Shafinaj Jerin Asha
IT-21031

Ques-1: How does Shor's Algorithm threaten the security of RSA and Elliptic Curve Cryptography (ECC), and what are the potential consequences for current digital infrastructure?

IT-21031

Ans:

Shor's Algorithm threatens the security of RSA and Elliptic Curve Cryptography (ECC) because it can efficiently solve the hard mathematical problems - integer factorization (for RSA) and elliptic curve discrete logarithm (for ECC) on which these cryptographic systems are based. While classical computers take an impractically long time to solve these problems, a quantum computer running shor's algorithm could break RSA and ECC in polynomial time, making them vulnerable.

The potential consequences for current digital infrastructure include:

- Decryption of secure communications (HTTPS, VPNs)
- Forgery of digital signatures used in authentication.

• Exposure of sensitive data in banking, government, and healthcare.

• Long-term risk from data harvested now and decrypted later when quantum computers are ready.

IT-21031

This creates an urgent need to adopt quantum resistant (post-quantum) cryptographic systems.

Ques-2: Discuss the role of quantum key distribution (QKD) in future cryptographic systems. How does it differ from classical public-key encryption?

Ans:

Quantum key Distribution (QKD) allows two parties to securely share a cryptographic key using the principles of quantum mechanics. The most well-known QKD protocol is BB84, which uses quantum bits (qubits) to detect any

eavesdropping attempts.

Differences from Classical Public-key Encryption:

* Classical encryption (RSA) relies on hard mathematical problems for security, while QKD relies on the laws of physics. IT-21031
* QKD can detect eavesdropping (due to quantum no-cloning and measurement disturbance) but classical methods cannot.
* QKD is used for key distribution, not for encrypting data itself.

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

~~* Lattice-based~~

Ans:

* Lattice-based cryptography is based on geometric problems in high-dimensional

space (shortest vector problem) which are believed to be hard even for quantum computers.

* RSA and ECC rely on integer factorization and discrete logarithms, which are vulnerable to shor's algorithm.

IT-21031

* Lattice cryptography is quantum-resistant and it's a strong candidate for post-quantum cryptography.

Ques-4: Develop a phython based PRNG that uses the current system time and a custom seed value. Write complete program and corresponding output.

Ans:

```
import time

def prng(seed):
    current_time = int(time.time()*1000)
    combined = seed ^ current_time
    combined = (combined * 1103515245+12345)
                & 0x7FFFFFFF
    return combined
```

```
seed_value = 12345
random_number = prng (seed_value)
print ("Generated pseudo_random number:"
                                random_number)
```

Sample output:

Generated pseudo_random number : 1620932541

IT-21031

Ques-5: Explain the sieve of Eratosthenes algorithm and use it to find all prime numbers less than 50. How does its time complexity compare to trial version?

Ans:

Sieve of Eratosthenes: An efficient algorithm to find all primes $\leq n$. It works by iteratively marking the multiples of each prime number starting from 2.

Algorithm:

1. Create a boolean list for numbers upto n.
2. start from p=2 mark all multiples of p as

False

3. Repeat until $p^2 > n$.

Python code for n = 50. :

IT-21031

```
def sieve(n):
    prime = [True for _ in range(n)]
    prime[0] = prime[1] = False
    p=2
    while p*p <n:
        if prime[p]:
            for i in range(p*p, n, p):
                prime[i]= False
        p+= 1
    return [i for i in False range(n) if prime[i]]
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 logn)
* Trial Division : o ($n\sqrt{n}$) (slower)

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

IT-21031

Ans:

A Carmichael number is a composite number n such that for all integers a coprime to n.

$a^{n-1} \equiv 1 \mod n$ (Fermat-like property)

Necessary and sufficient condition:

A composite number n is a Carmichael number if and only if:

* n is square-free
* For every prime p dividing n, (p-1) divides (n-1)

Check for:

* 561 = 3×11×17 ✓
* 1105 = 5×13×17 ✓
* 1729 = 7×13×9 ✓

In all cases:

* Composite
* Square-free
* (p-1) divides (n-1)

Ques-7: Determine whether the following are valid algebraic structures and justify your answer:

Is the set $\mathbb{Z}$ with operations + and × a ring

Yes, $\mathbb{Z}$ is a commutative ring with identity.

IT-21031

Are the sets $(\mathbb{R}, +)$ and $(\mathbb{Z}/n\mathbb{Z}, +)$ Abelian groups?

Yes, both are Abelian groups:

* $\mathbb{R}$ under addition: associative, identity 0, inverse $-a$, commutative.
* $\mathbb{Z}/n\mathbb{Z}$ under addition mod n: same properties hold.

Ques-8:

What is the remainder when $73 \times (-14)$ is reduced modulo 15?

Ans: $73 \times (-14) = -1022$

$-1022 \bmod 15$ = remainder when divided by 15

$-1022 \equiv 8 \pmod{15}$

Answer: 8

Ques-9: Determine the multiplicative inverse of 23 modulo 66 using Extended Euclidean Algorithm

Ans: Apply Extended Euclidean Algorithm:

gcd (66, 23) = 1 IT-21031

$23^{-1} \bmod 66 = x$ such that $23x \equiv 1 \pmod{66}$

using EEA:

$66 = 2 \times 23 + 20$

$23 = 1 \times 20 + 3$

$20 = 6 \times 3 + 2$

$3 = 1 \times 2 + 1$

$2 = 2 \times 1 + 0$

Back - substitute:

$1 = 3 - 1 \times 2$

$= 3 - 1 \times (20 - 6 \times 3)$

$= 7 \times 3 - 1 \times 20$

$= 7 \times (23 - 1 \times 20) - 1 \times 20 = 7 \times 23 - 8 \times 20$

$= 7 \times 23 - 8(66 - 2 \times 23) = 7 \times 23 - 8 \times 66 + 16 \times 23$

$= 23 \times (7 + 16) - 8 \times 66 = 23 \times 23 - 8 \times 66$

So, inverse = **23**

Multiplicative inverse of 23 mod 66 is 23.

Ques-10: State and prove Bezouts Theorem. Use it to find the multiplicative inverse of 97 modulo 385.

IT-21031

Bezout's Theorem:

For integers a and b, there exist integers x and y such that:

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

where, x and y are Bezout's coefficients

if $\gcd(a, b) = 1$, then x is the modular inverse of a ~~an~~ mod b.

Proof: Using the Extended Euclidean Algorithm.

Step-1: Apply the Euclidean Algorithm

Let a and b be integers and assume $a > b \geq 0$.

We perform repeated division:

$$a = bq_1 + r_1$$
$$b = r_1 q_2 + r_2$$
$$r_1 = r_2 q_3 + r_3$$
$$\vdots$$
$$r_{n-2} = r_{n-1} q_n + r_n$$
$$r_{n-1} = r_n q_{n+1} + 0$$

Then $\gcd(a, b) = r_n$

Step-2: Back-substitute to express gcd (a, b) a a linear combination of a and b.

Using the last few equations, we backtrack and express each remainder in terms of the previous remainders until we express gcd $(a,b) = r_n$ as:

$$r_n = ax + by$$

IT-21031

for some integers x and y. Hence, Bezout's identity is proven.

Use Extended Euclidean Algorithm for 97 and 385:

$385 = 3 \times 97 + 94$

$97 = 1 \times 94 + 3$

$94 = 31 \times 3 + 1$

Back substitute:

$1 = 94 - 31 \times 3$

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

$= 32 \times (385 - 3 \times 97) - 31 \times 97 = 32 \times 385 - 127 \times 97$

So,

$1 = 32 \times 385 - 127 \times 97$

$\Rightarrow -127 \times 97 \equiv 1 \mod 385 \Rightarrow$

$\Rightarrow$ Inverse of 97 is **258 ($-127 \equiv 258 \mod 385$)**

Ques: 11 - Use Bezout's identity to prove that the equation $385x + 97y = 1$ has integer solutions. Find x, y.

From earlier solution, we get

$$1 = 32 \times 385 - 97 \times 127$$

IT-21031

So, $x = 32$ and $y = -127$

Integer solution exists.

Ques - 12: Prove Fermat's Little Theorem and explain how it is used to test for primality. Is 561 a prime number based on this test? Evaluate $5^{20} \pmod{561}$.

Ans:

Fermat's little theorem:

if p is prime and a not divisible by p, then.

$$a^1(p-1) \equiv 1 \bmod p$$

Use in primality testing:

* If $a^1(n-1) \neq 1 \bmod n \rightarrow n$ is composite

* If = 1 → possibly prime (but carmichael number like 561 pass it)

check for 561:

gcd (5, 561) = 1 IT-21031

compute $5^{1}560 \bmod 561$:

Since 561 is carmichael, $5^{1}560 \equiv 1 \bmod 561$

But 561 is ** not prime**.

Ques-13: State and prove the Chinese Remainder Theorem. Then solve the following system:

CRT Ans:

CRT: For Pairwise coprime modulo $m_1, m_2, \cdots m_k$ and congruences:

$x \equiv a_1 \bmod m_1$

$x \equiv a_2 \bmod m_2$

....

There exits an unique solution modulo ~~$M = m_1 \times m_2$~~

$M = m_1 \times m_2 \times \cdots \times m_k$.

Example: Solve

$x \equiv 2 \mod 3$

$x \equiv 3 \mod 4$

$x \equiv 1 \mod 5$

Steps:

$M = 3 \times 4 \times 5 = 60$

IT-21031

* $M_1 = 60/3 = 20$, $M_2 = 15$, $M_3 = 12$

* Inverse: $20^{-1} \mod 3 = 2$, $15^{-1} \mod 4 = 3$

$12^{-1} \mod 5 = 3$

$x = (2 \times 20 \times 2) + (3 \times 15 \times 3) + (1 \times 12 \times 3) = 80 + 135 + 36 = 251$

$x \equiv 251 \mod 60 \Rightarrow x = 11$

Answer: ~~$x = 1 \mod 60$~~ $x = 11 \mod 60$

Ques-14: Briefly explain the CIA triad in information security.

Ans:

* Confidentiality: Protect data from unauthorized access (encryption)

* Integrity: Ensure data is accurate and

and unaltered (hashing)

* Availability : Ensure systems / data are accessible when needed (backups, uptime)

Together, they form the foundation of a secure system.

IT-21031

15

Ques-15: How does steganography differ from cryptography, and what are common techniques used ?

Ans: * cryptography : Scrambles data to hide meaning

* steganography : Hides the existance of data (embedding in images / audio).

Techniques :

* LSB (Least significat Bit) insertion
* Masking and filtering
* Audio / Video embedding

Ques-16: key differences between phishing, malware and DoS attacks:

Attack Type	Method	Impact
Phishing	Social Engineering via emails or fake sites	Credential theft
Malware	Malicious Software	Data theft, Spying, damage
DoS	Overloading a system	service unavailability

IT-21031

Ques-17: Explain GDPR and its role in cyber-security.

Ans: General Data Protection Rg Regulation (GDPR) is an EU law that protects personal data and privacy.

Key Roles:

* Enforces user consent
* Grants data access and deletion rights
* Imposes fines for data breaches
* Promotes secure data handling practices.

Ques-18: Explain DES using a 64-bit plaintext and a 56-bit key.

Ans:

DES (Data Encryption standard):

IT-21031

- Symmetric-key block cipher
- Input: 64-bit block, 56-bit key
- Steps:
  1. Initial permutation (IP)
  2. 16 rounds of Feistel function (expansion, substitution using S-boxes, permutation)
  3. Final permutation (IP^{-1})

Each round uses a subkey from the original key.

Ques-19: Evaluate $(-8 \times 5) \bmod 17$ and explain how to simplify negative modular multiplication

Ans: We have: $(-8 \times 5) \bmod 17$

First multiply: $-8 \times 5 = -40$

Step-2 Then reduce negative value modulo 17

we want: $-40 \bmod 17$

A remainder must be non-negative (in standard modular arithmetic).

To convert a negative number into positive equivalent:

IT-21031

If $r < 0$, add the modulus until it's positive.

$-40 + 17 = -23$ (still negative)

$-23 + 17 = -6$ (still negative)

$-6 + 17 = 11$ (positive)

Final answer: $(-8 \times 5) \bmod 17 = 11$

4) How to simplify negative modular multiplication:

1. Reduce each operand mod n before multiplying:

2. Or multiply first. then adjust the negativ result by adding the modulus until you get a positive equivalent.

Ques-20:

In the DES algorithm

IT-21031

Given,

- $R_0 = 0xF0F0F0F0$
- round key $K_1 = 0x0F0F0F0F$
- first round function $f(R_0, K_1)$ assumed to be bitwise XOR (simplified)
- $L_0 = 0xAAAAAAAA$
- $L_1 = R_0$ and $R_1 = L_0 \oplus f(R_0, K_1)$

Compute:

$$f(R_0, K_1) = R_0 \oplus K_1 = 0xF0F0F0F0 \oplus 0x0F0F0F0F$$
$$= 0xFFFFFFFF$$

So,

- $L_1 = R_0 = 0xF0F0F0F0$
- $R_1 = L_0 \oplus f(R_0, K_1) = 0xAAAAAAAA \oplus 0xFFFFFFFF$
$$= 0x55555555$$

Ans: $f(R_0, K_1) = 0xFFFFFFFF$

$L_1 = 0xF0F0F0F0$

$R_1 = 0x55555555$

Ques-21: Use the partial AES S-box to perform SubBytes on [0x23, 0xA7, 0x4C, 0x19]

Ans: lookups use high nibble = row
low nibble = column

- 0x23 → row 2, col 3 = 0xD4
- 0xA7 → row A, col 7 = 0x63
- 0x4C → row 4, col C = 0x2E
- 0x19 → row 1, col 9 = 0xC6

IT-21031

Resulting output: [0xD4, 0x63, 0x2E, 0xC6]

Ques-22: In AES, apply the Add Round key step only. Given Input word [0x1A, 0x2B, 0x3C, 0x4D] and round key [0x55, 0x6C, 0x77, 0x88] compute the XOR.

Ans: bytewise XOR:

- 0x1A ⊕ 0x55 = 0x4F
- 0x2B ⊕ 0x6C = 0x47
- 0x3C ⊕ 0x77 = 0x4B
- 0x4D ⊕ 0x88 = 0xC5

Output word: [0x4F, 0x47, 0x4B, 0xC5]

Ques-23: Show how MixColumns uses the fixed matrix over GF (2^8) to transform column

[0x01, 0x02, 0x03, 0x04]

Matrix :

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

IT-21031

Compute each output byte as gf-mul/xor (multiplication by 0x02/0x03 done in Rijndael field with reduction poly 0x11B).

Step results (hex):

- Row 1 = $02 \cdot 0x01 \oplus 03 \cdot 0x02 \oplus 01 \cdot 0x03 \oplus 01 \cdot 0x04$
  $= 0x02 \oplus 0x06 \oplus 0x03 \oplus 0x04 = 0x03$
- Row 2 = $0x01 \oplus 0x04 \oplus 0x05 \oplus 0x04 = 0x04$
- Row 3 = $0x01 \oplus 0x02 \oplus 0x06 \oplus 0x0C = 0x09$
- Row 4 = $0x03 \oplus 0x02 \oplus 0x03 \oplus 0x08 = 0x0A$

Output column : [0x03, 0x04, 0x09, 0x0A]

Ques-24: Describe AES-OFB mode and how it ensures synchronization.

Ans:

How it works: OFB (Output Feedback) turns the block cipher into a synchronous stream cipher. Starting from an IV:

IT-21031

$O0 = E(K, IV)$, $O1 = E(k, O0)$, $O2 = (k, O1)$, ...

Ciphertext $C_i = P_i \oplus O_i$. Decryption uses the same O_i to recover P_i.

- Synchronization: Both sides must share the same IV and key and process blocks in the same order. OFB keystream depends only on IV and key (not plaintext / ciphertext), so as long as sender and receiver use the same IV and no blocks are lost/inserted, keystreams align and decryption succeeds. Loss or reordering of blocks breaks sync until a new IV or explicit resync is used

Ques-25: Which AES modes cause error propagation during decryption ? Illustrate with CBC and CFB and explain integrity impact.

•CBC

IT-21031

Ans:

• CBC: A single-bit error in ciphertext block C_i causes P_i (after decryption of C_i) to be completely garbled (because $D(C_i)$ is wrong). The subsequent plaintext block P_{i+1} will have a single bit flip at the same bit positions (because $P_{i+1} = D(C_{i+1}) \oplus C_i$). So, curruption affects two plaintext blocks (one fully garbled, the next with bit-flips). CBC is malleable : attackers can flip bits in ciphertext to cause predictable changes. in decrypted plaintext → integrity is not ~~gua~~ guaranteed without authentication.

• CFB: Error propagation depends on segment size; a bit error in C_i will directly flip corresponding bits in decrypted P_i, and because of feedback, a limited number of

subsequent bits/blocks may be affected until the error shifts out of the feedback register. So, CFB has limited error propagation.

Intggrity implication: Neither mode provides integrity by itself, both need MAC/AE (HMAC or AES-GCM) to detect tampering.

IT-21031

Ques-26: Which AES mode for encrypting large files with parallel processing: ECB, CBC or CTR ? Justify

Recommendation: CTR (counter) mode

Why:

- keystream blocks are E(K, nonce || counter+i) so, each block encryption/decryption is independent and fully parallelizable.
- No block-pattern leakage like ECB (which is insecure).
- No sequential dependency like CBC (which prevents parallel encryption)
- Error effect is local (no long propagation).

Caveat: use unique/counter per key to avoid reuse.

Ques-27: RSA example : Given message M=1 public key e=5, n=14. Encrypt and decrypt with private d=11.

IT-21031

Ans:

Encrypt : $C = M^e \bmod n = 1^5 \bmod 14 = 1$

Decrypt : $M = c^d \bmod n = 1^{11} \bmod 14 = 1$

ciphertext = 1 (Note : n=14 is not secure-small/ composite - this is purely illustrative.)

Ques-28: RSA signature : Given H(M)=5 private key d=3, n=33. Generate signature.

Ans:

Signature $s = H(M)^d \bmod n = 5^3 \bmod 33$

$= 125 \bmod 33 = 26$

Signature = 26.

Ques :-29 : Diffie - Hellman example : p=17 g=3, a=4 (Aleya), b=5 (Badol). Compute public keys and shared secret.

- Aleya public $A = g^a \bmod p = 3^4 \bmod 17$
  $= 81 \bmod 17$
  $= 13$
- Badol public $B = g^b \bmod p = 3^5 \bmod 17$
  $= 243 \bmod 17 = 5$
- Shared secret $K = B^a \bmod p = 5^4 \bmod 17$
  $= 625 \bmod 17$.
  $= 13$ (or $A^b \bmod p$ yields same)

public keys : Aleya = 13, Badol = 5

Shared secret = 13

IT-21031

Ques-30: Hash $H(x) = (\text{Sum ASCII chars}) \bmod 100$. compute H("A") and H("BA") what does this imply about collision resistance?

Ans:

- ASCII ('A') = 65, ASCII ('B') = 66, sum = 66+65 = 131
- H("A") = 131 mod 100 = 31
- H("BA") = 66+65 = 131 mod 100 = 31

Both produce same hash (collision).

Implication : The function is not collision resistant

many different inputs map to the same short output. A cryptographic hash must make finding collisions computationally infeasible.

Ques: Mac = (Message + Secret key) mod 17.

Given Message = 15, key = 7, compute MAC. If attacker changes message to 10 without key, can they forge MAC?

IT-21031

Ans:

- MAC = (15+7) mod 17 = 22 mod 17 = 5
- For message 10, correct MAC would be (10+7) mod 17 = 17 mod 17 = 0.

Attacker without key cannot compute the correct MAC unless they guess the key, or brute-force (small modulus makes brute-force trivial). Thus conceptually secure if key secret and large enough, but this construction is insecure in practice (too simple, no cryptographic strenght). Use HMAC or other secure MACs.

Ques-32: Explain TLS handshake steps and how symmetric keys are established using asymmetric cryptography.

IT-21031

Ans:

- ClientHello: client sends supported versions chiphersuites, client random Rc.
- ServerHello: Server picks ciphersuits, sends server Random Rs.
- Server Certificate: Server proves identity by sending certificate containing its public key.
- (Optional) Serverkey Exchange: for ephemeral DH/ECDH server sends params signed by server key.
- ClientkeyExchange: either client encrypts peramaster secret with server public key (RSA key-exchange) or sends client DH public value (DHE/ECDHE).
- Both compute premaster → master secret:

master = PRF (premaster, RC, Rs);

derive symmetric keys from master secret,

• Change Cipher Spec & Finished : both sides switch to negotiated symmetric ciphers and verify handshake integrity.

IT-21031

Symmetric keys come from the shared premaster secret produced by asymmetric operations (RSA encryption of premaster or Diffie-Hellman shared secret) combined with nonces and a PRF to derive final session keys.

Ques-33:

Explain SSH layered architechture (protocol stack) and roles of each layer.

Ans:

• Transport layer (SSH-TRANSPORT) : establishes encrypted, authenticated channel, negotiates algorithms, provides confidentiality integrity optional compression.

• User Authentication Layer (SSH-USERAUTH)

authenticates the user over the secure transport (password, public-key, host-based).

- Connection layer (SSH-CONNECTION): multiplexes channels (shells, exec, port forwarding, subsystems like SFTP) over the authenticated transport.

IT-21031

Each layer builds on the lower: transport secures, userauth authenticates, connection carries application sessions.

Ques-34:

Explain the steps involved in the TLS handshake process.

Ans: (Concise stepwise recap)

- ClientHello (version, chipers)
- ServerHello (chosen ciphers)
- Server sends certificate (and serverkeyExchange if needed).
- ServerHello Done.

- ClientkeyExchange (premaster via RSA or client DH value)
- (Optional) Client Certificate + CertificateVerify
- Both compute master secret from premaster and derive session keys.

IT-21031

- Client sends ChangeCipherSpec and Finished
- Server sends Change Cipher Spec and Finished.
- Secure application traffic begins under negotiated symmetric keys.

Ques-35:

General form of elliptic curve equation over a finite field and Why used in cryptography

Ans:

- Over a prime field F_p: $y^2 \equiv x^3 + ax + b \pmod p$. with discriminant $4a^3 + 27b^2 \neq 0 \pmod p$ to avoid singularities.
- Points on the curve plus a point at infinity form an abelian group (point addition/doubling).

Why used : The Elliptic Curve Discrete Logarithm Problem (ECDLP) - given P and Q = kP, find k is believed hard. Ecc gives strong security per bit, enabling smaller keys and faster operations than equivalent RSA schemes.

IT-21031

Ques-36:

How does ECC achieve same level of security as RSA with much smaller keys?

Ans:

- Different hard problems: RSA security rests on integer factorization (sub-exponential GNFS algorithms), while ECC security rests on ECDLP (best general attacks like Pollard's rho run in roughly $O(\sqrt{n})$ exponential time).
- Consequence: For equivalent security, ECC needs much smaller key sizes (256-bit ECC $\approx$ 3072-bit RSA).
- Rr Practical benefits: smaller keys / certificates,

faster key generation / signing, lower storage / communication overhead - useful for constrained devices.

IT-21031

Ques-37:

Given the elliptic curve $y^2 = x^3 + 2x + 3 \pmod{97}$ determine whether the point $P = (3,6)$ lies on the curve.

Ans:

Compute both sides modulo 97:

- Left: $y^2 = 6^2 = 36 \pmod{97}$
- Right: $x^3 + 2x + 3 = 3^3 + 2 \cdot 3 + 3 = 27 + 6 + 3 = 36 \pmod{97}$

Since $36 \equiv 36 \pmod{97}$, the equality holds.

Yes. $P = (3,6)$ lies on the curve.

Ques-38: Given public key ($p = 23$, $g = 5$, $h = 8$) and message $m = 10$, compute the ElGamal ciphertext using random $k = 6$.

Ans: ElGamal encryption (mod p)

- $C_1 = g^k \bmod p$
- $C_2 = m \cdot h^k \bmod p$

Compute:

- $C_1 = 5^6 \bmod 23 = 15625 \bmod 23 = 8$.
- $h^k = 8^6 \bmod 23$ = compute ≡ (gives)?

IT-21031

→ $8^2 = 64 \equiv 18$, $8^4 \equiv 18^2 = 324 \equiv 2$,

then $8^2 \equiv 2 \cdot 18 = 36 \equiv 13$. so, $h^6 \equiv 13$.

- $C_2 = 10 \cdot 13 \bmod 23 = 130 \bmod 23 = 15$.

∴ ciphertext : $(C_1, C_2) = (8, 15)$.

Ques-39:

Explain why lightweight cryptography matters for IoT, and give one example of a lightweight algorithm used in IOT.

Ans:

why it matters:

IoT devices often have severe constraints - low CPU, little RAM, limited energy, small

Storage - so full-scale conventional crypto (large-block ciphers, heavyweight authenticated protocols) may be too slow, power-hungry, or memory-intensive. Lightweight cryptography provides secure primitives tailored to those constraints: lower computational cost, smaller code size, and lower memory use while still offering adequete security for the device's threat model.

IT-21031

Example algorithm:

Ascon (an authenticated encryption and hashing family) - winner in the NIST lightweight cryptography project - is designed for constrained devices and offers AEAD & hash functionality with small footprint and good performance on low-end microcontrollers. (other examples often used historically: lightweight AES variants, chacha 20-based constructions for low-cost devices and

and specialized ciphers like simon/speck note that simon/speck are controversial and not widely recommended for new standardized deployments.)

IT-21031

Ques-40:

List and breifly explain three common IoT-speific attacks and mitigation strategies

Ans:

1. Firmware hijacking / malicious firmware updates.

- What: Attacker replaces legitimate firmware with malicious firmware (backdoors, botnet clients).
- Mitigations: secure boot, digitally signed firmware updates, code-signature verification Strict update channels, rollback protection, and regular signed update audits.

2. Physical tampering / device compromise

• What : Attacker with physical access extracts keys . modifies hardware , or installs implants.

• Mitigations : Use tamper - evident / tamper - resistant enclosures , secure elements or TPMs to protect keys , disable debug ports in production hardware - based key storage (root of trust) and device attstation.

3. Botnets / mass compromis IT-21031

• What : Default / weak credentials or vulnerable services are exploited at scale to enslave devices into DDoS botnets.

• Mitigations: Enforce strong unique credentials disable unused services / ports , network segmentation , rate - limiting automated patching use of device identity + certificate - based authentication and monitoring / IDS to detect

unusual outbound traffic.

Additional general strategies: least privilege, encrypted communications (TLS with proper cert / nonce handling) periodic security audits · supply - chain security and incident response planning.

IT-21031