:
LFSR recurrence Relation:

A linear feedback shift register (LFSR) generates the keystream using:

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

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

key Idea: characteristic Polynomial and maximum period:

→ The characteristic polynomial of the LFSR

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

→ 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 repeating.

Question: 25:-

Lattice-Based cryptography and LWE-Based signature scheme

(i) Process of signing a Message Using LWE-Based signature scheme

1. Key Generation

- Generate a Public Key PK and Private Key SK .
- Private Key: SK is a secret vector chosen from a lattice.
- Public Key: PK is derived from SK , usually by creating noisy lattice vectors using the Learning with Error (LWE) problem.

2. Signing the Message

- Message hashing: Hash the message M to obtain a fixed-size value $h(M)$.
- Signature Generation:
- Solve a system of equations based on the LWE problem.
- Use SK and the $h(M)$ to create a valid signature ensuring that the signature is tied to the message and private key.

3. Verification

- Signature verification
- The verifier uses PK to check that the signature σ is valid for the message M and the corresp. pending hash $h(M)$.
- 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 s_k be a vector in a lattice, and the public key pk be a noisy lattice vector derived using the LWE problem.
2. Message signing process
 - Given message m , compute its hash $h(m)$.
 - Use the private key s_k and solve the LWE problem to generate a signature σ for $h(m)$.
→ This involves solving a system of linear equations where the noise (errors)

is controlled, ensuring the difficulty of solving the problem for an adversary.

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 s_k , even if they know the public key P_k .

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