IT-24668

1. Answer to the question no. 1

Ans: Quantum computing threatens to break RSA and ECC Shor's algorithm can efficiently solve the mathematical problems (integer factorization and discrete logarithms) they rely on.

Proposed replacements (post-Quantum Cryptography):

Lattice-based (Kyber, Dilithium): Hardness based on shortest vector problems in high-dimentional lattices.

Hash-based (SPHINCS+): Relies on the security of cryptographic hashes.

code-based (McEliece): Based on the difficulty of decoding general linear codes.

These resist quantum cryptanalysis because unlike factoring, these problems are currently believed to be unsolvable in polynomial time by quantum computers.

2.

Answer to the question no. 2

Ans: Here is a minimal implementation using a linear congruential Generator (LCG) approach seeded by the system state:

```
import time
import os

class CustomPRNG:
    def __init__(self, modulus=100000):
        self.modulus = modulus
        self.state = int(time.time_ns()) ^ os.getpid()

    def random(self):
        self.state = (self.state * 1103515245 + 12345 +
                      os.getpid()) % self.modulus

        return self.state


# Example usage
prng = CustomPRNG(modulus=1000)
for _ in range(5):
    print(prng.random())
```

3

Answer to the question no. 3

Ans: Traditional ciphers (Caesar, Vigenere) are character-based, manual, and easily broken by frequency analysis due to short keys and predictable pattern.

Modern ciphers (DES, AES) are bit-based, highly complex and mathematically robust.

Comparison:

Key length: Traditional (Tiny/variable) vs DES (56-bit) vs AES (128-256 bit)

Speed: Traditional is slow manual; modern is extremely fast.

Security: Traditional is insecure; DES is obsolete (brute-forceable); AES is the current global gold standard.

IT-24668

4.

Ans: Action Definition: The action is defined by

$\sigma \{i,j\} = \{\sigma(i), \sigma(j)\}$

for any permutation $\sigma \in S_4$

Well defined: Since σ is a bijection, $\sigma(i) \neq \sigma(j)$ if any $i \neq j$. Thus the image is always a 2-element subset of $\{1,2,3,4\}$

Orbit size: The set of 2-element subsets is

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

Since S_4 can map any pair to any other pair, the action is transitive. Orbit size $|O_{\{1,2\}}| = 6$.

Stabilizer size:

$|S_4| = |\text{orbit}| \times |\text{stabilizer}|$

$24 = 6 \times |\text{Stabilizer}|$

Stabilizer $|G_{\{1,2\}}| = 4$

The elements are: id, (12), (34) and ((12)(34))

5. Ans:

i) Multiplication group: The non-zero elements are $G^* = \{1, \alpha, \alpha+1\}$

Closure: $\alpha^2 = \alpha+1$; $\alpha(\alpha+1) = 1$

identity: 1 exists

Inverses: Every element has a partner that multiplies to 1 ($\alpha \cdot (\alpha+1) = 1$).

G^* satisfies all group axioms.

ii) Cyclicity Yes, it is cyclic. The power of α are: $\alpha^1 = \alpha$, $\alpha^2 = \alpha+1$ and $\alpha^3 = 1$.

Since α generates all non-zero elements the group is cyclic.

IT-24668

6

Ans: Normal subgroup proof: Let Z be the set of scalar matrices λI (where $\lambda \neq 0$) for any $A \in GL(2,R)$ and $S = \lambda I \in Z$: $ASA^{-1} = A(\lambda I)A^{-1} = \lambda(AIA^{-1}) = \lambda I$

Since, $ASA^{-1} \in Z$; Z is a normal subgroup.

Factor Group: The factor group is $GL(2,R)/Z$ which is the Projective General Linear Group, $PGL(2,R)$

Interpretation: $PGL(2,R)$ represent transformation where scaling does not matter. Geometrically, it describes the projective transformations of the real projective line (RP^1), mapping lines through the origin to other lines.

IT-24668

7. Ans: Protocol: Two parties, Alice and Bob, agree on a large prime p and a generator g. Alice sends $g^a \pmod p$ and Bob sends $g^b \pmod p$. Both compute the shared secret $k = g^{ab} \pmod p$

Security Basis: It relies on the Discrete logarithm Problem (DLP); it is computationally infeasible to find a or b from g^a or g^b given a large enough p.

Attacks & Risks:

Man-in the middle (MitM): The protocol is vulnerable because it lacks authentication.

Small Prime Modulus: If p is too small, attackers can solve the DLP using algorithms like the Number Field sieve or brute-force.

IT-24668

6. Ans: Proof: Let H and K be subgroup of G.
To prove $H \cap K$ is a subgroup, we check the three axioms.

1. identity: Since $e \in H$ and $e \in K$ then $e \in H \cap K$

2. closure: If $a, b \in H \cap K$, then
$a, b \in H$ and $a, b \in K$ and
H, K are subgroups, $ab \in H$ and
$ab \in K$ so, $ab \in H \cap K$

3. Inverses: If $a \in H \cap K$, then $a \in H$ and
$a \in K$. Since H and K are
subgroups, $a^{-1} \in H$ and $a^{-1} \in K$, So,
$a^{-1} \in H \cap K$

Example: In the group integer $(\mathbb{Z}, +)$: Let $H = 2\mathbb{Z}$
and $K = 3\mathbb{Z}$. The intersection $H \cap K = 6\mathbb{Z}$.
Which is also a subgroup of $\mathbb{Z}$.

IT- 29668

9 Ans: Commutativity: In Z_n, multiplication is defined as $[a][b] = [ab \pmod n]$. Since integer multiplication is commutative ($ab = ba$), it follows that $[a][b] = [b][a]$. Thus, Z_n is a commutative ring.

Zero Divisors: Z_n has zero divisors if and only if n is composite. Example: In Z_6, $[2] \cdot [3] = [6] = [0]$. Since $[2][3] \neq [0]$, They are zero divisors.

Condition for a field: Z_n is a field if and only if n is a prime number. If n is prime, every non-zero element has a multiplicative inverse and there are no zero divisors.

IT-24868

10.

Ans: Vulnerabilities: DES is insecure primarily due to its short 56-bit key, which is vulnerable to brute-force attacks. Modern hardware can crack a DES key in hours. It is also susceptible to differential and linear cryptanalysis.

AES improvements:

Key-size: AES offers 128, 192, 256 bit keys, making brute force mathematically impossible.

Block-size: AES uses 128-bit blocks (vs DES's 64 bit) preventing data collision attacks

Design: AES uses a Substitution-Permutation network, specifically designed to resist the cryptanalytic techniques that threatened DES.

11 Ans: DES : Resistance relies solely on the Sboxes. The Feistel structure only modifies that the half the back pen round, requiring more rounds to achieve full diffusion. While DES-S boxes were designed to minimize differential probability, the small 64 bit block size still allows attackers to tracle difference pattern.

ii) AES Resistance:

1. Wide Trail strategy: Unlike DES every bit of the state is transformed every round.

2. Diffusion: ShiftRows and MixColumns ensure a single bit change spread across the entire block rapidly.

3. Subbytes - AES-Sboxes have a mathematically lower maximum differential probability making it much harder to find high-probability differential paths.

12

Answer: Algorithm: Solve $ax+ny=1$ using back-substitution from the Eudlidean Algorithm. The value $x \pmod n$ is the inverse.

RSA Application: It is used to calculate the private key d, where $d \equiv e^{-1} \pmod{\phi(n)}$.

Efficiency: Because RSA uses huge numbers (2048-bit +); the algorithms logarithmic speed is essential for generating keys and performing decryptions in milliseconds.

IT-24668

13. Answer: i) ECB Insecurity:

Mathematical Proof: ECB encrypts blocks independently $C_i = E_k(P_i)$. If $P_i = P_j$ then $C_i = C_j$

Result: ECB preserves patterns in data, identical blocks in an image remain (identical in ciphertext) leaking structural information.

ii) CBC Relations & Error Propagation

Relations:

Encryption: $C_i = E_x(P_i \oplus C_{i-1})$

Decryption: $P_i = D_k(C_i) \oplus C_{i-1}$

Error Propagation Proof: If ciphertext C_i is corrupted.

1. P_i is totally garbled (since C_i is the input to D_k)
2. P_{i+1} has bit-errors only where C_i was flipped (XOR property)
3. P_{i+2} and beyond are unaffected because they depend on C_{i+1}, C_{i+2} etc, which are correct. Result: Errors are limited to exactly two blocks.

14 Ans: Linearity Vulnerability

LFSRs follow a linear recurrence:

$$s_i = \sum c_j s_{i-j} \pmod 2$$

1. Attack: An attacker with $2L$ bits of keystream can use the Berlekamp Massey algorithm.

2. Outcome: The algorithm solves a system of linear equations in $O(L^2)$ time to recover the feedback polynomial and initial state, breaking the key.

Mitigation: Non-linear filtering, Apply a non-linear Boolean function $f(x)$ to several taps of the LFSR.

Result: This destroys the linear relationship and increases the linear complexity, making it possible for simple linear solvers to reconstruction

15.

Ans: Definition: Perfect secrecy means:

$$P(M=m \mid C=c) = P(M=m)$$

knowing the chiphertext reveals nothing about the message.

ii) OTP Proof:

1. Relation: $C = m \oplus K$ $\Rightarrow$ for every (m, c) there is exactly one K.
2. Probability: If K is uniform, $P(C=c \mid M=m) = \frac{1}{|K|}$
3. independence: Since this value is the same for all m, M and C are independent

iii) Why Impractical: The key must be at least as long as the message. sending a 1TB key is as hard as sending a 1 TB message Reusing the keys "on-time" only that means reuse destroy secrecy.

IT-24668

16. Ans: To compute the sequence, let's use the parameters: $a = 3$, $c = 1$ and $m = 10$

Recurrence: $x_{n+1} = (3x_n + 1) \pmod{10}$

Calculation:

$x_0 = 7$ (seed)

$x_1 = (3 \cdot 7 + 1) \pmod{10} = 22 \pmod{10} = 2$

$x_2 = (3 \cdot 2 + 1) \pmod{10} = 7 \pmod{10} = 7$

$x_3 = (3 \cdot 7 + 1) \pmod{10} = 22 \pmod{10} = 2$

$x_4 = (3 \cdot 2 + 1) \pmod{10} = 7 \pmod{10} = 7$

Sequence: 7, 2, 7, 2, 7

17 Ans: Rings in Abstract Algebra

Definition: A ring is a set with two operations (+ and ·) where "+" is addition and "·" is multiplication.

Distributive : $a(b+c) = ab+ac$

Example

Commutative: Integers ($\mathbb{Z}$) where, $a \cdot b = b \cdot a$

Non-commutative: Square matrices, where order matters ($AB \neq BA$)

Relation to finite fields: A field is a commutative ring where every non-zero element has a multiplicative inverse.

Role in RSA: RSA works within the ring $\mathbb{Z}_n$. Encryption and Decryption are modular exponentiation operations.

18. Ans: Part 1: RSA Enanyption & Deanyption

set up ($P=5$, $2=11$, $M=2$); $n=55$, $\phi(n)=96$

let, $e=3$, then $d=27$ ($3 \times 27 \equiv 1 \pmod{96}$)

Enanyption: $C = 2^3 \pmod{55} = 8$

Deanyption: $M = 8^{27} \pmod{55} = 2$

Part-2: RSA signature

setup ($P=7$, $2=3$, $H(m)=3$):

$n = 21$, $\phi(n) = 12$

let $e=5$ then $d=5$ ($5 \times 5 \equiv 1 \pmod{12}$)

sign: $S = 3^5 \pmod{21} = 243 \pmod{21} = 12$

Verify: $V = 12^5 \pmod{21} = 3$ (Matches $H(m)$)

Why it works

1. Authenticity
2. Integrity

IT-24668

19 Ans: i) Verify P(3,10)

$y^2: 10^2 = 100 \equiv 8 \pmod{23}$

$x^3: x^3 + x + 1 : 3^3 + 3 + 1 = 31 \equiv 8 \pmod{23}$

result: $8 = 8$

ii) Double P(2P)

1. Slope (λ): $\frac{3(3^2+1)}{2(10)} = \frac{28}{20} \equiv \frac{5}{20} = \frac{1}{4} \equiv 6 \pmod{23}$

2. x_3: $6^2 - 2(3) = 30 \equiv 7$

3. y_3: $6(3-7) - 10 = -34 \equiv 12$

result: $2P = (7, 12)$

iii) Addition P+Q

1. Slope (λ): $\frac{7-10}{9-3} = \frac{-3}{6} \equiv \frac{-1}{2} \equiv 11 \pmod{23}$

2. x_3: $11^2 - 3 - 9 = 109 \equiv 17$

3. y_3: $11(3-17) - 10 = -164 \equiv 20$.

result: $P + Q = (17, 20)$

20

Ans: i) Public key Q

$Q = 9G$; calculated via scalar multiplication of $(2,5)$

Result : $Q = (3,3)$

ii) ECDSA signing

1. compute: calculate $kG = 3G = (15,20)$

$r = 15 \pmod{19} = 15$

2. compute s:

$s = k^{-1}(H(M) + rd) \pmod{n}$

$s = 3^{-1}(8 + 15 \cdot 9) \pmod{19} = 13(14\ 3)$

$(\text{mod } 19) = 16$

signature: $(r, s) = (15, 16)$

iii) Verification

1. calculate weights: $w = s^{-1} \pmod{19} = 6$

$u_1 = H(M)w = 8 \cdot 6 \equiv 10 \pmod{19}$

$u_2 = rw = 15 \cdot 6 \equiv 14 \pmod{19}$

2. Compute Point: $X = u_1 G + u_2 Q = 10G + 14Q$

$= (15, 20)$

3. check : The x-coordinate (15) matches r . Valid

IT-24668

21 Ans: i) Core Properties

- Pre-image-Resistance: Given h, Impossible to find m where $H(m)=h$
- Second-Pre-image Resistance: Given m_1, impossible to find m_2 that hashes to the same value
- Collision Resistance: Impossible to find any two inputs that hash to the same value.

ii) Output Length & Security

- Brute force: Length n dictates attack difficulty
- Birthday Paradox: Finding collisions takes $2^{n/2}$ operations.

iii) Application:

- Digital signature: Signs the hash of a message for speed and integrity.
- Blockchain: Hashes link blocks together any change breaks the chain.
- Proof of work: Miners must find a hash meeting specific criteria to validate blocks.

22. Ans: finite Fields (GF), A Galois field is a finite set where +, −, × and ÷ are always defined.

Types.

- GF(P) (Prime fields)
- GF(2^n) (Binary Extension field)

Cryptographic use:

- AES (GF(2^8)): The S-Box uses multiplicative inversion in this field to scramble data.

GF(2^n) is used because addition is a simple, fast bitwise XOR.

ECC: Curve coordinates are elements of GF(P) or GF(2^n).

Importance:

1. Precision
2. Hardware Speed

IT-24668

23. Ans: i) Shortest Vector Problem (SVP): A lattice is a grid of points in n-dimensional space. The SVP asks to find the shortest non-zero vector (point closest to the origin) in that lattice.

ii) comparison between Post Quantum and Traditional Quantum.

feature	Traditional (RSA/Ecc)	Lattice-Based Cryptography
Hard Problem	Factoring / Discrete Log	SVP / Learning with Errors (LWE)
Shor's Algorithms	Broken: Can solve these in polynomial time.	Resistant: No known efficient quantum algorithm exists.
Key Size	Small / efficient	Larger than RSA/Ecc but manageable

IT-24668

29. Ans: Proof : Maximum Period of an LFSR

1. State Constraints An m-bit register has 2^m total bit combinations.

   - The zero state: (0, 0, ... 0) must be excluded because the XOR sum of zeros is always zero, preventing the sequence from even changing.

   - Non-zero States: This leaves exactly $2^m - 1$ possible states.

2. Algebraic Property: The LFSR's recurrence corresponds to a characteristics polynomial over GF(2)

3. Conclusion Since the LFSR visits all available non-zero states before returning to the start, the maximum period T is:

$$T = 2^m - 1$$

25. Ans: LWE signature : Quick so

i) Process :

1. KeyGen : Secret s (private), $t = As + e$ (public)
2. Sign : Create mask y, challenge $c = Hash(M, Ay)$ and response $z = y + cs$
3. Verify : Check if $Az - ct \approx Ay$ and if z is small

ii) Signing & Security

Commit : Pick random, y compute $w = Ay$

challenge : $c = Hash(M, w)$

output : (c, z) where $z = y + cs$