(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?

⇒ Shor's algorithm threatens the foundational security of RSA and ECC, putting virtually all public key encryption at risk quantum hardware matures.

IT-21001

Here given the Shor's Algorithm undermines RSA and ECC -

- RSA's vulnerability: It reliance on the difficulty of factoring large numbers and efficiently solvable but classically extremely hard using Shor's algorithm.
- ECC: It depends on the hardness of the elliptic-curve discrete logarithm problem.

Shor's algorithm can also solve discrete logarithm problem in quantum polynomial time.

IT-21001

Potential consequences for digital Infrastructur

1. Breakdown of Secure Communication → Protocols like TLS/SSL, email encryption, VPN, digital signatures. Shor-based attacks could allow attackers to decrypt past or ongoing communications.

2. Impacted critical infrastructure and IoT devices

3. Risks to blockchain and cryptocurrencies

4. National security and financial system destabilization.

5. Undermined trust in PKI and certificates.

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

⇒ Quantam key distribution (QKD) is expected to play an important role in future cryptographic system. It is a way to securely share a secret encryption key using the rules of quantam physics.

IT-21001

Role of 'QKD:

1. QKD + Symmetric encryption: ⇒ Use QKD to securely share a symmetric key, then use fast classical method for actual encryption.

2. Quantum-safe networks ⇒ Government and some companies are already building QKD-secured communication systems.

IT-21001

3. Complementary to post Quantum Cryptography (PQC) ⇒ while PQC works on regular device QKD provides ultimate physical-level-security

Different from classical Public-key Encryption

Feature	Classical Public key	QKD
Based on	Hard math problems (factoring, elliptic curve)	Quantum physics laws
Security depends on	Difficulty of solving math problems	Laws of quantum mechanics
Vulnerable to quantum computers	Yes	No
How keys are shared	Public and private key pairs	Directly shared via quantum channel
used for	Encryption and authentication	only for secure key exchange, not encryption itself.

IT-21001

(b) What are the main different between lattice based cryptography and traditional number-theoretic approaches like RSA, particularly in the context of quantum resistance?

⇒

Lattice Based	RSA / ECC (Number-Theoret
(i) Based on Hard lattice problems (learning with errors - LWE, Shortest Vector Problem - SVP)	(i) Based on Integer factorization (RSA), discrete logarithms (ECC)
(ii) Problem-Domain are High-dimensional vector spaces.	(ii) Number theory numbers, modular exponentiation.
(iii) Structure often involves matrices, vectors and linear algebra	(iii) Involves prime numbers modular exponentiation.
(iv) Vulnerability to Shor's algorithm are not affected.	(iv) Broken by Shor's algorithm.

(4) Develop a python-based PRNG that uses the current system time and a custom seed value. Write complete program and output

⇒

IT-21001

PRNG Code:

```
import time
seed = int(input("Enter a seed Number:"))
time_seed = int(time.time() * 1000)
combined_seed = seed + time_seed
def simple_Prng(n, seed):
    a = 1103515245
    c = 12345
    m = 2**31
    numbers = []
    for _ in range(n):
        seed = (a* seed + c) % m
        numbers.append(seed)
    return numbers
```

```
random_numbers = simple_prng(5, combined_seed
print("\nGenerated random numbers:")
for num in random_numbers:
    print(num)
```

IT-21001

Output:

```
Enter a seed number : 100

Generated random numbers:
1523601 73
1009397642
1704568083
1888490152
1412922361
```

(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?

⇒

Algorithm Steps

IT-21001

For, n = 50

1. Write down numbers 2 to 50

   2, 3, 4, 5, 6, 7, 8 ..... , 50

2. Start with the first number, 2 — it's prime. cross out all multiples of 2 greater than 2

   4, 6, 8, 10, 12, ..., 50

3. Next uncrossed number is 3 — it's prime cross out all multiples of 3 greater than 3

4. Next uncrossed number is 5, — it's prime cross out all multiples of 5 greater than 5

   10, 15, 20, 25, 30, ..., 50

IT-21001

5. Next unerossed number is 7 — it is prime cross out multiple of 7

14, 21, 28, 35, 42, 49

6. Stop when the current number is square is greater than 50

Remaining Numbers (Primes < 50)

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

Time Complexity

- Sieve of Eratosthenes: $O(n \log \log n)$
- Reason: we mark multiples for each prime and the marking work becomes progressively smaller.
- Trial Division (Checking each number by dividing with all smaller numbers up to $\sqrt{n}$)

Time complexity = $O(n\sqrt{n})$

For each number up to n, you might check up to $\sqrt{n}$ possible factors.

IT-21001

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

⇒ A composite integer $n > 1$ is a Carmichael number (i.e $a^{n-1} \equiv 1 \pmod{n}$ for every a with $\gcd(a, n) = 1$) if all of the following hold:

1. n is squarefree (no prime square divides n)
2. For every prime p dividing n we have:

$$(p-1) \mid (n-1).$$

1. $n = 561$

$$3 \mid 561$$
$$11 \mid 187$$
$$17$$

Factor of $561 = 3 \times 11 \times 17$ (Distinct primes ~~sequence~~)

• $n-1 = 561-1$
$= 560$

• For $p=3$; $p-1 = 2$, $560 \div 2 = 280$
$\therefore (p-1) \mid (n-1)$
$\Rightarrow (3-1) \mid (264-1)$
$\Rightarrow 2 \mid 250$

IT-21001

• For $p=11$; $p-1 = 10$; $560 \div 10 = 56 \rightarrow 10 \mid 260$
• For $p=17$; $p-1 = 16$; $560 \div 16 = 35 \rightarrow 16 \mid 260$

All conditions hold ⇒ 561 is Carmichael num.

2. $n = 1105$

$5 \mid 1105$
$13 \mid 221$
17

Factor $1105 = 5 \times 13 \times 17$

• $n-1 = 1105-1$
$= 1104$

• For $p=5$; $p-1=4$; $1104 \div 4 = 276 \rightarrow 4 \mid 1104$
• For $p=13$; $p-1=12$; $1104 \div 12 = 92 \rightarrow 12 \mid 1104$
• For $p=17$; $p-1=16$; $1104 \div 16 = 69 \rightarrow 16 \mid 1104$

All conditions hold ⇒ 1105 is a Carmichael number.

IT-21001

(7) Determine whether 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 ?

$\Rightarrow$

Yes, More than that, Z_{11} (integers modulo 11) is a commutative ring with unity, and since 11 is a prime it is actually a field.

Why:

- $(Z_{11}, +)$ is an abelian group: closure, associativity, identity 0, inverse (additive inverse of a is $11 - a$)
- Multiplication mod 11 is associative and closed
- distributive laws hold between + and $\cdot$
- There is a multiplicative identity 1

(a) $(Z_{37}, +)$ is an abelian group Because-

- Addition modulo 37 is closed and associative
- Identity is 0
- Every element a has additive inverse 37-a
- Addition is commutative.

Moreover, since 37 is prime, this group is cyclic (generated by 1) so $(Z_{37}, +1)$ is an abelian group.

(b) $(Z_{35}, +)$ is not an abelian group:

- The element 0 has no multiplicative inverse,
- There are zero divisor (e.g. $5 \times 7 \equiv 35 \equiv 0 \pmod{35}$)

Neither 5 nor 7 is invertible modulo 35.

IT-21001

IT-21001

(8) What is the remainder when -52 is reduced modulo 31?

$\Rightarrow$

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

The idea: find r such that

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

and $0 \leq r < 31$

Step 1: Add multiples of 31 to -52 until it become nonnegative

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

$-21 + 31 = 10$ (nonnegative and < 31)

Step 2: Conclusion

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

So, the remainder is 10

(9) Determine the multiplicative inverse of 17 mod 26. If it exists. (use extended Euclidean algorithm)

⇒ We are solving for x in:

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

Step 1 | Apply Euclidean Algorithm

we find gcd: (7, 26)

1. $26 = 3 \times 7 + 5$
2. $7 = 1 \times 5 + 2$
3. $5 = 2 \times 2 + 1$
4. $2 = 2 \times 1 + 0 \rightarrow$ gcd $= 1$

IT-21001

Step 2 | Extended Euclidean Algorithm,

we work backward to express 1 as a combination of 26 and 7

From Step 3:

$$1 = 5 - 2 \times 2$$

From Step 2 ($2 = 7 - 1 \times 5$):

$$1 = 5 - 2 \times (7 - 1 \times 5)$$
$$1 = 5 - 2 \times 7 + 2 \times 5$$
$$1 = 3 \times 5 - 2 \times 7$$

From step 1 ($5 = 26 - 3 \times 7$).

$$1 = 3 \times (26 - 3 \times 7) - 2 \times 7$$

$$1 = 3 \times (26 - 9 \times 7) - 2 \times 7$$

$$1 = 3 \times 26 - 11 \times 7$$

Step 3: Extract inverse

$$1 = (-11) \times 7 + 3 \times 26$$

So,

$$-11 \times 7 \equiv 1 \pmod{26}$$

Since $-11 \bmod 26 = 15$

inverse = 15

IT-21001

(10) Evaluate $(-8 \times 5) \bmod 17$ and explain how to simplify negative modular multiplication?

⇒

Compute directly

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

Reduce modulo 17. Add a multiple of 17 to bring -40 into the range $0, \ldots, 16$.

~~$-40 + 8 \times 17 = -40 + 51$~~

~~$= 11$~~

$$-40 + 17$$

$$= -23 + 17$$

$$= -6 + 17$$

$$= 11$$

IT-21001

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

A quicker way are replace the negative factor by its congruent positive reside first.

IT-21001

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

⇒ For integers a, b not both zero, there exist integers x, y such that

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

Proof:

Let $a, b \in Z$, not both zero, Define

$$r_0 = a, \quad r_1 = b$$

and perform the Euclidean division steps

$$r_{i-2} = q_i r_{i-1} + r_i;$$

where r_n is the last nonzero remainder

$$\therefore r_{n+1} = 0$$

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

we now show that each r_i can be written as linear combination of a, b.

$r_i = s_i a + t_i b$

IT-21001

$s_0 = 1, t_0 = 0;$

$s_1 = 0, t_1 = 1$

$\therefore s_i = s_{i-2} - q_i s_{i-1}$; $t_i = t_{i-2} - q_i t_{i-1}$

In particular, for the last nonzero remainde

$$r_n = s_n a + t_n b$$

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

$\therefore \gcd(a, b) = s_n a + t_n b$

$\Rightarrow \gcd(a, b) = ax + by$ $[x = s_n, y = t_n]$

Find the multiplicative inverse of 97 modullo 385

we need x such that,

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

$\Rightarrow 97x - 385y = 1$

Step 1] Euclidean Algorithm,

$$385 = 97 \times 3 + 94$$

$$97 = 94 \times 1 + 3$$

$$94 = 3 \times 31 + 1$$

$$3 = 1 \times 3 + 0$$

So, gcd (97, 385) = 1

IT-21001

Step 2 Back substitution

From $94 = 385 - 97 \times 3$

$$1 = 94 - 3 \times 31$$

Now

$$3 = 97 - 94$$

$$= 97 - (385 - 97 \times 3)$$

$$= 97 - 385 + 97 \times 3$$

$$= 4 \times 97 - 385$$

Now, $1 = 94 - 3 \times 31$

$= (385 - 97 \times 3) - (97 \times 4 - 385) \times 31$

$= 385 - 97 \times 3 - 31 \times 97 \times 4 + 31 \times 385$

$= 385 + 31 \times 385 - 97 \times 3 - 124 \times 97$

$= 385 (1 + 31) - 97 (3 + 124)$

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

IT-21001

∴ Step 3

we have, $-127 \times 97 + 32 \times 385 = 1$

comparing it ~~to~~ $97x - 385y = 1$

we find $x = -127$

$y = 32$

∴ $97 \times (-127) \equiv 1 \mod (385)$

The positive inverse is

$x \equiv -127 \equiv 258 \pmod{385}$

∴ The multiplicative inverse of 97 mod 385 is 258

IT-21001

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

⇒ Let $d = \gcd(a,b)$

By the Euclidean algorithm there exist integers x, y (found by back-substitution) wit

$$ax+by = d$$

Hence the linear Diophantine equation $ax+by=c$ always has integer solutions

To find $43x \equiv 1 \bmod 240$

Step-1

Bezout's identity says:

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

for some integers x, y

Here $a = 43$, $b = 240$ and $\gcd(43, 240)$:

$$43x + 240y = 1$$

Step-2

Extended Euclidean Algorithm

$240 = 43 \times 5 + 25$

$43 = 25 \times 1 + 18$

$25 = 18 \times 1 + 7$

$18 = 7 \times 2 + 4$

$7 = 4 \times 1 + 3$

$4 = 3 \times 1 + 1$

$3 = 1 \times 3 + 0$

IT-21001

GCD = 1

Step-3 Back substitution

From $4 = 3 \times 1 + 1$

$\Rightarrow 1 = 4 - 3$

From $3 = 7 - 4$

From $4 = 18 - 7 \times 2$

From $7 = 25 - 18 \times 1$

From $18 = 43 - 25 \times 1$

From $25 = 240 - 43 \times 5$

IT-21001

Now $1 = 4 - 3$

$= 4 - (7 - 4)$

$= 2 \times 4 - 7$

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

$= 2 \times 18 - 4 \times 7 - 7$

$= 2 \times 18 - 5 \times 7$

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

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

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

$= 7 \times 43 - 7 \times 25 - 5 \times 25$

$= 7 \times 43 - 12 \times 25$

$= 7 \times 43 - 12 \times (240 - 43 \times 5)$

$= 7 \times 43 - 12 \times 240 + 60 \times 43$

$= 67 \times 43 - 12 \times 240$

$= 1$

So, $43 \times 64 \equiv 1 \pmod{240}$

Thus the modular inverse of 43 mod 240 is

$x \equiv 67 \pmod{240}$

(13) 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^{123} mod 175 using Fermat's Little Theorem. Show all steps.

IT-21001

⇒

Statement: If p is prime and a any integer, then $a^p \equiv a \pmod{p}$

if $\gcd(a, p) = 1$ this gives $a^{p-1} \equiv 1 \pmod{p}$

Short proof:

If $\gcd(a, p) = 1$ the nonzero residues mod p form a multiplicative group of order p-1. By Lagrange's theorem $a^{p-1} = 1$ in that group, so $a^{p-1} \equiv 1 \pmod{p}$

IT-21001

How used for primality testing?

For a candidate n, pick a with $1 < a < n$.
If $a^{n-1} \not\equiv 1 \pmod{n}$ then n is definitely composite. If $a^{n-1} \equiv 1 \pmod{n}$ than n is a probable prime for base a

$561 = 3 \times 11 \times 17$ is composite

$\gcd(a, 561) = 1$

$a^{560} \equiv 1 \pmod{561}$.

So a simple FLT test fail (declare 561 probable prime) even though 561 is composite FLT gives a necessary but not sufficient test

Evaluate: $5^{123} \bmod 175$

$175 = 25 \times 7$

Step 1

$5^r = 25$, so any power of 5 above 2 will be a multiple of 25

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

Step 2

Here gcd (5,7) = 1, so we can use fermat's Little Theorem:

$$5^6 \equiv 1 \pmod 7$$

$$123 \div 6 = 20 \text{ remainder } 3$$

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

Step 3 Combine using CRT

$$x \equiv 0 \pmod{25}, \quad x \equiv 6 \pmod 7$$

Let $x = 25k$, then,

$$25k \equiv 6 \pmod 7$$

Since $25 \equiv 4 \pmod 7$, we have

$$4k \equiv 6 \pmod 7$$

multiply both sides by 2 (because 2 is the inverse of 4 mod 7)

$$\therefore k \equiv 12 \equiv 5 \pmod 7$$

Smallest $k = 5$ given

$$x = 25 \times 5$$
$$= 125$$

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

IT-21001

(14) State and prove the Chinese Remainder Theorem. Then solve the following system of congruences.

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

⇒ If $n_1 \dots n_k$ are pairwise coprime positive integers and $a_1, \dots a_k$ are any integers, then the system

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

Here $x \equiv 2 \pmod 3$

$x \equiv 3 \pmod 5$

$x \equiv 2 \pmod 7$

• Consider

$a_1 = 2, a_2 = 3, a_3 = 2$

$m_1 = 3, m_2 = 5, m_3 = 7$

$m = 3 \times 5 \times 7$

$= 105$

$m_1 = \frac{105}{3} = 35$

$Y_1 = 35 \text{ modulo } 3$

$= 2$

$35 = 11 \times 3 + 2$

$3 = 1 \times 2 + 1$

$\Rightarrow 1 = 3 - 1 \times 2$

$= 3 - (35 - 11 \times 3)$

$= 12 \times 3 - 35$

$= -1$

$(-1) \bmod 3 \equiv 2$

$m_2 = \frac{105}{5} = 21$

$Y_2 = 21 \text{ modulo } 5$

$= 1$

$m_3 = \frac{105}{7} = 15$

IT-21001

$Y_3 = 15 \text{ modulo } 7$

$= 1$

$\therefore x = a_1 m_1 Y_1 + a_2 m_2 Y_2 + a_3 m_3 Y_3$

$= 2 \times 35 \times 2 + 2 \times 21 \times 1 + 2 \times 15 \times 1$

$= 233$

$= 23 \pmod{105}$

(15) Briefly explain the CIA triad in informat security. How does each component contribu to building a secure system?

⇒

CIA triad means Confidentiality, Integrity, Availability.

Confidentiality: It keep data secret by encryption, access control. Only authorized people / process can read or learn information.

common controls / technique :-

- Encryption
- Access control
- Authentication & strong password
- Network segmentation
- Data classification

IT-21001

Integrity: It means data is accurate and unmodified except by authorized action.

common controls/ technique:

- Checksums and cryptographic hashes
- Digital signatures
- Versioning and write-once logs
- Access controls and separation of duties

Availability: It means system and data are accessible to authorized users when needed.

common controls/ technique:

IT-21001

- Redundancy
- Backups and tested recovery plans
- load balancing & autoscaling.
- DDos protection and network safeguards
- Regular maintenance and patching.

IT-21001

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

⇒

Steganography and cryptography are both used in information security, but they project information in different ways.

Steganography	Cryptography
(i) Main goal hides and exitence of the message	(i) Makes the content unreadable without the key
(ii) The data is embedded in a carrier	(ii) The encrypted data is clearly visible as ciphertext
(iii) Security through concealment	(iii) Security through mathematical transformation
(iv) Convert communication, watermarking	(iv) Secure data transfer, authentication

IT-21001

(17) What are the key different between phishing, malware and denial-of-service (DoS) attacks in terms of their method and impact on system security?

Phishing	Malware	DoS
(i) Tricks people into revealing sensitive information	(i) Malicious software that infiltrates a system to steal, damage data	(i) Overwhelms a system with excessive request
(ii) Its Target human	(ii) Devices and data	(ii) Services
(iii) Its Primary goal steal credentials or sensitive data	(iii) steal data, damage files, spy on, gain control	(iii) Make a system or service unavailable
(iv) It compromises confidentiality	(iv) Can harm confidentiality, integrity and availability	(iv) Damages availability.
(v) For Prevention, user awareness training, spam filters are need	(v) Antivirus, firewalls regular patching are need.	(v) Rate limiting, firewalls, traffic filtering, redundancy.

IT-21001

(18) Explain how legal framworks such as the General Data Protection Regulation (GDPR) help mitigate cyber attacks and protect user privacy?

⇒ The General Data Protection Regulation is an EU law that sets strict rules on how organizations collect, store and use perso data.

Way GDPR helps Mitigate Cyber Attacks:

GDPR Requirement	How it reduces Risk
(i) Data Minimization	→ Less data stored = smaller target for attackers
(ii) Security by Design and Default	→ Forces systems to built with encryptio secure, coding and access control.
(iii) Access Controls & Accountability	→ Limits insider thre by tracking who ca view or modify pers information.

Way of GDPR Protects Users Privacy

Consent → Users must agree before their Personal data is collected or used

Right to Access → Users can see what data is stored about them

Data Portability → Users can transfer their data between services safely

Purpose Limitation → Data can only be used for the purpose it was collect

IT-21001

IT-21001

(19) Explain the basic working of the DES algorithm using a simple 64-bit plaintext block and a 56-bit key. Show how the initial permutation, round functions, and final permutation contribute to the encryption process.

⇒ DNS is a 64-bit block cipher with a 56-bit effective key (the 64-bit key input actually contains 8 parity bits)

High Level flow:

1. Initial Permutation (IP) → a fixed bit-recording of the 64-bit plaintext. IP does not add security. It just rearranges bits so the rest of the algorithm operates on a different ordering.
2. Split → result of IP is split into two 32-bit halves: L_0 and R_0

3. 16 Feistel rounds → each round i applies:

- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$

4. Swap and Final Permutation → after 16 rounds the left/right are swapped and then the inverse initial permutation (IP^{-1}) is applied to form the 64-bit ciphertext. IP and IP^{-1} are inverses, they only recoder bits and don't provide cryptographic strength by themselves.

IT-21001

IT-21001

(20) In the DES algorithm, a 64-bit plaintext block is divided into two 32-bit halves: L_0 and R_0. Given $R_0 = 0xF0F0F0F0$ and round key $k_1 = 0x0F0F0F0F$,
Compute the output of the 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 $L_0 = 0xAAAAAAAA$.

⇒ Given,

$R_0 = 0xF0F0F0F0$ (32 bits)

$K_1 = 0x0F0F0F0F$

$L_0 = 0xAAAAAAAA$

we assume $f(R_0, k_1) = R_0 \oplus k_1$

Compute $f(R_0, k_1)$ byte-wise

$0xF0 \oplus 0x0F = 0xFF$

So for the 4 bytes $f = 0xFFFFFFFF$

Now Apply feistel round formulas:

$L_1 = R_0 = 0xF0F0F0$

$R_1 = L_0 \oplus f(R_0, k) = 0xAAAAAAAA + 0xFFFFF$

Compute $0xA \oplus 0xF = 0x5$ nibble-wise ; no

$R_1 = 0x55555555$

Final numeric answers:

$f(R_0, k_1) = 0xFFFFFFFF$

$L_1 = 0xF0F0F0F0$

$R_1 = 0x55555555$

IT-21001

(21) Given the input word:

[0x23, 0xA7, 0x4C, 0x19]

Use the following partial AES S-box to perform the SubBytes transformation. Provide the resulting output word:

Row ↓ \Col →	3	4	7	9	A	C
1	6D	·	·	C6	·	·
2	D4	·	·	·	·	·
4	·	A1	·	·	·	2E
A	·	·	63	·	D2	·

IT-21001

⇒ Given input word:

[0x23, 0xA7, 0x4C, 0x19]

SubBytes lookup rule: high nibble = row,
low nibble = column.

Using the provided S-box.

1. Byte = 0x23 → row = 2, col = 3
   From table: (row 2, col 3) = D4
   ∴ S(0x23) = 0xD4
2. Byte = 0xA7 → row = A, col = 7
   From table: (row A, col 7) = 63
   ∴ S(0xA7) = 0x63

3. Byte = 0x4C → row = 4, col = C
From table: (row 4, col C) = 2E
∴ S(0x4C) = 0x2E

4. Byte = 0x19 → row = 1, col = 9
From table: (row 1, col 9) = C6
∴ S(0x19) = 0xC6

IT-21001

Resulting output word (after SubBytes):
[0xD4, 0x63, 0x2E, 0xC6]

(22) In AES encryption, apply the AddRoundKey step only. Given:

- Input word: [0x1A, 0x2B, 0x3C, 0x4D]
- Round key word: [0x55, 0x66, 0x77, 0x88]

Perform XOR between the input word and the round key, and write the resulting output word.

⇒ Given:
Input word = [0x1A, 0x2B, 0x3C, 0x4D]
Round key = [0x55, 0x66, 0x77, 0x88]
AddRoundKey (byte-wise XOR):

1. 0x1A ⊕ 0x55 = 0x4F
(0001 1010 ⊕ 0101 0101 = 0100 1111)

2. $0x2B \oplus 0x66 = 0x4D$
   $(0010\ 1011 \oplus 0110\ 0110 = 0100\ 1101)$
3. $0x3C \oplus 0x77 = 0x4B$
   $(0011\ 1100 \oplus 0111\ 0111 = 0100\ 1011)$
4. $0x4D \oplus 0x88 = 0xC5$
   $(0100\ 1101 \oplus 1000\ 1000 = 1100\ 0101)$

Resulting output word (after AddRoundKey):

$$[0x4F, 0x4D, 0x4B, 0xC5]$$

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

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

to transform an input column. Use an example column with values $[0x01, 0x02, 0x03, 0x04]$. (Convert Hex to Binary)

⇒ Convert input bytes to binary:

$0x01 = (0000\ 0001)_2$
$0x02 = (0000\ 0010)_2$
$0x03 = (0000\ 0011)_2$
$0x04 = (0000\ 0100)_2$

IT-21001

Reminder: multiplication by 02 and 03 in AES.

- xtime(x) = 02·x is left shift (one bit) with reduction by 0x1B if MSB was 1. For small values here MSB=0 so no reduction is needed.
- 03·x = (02·x) ⊕ x = xtime(x) ⊕ x.

Compute needed multiples (hex and binary):

02. 0x01 = 0x02 (0000 0010)
03. 0x01 = 0x02 ⊕ 0x01 = 0x03 (0000 0011)
02. 0x02 = 0x04 (0000 0100)
03. 0x02 = 0x04 ⊕ 0x02 = 0x06 (0000 0110)
02. 0x03 = 0x06 (0000 0110)
03. 0x03 = 0x06 ⊕ 0x03 = 0x05 (0000 0101)
02. 0x04 = 0x08 (0000 1000)
03. 0x04 = 0x08 ⊕ 0x04 = 0x0C (0000 1100)

Matrix multiplication over GF(2^8)

Let output column be $b = [b_0, b_1, b_2, b_3]^T$. Compute each row:

Row 0 → $b_0 = 02 \cdot a_0 \oplus 03 \cdot a_1 \oplus 01 \cdot a_2 \oplus 01 \cdot a_3$

$b_0 = 0x02 \oplus 0x06 \oplus 0x03 \oplus 0x04$

$= 0x03$

Row 1 → $b_1 = 01 \cdot a_0 \oplus 02 \cdot a_1 \oplus 03 \cdot a_2 \oplus 01 \cdot a_3$

$b_1 = 0x01 \oplus 0x04 \oplus 0x05 \oplus 0x04$

$= 0x04$

IT-21001

Row 2 → $b_2 = 01 \cdot a_0 \oplus 01 \cdot a_1 \oplus 02 \cdot a_2 \oplus 03 \cdot a_3$

$b_2 = 0x01 \oplus 0x02 \oplus 0x06 \oplus 0x0C$

$= 0x09$

Row 3 → $b_3 = 03 \cdot a_0 \oplus 01 \cdot a_1 \oplus 01 \cdot a_2 \oplus 02 \cdot a_3$

$b_3 = 0x03 \oplus 0x02 \oplus 0x03 \oplus 0x08$

$= 0x0A$

Final Result (hex and binary):

$$b = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 0x03 \\ 0x04 \\ 0x09 \\ 0x0A \end{bmatrix} = \begin{bmatrix} 0000\ 0011 \\ 0000\ 0100 \\ 0000\ 1001 \\ 0000\ 1010 \end{bmatrix}$$

(24) Describe how AES-OFB mode works. How does it ensure synchronization between encryption and decryptions streams?

⇒ OFB turns AES into synchronous stream cipher by repeatedly encrypting an internal feedback value (starting from an IV) to produce a keystream which is XORed with plaintext.

IT-21001

Notation Encryption:

For each block i:

1. Generate keystream: $O_i = E_K(O_{i-1})$.
2. XOR with plaintext: $C_i = P_i \oplus S_i$ where $S_i = O_i$
3. No padding required for final partial block - just XOR the needed bytes.

Decryption:

Receiver (knowing K and IV) repeats keystream generation:

1. $O_i = E_K(O_{i-1})$
2. Recover plaintext: $P_i = C_i \oplus S_i$

Properties:

- Synchronous stream cipher: keystream depends only on K and IV, not on plaintext or ciphertext.
- No error propagation: a bit error in ciphertext affects only the corresponding bit(s) of plaintext (no cascade).
- No padding needed: works on partial blocks by XORing only required bytes.
- Malleability: Flipping ciphertext bits flips corresponding plaintext bits - integrity needs MAC.
- IV reuse is catastrophic: reusing the same IV and key for two messages leaks $P_1 \oplus P_2$.

IT-21001

How OFB ensures synchronization:

Both sender and receiver must start with the same IV and key and generate keystream blocks in the same order. If both process the same number of blocks, Oi will match at both ends, so XOR recovers the plaintext.

(25) Which AES modes cause error propagation during decryption, and how does this affect the integrity of the decrypted message? Illustrate with CBC and CFB modes.

⇒ Some AES modes cause an error in a single ciphertext block to affect multiple decrypted blocks. This is called error propagation.

CBC (Cipher Block Chaining) Mode:

→ If one ciphertext block is corrupted,
- the corresponding plaintext block is completely garbled.
- The next plaintext block will have bit errors in the same positions as the corrupted bits.

IT-21001

→ Effect: Errors spread to two consecutive blocks during decryption.

CBF (Cipher Feedback) Mode:

→ If one ciphertext segment is corrupted,
- The corresponding plaintext segment is completely wrong.
- Following plaintext segment will have bit errors in the same bit positions.

→ Effect: Errors propagate to two segments.

Impact on Integrity: Error propagation prevents recovery of full plaintext without error detection mechanisms (e.g., MAC), making integrity verification essential.

Illustration:

Error in C2 → P2 (garbled), P3 (bit errors)

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

IT-21001

⇒ CTR (Counter) mode is the recommended AES mode for large files with parallel processing.

Justification:

→ ECB: Supports parallel encryption but insecure — identical plaintext blocks produce identical ciphertext blocks (pattern leakage).

→ CBC: Secure but cannot encrypt in parallel (each block depends on the previous ciphertext).

→ CTR:
- Secure (no pattern leakage).
- Each block uses a unique counter value, so encryption/decryption can be fully parallelized.
- Suitable for large files due to high speed and efficiency in multi-core systems.

So, CTR is the best for large file encryption.

(27) 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.

⇒ RSA Encryption & Deciyption:

Given:

M=1,

Public key: e=5, n=14.

private key: d=11.

Encryption:

$C = M^e \bmod n$;

$\Rightarrow C = 1^5 \bmod 14$

$\Rightarrow C = 1 \bmod 14 = 1$

∴ ciphertext, C=1

IT-21001

Decryption:

$M = C^d \bmod n$

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

$\Rightarrow M = 1 \bmod 14 = 1$

∴ Recovered Message: M=1

so, encryption and decryption didn't change the value.

(28) Given: Message hash: H(M)=5, RSA Private Key: d=3, n=33. Generate the digital signature.

⇒ Given

$H(M) = 5$

Private key: $d = 3, n = 33$

Formula:

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

$\Rightarrow S = 5^3 \bmod 33$

$\Rightarrow S = 125 \bmod 33$

$\therefore S = 26 \quad [\because 33 \times 3 = 99,\ 125 - 99 = 26]$

$\therefore$ Digital Signature, $S = 26$

IT-21001

(29) Aleya and Badol are using the Diffie-Hellman key exchange protocol. They agree on the following public values: Prime modulus: $p = 17$, Base (generator): $g = 3$, Aleya chooses a private key, $a = 4$, Badol chooses a private key $b = 5$. Compute public key of Aleya and Badol.

⇒ Given:

$p = 17,\ g = 3$

Aleya's private key, $a = 4$

Badal's private key, $b = 5$

IT-21001

Formula for Public key:

$$\text{Public key} = g^{\text{private}} \bmod p$$

Aleya's public key (A):

$$A = 3^4 \bmod 17$$

$$3^4 = 81$$

$$81 \bmod 17 = 81 - (17 \times 4) = 81 - 68 = 13$$

$$\therefore A = 13$$

Badol's public key (B):

$$B = 3^5 \bmod 17$$

$$3^5 = 243$$

$$243 \bmod 17 = 243 - (17 \times 14) = 243 - 238 = 5$$

$$\therefore B = 5$$

Therefore, Aleya's Public key = 13

Badol's Public Key = 5

(30) A simple (non-cryptographic) hash function $H(x)$ is defined as the sum of ASCII values of the characters in a message, modulo 100:

$$H(x) = (\Sigma \text{ ASCII of characters in } x) \bmod 100$$

Compute the hash value of the message "AB" and "BA" using this function. Do the two messages produce

the same hash? What does this imply about collision resistance in weak hash functions?

⇒ Given:

Hash function:

$$H(x) = (\Sigma \text{ ASCII values of characters in } x) \bmod 100$$

step-1: Calculate hash of "AB":

ASCII('A') = 65

ASCII('B') = 66

IT-21001

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

$$= 131 \bmod 100 = 31$$

step-2: calculate hash of "BA":

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

$$= 131 \bmod 100 = 31$$

step-3: Do they produce the same hash?

Yes, both "AB" and "BA" produce the same hash value: 31.

step-4: What does this imply about collision resistance?

→ This hash function is weak because different messages can produce the same hash (collision).

→ It lacks collision resistance, meaning it's easy to find two different inputs with the same hash.

→ Therefore, this hash function is not secure for cryptographic use.

IT-21001

(31) In a secure message system, a simple Message Authentication Code (MAC) is computed using modular addition:

$$MAC = (Message + SecretKey) \bmod 17$$

Given: Message: 15, SecretKey: 7. 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 MAC easily? Explain briefly.

⇒ Given:

Message, $M = 15$

Secretkey, $K = 7$

$$MAC = (M + K) \bmod 17$$

Computing MAC for original message:

MAC = (15 + 7) mod 17

MAC = 22 mod 17 = 5

original MAC = 5

IT-21001

After attacker changes message to M' = 10:
Without knowing K, attacker cannot directly compute correct MAC.

However, because this MAC uses simple modular addition and small modulus, attacker could try all possible keys easily (brute force). Thus, this method is not secure for real-world use.

(32) Explain the steps involved in the TLS handshake process. How are symmetric keys established securely using asymmetric cryptography during the handshake?

IT-21001

⇒ TLS Handshake Steps!

1. ClientHello
   - client sends! supported TLS version, cipher suites, random number R_c.
2. ServerHello
   - Server sends! chosen cipher suite, random number R_s, server's digital certificate (contains public key).
3. Server Authentication
   - Client verifies server's certificate using CA's public key to ensure authenticity.
4. Key Exchange
   - RSA-based! client generates a pre-master secret, encrypts it with server's public key, sends to server.
   - Diffie-Hellman/ECDH! Client and server exchange public parameters to compute the same shared secret.
5. Session Key Derivation
   - Both sides use!

     Pre-master secret + R_c + R_s → Session (symmetric) keys
6. Finished Messages
   - Both sides send encrypted "Finished" messages to confirm handshake success.

How Symmetric keys are Established Securely:

- Asymmetric cryptography (RSA or Diffie-Hellman) is used only to exchange or agree upon a shared secret over an insecure network.
- This shared secret is then used to derive symmetric keys (AES, chacha20, etc.) for fast, secure data encryption.
- Even if intercepted, the asymmetric encryption prevents an attacker from recovering the symmetric key.

IT-21001

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

⇒ SSH has three main layers, each with specific roles:

1. Transport Layer (SSH-TRANS)
   - Purpose: Provides encryption, integrity, and server authentication.
   - Functions:
     - Establishes secure channel over TCP (port 22)

- Negotiates algorithms (encryption, MAC, compression).
- Performs key exchange (Diffie-Hellman / ECDH).

2. User Authentication Layer (SSH-USERAUTH)
   - Purpose: Authenticates the client (user) to the server.
   - Methods:
     - Password authentication
     - Public key authentication
     - Keyboard-interactive

IT-21001

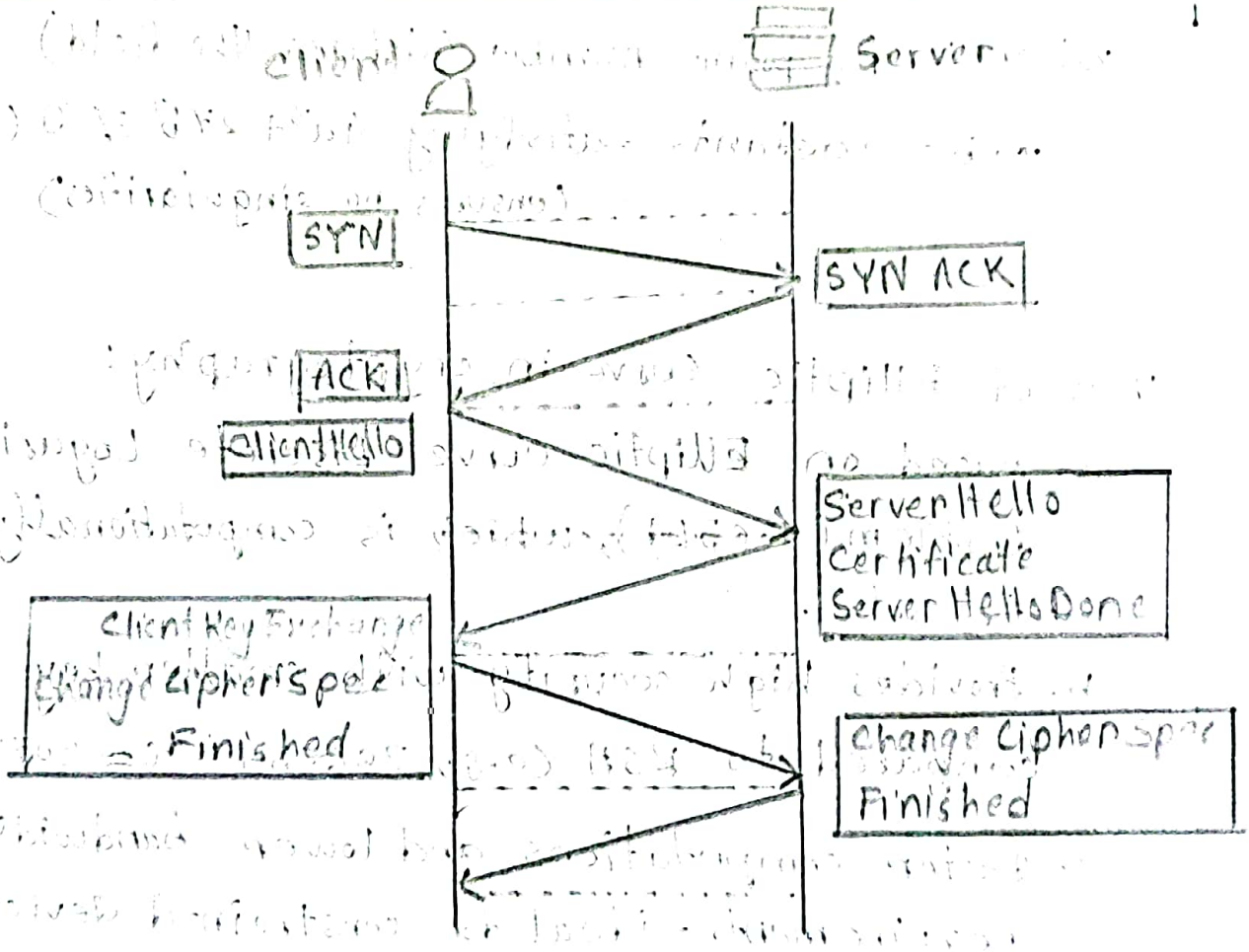
3. Connection Layer (SSH-Connect)
   - Purpose: Manages multiple logical channels over the secure tunnel.
   - Functions:
     - Multiplexing (run shell, execute commands, port forwarding).
     - Manages session control.

(34) Explain the steps involved in TLS handshake process.

⇒ TLS Handshake steps are –

1. ClientHello
2. ServerHello
3. Server Authentication
4. Key Exchange
5. Session Key Derivation
6. Finished Messages

IT-21001



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

⇒ Elliptic Curve Equation General Form (over a finite field F_p):

$$y^2 \bmod p = (x^3 + ax + b) \bmod p$$

where: p = prime number (defines the field)

a, b = constants satisfying $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ (ensures no singularities)

Uses of Elliptic curve in cryptography:

a. Based on Elliptic Curve Discrete Logarithm Problem (ECDLP), which is computationally hard to solve.

b. Provides high security with smaller key sizes compared to RSA (e.g., 256-bit ECC ≈ 3072-bit RSA)

c. Faster computations and lower bandwidth requirements – ideal for constrained devices.

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

⇒ ECC (Elliptic curve cryptography) is based on the Elliptic curve Discrete Logarithm Problem (ECDLP), which is much harder to solve than

the integer factorization problem used in RSA.

Because of this higher mathematical hardness, smaller ECC keys can provide the same security as much larger RSA keys.

Example:

- ECC 256-bit key ≈ RSA 3072-bit key in security strength.
- ECC 384-bit key ≈ RSA 7680-bit key.

Advantages:

a. Less computation time
b. Lower memory and bandwidth usage.
c. Suitable for mobile, IoT, and constrained devices.

IT-21001

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

⇒ Let's check if $P = (3,6)$ lies on $y^2 \equiv x^3 + 2x + 3 \pmod{97}$

Compute LHS:

$$y^2 \bmod 97 = 6^2 \bmod 97 = 36$$

Compute RHS:

$$x^3 + 2x + 3 = 3^2 + 2 \cdot 3 + 3 = 36$$

$$36 \bmod 97 = 36$$

Since LHS = RHS = 36,

(3,6) Lies on the curve over F_{97}.

IT-21001

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

⇒ Given:

$p=23$, $g=5$, $h=8$ (public key)

$m=10$ (message)

$k=6$ (random number)

IT-21001

Step-1: Compute C_1:

$$C_1 = g^k \bmod p$$

$$C_1 = 5^6 \bmod 23$$

$$5^2 = 25 \equiv 2 \pmod{23}$$

$$5^4 \equiv 2^2 = 4$$

$$5^6 \equiv 4 \times 2 = 8$$

So:

$$C_1 = 8$$

Step-2: Compute C_2:

$$C_2 = (m \times h^k) \bmod p$$

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

IT-21001

$8^2 = 64 \equiv 18$

$8^4 \equiv 18^2 = 324 \equiv 2$

$8^6 \equiv 2 \times 18 = 36 \equiv 13$

So: $C_2 = (10 \times 13) \bmod 23 = 130 \bmod 23$

$130 - (23 \times 5) = 130 - 115 = 15$

$C_2 = 15$

Final Ciphertext:

$(C_1, C_2) = (8, 15)$

(39) Explain how lightweight cryptography is important for securing IOT devices. Give one example of a light weight encryption algorithm used in IOT.

⇒ Importance of Lightweight Cryptography for IoT:

i. IoT devices (sensors, smart meters, wearables) have limited CPU power, memory, and battery.

ii. Traditional algorithms like AES-256 or RSA require high computation and memory, which can drain power and slow performance.

iii. Lightweight cryptography is designed to use less processing power, smaller keys, and lower energy while still providing adequate security.

iv. Essential for secure communication, authentication, and data integrity in IoT networks.

Example Algorithm:

- PRESENT cipher: 64-bit block cipher with 80/128-bit key sizes.
- Very small hardware footprint, low power usage, and suitable for constrained IoT devices.

IT-21001

IT-21001

(40) List and briefly explain any three common IoT-specific attacks (e.g., firmware hijacking, physical tampering, botnets like Mirai). What mitigation strategies can be applied?

⇒ Common IoT specific Attacks & Mitigation

1) Firmware Hijacking
   → Attack: Malicious firmware update replaces legitimate code, giving attacker full control.
   → Mitigation: Use digitally signed firmware and secure update protocols.

2) Physical Tampering
   → Attack: Direct access to IoT device hardware to steal data or modify functions.
   → Mitigation: ~~Change~~ Tamper-evident seals, strong physical casing, and sensor-based tamper detection.

3) Botnet Attacks (e.g., Mirai)
   → Attack: Malware infects IoT devices, linking them into a botnet for DDoS or spam

attacks.

→ Mitigation: Change default passwords, apply regular security patches, and use network firewalls.

Therefore, Securing IoT devices requires both physical and software defenses to reduce attack risks.

IT-21001