1. How does shore's algorithm threaten the security of RSA and Eliptic Curve Cryptography (ECC), and what are the potential consequences of current digital infrastructure

Answer:

shore's algorithm is a quantum algorithm capable of factoring large integers and solving discrete logarithms exponentially faster then the best known classical algorithm.

RSA's security relies on the difficult of factoring large numbers, and Elliptic Curve Cryptography (ECC) depend on the hardness of the discrete algorithm problem. On classical computers, these problems are computationally infeasible for sufficiently large key sizes, ensuring security. However, shore's algorithm, when executed on the sufficient powerful quantum computer

• Can solve both problems efficiently

This means that if large-scale quantum computers become practical they could break RSA and ECC encryption, rendering most of today's public-key cryptography obsolete. The potential consequences for current digital infrastructure are severe. — secure communications, online banking, digital signatures, and data confidentiality could all be compromised. Past encrypted data could also be decrypted if it has been stored ("harvest now, decrypt later"), posing long-term risks. This threat underscores the urgent need to transition to post-quantum cryptographic algorithm that are resistant to quantam attacks.

Discuss the role of quantum key distribution (QKD) in future cryptographic system. How does it differ from classical public key encryption?

Role of quantum key distribution (QKD) in future Cryptographic systems

1. Secure the Exchange using Quantum physics- QKD uses quantum mechanics principles to securely exchange encryption keys between parties.

2. Unconditional security: Any attempt to intercept Unlike classical encryption, whose security relies on computational difficulty QKD's security is based on the law of physics, making it immune to both classical and quantum computational attacks.

3. Eavesdropping Detection: Any attempt to intercept quantum keys distribution the quantum states, alerting the

the communicating parties to the presence of an eavesdropper.

Complement to post quantum Cryptography- QKD can be integrated with classical encryption methods to create hybrid systems, probide additional layer of security in a post quantum coorld.

Protection against future quantum computers- QKD enswes that even if quantum computers break classical algorithm like RSA or ECC, the exchange key remains secure.

Key differences from classical public-key encryption:

- Security Basis: classical public-key encryption relies on mathmatical problem, QKD relies on quantum mechanics.

- **Vulnerability:** Classical Methods can be broken by shor's algorithm on quantum computers, QKD is immune to such attacks.

- **Eavesdropping detection:** classical systems cannot inherently detect interception. QKD detects it automatically.

Infrastructure Requirement: QKD requires specialized quantum communication channels, whereas classical systems work over standard internet infrastructure.

3. What are the main difference between lattice based cryptography and traditional number-theoretic approaches like RSA, particularly in the context of quantum resistance.

Answer:

1. Underlying hard problems:

- RSA:
  → security is based on difficulty of integer factorization.
- Lattic-based Cryptography:
  → security reelies on hard lattic Problems, such as shortes vector problem (SVP) or learning with errors (LIWE), which mean remain hard even for quantum computer.

2. Vulnerability to Quantum Algorithm:

- RSA:
  → Not quantum-safe - shor's algorithm can factor large integers efficiently, breaking RSA entirely.

* Lattice-based
  → Quantum-resistant — No known quantum algorithm can efficiently solve lattice problems at the scales used in cryptography.

3. Performance and efficiency:
   - RSA:
     → smaller key size for classical security.
     → slower key generation and encryption for large key sizes
   - Lattice-based:
     → larger keys (often ten of kilobytes) but faster encryption/decryption and better scalability for post-quantum use.

4. Standardization and Adoption
   - RSA:
     → Mature, widely deployed in TLS, digital signatures, etc.
     → No future-proof against quantum attacks.

- lattice-based:
  - → Actively being standardized
  - → designed for post-quantum security

Security Assumptions:

- RSA:
  - → signal assumption: factoring large integers is hard.

- Lattice-based:
  - → Based on worst-case hardness assumption over lattices, which are generally considered more robust and versatile

4. Developed a python-based PRNG that uses the current system time and a custom base seed value. Write complete program and corresponding output value.

Answer:

Python Program:

```
import time
def custom_prng(seed, count=5):
    current_time = time.time_ns()
    mixed_seed = (seed ^ current_time) &
    0xFFFFFFFFFFFFFFFF

    a = 6364136223846793005
    c = 1
    m = 2**64

    numbers = []
    for _ in range(count):
        mixed_seed = (a* mixed_seed + c)%m
        numbers.append(mixed_seed &
        0xFFFFFFFF)
    return numbers
```

```
if __name__ == "__main__":
    seed = int(input("Enter your custom seed value: "))
    random_numbers = custom_prng(seed, count=5)
    print("\nGenerated Random Numbers:")
    for num in random_numbers:
        print(num)
```

Sample Output:

```
Enter your custom seed value: 12345

Generated Random Numbers:
3093706321
3243939968
1602667065
2262141050
2866349687
```

How it works:

- Current time source → time.time_ns() give nanoseconds since epoch.

2. Custom seed Mixing → XOR with user seed to enswre unique starting state.

3. LCG Algorithm – Uses a deterministic formula to generated asequence.

$$X(n+1) = (a * Xn + c) \bmod m$$

4. Output → Lower 32 bits for manageable integers.

5. Explain the sieve of Eratosthenes algorithm and use it to find all prime number. to be a Carmichale number.

Answer:

Sieve of Eratosthenes:

Efficient method to find all primes $\leq n$ by repeatedly marking multiples of known primes.

step:

1. List numbers from 2 to n (all prime at first).
2. Start at p=2, mark its multiples as composite.
3. Move to next unmarked number repeat until unit $p^2 > n$.
4. Remaining unmarked number are primes.

Python Example (n=50):

```
def sieve(n):
prime = [True] * (n+1)
Prime[0] = Prime[1] = False
p = 2
while p*p <= n:
    if prime[p]:
        for i in range(p*p, n+1, p):
        Prime[i] = False
p += 1
```

return [i for i,v in enmerate (Prime) if v]

Print [sieve (50)).

Output:

[2,3,5,7,11,13,17,19,23,29,31,37,41,43,47]

Time Complexity:

- Sieve: $O(n \log \log n)$ - faster for large ranges
- Trial division: $O(n\sqrt{n})$ - slower.

6. State and explain the necessary and sufficient conditions for a composite number to be a Carmichael number. Then verify whether the number $n = 561$ and $n = 1105$ and $n = 1729$ are carmichael number.

Answer:

Korselt's criterion (necessary and sufficient)

A composite integer n is a Carmichael number iff:

1. n is square-free (no prime square divides n) and

2. For every prime p dividing n, $(p-1)$ divides $(n-1)$

if both hold then $a^{n-1} \equiv 1 \pmod{n}$ for all $\gcd(a, n) = 1$, so n is Carmichael).

Verify the three Number:

1. $n = 561$. Factor $561 = 3 \cdot 11 \cdot 17$ (square-free)
   check: $2 \mid 560$; $10 \mid 560$; $16 \mid 560$. All true $\Rightarrow$
   561 is carmichael.

2. $n = 1105$. Factors: $1105 = 5 \cdot 13 \cdot 17$ (square-free)
   check: $4 \mid 1104$; $12 \mid 1104$; $16 \mid 1104$ All true
   $\Rightarrow$ 1105 is carmichael.

3. $n = 1729$. Factor: $1729 = 7 \cdot 13 \cdot 19$ (square-free).

All three - 561, 1105 and 1729 - satisfy Korselt's criterion, so each is a carmichael number.

Answer to the Q: No: 7

1. Z_{11} with $(+, \cdot)$

→ Since 11 is a prime, Z_{11} forms a filed under addition and multiplication modulo 11.

→ A filed is also a ring, so yes, it is a ring.

2. $(Z_{37}, +)$

→ Under additional modulo 37: closure, associativity, identify element (0), inverses, and communtativity all hold.

→ therefore, yes it's an Abelian group.

3. $(Z_{35}, \times)$

→ multiplication modulo 35 is not a group because 0 lack multiplicative inverse

→ Hence, no, it's not an Abelian group under multiplication.

Ans to the Q: NO: 8

We want the remainder when -52 is reduced modulo 31.

In modular arithmetic, the remainder r is defined such that:

$$-52 \equiv r \pmod{31} \text{ and } 0 \leq r < 31$$

step-1: Find a multiple of 31 close to -52.

- $31 \times (-2) \equiv -62$ is the nearest multiple of 31, less than -52.

step-2: Adjust to find the remainder

: Difference

$$-52 - (-62) = -52 + 62 = 10$$

step 3:

- This means:

$$-52 \equiv 10 \pmod{31}$$

- So when -52 is reduced modulo 31, the remainder is 10.

Ans to the Q: No: 9

Multiplicative inverse of 7 mod 26 (Extended Euclidean Algorithm):

we want $7x \equiv 1 \pmod{26}$

using extended Euclidean Algorithm:

$$26 = 3(7) + 5$$

$$7 = 1(5) + 2$$

$$5 = 2(2) + 1$$

$$2 = 2(1) + 0$$

Now back-substitute:

$$1 = 5 - 2(2)$$

$$1 = 5 - 2(7 - 1(5)) = (5)3 - 2(7)$$

$$1 = 3(26 - 3(7)) - 2(7) = 3(26) - 11(7)$$

So,

$$-11(7) \equiv 1 \pmod{26}$$

$$\Rightarrow x \equiv -11 \equiv 15 \pmod{26}$$

Inverse: 15

Answer to the Q: No: 10

To Evaluate $(-8 \times 5) \bmod 17$:

step 1: Multiply

$$(-8) \times 5 = -40$$

step 2: Convert negative to positive modulo:

$$-40 \bmod 17$$

$$= -40 + 3(17)$$

$$= -40 + 51$$

$$= 11$$

Result: 11

Simplifying negative modular multiplication:

- Replace negative numbers with thier positive equivalents:

$$-8 \bmod 17 = 9$$

Then multiply:

$$9 \times 5 = 45$$

$$= 45 \bmod 17$$

$= 11$

so both was given 11.

Ans to the Q: No: 11

Bézout's Theorem (Statement):

For any two integers a and b, not both zero, there exist integers x and y such that:

$$\gcd(a,b) = ax + by$$

Thes integers x and y called Bézout's coefficients.

Proof:

Using the euclidean Algorithm:

① let $d = \gcd(a,b)$

② By the euclidean Algorithm

$$a = bq_1 + r_1 \; ; \; 0 \leq r_1 < b$$

$$b = r_1 q_1 + r_2$$

Continoue until $r_K = 0$. The last

non-zero remainder is d.

3. Back-substitute each remainder in terms of a and b.

4. This gives: $d = ax + by$ for some integers x, y.

Thus Bézout's identity is proved.

Using Bézout's theorem to find Inverse of 97 mod 385:

We want x such that

$$97x \equiv 1 \pmod{385}$$

This means:

$$97x + 385y = 1$$

Here, x will be the multiplicative inverse of 97 modulo 385.

Step 1: Apply extended Euclidean Algorithm

$$385 = 97(3) + 94$$
$$97 = 94(1) + 3$$
$$94 = 3(31) + 1$$
$$3 = 1(3) + 0$$

gcd = 1 (so, inverse exists).

Step 2: Back-substitute

From $94 = 3(31) + 1$:

$$1 = 94 - 3(31)$$

From $97 = 94(1) + 3$

$$\Rightarrow 3 = 97 - 94:$$

$$1 = 94 - (97 - 94)(31)$$

$$1 = 94 - 97(31) + 94(31)$$

$$1 = 94(32) - 97(31)$$

From $385 = 97(3) + 94$

$$\Rightarrow 94 = 385 - 97(3):$$

$$1 = (385 - 97(3)) - 97(31)$$

$$1 = 385(32) - 97(96) - 97(31)$$

$$1 = 385(32) - 97(127)$$

step 3: Identify Inverse

we have:

$$1 = 385(3'2) - 97(127)$$

so:

$$-127 \cdot 97 \equiv 1 \pmod{385}$$

Thus:

$$x \equiv -127 \pmod{385}$$

$$x \equiv 258 \pmod{385}$$ (since $-127 + 385 = 258$)

Multiplicative inverse : 258

Ans to the 8.12

Bezout's identity and $43x \equiv 1 \pmod{240}$

Bezout's identity: for integers a, b there exist integers x, y such that

$$ax + by = \gcd(a, b)$$

If $\gcd(a, b) = 1$, this gives $ax \equiv 1 \pmod{b}$, meaning x is the modular inverse of a mod b.

Here $\gcd(43, 240) = 1$ using the extended euclidean Algorithm:

$$240 = 5 \cdot 43 + 25$$
$$43 = 1 \cdot 25 + 18$$
$$25 = 1 \cdot 18 + 7$$
$$18 = 2 \cdot 7 + 4$$
$$7 = 1 \cdot 4 + 3$$
$$4 = 1 \cdot 3 + 1$$

Back-substitution gives:

$$1 = 67 \cdot 43 - 12 \cdot 240.$$

So. $x \equiv 67 \pmod{940}$.

Answer to the q: 13

Fermat's Little theorem (FLT).

If p is prime, then for any integer a,

$$a^p \equiv a \pmod{p}$$

If $\gcd(a, p) = 1$ then

$$a^{p-1} \equiv 1 \pmod{(p)}$$

FLT is used for primality testing: if $a^{n-1} \not\equiv 1 \pmod{n}$, then n is composite.

However, Carmichael number (e.g $561 = 3 \cdot 11 \cdot 17$) can pass the test for many a even though they are composite.

Compute $5^{123} \pmod{175}$

factor $175 = 25 \cdot 7$

- Mod 25: $5^2 \equiv 0$, so, $5^{123} \equiv 0 \pmod{25}$
- Mod 7: By FLT, $5^6 \equiv 1$ since $123 \equiv 3 \pmod{6}$

$$5^{123} \equiv 5^3 = 125 \equiv 6 \pmod{7}$$

Chinese Remainder Theorem:

We need $N \equiv 0 \pmod{25}$ and $N \equiv 6 \pmod{7}$

Testing multiples of 25: $125 \equiv 6 \pmod{7}$

$$5^{123} \pmod{175} = 125$$

Final Answer:

- $x \equiv 67 \pmod{240}$
- $5^{123} \pmod{175} = 125$

= o =

Answer to the 9:14

Chinese Remainder Theorem (Statement and proof sketch) and solution

statement:

Let, $n_1, \ldots\ldots, n_k$ be pairwise coprime positive integers and set $N = n_1 \cdots n_k$ for any integers $a_1, \ldots\ldots, a_k$ the system

$$x \equiv a_i \pmod{n_i} \quad (i = 1, \ldots\ldots, k)$$

has a unique solution modulo N.

Proof sketch (constructive):

For each i put $N_i = N/n_i$. Because $\gcd(N_i, n_i) = 1$ there exists an inverse y_i with $N_i y_i \equiv 1 \pmod{n_i}$. Then

$$x \equiv \sum_{i=1}^{k} a_i N_i y_i$$

satisfies $x \equiv a_i \pmod{n_i}$ (all other terms vanish mod n_i since they contain the factor n_i). Uniqueness

modulo N follows because any two solution differ by a multiple of every n_i, hence by a multiple of N.

Solve

$$x \equiv 2 \pmod 3$$
$$x \equiv 3 \pmod 5$$
$$x \equiv 2 \pmod 7$$

Here $n_1 = 3$, $n_2 = 5$, $n_3 = 7$, are pairwise coprime. $N = 3 \cdot 5 \cdot 7 = 105$

Compute

$N_1 = 35$

$N_2 = 21$

$N_3 = 15$

find inverses:

- $35 \bmod 3 \equiv 2$. Inverse of 2 mod 3 is 2 (since $2 \cdot 2 \equiv 1$)
  so, $y_1 = 2$
- $21 \bmod 5 \equiv 1$. Inverse is 1 also
  $y_2 = 1$

• $15 \bmod 7 = 1$; inverse is 1,
so, $y_3 = 1$

for solution:

$x = 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 = 140 + 63 + 30$

$= 233$

Reduce modulo 105: $233 \equiv 23 \pmod{105}$

Check: $23 \equiv 2 \pmod 3$,

$23 \equiv 3 \pmod 5$

$23 \equiv 2 \pmod 7$

So solution is,

$x \equiv 23 \pmod{105}$.

= o =

Ans for the 9:15

CIA triad:

The CIA triad = Confidentiallity, Integrity, Availability, three core goals of information security.

- **Confidentiallity:** Ensure data is seen only by authorized parties. It prevents information disclosure

- **Integrity:** Ensure data and systems are accurate and unchanged except by authorized actions (examples: checksums, hashes, digital signatures, versioning). It prevents tampering and undetected modification.

- **Availablity:** Ensure authorized users can access data/services when needed (examples: redundancy, backups, load balancing, DDos protection). It prevents outages.

and service denial.

Together they guide design decisions. A secure system balances all three according to risk and business needs.

=:o:=

Ans to the Q: NO: 16

Steganography vs Cryptography & common techniques

- **Difference:**
- Steganography hides the existence of a message within another medium so that outsiders don't even know a message is present.

- Cryptography scrambles the message into an unreadable formate so that even if it's intercepted it cannot be understood without a decryption key.

Common steganography techniques in digital Media:

1. Least significant Bit (LSB) Intersection - Modifying the last bits of pixel or sound values to hide data.

2. Image Palette Modification - slightly altering the color palette to encode information.

3. Audio Echo Hiding: Embedding data in sound by introducing subtle echoes.

4. Video Frame Manipulation - High hiding data in specific frames or pixel patterns.

5. Whitespace Steganography - Using spaces and tabs in text files to hide binary data.

= o =

Ans to the Q. No. 11

Aspect	phishing	Malware	Denial-of service
Method	Tricking users into giving sensitive information via fake emails, website or messages.	Malicious software that infiltrates system	Overwhelming a network/server with excessive requests to make it unavailable.
Goal	steal credentials, personal data or financial info	Disrupt, damage steal, or control system/data.	Render services unavailable and disrupt operation
Impact	Loss of personal or financial data identity theft	Data loss, system damage, unauthorized access, ransom demands	service downtime loss of revenue, customer dissatisfaction.
Detection Difficulty	Medium - can often be spotted with awareness	Medium to high - may be hidden in files or process	Low - effects are immediate and visible.

Ans to the Q. No. 12

- Data minimization: Organizations should collect only the necessary data for a specific purpose, reducing the amount of sensitive information they store.
- Security measures: Requires encryption, pseudonymization, and strict access controls to safeguard personal data.
- Breach notification: Any data breaches must be reported to authorities within 72 hours and affected users must be informed promptly.
- User rights: Individuals have the right to access, correct, delete, or restrict the processing of their data.
- Accountability: Requires organizations to maintain records of data processing activities, conduct regular risk assessments, and appoint a Data Protection Officer (DPO) to oversee compliance.
- Penalties: Heavy fines can be imposed for non-compliance, encouraging organizations to adopt robust security practices and

Ans to the Q: No: 18

- Data minimization: Organizations can only collect strictly necessary for a specific purpose, reducing the amount of data
- Security measure: Requires encryption, pseudonymization, and strict access control to safeguard personal data.
- Breach notification: Any data breech must be reported to authorities within 72 hours and affected users must be informed promptly.
- User Rights: Individuals have the right to access, correct, delete, and transfer their data.
- Accountability: Requires documentation, risk assesment, somtimes, a data protection officer to ensure compliance
- Penalties: Heavy fines encourage organization to adopt strong cyber-security practices and prevent attacks

Ans to the Q: No: 19

DES (Data encryption standard):

- Input: 64-bit plaintext block; key is 64 bits only 56 bits are used (8 bits for parity)

- Step:

1. Initial permutation (IP):
Rearrange the bits into a standardize order

2. 16 Feistel round: In each round
   - split into L and R halves.
   - $L_i = R_{i-1}$
   - $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

3. Round function f:
   - Expand R from 32 to 48 bits (E-box)
   - XOR with 48 bit round key K_i

- Pass through S-boxes for non-linear substitution (48 bits → 32 bits).
- Apply a fixed permutation (P-box)

4. final permutation (IP^{-1}):
   Reverses IP to get ciphertext.

- Keys: subkeys K_1 ... K_{16} generated from the 56-bit key using shifting and permutation.

=o=

Ans to the Q: 20.

DES Round Calculation (Simplified with XOR-only f):

Given:

$R_0 = F0\,F0\,F0\,F0$

$K_1 = 0F\,0F\,0F\,0F$

$L_0 = AA\,AA\,AA\,AA$

Step:

1. Round function:

$f(R_0, K_1) = R_0 \oplus K_1 = F0F0F0F0$

$\oplus$ 0F0F0F0F

$= FFFFFFFF$

Answer:

$f(R_0, K_1) = FFFFFFFF$

$L_1 = F0F0F0F0$

$R_1 = 55555555$

Ans to the Q: 22

AddRoundKey (bytewise XOR only).

Input word: [0x1A, 0x2B, 0x3C, 0x4D]

Output word: [0x55, 0x66, 0x77, 0x88]

XOR per type:

- $0x1A \oplus 0x55 = 0x4F$
- $0x2B \oplus 0x66 = 0x4D$
- $0x3C \oplus 0x77 = 0x4B$
- $0x4D \oplus 0x88 = 0xC5$

Resulting word: [0x4F, 0x4D, 0x4B, 0xC5]

Ans to the Q: NO: 21

subbyte (using the given partial S-box) we look up each byte as row = high nibble, col = low nibble

- S[23] → 0xD4
- S[A7] → 0x63
- S[9C] → not provided in the partial table (Entry is missing)
- S[19] → 0xC6

Resulting word: [0xD4, 0x63, (unknown), 0xC6]

The third byte cannot be determined from partial S-box given.

=o=

Ans to the Q: No: 23.

Mix Colums with input Column

[01, 02, 03, 04) (hex)

AES matrix (row):

[02, 03 01 01 ; 01 02 03 01 ; 01 01 02 03 ; 03 01 01 02]

Over GF (2^8):

02 · x = xtime(x) and 03 · x = xtime(x) ⊕ x

Pre Compute:

Now:

- b0 = 02 · 01 ⊕ 03 · 02 ⊕ 01 · 03 ⊕ 01 · 04
  = 02 ⊕ 06 ⊕ 03 ⊕ 04
  = 03
- b1 = 01 · 01 ⊕ 02 · 02 ⊕ 03 · 03 ⊕ 01 · 04
  = 01 ⊕ 04 ⊕ 05 ⊕ 04
  = 04

- $b2 = 01.01 \oplus 01.02 \oplus 02.03 \oplus 03.04$
  $= 01 \oplus 02 \oplus 06 \oplus 0C$
  $= 09$

- $b3 = 03.01 \oplus 01.02 \oplus 01.03 \oplus 02.04$
  $= 03 \oplus 02 \oplus 03 \oplus 08$
  $= 0A$

output column: [03, 04, 09, 0A)

in binary: (00000011, 00000100, 00001001, 00001010)

Ans to the Q: No 24

How AES-OFB works and synchronization

- OFB turns AES into a synchronous stream cipher:

$S0 = IV$

$Si = AES_K(Si-1)$

Chiphertext $Ci = Pi \oplus Si$

Decryption is the same XOR with same keystream.

- Sync requirement: Sender and reciver must share the same IV/nonce and be aligned to the same block index Bit errors flip only the corresponding bits of Pi ; no further propagation but lost/inserted blocks breaks alignment until re-synced.

- $b2 = 01.01 \oplus 01.02 \oplus 02.03 \oplus 03.04$
  $= 01 \oplus 02 \oplus 06 \oplus 0C$
  $= 09$

- $b3 = 03.01 \oplus 01.02 \oplus 01.03 \oplus 02.04$
  $= 03 \oplus 02 \oplus 03 \oplus 08$
  $= 0A$

output column: [03, 04, 09, 0A)

in binary: (00000011, 00000100, 00001001, 00001010)

Ans to the Q: No 24

How AES-OFB works and synchronization

- OFB turns AES into a synchronous stream cipher:

$S0 = IV$

$S_i = AES\_K(S_{i-1})$

Chiphertext $C_i = P_i \oplus S_i$

Decryption is the same XOR with same keystream.

- Sync requirement: Sender and reciver must share the same IV/nonce and be aligned to the same block index Bit errors flip only the corresponding bits of P_i; no further propagation but lost/inserted blocks breaks alignment until re-synced.

Ans to the q: NO: 25

Error Propagation and integrity (CBC vs CFB)

- CBC: If C_i has a 1-bit error then
  - → P_i (block i) become completly grabled, and
  - → the same bit in P_{i+1} flips (due to XOR with corrupted C_i),
  - → recovery after block i+1.
  - → ⇒ some propagation → hurts integrity beyond a single block.

- CFB (Full block): If C_i has a 1-bit error, then
  - Only that bit in P_{i+1} filips but

- Pi+1 becomes completly garbled (since AES-K(Ci) is corrong);
- recovery after block i+1. (For s-bit CFB, corruption lasts for one block of shift-register length)
- ⇒ Propagation to the next block
- Compare:
  - Propagating modes: CBC, CFB (and PCBC)
  - Non-Propagating (bit-flip only): OFB, CTR (and ECB confines damage to its own block)

Overall propagation means a single transmission error degrades multiple decrypted blocks, reducing the integrity of the recovery message.

Ans to the Q: 26

I would recommend AES in CTR (Counter) mode for encrypting large files with parallel processing.

- Reasoning:
  - → ECB: ECB mode encrypts each block independently, but identical plaintext blocks produce identical ciphertext blocks. This leaks patterns, making it insecure for most use cases.
  - → CBC: CBC is secure, but encryption is sequential each block depends on the previous one - so it cannot be parallelized efficiently (although decription can)

Extra advantage: Supports random access encryption (you can decrypt a block without processing

previous blocks), which is useful for large files.

Ans to the 9:27

RSA Encryption & Decryption:

Given, $M = 1$ (message "A")

public key $e = 5$, $n = 14$

Private key $d = 11$

Encryption:

$$C = M^e \bmod n$$

$$= 1^5 \bmod 14$$

$$= 1 \bmod 14$$

$$= 1$$

Decryption:

$$M = C^d \bmod n$$

$$= 1^{11} \bmod 14$$

$$= 1 \bmod 14$$

$$= 1$$

Ans to the Q: 26

I would recommend AES in CTR (Counter) mode for encrypting large files with parallel processing.

- Reasoning:
  - → ECB: ECB mode encrypts each block independently, but identical plaintext blocks produce identical ciphertext blocks. This leaks patterns, making it insecure for most use cases.
  - → CBC: CBC is secure, but encryption is sequential each block depends on the previous one - so it cannot be parallelized efficiently (although decription can)

Extra advantage: Supports random access encryption (you can decrypt a block without processing

previous blocks), which is useful for large files.

Ans to the Q: 27

RSA Encryption & Decryption:

Given,

$M = 1$ (message "A")

Public key $e = 5$, $n = 14$

Private key $d = 11$

Encryption:

$$C = M^e \bmod n$$

$$= 1^5 \bmod 14$$

$$= 1 \bmod 14$$

$$= 1$$

Decryption:

$$M = C^d \bmod n$$

$$= 1^{11} \bmod 14$$

$$= 1 \bmod 14$$

$$= 1$$

So, the ciphertext is 1, and after decryption, we recover 1 (which correspond to 'A')

Ans to 9.28

Given:

$H(M) = 5$ (message hash)

private key $d = 3$

$n = 33$

Signature generation:

$$S = H(M)^d \bmod n$$

$$= 5^3 \bmod 33$$

$$= 5^3$$

$$= 125$$

$125 \bmod 33$

$= 125 - (33 \times 3)$

$= 125 - 99 = 26$

signature = 26

The digital signature is 26, which the recipent can verify using the public key.

Ans to Q: No 29

Diffie-Hellman Key Exchange:

Given,

prime modulus $p = 17$

Base (generator) $g = 3$

Alega's private key $a = 4$

Badol's private key $b = 5$

Alega's public key:

$$A = g^a \bmod p$$
$$= 3^4 \bmod 17$$
$$= 3^4$$
$$= 81$$

So, the ciphertext is 1, and after decryption, we recover 1 (which correspond to "A")

Ans to Q: 28

Given:

H(M) = 5 (message hash)

private key d = 3

n = 33

Signature generation:

$$S = H(M)^d \bmod n$$

$$= 5^3 \bmod 33$$

$$= 5^3$$

$$= 125$$

$125 \bmod 33$

$= 125 - (33 \times 3)$

$= 125 - 99 = 26$

signature = 26

The digital signature is 26, which the recipent can verify using the public key.

Ans to Q: No 29

Diffie-Hellman key Exchange:

Given,

prime modulus $p = 17$

Base (generator) $g = 3$

Alega's private key $a = 4$

Badal's private key $b = 5$

Alega's public key:

$$A = g^a \pmod p$$
$$= 3^4 \bmod 17$$
$$= 3^4$$
$$= 81$$

81 mod 17

$= 81 - (17 \times 4)$

$= 81 - 68$

$= 13$

So, A = 13

Badol's public key:

$B = g^b \bmod p$

$= 3^5 \bmod 17$

$= 35$

$= 243$

243 mod 17

$= 243 - (17 \times 14)$

$= 243 - 238$

$= 5$

So, B = 5

(a) Ans to the Q: 30

Given,

$H(x) = (\sum$ ASCII values of characters in $x) \bmod 100$

Message "AB":

ASCII(A) = 65

ASCII(B) = 66

$H("AB") = (65+66) \bmod 100$

$= 131 \bmod 100$

$= 31$

Message "BA":

ASCII(B) = 66

ASCII(A) = 65

$H("BA") = (66+65) \bmod 100$

$= 131 \bmod 100$

$= 31$

Both messages have the same hash value (31) even though the order of characters is different

Implication: This function is not collision-resistant - different input can produce the same output easily. This is why cryptographic hash functions are more complex!

=o=

Ans to the Q: 31

Formula:

$$MAC = (Message + Secret\ key) \bmod 17$$

- Message = 15
- Secret key = 7

$$MAC = (15+7) \bmod 17$$
$$= 22 \bmod 17$$
$$= 5$$

So, MAC = 5

Now if the attacker change the message to 10 but doesn't know the key:

→ New message = 10

→ Without key, the cannot compute MAC correctly.

→ Correct MAC would be:

$$= (10+7) \bmod 17$$

$= 17 \bmod 17$

$= 0$

→ Since the attacker doesn't know the key (7), they can't forget the correct MAC easily. They would have to guess, which has only a 1/17 chance of being correct.

Ans to the Q:32

Step in TLS handshake and key establishment ←

① ClientHello → client proposes supported cipher suites, TLS version random value.

2. ServerHello: server selects cipher suits, sends random value, and server's certificate

3. Authentication: Client server verifies server's certificate using CA.

4. Key exchange: Using asymmetric cryptography, client and server agree on a shared session ~~day~~ key.

5. Session key established:- Both send "Finished" message encrypted with the session key to confirm handshake completion.

Symmetric keys are established securely because only the legitimate server with the private key (or ephemeral diffie-Hellman secret) can derive the same session key.

=:=

33 Answer:

Layered Architecture (Protocol stack) of SSH.

- Transport layer protocol (TCP + SSH-Transport): provides, encryption integrity, and server authentication.
- Users Authentication layer: verify client identity.
- Connection layer: Manages multiple logical channels over one encryption connection.

Each layer ensures modularity: secure channel first, then authentication, then communication services.

Ans to the Q: 34

TLS Handshake steps -

1. Client Hello
2. Server Hello and Certificate
3. Key Exchange (RSA/DH/ECDHE)
4. Authentication of server (and client optionally)
5. Session Key derivation
6. Secure channel established, Finished message exchanged

After this encrypted communication start.

Ans to the Q: 3b

General form of Elliptic curve equation and use in cryptography

General form over a finite filed:

$$y^2 = x^3 + ax + b \pmod{p}$$

where $a, b \in F_p$ and $4a^3 + 27b^2 \neq 0$ (to avoid singular curves).

Used in cryptography because:

- Provides high security with small key size
- Efficient computation
- Foundation for secure key exchange (ECDH), digital signature (ECDSA), etc.

= 0 =

Answer to the Q: 36

Because the core hard problem in ECC (the Elliptic-curve - Discrete logarithm problem) scales much harder than integer factorization (RSA's problem). You get far more "security per bit". Rule of ~~xxx~~ thumb: ~256-bit ECC ≈ ~3072-bit RSA in classical security.

Ans to the Q: 37

Check if $P = (3, 6)$ lies on

$y^2 \equiv x^2 + 2x + 3 \pmod{97}$.

RHS for $x = 3$: $3^2 + 2 \cdot 3 + 3$

$= 27 + 6 + 3$

$= 36$.

LHS for $y = 6$: 6^2

$= 36$

Since $36 \equiv 36 \pmod{97}$, yes P is on curve

Ans to the Q: 38

ElGamal with $(P, g, h) = (23, 5, 8)$

message $m = 10$, random $k = 6$

$c_1 = g^k \bmod P$

$= 5^6 \bmod 23$

$5^2 = 25 \equiv 2$

$5^4 \equiv 4 \cdot 2 = 8 \Rightarrow c_1 = 8$

$h^k = 8^6 \bmod 23.$

$8^2 = 64 \equiv 18, \quad 8^4 \equiv 18^2 = 326 \equiv 2$

$8^6 \equiv 2 \cdot 18 = 36 \equiv 13.$

$c_2 = m \cdot h^k \bmod P = 10 \cdot 13 = 130 \equiv 15.$

Ciphertext $(c_1, c_2) = (8, 15).$

IoT devices are constrained (CPU, RAM, power, bandwidth). Lightweight crypto delivers confidentiality / integrity with the code size, low RAM, and low energy, so devices can stay secure without killing battery or boosting cost.

Example:

Ascon (NIST's LWC winner) for ~~AHI~~ AEAD and hashing, efficient on small MCUs.

Ans to the Q: No 40

Three common IoT-specific attack + Mitigation.

1. Firmware hijacking / malicious updates.

   Mitigations: secure boot, signed firmware and verified OTA, version pinning / rollback protection, supply chain code signing

2. physical tampering

   Mitigation: Disable / lock debug ports, secure enclosures and tamper-evident seals, secure elements / TPM,

3. Botnet.

   Mitigation: Unique strong credentials at manufacture, remove default passwords, disable unused services

Ans to the Q: No 40

Three common IoT-specific attack + Mitigation.

1. Firmware hijacking / malicious updates.

   Mitigations: secure boot, signed firmware and verified OTA, version pinning / rollback protection, supply chain code signing

2. physical tampering

   Mitigation: Disable / lock debug ports, secure enclosures and tamper-evident seals, secure elements / TPM,

3. Botnet.

   Mitigation: Unique strong credentials at manufacture, remove default passwords, disable unused services

(Telnet/UpnP); rate-limit and firewall; network segmentation; automatic patching.

=0=