Md. Lutfor Rahaman, II-21047

Question-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 insfructure ?

Answer

Shor's Algorithm is a quantum computing algorithm that can factor large integers and compute discrete logarithms exponentially faster than classical Algorithm.

Impact on RSA:

→ RSA security is based on the difficulty of factoring large numbers.

→ Shor's algorithm can factor these large numbers efficiently on a powerful quantum computer → breaking RSA encryption.

Impact on ECC:

→ ECC security is depend on the Elliptic curve Discrete Logarithm problem (ECDLP)

→ Shor's algorithm can also solve ECDLP quickly, breaking ECC based schemes like ECDSA and ECDH.

potential consequences:

→ Loss of confidentiality: Encrypted communica can be decrypted.

→ Loss of authentication: Digital signatures can be forged.

→ collapse of trust in online banking, e-commerce, government communication and IOT security.

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

Ans: Role of QKD:

→ Quantum key Distribution (QKD) uses

quantum mechanics principles to let two parties share a ser secret symmetric key securely.

→ It provides unconditional security based on physics, meaning any eavesdropping attempt disturbs the quantum state and can be detected.

→ QKD is important for protecting communications against future quantum computer attacks.

Difference from classical public key Encryption

Quantum key Distribution (QKD)	classical public key Encryption
(i) Security based on physical laws of quantum mechanics, providing provable unconditional security.	(i) Security based on computational hardness assumptions.
(ii) Keys are securly distributed over quantum channels using quantum states like photons.	(ii) Keys are exchanged or encrypted mathematically over classical communication channels.

P.T.O

(iii) Any attemps to eavesdrop ~~distrib~~ disturb the quantum state, which can be immediately detected.

(iv) produces symmetric keys to be used ~~for~~ for encryption/decryption after secure exchange

(v) unconditional secure against quantum computer attacks

(vi) Require specialized quantum Hardware

(iii) Eavesdropping cannot be detected directly, Security depends on difficulty of decryption with the private key.

(iv) uses public private key pairs for encryption & decryption and digital signatures.

(v) Vulnerable to quantum algorithm's like shor's which can break many classical cryptosystems.

(vi) Implementation with software and classical hardware widely deployable today.

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

Ans:

Lattice Based cryptography	Traditional cryptography
(i) Based on math problems about points in space called lattices.	(i) Based on hard math problems like factoring large numbers.
(ii) considered safe against quantum computers, no known quantum attack can easily break it.	(ii) Vulnerable to quantum computers using shor's algorithm which can break it fast.
(iii) Keys and ciphertext are usually large in size	(iii) Keys are smaller and faster to use.
(iv) Algorithms often use simple math operations like addition and multiplication on vectors	(iv) Uses complex operations like modular exponentiation
(v) Supports advanced functions like homomorphic encryption and identity based encryption	(v) Mostly supports basic encryption and digital signatures.

Question-4 : Develop a python-based PRNG that uses the current system time and a customer seed value. Write a complete program and corresponding output.

Ans:

```
import time
# Simple PRNG using linear congruence
def prng(customseed):
    m = 2**48
    a = 25214903917
    c = 11
    # mix time and seed
    state = (customseed ^ time.time_ns()) % m
    # generate 10 Random integers
    for _ in range(10):
        state = (a*state+c) % m
        print(state)
```

```
# Example run
custom_seed = 2025
prng(custom_seed)
```

Example output

```
187585950688412
13739027904398
105231355068293
2943349096612
...
```

Question-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 division?

Ans:

→ Sieve of Eratosthenes Algorithm is an effective way to find all prime numbers up to a given number n.

→ It works by iteratively marking the multiple of each prime number starting from 2.

How it workes:

(I) create a list of numbers from 2 to n.

(II) Start with the first prime number 2.

(III) Mark all multiples of 2 (4, 6, 8. --) as not prime.

(IV) Move to the next unmarked number (3) and mark all multiples of 3 (6, 9, 12 ---).

(V) Repeat this for the next unmarked numbers (5, 7, 11, · ---) up to $\sqrt{n}$.

(VI) Remaining unmarked numbers are prime.

IT-21048

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

primes less than 50 : 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47.

Time complexity :-

Sieve of Eratosthenes:
Time complexity is approximately $O(n \log \log n)$, which is very efficient for large n.

Trial division:
Time complexity $O(n\sqrt{n})$ - much slower for large n.

Question 6: State and explain the necessary and sufficient conditions for a composite number to be a Carmichael number. Then verify wether the number $n=561$ and $n=1105$ and $n=1729$ are ea Carmichael number?

Ans:

Necessary and sufficient condition (Korselt's criterion)

A composite integer $n>1$ is a Carmichael number if all three hold::

1. n is composite (not prime)
2. n is square-free (no prime square divides n)
3. For every prime p dividing n; $(p-1) \mid (n-1)$

~~Korselt's criterion:~~

Verify the three numbers:

We apply Korselt's criteria to each n.

1. $n=561$
   → factorization $561 = 3 \times 11 \times 17$ (composite)
   → square free: yes (distinct prime)

→ n-1 = 560

for p=3 : p-1 = 2, 560/2 = 280 (divides)

for p=11 : p-1 = 10, 560/10 = 56 (divides)

for p=17 : p-1 = 16 560/16 = 35 (divides)

All condition satisfied so 561 is a Carmichael number.

2. n = 1105

→ factorization : 1105 = 5×13×17 (composite)

→ square free : yes

→ n-1 = 1104

p=5, p-1 = 4, 1104/4 = 276 (divides)

p=13, p-1 = 12, 1104/12 = 92 (divides)

p=17, p-1 = 16, 1104/16 = 69 (divides)

All condition satisfied so 1105 is a Carmichael number.

3. n = 1729

→ factorization : 1729 = 7×13×19 (composite)

→ square free : yes

→ n-1 = 1728

p=7, p-1 = 6, 1728/6 = 288 (divides)

p=13, p-1 = 12, 1728/12 = 144 (divides)

p=19, p-1 = 18, 1728/18 = 96 (divides)

All three numbers are Carmichael number.

Question 2: Determine weather the following are valid Algebraic structures and justify your answer.

- Is the set Z_{11} with operations $(+, \cdot)$ a ring?
- Are the sets $(Z_{37}, +)$ and $(Z_{35}, \times)$ Abelian groups?

Ans: Is the set Z_{11} with operations $(+, \cdot)$ a ring?

→ yes, Z_{11} is a ring. specifically, it is the ring of integers modulo 11. This means it consists of the equivalence classes of integers under the equivalence relation of congruence modulo 11. Addition and multiplication are performed in the usual way, but with the result reduced modulo 11.

for example, in Z_{11}, $6+2 = 2 \pmod{11}$

similarly, $6 * 5 = 8 \pmod{11}$.

Z_{11} also processed the properties of a field.

Z_{11} is a finite ring with 11 elements.

The elements are represented by the set $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

P.T.O

→ $(Z_{32}, +)$ is an abelian group? IT-21042

yes, $(Z_{32}, +)$ is an abelian group.

justify:

→ operation is addition modulo 32

→ closure, associativity identity 0, and inverse (for a the inverse is $(32-a)$ hold.

→ Addition is commutative.

→ So $(Z_{32}, +)$ is a finite cycle group of order 32 and therefore abelian.

→ Is $(Z_{35}, \cdot)$ is abelian group?

No, $(Z_{35}, \cdot)$ is not an abelian group.

Justification:

→ multiplication modulo 35 is associative, commutative and has identity 1, but not every element has a multiplicative inverse.

→ Example: $5 \in Z_{35}$ has $\gcd(5, 35) = 5 > 1$, so no x satifies $5x \equiv 1 \pmod{35}$

Therefore $(Z_{35}, \cdot)$ is not a group.

So it not an abelian group.

Question 8: What is remainder when -52 is reduced modulo 31?

Step 1: Write the congruence idea.

We want r such that

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

and $0 \le r < 31$

Step 2: Add multiples of 31 until we get a non negative remainder

$-52 + 31 = -21$ (still negative)

$-21 + 31 = 10$ (non negative and less than 31)

Step 3: $-52 \equiv 10 \pmod{31}$

Question-9 Determine the multiplicative inverse of 7 mod 26, If it is exists. (use extended Euclidean Algorithm)

Ans: We want to find the multiplicative inverse of 7 mod 26 - meaning we want an integer x such that:

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

Step 1 : checks if inverse exists

An inverse exists if gcd $(7, 26) = 1$

$$26 = 3 \times 7 + 5$$
$$7 = 1 \times 5 + 2$$
$$5 = 2 \times 2 + 1$$
$$2 = 2 \times 1 + 0$$

GCD $(7, 26) = 1$ means Inverse exists.

Step-2: Work backwards. (EEA)

$$1 = 5 - 2 \cdot 2$$
$$\Rightarrow 1 = 5 - 2 \cdot (7 - 1 \cdot 5)$$
$$\Rightarrow 1 = 5 - 2 \cdot 7 + 2 \cdot 5$$
$$\Rightarrow 1 = 3 \cdot 5 - 2 \cdot 7$$
$$= -2 \cdot 7 + 3 \cdot (26 - 3 \cdot 7)$$
$$= -2 \cdot 7 + 3 \cdot 26 - 9 \cdot 7$$
$$= 3 \cdot 26 - 11 \cdot 7$$

Step 3 Identify inverse

$$1 = 3 \times 26 - 11 \cdot 2$$

$$-11 \cdot 2 \equiv 1 \pmod{26}$$

Inverse of $2 \bmod 26 \equiv -11 \equiv 15 \pmod{26}$

$$7^{-1} \pmod{26} = 15$$

check: $2 \times 15 = 105 \equiv 1 \pmod{26}$

Question 10:- Evaluate $(-8 \times 5) \bmod 12$ and explain how to simplify negative modular multiplication.

Ans:

Step 1: Multiply the numbers

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

Step 2 Reduce modulo 12

$$-40 \bmod 12$$

Since modulo results are usually taken a

non-negative remainder, we add multiples of 17 until the result become positive:

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

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

$-6+17 = 11$ (positive)

$\therefore (-8 \times 5) \bmod 17 = 11$

Question 11 State and prove Bezout's Theorem. Use it to find the multiplicative inverse of 97 modulo 385.

Lemma: If a, b and c are positive integers such that $\gcd(a,b) = 1$ and $a \mid bc$ then $a \mid c$.

proof:

Assume $\gcd(a,b) = 1$ and $a \mid bc$, since $g(a,b) = 1$ by Bezout's theorem there are integers s and t such that $sa + tb = 1$

→ multiplying both sides of the equation by c, now, $sac + tbc = c$

→ We know that $a|bc$ and a divides $sac+tbc$ since $a|sac$ and $a|tbc$.

We conclude $a|c$ since $sac+tbc=c$.

Find the multiplicative inverse of 92 modulo 385.

We want integers x and y such that

$92x+385y=1$

Since $\gcd(92, 385)=1$, inverse exist

Step 1 Apply Euclidean Algorithm to find $\gcd(92, 385)$

$$385 = 92\times 3 + 94$$
$$92 = 94\times 1 + 3$$
$$94 = 3\times 31 + 1$$
$$3 = 1\times 3 + 0$$

$\gcd(92, 385) = 1$

Step 2 Back substitution to express 1 as linear combination.

$1 = 94 - 3 \times 31$

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

$1 = 94 - 31 \cdot 97 + 31 \cdot 94$

$1 = 32 \cdot 94 - 31 \cdot 97$

$1 = -31 \cdot 97 + 32 \cdot (385 - 3 \times 97)$

$= -31 \cdot 97 + 32 \cdot 385 - 96 \cdot 97$

$= 32 \cdot 385 - 127 \cdot 97$

Step 3

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

$\Rightarrow 97 \times (-127) \equiv 1 \pmod{385}$

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

$x = 385 - 127$

$x = 258 \pmod{385}$

The multiplicative inverse of $97 \pmod{385}$ is 258

11-21042

Question 12: Using Bezout's identity, prove that the equation $ax+by=\gcd(a,b)$ has integer solutions. Find x such that $43x \equiv 1 \bmod 240$.

Ans.

For any integers a and b, there exists integers x and y such that

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

proof:

(I) consider all numbers of the form $ax+by$, where x, y are integers. Among these, pick the smallest positive number d.

(II) Show that d divides both a and b:

→ Divide a by d to get remainder r.

→ Since $r = a - qd$ is also a combination of a and b, if $r > 0$ it contradicts d being smallest.

→ So, $r = 0$; meaning d divides a.

→ Similarly, d divides b.

(iii) Since d divides both a and b, d must divide their greatest common divisor

$g = \gcd(a,b)$

Also, since g divides any combination of a and b, it divides d.

(iv) Therefore, $d = g = \gcd(a,b)$.

This shows integers x and y exist such that $ax + by = \gcd(a,b)$

Find x such that $43x \equiv 1 \pmod{240}$

This means solve $43x + 240y = 1$

use Extended Euclidean Algorithm.

$240 = 43 \times 5 + 25$

$43 = 25 \times 1 + 18$

$25 = 18 \times 1 + 7$

$18 = 7 \times 2 + 4$

$7 = 4 \times 1 + 3$

$4 = 3 \times 1 + 1$

$3 = 1 \times 3 + 0$

$\gcd(43, 240) = 1$, so inverse exists.

Back substitution:

$1 = 4 - 3 * 1$

$1 = 4 - 1(7 - 4)$

$= 4 - 7 + 4$

$= 2 \cdot 4 - 7$

$= 2 \cdot (18 - 7 \times 2) - 2$

$= 2 \cdot 18 - 14 \cdot 2 - 7$

$= 2 \cdot 18 - 5 \cdot 2$

$= 2 \cdot 18 - 5 \cdot (25 - 18)$

$= 7 \cdot 18 - 5 \cdot 25$

$= 7 (43 - 25) - 5 \cdot 25$

$2 \cdot 43 - 7 \cdot 25 - 5 \cdot 25$

$2 \cdot 43 - 12 \cdot 25$

$2 \cdot 43 - 12 \cdot (240 - 43 \times 5)$

$2 \cdot 43 - 12 \cdot 240 + 60 \cdot 43$

$\Rightarrow 62 \cdot 43 - 12 \cdot 240$

$43 \times 62 \equiv 1 \pmod{240}$

So, multiplicative inverse of 43, modulo 240 is 62.

Lutfor

Question 13: Prove Fermat's Little Theorem and explain how it is used for test primality. Is 561 a prime number based on this test?

Evaluate $5^{123} \bmod 125$ using Fermat's Little Theorem. Show all steps.

→ If p is prime and a is not divisible by p, then $a^{p-1} \equiv 1 \bmod p$

proof:

→ consider the numbers $a, 2a, 3a, \ldots, (p-1)a$.

→ These numbers are distinct modulo p (since p is prime and $\gcd(a,p)=1$

→ Their product is congruent to $(p-1)! \bmod p$.

$$a, 2a, 3a, \ldots (p-1)a \equiv (p-1)! \bmod p$$

$$a^{p-1} (p-1)! \equiv (p-1)! \bmod p.$$

→ Since $(p-1)!$ is not divisible by p, we can cancel it $a^{p-1} \equiv 1 \bmod p$

IT-2104

Fermat's primality Test:

→ For a candidate n, pick a random a where $1 < a < n$.

→ If $a^{n-1} \not\equiv 1 \bmod n$, n is composite.

→ If $a^{n-1} \equiv 1 \bmod n$, n is likely prime.

* Is 561 prime?

$561 = 3 \times 11 \times 17$

So 561 is composite

Now check for fermets behavior:

$561 - 1 = 560$. For base a coprimes to 561, one can show that $a^{560} \equiv 1 \pmod{561}$. Indeed 561 is a carmichael number ~~becau~~ because it is squarefree and for each prime divisor p of 561, $p-1$ divides 560:

$3-1 = 2 \mid 560$, $11-1 = 10 \mid 560$, $17-1 = 16 \mid 560$

By the carmicheal property, every a with $\gcd(a, 561) = 1$ will satisfy $a^{560} = 1$

$a^{560} \equiv 1 \pmod{561}$. So for such bases fermat's test declares probably prime.

Evaluate $5^{123} \bmod 125$.

Step 1: Factorize 125. $125 = 5^2 \times 2$

Step 2: compute $5^{123} \bmod 25$

Since $25 = 5^2$ for $k \geq 2$, $k^2 \equiv 0 \bmod 25$

Thus $5^{123} \equiv 0 \pmod{25}$

Step 3 compute $5^{123} \bmod 2$

By fermats. Little Theorem, $5^6 \equiv 1 \bmod 7$

Write $123 = 6 \times 20 + 3$

$$5^{123} \equiv (5^6)^{20} \times 5^3 = 1^{20} \times 125 \equiv 6 \bmod 7$$

Step 4 combine result via CRT

we need x such that,

$x \equiv 0 \bmod 25$, $x \equiv 6 \bmod 7$

Let $x = 25k$ then

$25k \equiv 6 \bmod 7$

$4k \equiv 6 \bmod 7$

$k \equiv 5 \bmod 7$

Thus $k = 7m + 5$ and $n = 25(7m + 5)$

$= 125m + 125$

The smallest non-negative solution is 125.

Lutfor

Question-14 State and prove the Chinese Remainder Theorem. Then solve the following system of congruence.

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

State: Let $m_1, m_2 \cdots m_k$ be pairwise coprime positive integers $\gcd(m_i, m_j) = 1$ for $i \neq j$.

For any integers $a_1, \cdots a_k$ the system

$$x = a_i \pmod{m_i} \qquad i = 1, \cdots, k$$

has a unique solution modulo $M = m_1 m_2 \cdots m_k$

Proof:

Let $M = \prod_{i=1}^{k} m_i$ and for each i put $M_i = \frac{M}{m_i}$.

Since $\gcd(M_i, m_i) = 1$, there exists an inverse y_i with

$$M_i y_i \equiv 1 \pmod{m_i}.$$

Then $x = \sum_{i=1}^{k} a_i M_i y_i$

Satisfies $x \equiv a_i \pmod{m_i}$ for every i (because all terms with $j \neq i$ vanish modulo m_i and $M_i y_i \equiv 1 \pmod{m_i}$). So a solution exists.

$$\begin{cases} x \equiv 2 \pmod 3 \\ x \equiv 3 \pmod 5 \\ x \equiv 2 \pmod 7 \end{cases}$$

Here, $m_1 = 3$, $m_2 = 5$, $m_3 = 7$ are pairwise coprime

and $M = 3 \times 5 \times 7 = 105$

i	m_i	a_i	$M_i = M/m_i$	$y_i = M_i^{-1} \pmod{m_i}$	$a_i M_i y_i$
1	3	2	35	$35 \equiv 2 \bmod 3$ $y_1 = 2$	$2 \times 35 \times 2 = 140$
2	5	3	21	$y_2 = 1$	63
3	7	2	15	$y_3 = 1$	30

Now $140 + 63 + 30 = 233$

$= 233 \bmod 105$

$= 23$

$x \equiv 23 \pmod{105}$

Question 15: Briefly explain the CIA triad in information security. How does each component contribute to building a secure system?

Ans: The CIA triad is a fundamental model for designing and evaluating security policies, consisting of:

Confidentiality:

Defination: Ensures that information is accessible only to authorized individuals.

Goal: Prevent unauthorized access or disclosure.

Example: Using encryption and access control

contribution in security: protects sensitive data from hackers, or eavesdroppers, and unauthorized access (users).

Integrity:

Defination: Ensures data is accurate complete and unaltered except by

authorized persons.

Goal: Prevent unauthorized modification.

Example: Digital signatures, checksums and hashing hashing.

contribution to security: Maintain trust in data by ensuring it's not tampered with or corrupted.

Lutfor

Availability:

Defination: Ensures authorized users have reliable and timely access to information and resources.

Goal: prevent disruption of services.

Example: Redundant systems, backups and DDos protection.

contribution to security: Ensures that data and systems are accessible when needed, even during failure or ~~attacks~~ attacks.

Question 16: How does steganography differ from cryptography in the context of information security, and what are common techniques used for hiding data in digital media?

Ans:

Cryptography: is the process of converting information into an ~~unauthori~~ unreadable format (ciphertext) to protect its content from unauthorized access. The presence of the message is visible, but its meaning is hidden.

Steganography: is the practice of hiding the ~~vary~~ very existence of information by embedding it inside another medium.

Lutfor

Cryptography	Steganography
(i) protect the content of a message.	(i) Hide the existance of a message.
(ii) Message is visible but unreadable.	(ii) Message is invisible to eau casual observers
(iii) Message is known to exist.	(iii) Message is hard to detect

common Steganography Techniques in digital media:

1. Least Significant bit (LSB) insertion: change the smallest bits in image pixels to hide data.
2. Masking and filtering: Hide data in important parts of an image (like watermark) so it's harder to remove.
3. Transform domain Techniques: Hide data in frequency parts of media instead of directly in pixels.
4. Meta data Hiding: put secret data into unused file information fields (metadata) without changing the visible content.

Lutfor

Question 12 : What are the key difference between phishing, malware, and denial of Service (DoS) attacks in terms of their method and impact on system security?

Here is the difference between phishing malware, and denial of Service (DoS) Attacks:

Attack Type	method	Impact on Security
Phishing	Tricking users (usually via fake emails, websites, or messages) into revealing sensitive information like passwords, bank details, or credit card numbers.	compromise confidentiality by stealing personal or login data.
Malware	Installing malicious software (virus, worm, Trojan, ransomware, spyware) on a system often via downloads, email attachments, or infected sites.	can harm confidentiality, integrity, and availability by stealing data, modifying files or disabling systems.

Denial of service (DoS)	Flooding a network server or website with excessive traffic to overload and make it unavailable to users	Targets availability by disrupting normal access to services.

Question 18: Explain How legal frameworks such as the General data protection Regulation (GDPR) help ~~min~~ mitigate cyber attacks and protect users privacy.

Ans: The explanation of how legal frameworks like GDPR ~~kept~~ help protect against cyber attacks and safegurd privacy.

General data protection Regulation (GDPR):

→ A European union (EU) law effective since may 25, 2018.

→ ~~Reduces the risk of data theft in case of cyber attacks.~~

→ Focuses on protecting the personal data and privacy of individuals in the EU and affects any company handling EU citizens

data).

How GDPR helps mitigate cyber attacks and protect privacy:

1. Strict data protection Requirments:
→ organizations must secure personal data using encryption, access control and secure storage.
→ Reduces the risk of data theft in case of cyber attacks.

2. Data minimization principle:
→ only necessary data should be collected and stored.
→ Less stored data means attackers have fewer targets.

3. Mandatory Breach Notification
→ company must report data breaches to authorities within 72 hours.
→ Helps quick response to limit damage.

4. User Rights over data: IT-21042

→ Right to access, correct, delete or transfer personal data ("Right to be forgotten").

→ Reduces long term exposure of sensitive information.

5. Accountability and heavy penalties:

→ fines can be up to $ 20 million or 4% of annual global turnover for violations.

→ create strong incentives for company to invest in cybersecurity.

Question-19 Explain the basic working of the ~~AES~~ DES Algorithm using a simple 64 bit paintext block and a 56 bit key. Show how the initial permutation, round function, and final permutation contribute to the encryption process.

IT-21047

Ans:

DES basic working

→ using 64 bit plaintext and 56 bit key

Step 1 Input block and key

→ plaintext 64 bits

→ key 64 bits input but only 56 bit are used (8 bit for parity)

Step 2: Initial permutation (IP)

→ The 64 bit plaintext undergoes initial permutation (IP) using a fix table.

→ The bits are rearranged according to the IP table.

Output: Two 32 bit halves, L_0 and R_0

Step 3: 16 Round of feistel structure

Each Round has:

→ Expansion (E)

→ Eey mixing (xor)
→ substitution (s-box)
→ permutation (P)

IT-21042

Steps per Round:

a. Expansion (E-box)
R(n-1) (32 bits) → expand to 48 bits using expansion table.
→ This introduces redundancy

b. Key Mixing
→ The 48 bit expanded R(n-1) is xored with 48 bit subkey K(n).
→ subkeys are generated from the 56 bit main key, one for each round.

c. Substitution (s-box)
→ The XOR Result is divided 8 blocks of 6 bits.
→ Each block goes through an s-box (substitution box), outputing 4 bit each.
→ final result: 32 bits.

d. permutation (P-box)

→ The 32 bit output from sboxes is permuted again using a fixed table.

→ Increases confusion and diffusion.

e. Feistel Swap

⇒ New values:

$L(n) = R(n-1)$

$R(n) = L(n-1)$ XOR $f(R(n-1), k(n))$

4. Repeat for 16 Rounds:

→ Each round uses a different 48 bit subkeys generated from the original key using pc-1, shifts and pc-2 tables.

5. Final permutation (FP):

→ After Round 16, R_{16} and L_{16} are swapped and then the final permutation (inverse of IP) is applied.

→ The Result is 64 bit ciphertext.

Question 20: In the DES algorithm, a 64 bit plaintext block is divided into two 32-bit halves: L_0 and R_0. Given $R_0 = 0xF0F0F0F0F0$ and round key $K_1 = 0x0F0F0F0F0F$, compute the output of first round's function $f(R_0, K_1)$ assuming XOR operation only. Then, find $L_1 = R_0$ and $R_1 = L_0 \oplus f(R_0, K_1)$, where $L_0 = 0xAAAAAAAA$

Ans: Given that,

IT-21042

$R_0 = 0xF0F0F0F0$

$K_1 = 0x0F0F0F0F$

$L_0 = 0xAAAAAAAA$

Step 1: First Round function (only XOR)

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

$= 0xF0F0F0F0 \oplus 0x0F0F0F0F$

$= 0xFFFFFFFF$

Step-2 compute L_1 and R_1

$L_1 = R_0 = 0xF0F0F0F0$

$R_1 = L_0 \oplus f(R_0, K_1) = 0xAAAAAAAA \oplus 0xFFFFFFFF$

$= 0x55555555$

$L_1 = 0xF0F0F0F0$ and $R_1 = 0x55555555$

Question-21 Given the input word [0x23, 0xA7, 0xAC, 0x19] use the following partial AES S-box to perform the SubBytes transformation. provide the resulting output/word.

Row\Col →	3	7	9	C
1	6D			C6
2	D4			
4		A1		2E
A			63	

Ans:

0x23 → Row = 2, Col = 3 → D4

0xA7 → Row = A, Col = 7 → 63

0x4C → Row = 4, Col = C → 2E

0x19 → Row = 1, Col = 9 → C6

Output word = [D4, 63, 2E, C6].

For when no value found given N/A (not Available)

Question 23 Show how MixColumns uses the following fixed matrix over GF(2^8):

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

Lut for

to transform an input column. use an example column with values [0x01, 0x02, 0x03, 0x04] (convert hex to Binary)

Ans: Given that,

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

= [00000001, 00000010, 00000011, 00000100]

Mix columns matrix $M = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$

Step1: First Row of output

$\Rightarrow$ 0x02 • 0x01 ⊕ 0x03 • 0x02 ⊕ 0x01 • 0x03 ⊕ 0x01 • 0x04

= 0x02 ⊕ 0x06 ⊕ 0x03 ⊕ 0x04

= 00000010 ⊕ 00000110 ⊕ 00000011 ⊕ 00000100

= 00000011

= 0x03

Step 2: Second Row output

= 0x01 ⊕ 0x02 ⊕ 0x03 ⊕ 0x04

= 0x01 ⊕ 0x02 ⊕ (0x03 • 0x02) ⊕ (0x04 • 0x03)

= 0x01 ⊕ 0x02 ⊕ 0x06 ⊕ (0x08 ⊕ 0x03)

= 0x01 ⊕ 0x02 ⊕ 0x06 ⊕ 0x0b

= 0000 0001 ⊕ 0000 0010 ⊕ 0000 0110 ⊕ 0000 1011

= 0000 0000

= 0x00

Step 3: Third Row output:

= (0x01 • 0x01) ⊕ (0x02 • 0x02) ⊕ (0x03 • 0x03) ⊕ (0x04 • 0x04)

= 0x01 ⊕ 0x04 ⊕ 0x09 ⊕ (0x10 - 0x01)

= 0x01 ⊕ 0x04 ⊕ 0x09 ⊕ 0x0f

= 0000 0001 ⊕ 0000 0100 ⊕ 0000 1001 ⊕ 0000 1111

= 0000 1001

= 0x09

Step 4: Fourth Row output:

= (0x03 • 0x01) ⊕ (0x01 • 0x02) ⊕ (0x02 • 0x03) ⊕ (0x04 • 0x04)

= 0x03 ⊕ 0x02 ⊕ 0x06 ⊕ 0x08

= 0x00 11 ⊕ 0x00 10 ⊕ 0x00 11 ⊕ 0x00 1000

= 0x0A

= 0x0A

output: [0x0A, 0x09, 0x00, 0x03]

IT-21042

Question 24: Describe how AES-OFB mode works. How does it ensure synchronization between encryption and decryption streams?

Ans: what is AES-OFB?

AES-OFB (output feedback mode) is a stream cipher mode of AES encryption.

→ It turns the block cipher (AES) into a synchronous stream cipher using feedback.

Working procedure:

1. Initialization vector (IV) is chosen randomly and shared with the receiver.
2. IV is encrypted using AES and a ~~Security~~ secret key → This generate the first output block.
3. This output block is xored with the plaintext block to produce the ciphertext.
4. The same output block is used as the input for AES in the next round (not the ciphertext)
5. This repeats for every blocks:
   Next_input = AES(previous_output)
6. Description uses the same output stream to XOR with the ciphertext → gives the

original plaintext.

Synchronization between encryption and decryption:

→ Both Sender and Receiver use the same IV and key, and perform the same AES encryption.

→ Since the same ~~square~~ sequence of output blocks is generated at both ends, they remain synchronized.

→ Even if plaintext/cipher text is lost, they will still stay in sync as long as IV is the same.

Lwtfon

Question 25 Which AES mode causes error propagation during decryption and how does this affect the integrity of the decrypted messages? Illustrate with CBC and CFB modes.

Ans:

→ CBC (cipher block chaining) and CFB (cipher feedback) modes both causes error propagation.
→ A single bit error in the ciphertext will affect multiple blocks in decryption.

In CBC mode:

→ Each ciphertext block is used to decrypt the next plaintext text.
→ So, if one ciphertext block is corrupted
  ① The current plaintext block become garbled.
  ② The next block is affected due to xor with corrupted block.

Effect: 1 error affects 2 blocks.

In CFB mode:

→ Each ciphertext block is feedback into the encryption process.
→ If a ciphertext block is corrupted
  ① The corresponding plaintext block become

wrong.

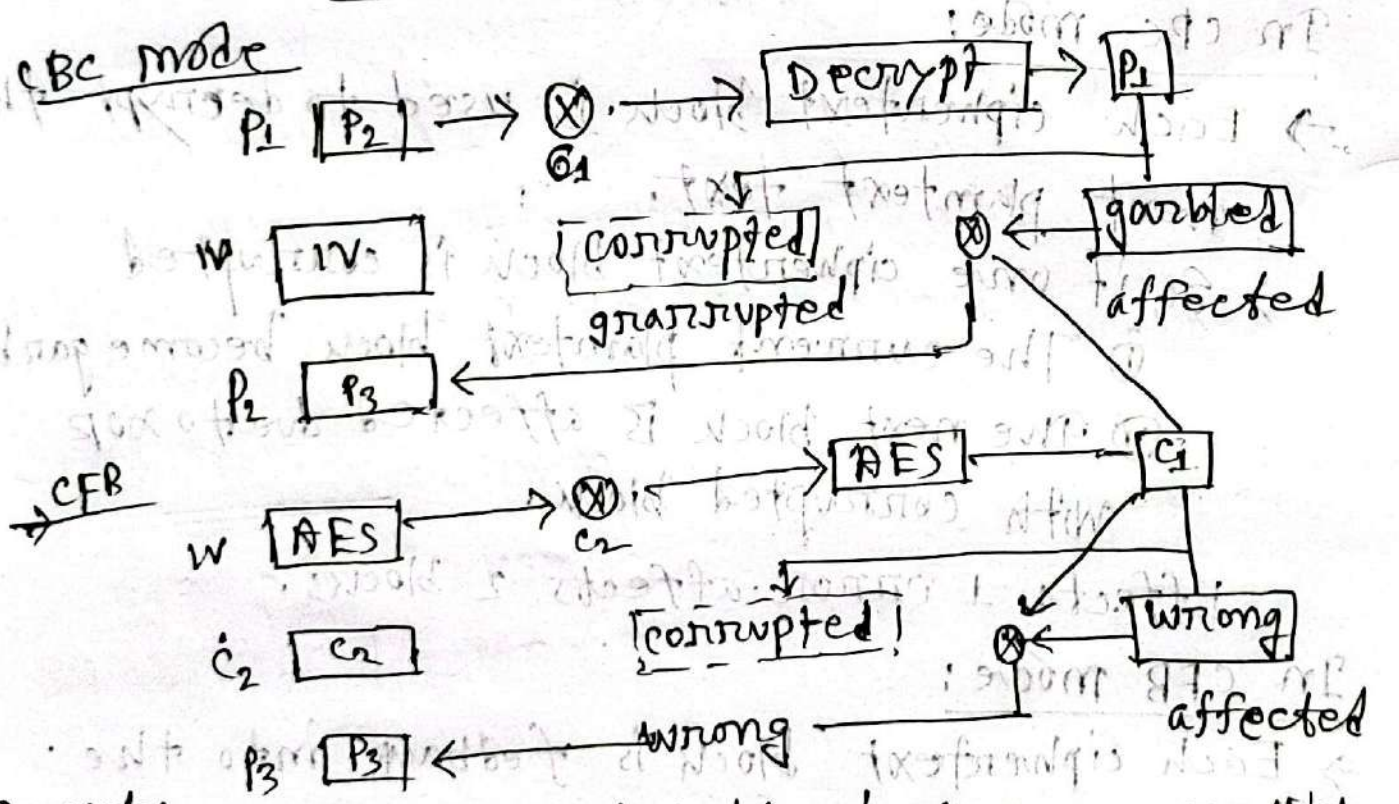
(iii) And the next one is also affected.

Impact on Integrity

→ Loss of Integrity means wrong plaintext is recovered.

→ Need of for Message Authentication code (MAC) or Authentication Encryption to ensure correctness.

Error propagation in AES



Impact:

→ Single bit error in ciphertext causes multiple bit errors in decrypted output. (CBC/CFB)

Question-26 Which AES mode would you recommend for encrypting large files with parallel processing and why? Justify your choice between ECB, CBC and CTR.

Ans: Recommended mode : AES-CTR (counter mode)

Justification:

(I) Support parallel processing:
Each block is encrypted independently using a counter value, allowing simultaneous encryption/decryption.

(II) No block dependency: CTR does not rely on previous ciphertext blocks like CBC, So its faster and more efficient.

(III) Maintain data confidentiality: CTR hides repeating patterns in plaintext, unlike ECB which leaks structure.

(IV) Ideal for large files: Works best for big files where speed and efficiency matter.

(V) Error does not propagate: A single bit error affects only the corresponding block, making recovery easier.

(VI) Flexible counter based design.

Question-22: Given a message "A" represented as M=1, encrypt it using public key e=5, n=14. What is the ciphertext? Then decrypt using private key d=11.

Ans: Using the RSA: encryption formula

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

$$c = 1^5 \bmod 14$$

$$c = 1 \bmod 14 = 1$$

ciphertext c=1

Decryption:

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

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

$$= 1 \bmod 14$$

$$= 1$$

Question-28 Given message hash H(M)=5 RSA private key d=3, n=33. Generate the digital signature.

Ans: Given that, H(M) = 5

d = 3

n = 33

Digital Signature formula:

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

$$= 5^3 \bmod 33$$

$$= 125 \bmod 33$$

$$S = 26$$

Digital Signature : 26

Question 29: Aleya and Badal are using the Diffie-Hellman key exchange protocol. They agree on the following public values: prime modulus $p = 12$, Base (generator): $g = 3$, Aleya choose a private key, $a = 4$ Badal choose a private key ~~b~~ $b = 5$. compute public key of Aleya and Badal

Ans: Given that

prime module $p = 12$

Base (Generator) $g = 3$

Aleya's private key $= 4$

Badal's private key ~~b = 6~~ $b = 5$

Step 1: Aleya's public key

$$A = g^a \bmod p$$

$$A = 3^4 \bmod 12$$

$$A = 13$$

~~Atyas~~ Aleya's public key $= 13$.

Step 2 Badal's public key

$$B = g^b \bmod p$$
$$= 3^5 \bmod 17$$
$$= 243 \bmod 17$$
$$= 5$$

Badal's public key = 5

Question 30 A simple (non-cryptographic) hash function H(x) is defined as the sum of ASCII values of the characteristics is a message, modulo 100: H(x) = (Sum(ASCII of the characteristics in x)) mod 100. Compute the hash value of the message "AB" and "BA" using this function. Do the two message produce the same hash? Do the two messages produce the same hash? What does this imply about collision resistance in weak hash function?

Ans: Given

$H(x)$ = (sum of ASCII values of characters in x) mod 100.

Message "AB"

ASCII A = 65

ASCII B = 66

Sum = 65 + 66 = 131

Hash: 131 mod 100 = 31

message "BA"

Sum = 66 + 65
= 131

Hash: 131 mod 100 = 31

→ Same hash value, this means "AB" and "BA" produce the same hash value.

→ This is a collision - two different input produce the same hash output. It shows that weak hash functions are not collision resistant and can easily lead to security vulnerabilities.

Question 31 In a secure message system a simple message Authentication code (MAC) is computed using modular addition MAC = (message + secretkey) mod 17.
Given message : 15, secret key : 12. compute the MAC for the message. Suppose an attacker changes the message to 10 but doesn't know the key. can they forge the ~~correct~~ correct MAC ~~or~~ easily ? Explain briefly.

Any Given :

Message $m = 15$

Secret key $k = 2$

Formula, $MAC = (m + k) \bmod 12$

Original MAC :

$MAC = (15 + 2) \bmod 12$

$MAC = 22 \bmod 12$

$= 5$

∴ Original MAC = 5

If attacker change message to 10.

then the MAC will be

$$MAC' = (10+2) \bmod 12$$
$$= 0$$

If they guess, probability of correct guess = $\frac{1}{12}$ which is very low.

If Attacker change the message then the forumal of $MAC_{new} = (10+k) \bmod 12$ where k is unknown

Now, $10+k \equiv 5 \bmod 12$

$\Rightarrow k = -5 \bmod 12$

$\Rightarrow k = 12$

which is not correct secret key.

Question 32 Explain the steps involved in the TLS handshake process. How are symmetric key established securely using ~~ass~~ asymetric cryptography during the handshake?

Ans:

TLS = Transport Layer Security, used in HTTPS and secure communication.

Main Idea: The TLS handshake is how two devices (like browser and server) agree on encryption keys and start a secure session. It uses asymmetric cryptography (public/private keys) first, then switches to symmetric cryptography for speed.

Steps

Client Hello

→ The client (browser) sends:

(i) Supported TLS version.

(ii) List of supported encryption algorithm (cipher suites)

(iii) A random number.

→ Server Hello

→ The server replies with

(i) Chosen TLS version

(ii) Selected cipher suite

(iii) Another random number

(iv) Its digital certificate (contain public key, signed by a trusted CA)

Certificate Verification

→ The client checks the server's certificate to ensure:

(i) It's issued by a trusted CA.

(ii) It matches the domain name.

(iii) It's not expired or revoked.

key exchange:

→ If using RSA key exchange: The client encrypts a "premaster secret" with the server's public key and sends it.

→ If using Diffie-Hellman/ECDH: Both sides exchanges public parameters and compute a shared secret.

Session key Generation:

→ Both client and server use the shared secret + random numbers to create identical symmetric session key.

Finished message:

→ Both send an encrypted "Finished" message to confirm key setup.

Secure communication Begins

→ From now on, all data is encrypted with symmetric keys.

Question: 33: Explain the layered architecture (protocol stack) of SSH. Briefly describe the role of each layer.

Ans: SSH (Secure Shell) has three main layers.

1. Transport Layer protocol:
   → Handles encryption, server authentication and integrity.
   → Negotiates algorithms and establishes a secure channel.
2. User Authentication protocol: Layer
   → Verifies the client's identity (password, public key, etc.)
3. connection protocol Layer
   → Manages multiple logical channels over the secure connection.

Question-3) Explain the steps involved in the TLS handshake process.

Ans:

1. client Hello

The client Sends

(i) Supported TLS version

(ii) Supported encryption Algorithm

2. Server hello:

The Server response with

→ chose TLS version

→ selected cipher suite.

3. Server certificate.

→ The Server send digital certificate.

4. Server key exchange

→ If needed the Server Sends extra key exchange parameters.

5. client certificate:

→ In mutual authentication the client also Sends its certificates.

6. Key exchange and pre Master secret

7. change cipher key

8. Finished message.

Question 35 What is the general form of an elliptic curve equation over a finite field, and why it used in cryptography?

Ans: The general form of an elliptic curve equation over a finite field is

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

Where,

→ a, and b are constants in the field Fp.

→ p is a prime number (for prime fields)

→ The discriminant condition $4a^3 + 27b^2 \neq 0 \pmod{p}$ ensures the curve has no singularities (no cusps or self interactions)

Why is it used in cryptography?

① Strong security per key size: Elliptic curve cryptography (ECC) provides the same security as traditional Algorithms (like RSA) but with a smaller keys, making it faster and more efficient.

2. Hard mathematical problem: The security relies on the Elliptic curve Discrete logarithm problem (ECDLP), which is extremely hard to solve.

IT-21042

3. Efficient computation: Suitable for devices with low processing power, like IoT devices and mobile systems.

4. Widely Adopted: use in TLS, digital signatures (ECDSA), and key exchange (ECDH).

Question-36: How does ECC achieve the same level of security as RSA with a smaller key size? Briefly explain.

Ans: Elliptic curve cryptography (ECC) achieves the same level security as RSA with much smatter key sizes becauses it is based on a harder underlying mathomatical problem.

Key Reason:

→ RSA security is based on the difficulty of integer factorization (factoring a large number into primes)

→ ECC security is based on the Elliptic curve Discrete logarithm problem (ECDLP) given two points P and Q = kP on an elliptic curve, it is extremly hard to find k. The EDLC is much harder to solve than factoring for the same key size, so ECC can use much smaller keys for equivalent security:

Security level comparison

RSA key size	ECC key size	Approx: security
1024 bits	160 bits	~ 80 bit security
2048 bits	224 bits	~ 112 bit security
3072 bits	256 bits	~ 128 bit security

Question 22 Given the elliptic curve $y^2 = x^3 + 2x + 3 \pmod{97}$ determine weather the point $P = (3, 6)$ lies on curve.

Ans: Given elliptic curve

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

point $P = (3, 6)$

Step 1: check Left Hand Side (LHS)

$$y^2 = 6^2$$
$$= 36 \pmod{97}$$
$$= 36$$

Step 2: checks Right-Hand Side (RHS)

$$= x^3 + 2x + 3$$
$$= (3)^3 + 2 \cdot 3 + 3$$
$$= 27 + 6 + 3$$
$$= 36 \pmod{97}$$
$$= 36$$

Hence, LHS = RHS, so point $P = (3, 6)$ lies on the curve.

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

Ans: Given that,

$p=23, g=5, h=8$

message $m=10$

Random $k=6$

Formula: $c_1 = g^k \bmod p$

$c_2 = m\, h^k \bmod p$

Step 1

$c_1 = 5^6 \bmod 23$

$= (5^2)^3 \bmod 23$ ——— (1)

$5^2 = 25 \bmod 23$

$= 2$

From (1) $c_1 = (2)^3 \bmod 23$

$= 8 \bmod 23$

$c_1 = 8$

Step 2

$c_2 = m.\, h^k \bmod p$

$= 10.8^6 \bmod 23$

$= 10 \times (8)^2 \bmod 23$

$= 10 \times 36 \bmod 23$

$8^6 \bmod 23$

$= (8^3)^2 \bmod 23$

$= 8^3 \bmod 23$

$= 512 \bmod 23$

$= 6$

$= 360 \bmod 23$

$C_2 = 15$

$\therefore (C_1, C_2) = (18, 15)$

Question-39:- Explain how light-weight cryptography is important for securing IoT devices. Give one example of a light-weight encryption algorithm used in IoT.

Ans: Lightweight cryptography is designed to use low memory, low-processing power and minimal energy, making it suitable for resource-constrained IoT devices.

Importance:

→ IoT device have limited cpu and low battery - cannot handle heavy algorithm like RSA easily

→ Ensures data confidentiality, integrity, and authentication without overloading the device.

Example:

PRESENT: A block cipher with 64 bit block size and 80/128 bit key, optimized for low hardware area and low energy usage.

Question 40: List and briefly explain any three common IOT specific attacks (firmware hijacking, physical tampering ~~bon~~ botnets like Mirai). What mitigation ~~Sta~~ strategies can be applied?

Ans: common IOT specific Attacks and Mitigation:

1. Firmware Hijacking

What is it: Attackers replaces legitimate firmware with malicious code.

Impact: can control the device remotely or steal data.

Mitigation: Secure boot, firmware signing and version verification before updates.

2. Physical Tampering Lutfor

What it is: Direct manipulation of IoT hardware (opening device, probing chips)

Impact: can extract encryption keys or modify circuits.

Mitigation: Tamper-evidant seals, hardware encryption, protective casing.

3. IoT Botnets (Mirai)

What it is: Malware infects IoT devices and uses them in large scale DDos attacks.

Impact: Disrupts services, causes internet outages.

Mitigation: change default passwords, regular firmware updates, network segmentation.