1. Quantam Computing & Post-Quantam Cryptograp

Implications of quantam Computing:

- Quantam computers can efficiently solve proble that classical computers find hard.
- Shor's algorithm can factor large integers and compute discrete logarithm's in polynomi time.
- This makes traditional public-key ciphers like RSA and ECC vulnerable.

Post-Quantam cryptography algorithms:

- Lattice-based cryptography: NTRU, kyber.
- Hash-based signatures: SPHINCS+, Merkle trees; secure under collision-resistant hash functions.
- Code-based cryptography: MCEliece trees; secure under collision-resistant hash functio

Resistance Quantam attacks:

- Shor's algorithm does not efficiently solve lattice, code or hash based functions.
- Security is based on NP-hard problems and remains hard for quantam computers

IT-24639

(2) Novel PRNG in Python:

Requirements: use timestamp, process ID, modules for range.

python:

```
import time, os
def custom_prng(modules):
    seed = int((time.time() * 1000) + os.getpid())
    return (1103515245 * seed + 12345) % modules
print(custom_range(1000))
```

- Timestamp → ensures time-based variation.
- Process ID → adds system-level randomness.
- Modules → constrains output to desired range.

(3) Cipher Comparison:

Cipher	key length	Speed	Security	Notes
Caesar	1-26	Very fast	Extremely weak	Easily brute-forced
Vigenere	length for keyword	Medium	weak	Vulnerable to frequency analysis
Playfair	25-characters key	Medium	weak	Slightly better than vigenere
DES	56-bit	Fast	weak	susceptible to linear attacks
AES	128/192/256 bit	Fast	strong	Resistant to all known classical attacks.

Strengths and Weaknesses:

- Traditional ciphers are simple, fast but insecure for modern standards.
- Modern symmetric ciphers (AES/DES) are computationally secure, scalable and designed to resist cryptanalysis.

(4) S_4 Action on 2-element subsets

- Definition of Action:
  $\sigma \cdot \{i,j\} = \{\sigma(i), \sigma(j)\}$ for $\sigma \in S_4$
- Well-defined: Permutation maps 2-element subsets to 2-element subsets.
- Orbit of $\{1,2\}$:
  2-element subsets of $\{1,2,3,4\}$ are:
  $\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\}$
- Orbit size = 6
- Stabilizer of $\{1,2\}$: Permutations that fix $\{1,2\}$ setwise.
- Examples: id, identity, $(1,2)$, $(3,4)$, $(1,$
  → size = 4

5. $GF(2^2)$ field

- Constructed with irreducible polynomial over $GF(2)$: x^2+x+1
- Elements : $\{0, 1, \alpha, \alpha+1\}$, $\alpha^2 = \alpha+1$

(i) Multiplicative Group

- Non-zero elements $\{1, \alpha, \alpha+1\}$
- Closure, identity (1), inverses exist : $\alpha(\alpha+1) = 1$, $\alpha^2 = \alpha+1$

(ii) Cyclicity :

- α generates $\{\alpha, \alpha+1, 1\}$ → group is cyclic.

(6) $GL(2,R)$ & Scalar Matrices :

- Scalar matrices : $S = \{kI \mid k \neq 0\}$
- Normal subgroup : For all $A \in GL(2,R)$, $AS = SA$
- Factor group : $GL(2,R) / \text{scalars} \approx PGL(2,R)$
  - Structure : Linear transformations modulo scalar multiplications.

⑦ Diffie-Hellman key Exchange:

- Protocol:
  - Choose prime p and generate g.
  - Alice: $a \rightarrow$ sends $g^a \bmod p$
  - Bob: $b \rightarrow$ sends $g^b \bmod p$
  - Shared key: $g^{(ab)} \bmod p$
- Security:
  - Based on discrete logarithm problem.
  - Vulnerable to man-in-the-middle → requires authentication.
  - Small modulus → easy discrete log → insecure

⑧ Intersection of Subgroups:

- Theorem: $H_1 \cap H_2$ is a subgroup.
  - Identity: $e \in H_1 \cap H_2$
  - Closure: if $x, y \in H_1 \cap H_2$, $xy \in H_1 \cap H_2$
  - Inverse: $x^{-1} \in H_1 \cap H_2$
- Example: Z_6; $H_1 = \{0, 2, 4\}$, $H_2 = \{0, 3\}$ →

$$H_1 \cap H_2 = \{0\}.$$

(9) Ring Z_n :
- Commutative under addition/multiplication.
- Zero divisors: Exist if n is not prim.
- Field condition: Z_n is a field $\Leftrightarrow$ n is prime.

(10) DES vulnerabilities and AES improvements
- DES: 56 bit key → brute-force attack feasible
- Weak S boxes → vulnerable to linear cryptanaly
- AES:
  - Longer keys (128/192/256)
  - Non-linear S boxes, more rounds
  - Resistant to known attacks.

(11) Differential Cryptanalysis:

(i) DES Feistel structure:
- Spreads input difference → only partial exposure to each round.

- AES Resistance:
  - SubBytes: Non-linear
  - ShiftRows + Mixcolumns: diffusion
  - AddRoundkey: key mixing
  - → Difficult to exploit difference

(12) Extended Euclidean Algorithm:

- Solve, $ax \equiv 1 \bmod n$ for x
- Steps: compute $\gcd(a, n)$ recursively
- RSA: Compute private key $d = e^{-1} \bmod \phi(n)$
- Efficient for large keys → practical RSA encryption/decryption.

(13) Block Cipher Modes

(i) ECB insecurity:

- Identical plaintext blocks → identical cipher text → leaks patterns.

(ii) CBC Mode:

- Encryption: $C_i = E_k(P_i \oplus C_{i-1})$
- Decryption: $P_i = D_k(C_i) \oplus C_{i-1}$
- Error propagation: only 1 block affected.

(14) LFSR Vulnerability:

- Linearity → predictable sequences under known plaintext attacks.
- Mitigation → Use Non-linear combination of multiple LFSRs (Geffe generator, filter functions)

(15) Perfect Secrecy

(i) Shannon : $P(M|C) = P(M)$

(ii) One-time pad:
- key random, $|K| \geq |M| \rightarrow$ perfect secrecy

(iii) Impracticality:
- Requires large, truly random keys for each message.
- key distribution is difficult.

(16) Linear Congruential Generator (LCG)
- formula : $x_{(n+1)} = (ax_n + c) \bmod m$
- Example: $a=5, c=3, m=16, x0=7 \rightarrow 6, 1, 8, 11, 10$

(17) Ring Definition
- Ring : $(R, +, \cdot)$ with additive identity, inverses, associative + distributive
- Commutative example: Z_n
- Non-commutative example: $M_2(R)$
- Use in cryptography: Provides modular arithmet

for RSA, finite fields, ECC.

(18) RSA Encryption / Decryption :

- $P=5, q=11 \rightarrow n=55, \varphi(n)=40, e=3$
- Encrypt $M=2$: $C=2^3 \bmod 55 = 8$
- Decrypt : $M = C^d \bmod n = 2$

(19) RSA signature :

- Sign $H(m)=3, d=7 \rightarrow s = 3^7 \bmod 21 = 3$
- Verify $s^e \bmod n = H(m) \rightarrow$ integrity / authenticity.

(20) Elliptic Curve Operations

- Equation : $y^2 = x^3 + ax + b \bmod p$
- Check $P=(3,10)$: Verify $10^2 = 3^3 + 1*3 + 1 = 31$

$\Rightarrow 100 \bmod 23 = 8$) Not on curve

- Doubling : $\lambda = (3x_1^2 + a) / (2y_1)$
- Addition : $\lambda = (y_2 - y_1)/(x_2 - x_1)$, $x_3 = \lambda^2 - x_1 - x_2$

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

(21) ECDSA Example:

- Base $G = (2,5)$, $n = 19$, private $d = 9 \rightarrow Q = dG$
- Sign $H(M) = 8$, random $k = 3 \rightarrow$ compute r, s
- Verification: check r, s with $Q \rightarrow$ signature valid.

(22) Cryptographic Hash functions:

(i) Properties: Pre-image resistance, collision resistance, second pre-image resistance.

(ii) Output length: Longer Output (SHA-256) → harder to find collisions

(iii) Applications: Digital signatures, block-chain.

(23) Galois Fields:

- GF(p): arithmetic mod prime p.
- GF(2^n): used in AES, ECC.
- Field arithmetic ensures invertibity, diffusion and secure cryptographic operations.

(24) Lattice Based Cryptography:

(i) SVP: Finding shortest nonzero vector in Lattice → NP-hard.

(ii) Security vs RSA/ECC: Resistant to Shor's algorithm; RSA/ECC broken.

(iii) Quantam Cryptography: QKD ensures secure key exchange, different from lattice based encryption.

(24) LFSR Maximum Period:

- Max period = $2^m - 1$ if characteristic polynomial is primitive.
- Ensures full utilization of all states except 0

(25) LWE-Based Signatures:

(i) key generation: lattice public/private key.

(ii) Signing: encode message as lattice vector, add small noise.

(iii) Verification: check lattice relation holds.

Example: • Message M → hash → map to lattice vector
• Add random small error → signature.
• Verifier checks relation with public key.