IT-24660

1. With the advent of quantum Computing, traditional public-key Cryptosystems such as RSA and ECC are vulnerable to shore algorithm. Discuss the implications on cryptographic protocols and suggest post quantum alternatives. How do they resist quantum attacks?

Ans: Quantum Computing threatens RSA and ECC because shor's algorithm can factor large number and solve discrete logarithms in polynomial ~~treats~~ time. This means attackers with quantum computers could break key exchange encryption, and digital signatures used in protocol like TLS, SSH, PKI. However, Symmetric algorithm like AES and hash functions remain mostly safe with larger key sizes.

To replace RSA and ECC, post-Quantum Cryptography (PQC) algorithm are proposed,

IT-24660

→ Lattice-based (Kyber, Dilithium) — based on learning with errors.
→ Code-based (McEliece) → based on decoding linear code.
→ Hash-based (SPHINCS+) — based on hash function security.
→ Multivariate — based on solving multivariate equations.

These algorithm resist quantum attacks because their security relies on mathmetical problems not affect by shors algorithm and no effect quantum solutions are known. Hence pqc provides a quantum-safe replacement for RSA and ECC.

IT-24660

2. Ans : A simple PRNG can be created by Combining the current time and processor ID (PID) to from a seed value. Then a modulus operation is used to keep the random number within a desire range.

python implementation:

```
python
import time
import os
def simple_prng(mod_range):
timestamp = int(time.time() * 1000)
pid = os.getpid()
seed = timestamp ^ pid
random number = seed % mod_range
return random_number
Print (simple_prng(100)).
```

IT-24660

3.

→ Tradition ciphers like Caesar, Vigenere and playfair use simple letter substitution methods. They have very short keys are easy to break using frequency analysis and are not secure against modern attacks. They are simple and fast but only useful for learning purposes.

→ Modern ciphers like AES and DES use complex mathematical operations and long keys. They are very secure, resist modern cryptanalysis and are used in real world applications like banking and internet security. AES is fast and secure, while DES is now considered weak due to it's short key.

IT-24660

Feature	Traditional ciphers	Modern ciphers
key length	Very short	long (128+ bits)
Speed	Very fast	Fast with hardware/software support.
Security	Very weak	Very strong
Attack resistance	Easily broken	Resistant to modern attacks.
usage	Educational purpose	Real-world security system.

4.

Step 1: Define the action

let A be the set of all 2-element subsets of $\{1,2,3,4\}$

For any permutation $\sigma \in S_4$ and any subset $\{a,b\} \in A$, define

$$\sigma:\{a,b\} = \{\sigma(a), \sigma(b)\}.$$

Step2: show the action is well defined

→ since σ is a permutation, it maps elements of the set to elements of the

same set.

→ So $\{\sigma(a), \sigma(b)\}$ is again a 2-element subset of $\{1, 2, 3, 4\}$

→ Identity permutation keeps the subset same.

→ Composition of permutation also works correctly.

Hence the action is well defined.

step-3: Find the orbit of $\{1, 2\}$

The orbit is all 2-elements that can be formed from $\{1, 2, 3, 4\}$

All possible two element subsets are:

$\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\}$

So, orbit size = 6.

step 4:

Find the stabilizer of $\{1, 2\}$

The stabilizer consists of all permutations in S_4 that keep $\{1, 2\}$ unchanged.

CamScanner

possible permutations:

→ Identity
→ (1 2)
→ (3 4)
→ (1 2)(3 4)

So stabilizer = 4

5.

The field $GF(2^2)$ has 4 elements. Let α be a root of $x^2+x+1=0$

from the equation : $\alpha^2+\alpha+1=0$

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

so, the element of $GF(2^2)$ are : $\{0, 1, \alpha, \alpha+1\}$

(1) Consider only non-zero elements :

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

check group properties :

→ closure :

Multiplying any two elements given another element in the set.

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

$\Rightarrow \alpha \times (\alpha + 1) = 1$

$\Rightarrow (\alpha + 1) \times (\alpha + 1) = \alpha$

→ Identity: 1 acts as multiplicative identity.

→ Inverse :

1. inverse of α is $\alpha + 1$
2. inverse of $\alpha + 1$ is α

→ Associative : True because it's a field.

Hence non Zero elements of GF (2^2) form a group under multiplication.

(11)

A group is cyclic if one element can generate all elements.

Check power of α :

$$\alpha^1 = \alpha$$
$$\alpha^2 = \alpha + 1$$
$$\alpha^3 = 1$$

CamScanner

So, α generate all non Zero elements.
Hence, the set $\{1, \alpha, \alpha+1\}$ is a cyclic group.

6.

Step-1 : Define scalar Matrices

A scalar matrix has the form

$$\lambda I = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}, \lambda \neq 0$$

let, $H = \{\lambda I \mid \lambda \in \mathbb{R}, \lambda \neq 0\}$

Step-2 : show H is a subgroup

→ closure : $(\lambda I)(\mu I) = (\lambda\mu) I \in H$

→ Identity : $I = I \in H$

→ Inverse : $(\lambda I)^{-1} = \frac{1}{\lambda} I \in H$

So, H is a subgroup of $GL(2, \mathbb{R})$

step 3 : Show H is normal

Take any matrix $A \in GL(2, \mathbb{R})$ and $\lambda I \in H$

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

So, H is a normal subgroup.

CamScanner

Step-4 : Factor group

The factor group is, $GL(2, \mathbb{R})/H$

In this group, two matrices are considered the same if they differ only by a scalar multiple.

The factor group is called the projective general linear group, written as,

$$PGL(2, \mathbb{R})$$

Interpretation : This group represents all invertible matrices ignoring scalar multiples. It studies the shap or direction of transformation, not their size.

CamScanner

IT-24660

7.

The diffie-Hellman key exchange is a method that allows two parties to securely share a secret key over an insecure channel (like the internet), without sending the key directly.

How the diffi-Helman protocol works:

→ Two users agree on a public prime num p and a base g.
→ Each user choose a private number.
→ They exchange public values created using their private numbers.
→ Both user calculate the same secret key, which is used for secure Communication.

Application in secure Communication:

→ Used in HTTP website, online banking and secure messaging.
→ Helps in secure data encryption.
→ Protect information sent over the internet.

CamScanner

IT- 24660

Role of discrete logarithm problem:

→ It is very hard to find the secret number from the public value.

→ This problem is called the discrete logarithm problem.

→ Because of this attackers cannot find the secret key easily.

Man-in-the-Middle Attack

→ An attacker may intercept messages between two users.

→ Diffi-Hellman alone cannot stop this attack.

→ Authentication needed to prevent it.

Effect of using a small prime number:

→ If the prime number is too small, the secret key can be broken.

→ Attackers can easily calculate the private key this makes the system unsafe.

CamScanner

IT-24660

8. Ans:

To ~~check~~ prove $H_1 \cap H_2$ is a subgroup, we check the subgroup conditions,

let $x, y \in H_1 \cap H_2$

this means x, y are in both H_1 and H_2

1. closure:

Since H_1 and H_2 are subgroups

$xy \in H_1$

$xy \in H_2$

So, $xy \in H_1 \cap H_2$

2. Identity:

Identity element e is both H_1 and H_2.

So, $e \in H_1 \cap H_2$

3. Inverse:

If $x \in H_1 \cap H_2$, then

$\rightarrow x^{-1} \in H_1$

$\rightarrow x^{-1} \in H_2$

So, $x^{-1} \in H_1 \cap H_2$

So, $H_1 \cap H_2$ is a subgroup of G.

Ex. Let $G = Z$

take two subgroups:

$H_1 = 2Z = \{\dots, -4, -2, 0, 2, 4 \dots\}$

$H_2 = 3Z = \{\dots, -6, -3, 0, 3, 6 \dots\}$

their intersection $H_1 \cap H_2 = 6Z$

this is also a subgroup of Z.

9. Ans:

In Z_n addition and multiplication are done modulo n.

For any $a, b \in Z_n$:

$$a + b \equiv b + a \pmod{n}$$

$$a \times b \equiv b \times a \pmod{n}$$

So both multiplication and addition are commutative.

Hence Z_n is a commutative ring.

CamScanner

IT-24660

A Zero divisor is a non-Zero element that gives 0 when multiplied by another non-Zero element.

Example in Z_6 : $2 \times 3 = 6 \equiv 0 \pmod 6$

Hence, 2, 3 are no zero and their product is 0.

So Z_6 has Zero divisor.

Z_n is a field : when a field has a no Zero divisors and every non zero element has a multiplicative inverse. This happens only when n is a prime number.

Ex. Z_5 is a field.

IT-24660

10.

DES uses a small key size of 56 bits, which makes it weak. With modern computers, DES can be broken easily. Hence it is not secure for modern use.

In a brute-force attack, an attacker tries all possible keys. Since DES has only 2^{56} key, attackers can test them quickly and find the correct key.

Short key length means less security. The smaller the key, the easier it is to break the encryption. DES fails because its key length is too small.

AES was developed to replace DES. It uses larger key size (128, 192, or 256 bits)

and stronger encryption methods. AES is more secure and resistant to attacks. making it suitable for modern applications.

11. Ans:

① DES usage uses a Feistel structure, where the data block is split into two halves. In each round, one half is changed using a function and then mixed with the other half.

These structure helps spread small changes in input across many rounds. Because of multiple rounds and mixing, it becomes difficult for an attacker to predict how input differences affect the output. However DES still has weaknesses due to its small block and key size.

CamScanner

IT-24660

(ii) AES is more resistant than DES because,

- AES uses several strong operations in each round.
  - → subBytes : Provides strong non-linearity using substitution tables.
  - → shiftRows : Moves data around to spread changes.
  - → MixColumns : Mixes data across columns for better diffusion.
  - → Add Round key : Combines data with the secret key in every round.

These step cause a small input change to affect many output bits quickly. Because of strong diffusion and non linearity, AES is much more resistant to differential cryptanalysis than DES.

IT-24660

12. Ans:

The modular inverse of a modulo n is a number x such that,

$$a \times x \equiv 1 \pmod{n}$$

This is possible only when gcd $(a, n) = 1$

So x is the modular inverse of a modulo n

In RSA,

→ choose e such that gcd $(e, \phi(n)) = 1$

Find d such that,

$$e \times d \equiv 1 \pmod{\phi(n)}$$

* why efficiency is important,

→ RSA uses very large numbers

→ Finding inverse must be fast for key generation.

→ The extended Euclidean Algorithm is efficient even for large numbers, so it is suitable for cryptography.

13. Ans :

(i) Why ECB mode is insecure,

In ECB mode, each plaintext block is encrypted separately,

$$C_i = E_K(P_i)$$

if two plaintext blocks are the same,

$$P_i = P_j \Rightarrow C_i = C_j$$

So, identical plaintext given identical ciphertext.

For highly redundant data (like image or repeated text), patterns in plaintext appear in ciphertext. An attacker can notice these patterns, so ECB leaks information and is insecure.

(ii)

In CBC mode, each block depends on the previous block,

IT-24660

Encryption :

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

with $C_0 = IV$

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

Error propagation in CBC decryption;

Suppose an error occurs in ciphertext block C_i

→ P_i becomes incorrect because it depends on C_i

→ P_{i+1} is also affected because it uses C_i

→ But P_{i+2} and later blocks are correct.

So, the error affects only two blocks, showing limited error propagation.

14. Ans :

A linear feedback shift Register produces bits again using a linear relation:

$$S_t = C_1 S_{t-1} \oplus C_2 S_{t-2} \oplus \cdots + C_k S_{t-k}$$

This means the next bit is a linear Combination of previous bits.

If an attacker knows some plaintext and the corresponding ciphertext, they can find the keystream. Since the LFSR is linear, the attacker can set up linear equations and solve them to find,

→ The feedback coefficient C_i

→ The internal state of the LFSR.

This can be done using simple linear algebra.

So, LFSRs are weak against known-plaintext attacks.

To make it secure, we must break the linearity.

A common mathematical method is,

→ Combine multiple LFRSs using a non-linear function.

→ Or use Nonlinear Feedback shift registers.

15. Ans:

(i)

A cryptosystem has perfect secrecy if the ciphertext given no information about the plaintext. Mathematically,

$$P(M|C) = P(M)$$

This means: Knowing the ciphertext does not change the probability of the plaintext.

(ii)

In OTP, $C = M \oplus K$

where k is,
→ truly random
→ same length as message
→ used only once.

For any plaintext M and ciphertext C there exist a ~~set~~ unique key,

$$K = C \oplus M$$

Since all key are equally likely

$$P(C|M) = \frac{1}{|K|}$$

So, $P(M|C) = P(M)$

(iii) Why perfect secrecy is impractical
→ key must be as long as the message
→ keys must be truly random
→ keys cannot be reused.
→ Hard to store and share very large keys securely.

So, For large communication systems, managing such keys is difficult.

IT-24660

16. Ans :

A linear Congruential Generator is given by

$$X_{n+1} = (aX_n + c) \bmod m$$

Choose the values,

$a = 5$, $c = 3$, $m = 16$

Given seed, $X_0 = 7$

Step 1, Find X_1 : $X_1 = (5 \times 7 + 3) \bmod 16$

$= 38 \bmod 16$

$= 6$

X_2 : $X_2 = (5 \times 6 + 3) \bmod 16$

$= 33 \bmod 16$

$= 1$

X_3 : $X_3 = (5 \times 1 + 3) \bmod 16$

$= 8 \bmod 16$

$= 8$

X_4 : $X_4 = (5 \times 8 + 3) \bmod 16$

$= 11$

X_5 : $X_5 = (5 \times 11 + 3) \bmod 16$

$= 10$

CamScanner

IT-24660

First 5 numbers of the sequence,

6, 1, 8, 11, 10

17.

Definition of a ring:

A ring is a set with two operations: addition (+) and multiplication (×).

It satisfies:
→ closed under addition and multiplication.
→ Addition is associative and has identity (0) and inverse.
→ Multiplication is associative.
→ Multiplication distributes over addition.

Example for commutative ring:

Z (integers) where $a \times b = b \times a$

Ex for non-commutative ring:

set of 2×2 matrices, where $AB \neq BA$

CamScanner

IT-24660

Finite fields are built from rings.

For example, Z_p (integers mod p) is a ring.

If p is prime, this ring becomes a field. (every non-zero element has inverse)

So, rings are the base structure used to create finite fields.

Role of rings in RSA:

RSA works in the ring Z_n (integers modulo n)

→ Encryption and decryption use modular arithmetic in this ring.

→ Properties of the ring are used to generate keys.

18.

(a)

Given, $p = 5$, $q = 11$

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

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

Choose $e = 3$ such that $\gcd(3, 40) = 1$

Find d such that,

$$e \times d \equiv 1 \pmod{40}$$
$$3 \times 27 = 81 \equiv 1 \pmod{40}$$

So, $d = 27$

public key : $(e, n) = (3, 55)$

Private key : $d = 27$

Encrypt $M = 2$

$$C = M^e \bmod n$$
$$= 2^3 \bmod 55$$
$$= 8$$

Decrypt :

$$M = c^d \bmod n$$
$$= 8^{27} \bmod 55$$
$$= 2$$

original message is recovered.

(b)

Given, $p = 7, q = 3$

$n = 21$, $\phi(n) = (6)(2) = 12$

choose $e = 5$, $\gcd(5, 12) = 1$

Find d such that, $5 \times 5 = 25 \equiv 1 \pmod{12}$

So, $d = 5$

Hash of message : $H(m) = 3$

Sign using private key :

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

Verify using public key :

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

Hash matches, so signature is valid.

→ integrity: If message changes, hash changes, signature fails.

→ Authenticity: Only the owner of private key can create the signature.

19. Ans:

(i) Given, $P = (3, 10)$

Check LHS and RHS.

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

RHS: $x^3 + x + 1 = 3^3 + 3 + 1$

$= 27 + 3 + 1$

$\equiv 8 \pmod{23}$

Since LHS = RHS, point P lies on the curve.

(ii) Formula for slope,

$$\lambda = \frac{3x_1^2 + a}{2y_1} \pmod{23}$$

For $P = (3, 10)$

$$\lambda = \frac{3x3^2+1}{2(10)} = \frac{28}{20} \pmod{23}$$

Find inverse of 20 mod 23: inverse is 15

$$\lambda = 28 \times 15 \equiv 6 \pmod{23}$$

Now find new point:

$$x_3 = \lambda^2 - 2x_1 = 6^2 - 6 = 30 \equiv 7$$

$$y_3 = \lambda(x_1 - x_3) - y_1 = 6(3-7) - 10$$
$$= -34$$
$$\equiv 12$$

So, $2P = (7, 12)$

(iii) Slope formula,

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{7-10}{9-3}$$
$$= -3/6 \pmod{23}$$

inverse of 6 mod 23 is 4.

$$\lambda = -3 \times 4 = -12 \equiv 11$$

Now, $x_3 = \lambda^2 - x_1 - x_2 = 11^2 - 3 - 9 = 109$
$\equiv$ to 17

$$y_3 = \lambda(x_1 - x_3) - y_1$$
$$= 11(3-17) - 10$$
$$= -164 \equiv 20$$

So, $P + Q = (17, 20)$

20. Ans:

Public key is found by scalar multiplication,

$$Q = dG = 9G$$

by repeated point addition on the curve (adding G nine times):

$$Q = (16, 9)$$

(1) Given, Hash $H(M) = 8$

Nonce $K = 3$

First Compute, $KG = 3G = (14, 24)$

take $r = x \bmod n$

$$r = 14 \bmod 19 = 14$$

(ii)

ECDSA formula,

$$S = k^{-1}(H(M) + d \cdot r) \bmod n$$

For inverse of $K = 3 \bmod 19$:

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

Now,

$$S = 13 \times (8 + 9 \times 14) \bmod 19$$

$$S = 13 \times (8 + 126) \bmod 19$$

$$S = 13 \times 134 \bmod 19$$

$$S = 13 \times 1 \bmod 19$$

$$= 13$$

So, signature is, $(r, s) = (14, 13)$

(iii) Compute: $w = s^{-1} \bmod 19$

inverse of 13 mod 19 is 3.

$$u_1 = H(M) \cdot w = 8 \times 3 = 24 \equiv 5$$

$$u_2 = r \cdot w = 14 \times 3 = 42 \equiv 4$$

now Compute,

$$u_1 G + u_2 Q = 5G + 4Q$$

CamScanner

After point addition on the curve

$= (14, 24)$

Take X- Coordinate,

$X \bmod 19 = 14$

This equals r, so the signature is valid.

21. Ans:

A cryptographic hash function (like SHA-256) takes an input of any size and produces a fixed-size output called a hash. It is designed to be secure and fast.

Essential characteristics of a secure hash functio

1. Pre-image resistance:

It is very difficult to find the original message from the hash value. So the hash works like a one-way function.

CamScanner

2. Second pre-image resistance: If we know one message and its hash, it is very hard to find another different message that given the same hash.

3. collision resistance: It is very difficult to find two different messages that produce the same hash.

4. Deterministic: The same output always given the same hash.

5. Fixed length output: Whatever the input size, the output size is fixed.

6. Avalanche effect: A small change in the input creates a completely different hash.

7. Fast computation: The hash value can be calculated quickly.

(11) Impact of hash output length on security:

The length of the hash directly affects security.

→ For an n-bit hash, breaking pre-image resistance needs about 2^n attempts.

→ For collision attack, it needs about $2^{n/2}$ attempts

So For SHA-256:

→ Pre-image attack $\approx 2^{256}$ attempts

→ Collision attack $\approx 2^{128}$ attempts

This is extremly difficult, so SHA-256 is very secure. A longer hash means stronger security, but it needs a little more storage and time.

(iii) Real world applications of SHA

→ Digital signature:
this ensure integrity, Authentication and non-repudiation.

→ Block chain systems
this provides security, Immutability, easy verification.

→ Password storage
website store the hash of a password instead of the real password for security.

→ Data integrity checking
used to verify that files or message are not changed.

IT-24660

22. Ans:

Galoin field: A Galoin field is a set of finite elements where addition, subtraction, multiplication and division are defined and closed.

There are two common types,

1. GF(P):
→ Here, p is a prime number
→ The field contains numbers from 0 to p-1
→ All calculation are done mod p.

2. GF(2^n):
→ Used mostly in digital systems.
→ Elements are represented as binary numbers (0 and 1)
→ Arithmetic is done using polynomials modulo an irreducible polynomial.
→ Very suitable for computer hardware because Computers use binary.

Use in cryptography

1. Elliptic Curve Cryptography (ECC):
ECC uses elliptic curve equations defined over

finite fields. All point addition and multiplications are done using field arithmetic. Because the field is finite, the curve has a limited number of points, which makes solving the discrete logarithm problem very hard — This provides strong security.

2. AES (Advanced Encryption Standard)

→ AES uses $GF(2^8)$
→ Each byte (8 bits) is treated as an element of $GF(2^8)$.
→ Operations like the subBytes step use multiplicative inverse in $GF(2^8)$

This finite field arithmatic makes AES secure and resistant to many attacks.

Importance of field arithmatic :

Field arithmetic is important because,

→ It ensures mathematical structure and Consistency.

→ It ensures mathematical structure and Consistency.
→ Every non Zero element has a multiple inverse, which is necessary for encyption and decryption.
→ It creates hard mathematical problems that attackers cannot easily solve.
→ It allows secure operations in both hardware and software.

23. Ans:
(i) Shortest Vector Problem (SVP)
A lattice is a grid of points in space made by combining some base vectors.
The SVP means find the shortest non-Zero vector in that lattice.
This problem is very hard to solve, especially when the lattice dimension is large.
lattic-based cryptography builds its security on this hardness.

If an attacker can solve SVP easily, they can break the cryptosystem.

(11)

→ RSA security depends on integer factorization.

→ ECC security depends on the elliptic curve discrete logarithm problem.

→ A quantum algorithm called Shor's algorithm can solve both of these problems quickly.

So,
Quantum Computer → Can break RSA and ECC.

Quantum Computer → Cannot efficiently solve lattice problem like SVP.

That is why lattice-based cryptography is safer for the post-quantum era.

CamScanner

(iii)

lattice Based Cryptography:

→ Uses hard mathematical problems (like SVP)

→ runs on normal Computers.

→ Designed to resist quantum attacks.

Quantum Cryptography:

→ Uses rules of quantum physics (like photon behavior)

→ Example: Quantum key Distribution

→ detect eavesdropping using physics laws.

So, LBC = math based security

QC = physics " " .

IT-24660

24. Ans:

→ The lsf LFSR has m memory bits.
→ Each new bit is made using XOR of previous m bits.
→ Since it works over GF(2), each bit is either 0 or 1.

So, the state of the LFSR at any time is an m bit vector.

Because each of the m bits can be 0 or 1

Total possible states = 2^m

But one state is special:

→ The all-Zero state always produces Zero again.
→ So once it enters this state, it stays there forever.

Therefore, this state is not useful for generating a long sequence.

So usable states, $2^m - 1$

The reduced recurrence relation defines a characteristic polynomial,

$$f(x) = x^m + c_1 x^{m-1} + c_2 x^{m-2} + \cdots + c_m$$

If the polynomial is primitive over GF(2) then,

→ it generates all possible non zero states before repeating.

→ The LFSR cycles through every non-zero m bit state exactly once.

That means the sequence visits,

$$2^m - 1$$

different states before repeating.

Since, there are only $2^m - 1$ non zero states, and a primitive polynomial makes the LFSR go through all of them,

The maximum possible period is $2^m - 1$

25. Ans:

1. key generation

→ choose random matrix A

→ choose small secret vector S

→ choose small error vector e

→ Compute,

$$b = A \cdot s + e \pmod{q}$$

→ public key : P_k = (A, b)

→ Private " : S_k = S

Security comes from learning with errors:

Recovering s from (A, b) is hard because of the added noise e.

This hardness relates to worst-case lattice problems like shortest vector problem (SVP)

2. Signing

→ Hash message m → h

→ Use private key S

→ Generate random small vector y.

CamScanner

→ Compute Commitment : $u = A \cdot y \pmod q$

→ Compute challange : $c = H(u, M)$

→ Compute response : $Z = y + cs$

Signature = (z, c)

3. Verification

Using public key (A, b):

→ Compute : $A \cdot Z - c \cdot b$

→ Check if result matches commitment condition.

→ Check if Z is small.

if valid → accept signature.

(ii) Signing Message M with p-k and s-k

Given:

→ Message M

→ public key P.k = (A, b)

→ Private key S-k = S

Step-1 : Compute hash

$h = H(m)$

Step-2 : Choose random small vector y

Step-3 : Compute $U = A \cdot y$

Step-4 : Compute challange

$$C = H(u, M)$$

Step-5 : Compute response

$$Z = y + cs$$

Signature = (z, c)

Role of LWE in security

→ Attacker sees (A, b)

→ $b = A \cdot s + e$

→ Finding s requires solving noisy linear equations.

→ This reduces to hard lattice problems like SVP.

→ No efficient classical or quantum attack known.