Shetu Saha
IT-21009

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?

Answer:

Shor's algorithm poses a significant threat to RSA and Elliptic Curve Cryptography (ECC) because it can efficiently solve the mathematical problems that form their security foundation. On a sufficiently powerful quantum computer, the algorithm could break these cryptosystems in a fraction of the time it would take a classical computer, rendering them obsolete.

How Shor's Algorithm Threatens RSA and ECC

The security of RSA and ECC relies on mathematical problems that are currently considered "hard" for classical computers to solve. Shor's algorithm, however, leverages the principles of quantum mechanics to solve these problems exponentially faster.

Assignment
CSE 412-II

Shetu Saha
IT-21009

- RSA : RSA's security is based on the integer factorization problem. The public key is the product of two very large prime numbers, and the private key consists of those two prime number's. It's easy to multiply two prime numbers to get the public key, but it's computationally intractable for classical computers to reverse the process and find the two original primes from their product. Shor's algorithm, on a quantum computer, can factor these large numbers in polynomial time, effectively cracking the private key and making the encryption useless.

- ECC : ECC's security is based on the elliptic curve discrete logarithm problem (ECDLP). This problem involves finding an integer K given two points on an elliptic curve, P and Q, such that $Q = KP$. Like integer factorization, this is extremely difficult for classical

Shetu Saha
IT-21009

computers to solve. Shor's algorithm can also solve the ECDLP in polynomial time, compromising the security of ECC-based systems.

Potential Consequences for Digital Infrastructure

The development of a practical quantum computer capable of running Shor's algorithm would have far-reaching and severe consequences for current digital infrastructure. This "Y2Q" or "Q-day" scenario would essentially break the backbone of modern public-key cryptography.

- Compromised Digital Security : Much of our online security relies on RSA and ECC. This includes secure websites (HTTPS), VPNs, digital signatures, and encrypted emails. An adversary with a powerful quantum computer could decrypt sensitive data that has been intercepted and stored; a threat known as "harvest now, decrypt later".

Shetu Saha
IT-21009

- Vulnerability of Critical Systems: Financial transactions, government communications, critical infrastructure (like power grids and water systems), and healthcare records are all protected by these vulnerable cryptographic systems. Their security could be severely undermined, leading to potential data & breaches, fraud, and a loss of trust in digital systems.

- Need for Post-Quantum Cryptography (PQC): The threat has spurred a race to develop and standardize new cryptographic algorithms that are resistant to quantum attacks. These post-quantum cryptography (PQC) algorithms are designed to be secure against both classical and quantum computers. However, the transition to PQC will be a massive undertaking, requiring widespread software and hardware updates across the globe.

Shetu Saha
IT-21009

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

Answer:

Role of Quantum Key Distribution (QKD) in future cryptographic systems:

Quantum Key Distribution (QKD) uses the principles of quantum mechanics - especially the no-cloning theorem and measurement disturbance - to securely share encryption keys between two parties. Its main role in the future will be to provide information-theoretic-security for key exchange, meaning the security doesn't depend on the hardness of a mathematical problem (like factoring or discrete logs) and is safe even against quantum computers.

It's expected to complement post-quantum cryptography (PQC) by offering an extra-secure

Shetu Saha
IT-21009

way to exchange ~~syste~~ symmetric keys for encryption in sensitive fields like banking, defense, and critical infrastructure.

Key differences ~~of~~ from classical public key encryption:

1. Security Basis:

- QKD: Relies on quantum physics laws; eavesdropping changes the quantum states, revealing intrusion.
- Classical Public Key: Relies on computational hardness assumptions (e.g. factoring large numbers).

2. Vulnerability to Quantum Attacks

- QKD: Immune to shor's algorithm and other quantum attacks (assuming ideal devices).
- Classical public key: Broken if a large quantum exists.

Shetu Saha
IT-21009

3. Purpose :

- QKD: Only exchanges symmetric keys securely; still needs classical algorithms (like AES) for data encryption.
- Classical public key : Directly used for encryption, signatures and authentication.

4. Infrastructure Needs:

- QKD: Requires specialized hardware (single-photon sources, detectors, quantum channels like fiber opties or St satellites).
- Classical Public key : Can be implemented entirely in software over existing internet infrastructure

In short QKD secures the key exchange by physics, not math, making it a strong candidate for the most sensitive communications in a post-quantum era.

Shetu Saha
IT-21009

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

Answer:

Main differences between lattice-based cryptography and RSA (number-theoretic):

1. Mathematical Foundation

- Lattice-based: Security relies on hard lattice problems (e.g. Learning with Errors, Shortest Vector Problem).
- RSA: Security relies on the hardness of integer factorization.

2. Quantum Resistance

- Lattice based: Believed to be resistant to quantum attacks; no known efficient quantum algorithm (including Shor's) Solves its core problems.

Shelu Saha
IT-21009

- RSA : Broken by Shor's algorithm on a large quantum computer.

3. Performance & Flexibility:

- Lattice-based : Often faster key generation, supports advanced features like fully homomorphic encryption.
- RSA : Mature and widely used, but generally slower for equivalent security levels.

4. Key Sizes:

- Lattice-based : Larger public/private keys than RSA but still practical.
- RSA : Keys grow very large for high security

In short Lattice-based cryptography shifts from factorization to hard geometric problems, offering strong candidates for post-quantum security, while RSA becomes insecure in a quantum era.

Shetu Saha
IT-21009

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

Answer:

Here's a python PRNG (Pseudo-Random Number Generator) that uses current system time and a custom seed value to generate random numbers. I'll also include sample output.

Python Program :

```
import time

def custom_prng(seed, count):
    current_time = int(time.time_ns())
    combined_seed = seed ^ current_time
    a = 1664525
    c = 1013904223
    m = 2 ** 32
```

Shetu Saha
IT-21009

```
random_numbers = [ ]
  X= combined_seed
for_in range (count):
    X= (a*X+c)% m
 random_numbers.append(X)
 return random_numbers
 seed_value = 12345
  count = 5
 numbers = custom_prng (seed_value, count)
 print = ("custom PRNG Output:")
for i, num in enumerate (numbers, 1):
  print ("Random Number {i}: {num} ")
```

Sample Output:

Custom PRNG Output:

Random Number 1: 1885951992

Random Number 2: 2910389135

Random Number 3: 24757 39278

Random Number 4: 33994785

Random Number 5: 1326364488

Shetu Saha
IT-21009

Explanation:

(i) Seed initialization: Combines custom seed and system time (nanoseconds) using XOR for randomness.

(ii) LCG formula: Uses $(a * X + c) \% m$ to generate pseudo-random numbers.

(iii) Quantum Safety: This PRNG is not cryptographically secure but is suitable for simulation/random tasks.

(iv) Dynamic Output: Even with the same seed different times give different results.

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?

Shetu Saha
IT-21009

Answer:

Sieve of Eratosthenes Algorithm:

The sieve of Eratosthenes Algorithm is an ancient and efficient method for finding all prime numbers up to a given limit n. It works by progressively eliminating the multiples of each prime number, starting from the smallest prime(2).

Algorithm Steps:

1. Create a list of integers from 2 to n
2. Start with the first number in the list (P=2), which is prime.
3. Eliminate all multiples of p.
4. Find the next number in the list that is not marked; this is the next prime.
5. Repeat Steps 3-4 until all multiples up to $\sqrt{n}$ have been processed.

6. The remaining unmarked numbers are all primes.

Shetu Saha
IT-21009

Find all primes less than 50

Let's use the Sieve of Eratosthenes to find all prime numbers less than 50.

1. Create a list of numbers from 2 to 49:

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

2. Start with the first prime 2 then eliminates all multiple of '2':

2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49.

3. The next unmarked number 3. Eliminate all multiple of 3:

2, 3, 5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49

Shetu Saha
IT- 21009

4. The next unmarked number 5. Eliminate all multiple of 5:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 49

5. The next unmarked number 7. Eliminate all multiple of 7:

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

6. The next unmarked number is 11, and $11^2 = 121$, which is greater than 50. The algorithm stops here.

The remain unmarked numbers are the prime numbers less than 50:

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

[P.T.O]

Time complexity comparison:

- Sieve of Eratosthenes: $O(n \log \log n)$
  - very efficient for large ranges
- Trial Division: $O(n\sqrt{n})$
  - much slower for large n.

Thus, the sieve is significantly faster than trial division when generating all primes up to a large number.

Shetu Saha
IT-21009

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

Shetu Saha
IT-21009

Answer:

Definition: A carmichael number is a composite integer n such that

$$a^{n-1} \equiv 1 \pmod{n}$$

for every integer a with $\gcd(a,n)=1$.

A positive composite integer n is a carmichael number if all three conditions hold:

1. n is composite
2. n is square free
3. for every prime p dividing n, $(p-1)$ divides $(n-1)$; i.e. $p-1 \mid n-1$.

1) $n = 561$

Factorization : $561 = 3 \times 11 \times 17$ (all exponents = 1 → square-free)

$n - 1 = 560$. Check divisibility :

- $p = 3$: $p - 1 = 2$, $560 \bmod 2 = 0$
- $p = 11$: $p - 1 = 10$, $560 \bmod 10 = 0$
- $p = 17$: $p - 1 = 16$, $560 \bmod 16 = 0$

All conditions satisfied ⇒ 561 is a Carmichael number.

Shetu Saha
IT-21009

2) $n = 1105$

Factorization : $1105 = 5 \times 13 \times 17$ (square-free)

$n - 1 = 1105 - 1 = 1104$ Check :

- $p = 5$: $p - 1 = 4$, $1104 \bmod 4 = 0$
- $p = 13$: $p - 1 = 12$, $1104 \bmod 12 = 0$
- $p = 17$: $p - 1 = 16$, $1104 \bmod 16 = 0$

All conditions satisfied ⇒ 1105 is a Carmichael number.

3) $n = 1729$

Factorization : $1729 = 7 \times 13 \times 19$ (square-free)

$n-1 = 1728$. Check:

- $p = 7 : p-1 = 6, \quad 1728 \bmod 6 = 0$
- $p = 13 : p-1 = 12, \quad 1728 \bmod 12 = 0$
- $p = 19 : p-1 = 18, \quad 1728 \bmod 18 = 0$

All satisfied $\Rightarrow$ 1729 is a Carmichael number.

All three numbers 561, 1105, 1729 are Carmichael numbers.

Shetu Saha
IT-21009

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

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

Answer:

Yes, In fact $\mathbb{Z}_{11}$ is a commutative ring with unity and moreover a field.

Shetu Saha
IT-21009

Justification:

- $(\mathbb{Z}_{11}, +)$ is an abelian group: closure, associatively, identity 0, inverses $(-a \equiv 11 - a)$, and commutativity hold (addition mod 11)
- Multiplication mod 11 is closed and associative, and distributes over addition.
- There is a multiplicative identity $1 \in \mathbb{Z}_{11}$.
- Because 11 is prime, every nonzero element $a \in \mathbb{Z}_{11}$ has a multiplicative inverse mod 11.

Thus all ring axioms hold and every

nonzero element is invertible → Z_{11} is a field.

Yes, $(Z_{37}, +)$ is an Abelian group.

Justification:

- Set $Z_{37} = \{0, 1, \ldots, 36\}$ with addition modulo 37.
- closure, associativity and commutativity follow from integer addition.
- Identity is 0. for each a, additive inverse is $37-a$.
- Thus $(Z_{37}, +)$ satisfies all group axioms and is abelian

Sheta Saha
IT-21009

No, $(Z_{35}, \times)$ is not a group.

Justification:

- Although multiplication mod 35 is associative and has Identity 1, not every element has a multiplicative inverse in Z_{35}.
- Example: $\bar{5} \in Z_{35}$, $\gcd(5, 35) = 5 > 1$. There is no x with $\bar{5}x \equiv 1 \pmod{35}$ because any product

$5x$ is divisible by 5 and can not be congruent to 1. Hence 5 has no inverse.

• Therefore the inverse axiom fails and (Z_{35}, X) is not a group, so it is not an abelian group.

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

Shetu Saha
IT-21009

Answer:

We want the remainder of $-52 \bmod 31$.

Step-by-Step:

1. Division: $-52 \div 31$ gives quotient -2 because $-2 \times 31 = -62$ is the nearest multiple of 31 not greater than -52.

2. Remainder:

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

3. Since we're taking remainder in the standard non-negative form, the answer is: 10

Final: $-52 \equiv 10 \pmod{31}$.

9. Determine the multiplicative inverse of 7 mod 26, if it exists. (Use extended Euclidean algorithm).

Answer:

We solve $7x \equiv 1 \pmod{26}$ or find integers x, y with $7x + 26y = 1$

Use the Euclidean algorithm:

Shetu Saha
IT-21009

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

Back-substitute to express 1 as a linear combination:

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

Thus $-11 \cdot 7 + 3 \cdot 26 = 1$. Therefore $x \equiv -11 \pmod{26}$

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

The multiplicative inverse of 7 modulo 26 is 15.

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

Shetu Saha
IT-21009

Answer:

Step-1: Multiply the numbers:

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

Step-2: Reduce modulo 17:

We find the equivalent positive remainder:

$$-40 + 3 \times 17 = -40 + 51 = 11$$

$$-40 \equiv 11 \pmod{17}$$

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

Explain of simplification method:

Negative numbers in modular arithmetic can be converted to their positive equivalent before multiplying.

$$-8 \equiv 9 \pmod{17}$$

$$9 \times 5 = 45$$

Reduce 45 modulo 17:

$45 - 2 \times 17 = 45 - 34 = 11$; This match the previous result.

∴ Final answer - 11 (Ans:)

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

Answer:

Bezout's Theorem Statement:

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

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

Equivalently, the set of all integer linear combinations $\{ax+by : x,y \in \mathbb{Z}\}$ is the ideal generated by $\gcd(a,b)$.

Shetu Saha
IT-21009

Proof:

Let $S = \{ax+by > 0 : x,y \in \mathbb{Z}\}$, S is a non-empty set of positive integers, so it has a least element d.

By construction $d = ax_0 + by_0$ for some integers x_0, y_0.

Use division with remainder: for any integer m (in particular $m=a$), write $m = qd+r$ with $0 \leq r < d$.

But $r = m - qd$ is also an integer linear combination of a and b. If $r > 0$ it belongs to S and is smaller than d, contradiction. Hence $r = 0$ and $d \mid m$. So d divides both a and b. Thus d is a common divisor. Any common divisor of a, b divides every combination $ax + by$, hence divides d. So $d = \gcd(a,b)$. That proves existence of integers x_0, y_0 with $ax_0 + by_0 = \gcd(a,b)$.

Sheta Saha
IT-21009

Use Bezout to find $97^{-1} \bmod 385$

We need integers x, y. with

$$97x + 385y = \gcd(97, 385)$$

Apply the Euclidean Algorithm (showing each step):

1. $385 = 3 \cdot 97 + 94$
2. $97 = 1 \cdot 94 + 3$
3. $94 = 31 \cdot 3 + 1$
4. $3 = 3 \cdot 1 + 0$

So gcd (97, 385) = 1 (hence an inverse exists).

Back-substitute to express 1 as a combination:

From step 3: $1 = 94 - 31 \cdot 3$

From step 2: $3 = 97 - 1 \cdot 94$

Substitute:

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

From step 1: $94 = 385 - 3 \cdot 97$. Substitute:

$$1 = 32(385 - 3 \cdot 97) - 31 \cdot 97$$
$$= 32 \cdot 385 - (96 + 31) \cdot 97$$
$$= 32 \cdot 385 - 127 \cdot 97$$

Shetu Saha
IT-21009

So we have

$$-127 \cdot 97 + 32 \cdot 385 = 1.$$

Reducing modulo 385 gives

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

So -127 is the multiplicative inverse of 97 mod 385. Make it the standard representative in $\{0, \ldots, 384\}$:

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

Answer: $97^{-1} \bmod 385 = 258$ (Ans:)

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

Answer:

Bezout Identity Statement:

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

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

Shetu Saha
IT-21009

Proof:

1. Let $d=\gcd(a,b)$ By definition, d divides, d divides both a and b, so $a=da'$ and $b=db'$, where a', b' are integers with $\gcd(a',b')=1$.

2. Since a' and b' are coprime, there exists integers x_0, y_0 such that

$$a'x_0+b'y_0=1.$$

3. Multiplying through by d gives:

$$a(dx_0) + b(dy_0) = d$$

Thus $x = x_0$ and $y = y_0$ are integer solutions to: $ax + by = \gcd(a, b)$

Hence proved.

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

This asks for the multiplicative inverse of 43 modulo 240. We solve the Diophantine equation.

$$43x - 240y = 1$$

by the extended Euclidean Algorithm.

Compute gcd chain (Euclidean algorithm):

$$240 = 5 \cdot 43 + 25,$$
$$43 = 1 \cdot 25 + 18,$$
$$25 = 1 \cdot 18 + 7,$$
$$18 = 2 \cdot 7 + 4,$$
$$7 = 1 \cdot 4 + 3,$$
$$4 = 1 \cdot 3 + 1,$$
$$3 = 3 \cdot 1 + 0.$$

Sheta Saha
IT-21009

So gcd $(43, 240) = 1$ and the inverse exist.

Now back-substitute to express 1 as a linear combination of 43 and 240.

Back-substitution:

$$1 = 4 - 1 \cdot 3$$
$$= 4 - 1 \cdot (7 - 1 \cdot 4) = 2 \cdot 4 - 1 \cdot 7$$
$$= 2 \cdot (18 - 2 \cdot 7) - 1 \cdot 7 = 2 \cdot 18 - 5 \cdot 7$$
$$= 2 \cdot 18 - 5 \cdot (25 - 1 \cdot 18) = 7 \cdot 18 = 5 \cdot 25$$
$$= 7 \cdot (43 - 1 \cdot 25) - 5 \cdot 25 = 7 \cdot 43 - 12 \cdot 25$$
$$= 7 \cdot 43 - 12 (240 - 5 \cdot 43)$$
$$= 7 \cdot 43 - 12 \cdot 240 + 60 \cdot 43$$
$$= 67 \cdot 43 - 12 \cdot 240$$

Shetu Saha
IT-21009

Thus $43 \cdot 67 - 240 \cdot 12 = 1$,

So, $x = 67$ is a solution.

$43^{-1} \equiv 67 \pmod{240}$ or $43 \cdot 67 \equiv 1 \pmod{240}$.

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 $5^{123} \bmod 175$ using Fermat's Little Theorem. Show all steps.

Answer:

Fermat's Little Theorem (FLT) Statement:

If p is a prime number and a is any integer with $\gcd(a, p) = 1$, then

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

Shetu Saha
IT-21009

Proof:

Consider the nonzero residues modulo p: $1, 2, \ldots, p-1$. Multiply each by a (with $\gcd(a, p) = 1$)

The set $a \cdot 1, a \cdot 2, \ldots, a \cdot (p-1)$

is a permutation of $1, 2, \ldots, p-1$ modulo p.

Taking the product of all elements in both sets and reducing mod p gives-

$$a^{p-1} \cdot (1 \cdot 2 \ldots (p-1)) \equiv 1 \cdot 2 \cdots (p-1) \pmod{p}$$

Cancelling the nonzero product $1.2. \cdots (p-1)$ (which is invertible mod p) yields $a^{p-1} \equiv 1 \pmod{p}$

Primality Test Using FLT:

Choose a with $1 < a < p$ and $\gcd(a, p) = 1$

If $a^{p-1} \not\equiv 1 \pmod{p} \Rightarrow p$ is composite.

If it is 1, p may be prime - but there are composites (carmichael numbers) that also pass.

Shetu Saha
IT-21009

Is 561 prime?

Tak $a = 2$, $\gcd(2, 561) = 1$;

$$561 = 3 \cdot 11 \cdot 17$$

By chinese Reminder theorem and FLT:

$$2^{560} \equiv 1 \pmod{3}, 1 \pmod{11}, 1 \pmod{17}$$

So, $2^{560} \equiv 1 \pmod{561}$ - 561 passes FLT but not prime. It's a carmichael number.

Evaluate $5^{123} \bmod 175$

1. Factor the modulus: $175 = 25 \cdot 7$. Work module 25 and 7 and combine by CRT.

2. Mod 25: $5^2 = 25 \Rightarrow 5^n \equiv 0 \pmod{25}$ for all $n \geq 2$. since $123 \geq 2$

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

Shelu Saha
IT-21009

3. Mod 7: $\gcd(5, 7) = 1$ so by Fermat $5^6 \equiv 1 \pmod 7$. Reduce the exponent.

$$123 \equiv 123 - 6 \cdot 20 = 3 \pmod 6,$$

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

4. Combine by CRT: Find x with

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

Check multiples of 25: $25 \cdot 5 = 125 \equiv 6 \pmod 7$

Thus the solution modulo 175 is

$$5^{123} \equiv 125 \pmod{175}.$$

Fermat's Little Theorem can not be applied directly modulo 175 because 5 and 175 are not coprime - that's why we used CRT.

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

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

$x \equiv 2 \pmod 7$

Shetu Saha
IT-21009

Answer:

Statement: The Chinese Remainder Theorem states that a system of linear congruences with pairwise coprime moduli has a unique solution modulo the product of the moduli.

If $m_1, m_2 \ldots m_K$ are pairwise coprime positive integers and $a_1, \ldots a_K$ are any integers, then the system

$x \equiv a_i \pmod{m_i}$. $(i = 1, \ldots K)$ has an unique solution modulo $M = m_1 m_2 \ldots m_K$

Proof (Constructive):

Let $M = \prod_{i=1}^{K} m_i$ and $M_i = M/m_i$. Since $\gcd(M_i, m_i) = 1$, there exists an inverse y_i such that $M_i y_i \equiv 1 \pmod{m_i}$ Then

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

Shetu Saha
IT-21009

Satisfies $x \equiv a_j \pmod{m_j}$ for each j (because for $i \neq j$, M_i is divisible by m_j). Uniqueness modulo M follows because any two solutions differ by a multiple of all m_i, hence by a multiple of M.

Solve the System:

Start with $x \equiv 2 \pmod 3$. put $x = 2 + 3K$

Plug into mod 5:

$$2 + 3K \equiv 3 \pmod 5 \Rightarrow 3K \equiv 1 \pmod 5$$

Inverse of 3 mod 5 is 2 (Since $3 \cdot 2 = 6 \equiv 1$) so,

$$K \equiv 2 \pmod 5$$

$$\Rightarrow K = 2 + 5t$$

Thus

$$x = 2+3k = 2+3(2+5t)$$
$$= 8+15t$$

Now impose mod 7:

$$8+15t \equiv 2 \pmod 7 \Rightarrow 15t \equiv 6 \equiv 1 \pmod 7$$

Since $15 \equiv 1 \pmod 7$, this gives $t \equiv 1 \pmod 7$

So, $t = 1+7s$. Then

$$x = 8+15(1+7s) = 8+15+105 = 23+105s$$

Therefore the solution modulo $3 \cdot 5 \cdot 7 = 105$

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

Shetu Saha
IT-21009

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

Answer: The CIA triad is a core model in information security that stands for confidentiality, Integrity and Availability – three key principles that work together to protect data and systems.

1. Confidentiality – Ensures that information is accessible only to authorized users.
- Protects against unauthorized access or disclosure.
- Achieved using encryption, access controls and authentication.

Sheta Saha
IT-21009

2. Integrity – Ensures that information is accurate, complete, and unaltered unless modified by authorized entities.
- Protects against unauthorized changes or corruption
- Achieved using hashing, digital signatures, and version control.

3. Availability- Ensures that authorized users can access data and systems when needed.

- Protects against downtime, outages, or denial-of-service attacks
- Achieved using redundancy, backups, and fault-tolerant systems.

Sheta Saha
IT-21009

In short:

- Confidentiality protects secrecy
- Integrity protects correctness
- Availability protects accessibility

Together, they form the foundation of a secure system.

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?

Shetu Saha
IT-21009

Answer:

Difference between Steganography and Cryptography:

- Cryptography focuses on protecting the content of a message by transforming it into unreadable ciphertext, so even if intercepted, it cannot be understood without the key.
- Steganography focuses on concealing the existence of the message itself by embedding it within another seemingly harmless medium (image, audio, video, text, etc.) So outsiders don't even suspect a hidden message exists.

Often the two are combined: first encrypt a message, then hide it with steganography for extra security.

Common Techniques for Hiding Data In Digital Media.

Shetu Saha
IT-21009

① Image Steganography:

- LSB (Least Significant Bit) Insertion: Replace the least significant bits of pixels values with message bits.
- Palette Modification: Alter the color palette of indexed images slightly to encode data.
- Transform Domain Methods: Embed data in frequency coefficients (DCT in JPEG).

② Audio Steganography:

- LSB in Audio Samples: Modify least significant bits of audio samples.
- Echo Hiding: Add a short echo to the audio signal to encode information.
- Phase Coding: Modify phase components of the signal.

③ Video Steganography:

- Combine image techniques on video frames.
- Embed in motion vectors or transform co-efficients.

Shetu Saha
IT-21009

④ Text Steganography:

- format-based: Adjust spaces, line breaks, or font styles to carry bits.
- Syntactic / Semantic: Replace words with synonyms in a controlled pattern.

In short:

- cryptography hides meaning
- steganography hides existence.

Both can be used together for stronger information security.

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

Answer:

Shetu Saha
IT-21009

- Phishing:
- Method: Social Engineering attack that impersonates legitimate sources (e.g., fake emails, cloned websites) to trick users into providing credentials, credit card numbers, or other sensitive data.
- Impact: Leads to unauthorized acess to accounts, identity theft, or further targeted attacks.

- Malware
- Method: Malicious code or software (viruses, worms, trojans, ransomware) secretly installed on a device through infected downloads, email attachments or compromised websites.
- Impact: Can corrupt or delete files, steal information, spy on user activity, or lock data until ransom is paid.

Shetu Saha
IT-21009

- Denial-of-Service (DoS) Attack
- Method: Overloads a target server or network with excessive requests, often using botnets, making it unable to process legitimate traffic.
- Impact: Disrupts availability of services, causes downtime, financial loss, and potential reputational damage.

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

Answer :

The General Data Protection Regulation (GDPR) helps mitigate cyber attacks and protect privacy by:

Shetu Saha
IT-21009

1. Enforcing strict data handling rules - Organizations must collect, store and process only necessary personal data, reducing the amount attackers can steal.

2. Requiring strong security measures - GDPR mandates encryption, acess controls and regular security assessments to prevent breaches.

3. Mandatory breach notifications - companies must report data breaches quickly (within 72 hours), allowing faster response to limit damage.

4. Holding organizations accountable - Heavy fines for non-compliance encourage proactive cyber defense and privacy protection.

5. Empowering users - Individuals gain rights to access, correct, or delete their data, reducing risks from outdated or unnecessary stored information.

Shetu Saha
IT-21009

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.

Answer: Data Encryption Standard (DES) with the key steps are given below:

Shetu Saha
IT-21009

1. Input & Key

- Plaintext : 64-bit block.
- Key : 56 bits (plus 8 parity bits in storage → total 64 bits)

2. Initial Permutation (IP)

- Rearrages the 64-bit plaintext according to a fixed table
- This doesn't add security by itself but prepares data for the round operations.

3. 16 Rounds of Feistel Structure

Each round has:

1. Split: Data is split into Left (L) and Right (R) halves (32 bit each).

2. Expansion (E): Right half expanded from 32 bits → 48 bits.

3. Round Key Mixing: XOR the expanded R with a 48-bit subkey (derived from the original key via a key schedule).

Shetu Saha
IT-21009

4. Substitution (S-boxes): Output from XOR is passed through 8 S-boxes, reducing it back to 32-bits with non-linear mapping.

5. Permutation (P): Rearranges bits from the S-box output.

6. Swap: L becomes R, and new R becomes $L \oplus f(R, \text{subkey})$.

4. Final Permutation (FP)

- After 16 rounds, swap L and R one last time, then apply the inverse of the initial permutation table to get the 64-bit ciphertext.

Shetu Saha
IT-21009

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 $L_0 = 0xAAAAAAAA$.

Answer:

In the DES algorithm

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

Shetu Saha
IT-21009

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

$L_1 = 0xF0F0F0F0$

$R_1 = 0x55555555$

Q 21. Use the partial AES S-box to perform sub bytes on [0x23, 0xA7, 0x4C, 0x19]

Shetu Saha
IT-21009

Answer:

~~Looks up~~
Lookups use high nibble = row
low nibble = column

- 0x23 → row 2, col 3 = 0x D4
- 0xA7 → row A, col 7 = 0x 63
- 0x 46 → row 4, col C = 0x2E
- 0x19 → row 1, col 9 = 0x66

Resulting output: [0xD4, 0x63, 0x2E, 0x66]

22. In AES, apply the Add Round Key step only. Given Input word [0x1A, 0x2B, 0x3C, 0x4D] and round key [0x55, 0x66, 0x77, 0x88] Compute the XOR.

Answer: Bytewise XOR:

- 0x1A ⊕ 0x 55 = 0x4F
- 0 x 2B ⊕ 0x 66 = 0x 4D
- 0x 3C ⊕ 0x 77 = 0 x 4B
- 0x 4D ⊕ 0x 88 = 0 x 65

Output word: [0x4F, 0x 4D, 0x 4B, 0x 65]

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

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

Shetu Saha
IT-21009

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

(24) AES-OFB (Output Feedback) Mode

- Works like a stream cipher: an initialization vector (IV) is encrypted with AES to produce a keystream block, which is XORed with plaintext to get ciphertext (and vice versa for decryption)

Shetu Saha
IT-21009

- For the next block, the previous AES output (not the ciphertext) is re-encrypted to generate the next keystream block.

- Synchronization: Both sender and receiver must use the same IV and process the same number of AES iterations so their keystreams match exactly; even a single mismatch breaks decryption.

(25)

AES Modes and Error Propagation

Some AES modes cause error propagation - meaning a single bit error in the ciphertext can affect multiple blocks of the decrypted plaintext. This impacts integrity, because corrupted ciphertext leads to more widespread corruption in the decrypted message.

1. CBC (Cipher Block Chaining) Mode

Sheta Saha
IT-21009

- Decryption Process:

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

(where P_i is plaintext block i, C_i is ciphertext block i)

- Error Effect:
  - If one bit of C_i is corrupted:
    - Block P_i → completely garbled after decryption (due to block cipher decryption).
    - Block P_{i+1} → single-bit error in the same position (due to XOR with corrupted C_i).
- Impact: Corruption spreads to two plaintext blocks.

2. CFB (Cipher Feedback) Mode

Shelu Saha
IT-21009

- Decryption Process:

  $P_i = C_i \oplus E_K(\text{shift register})$

- Error Effect:
  - A bit error in C_i causes:
    - Block P_i → one-bit error in the exact position
    - Several following blocks → temporary corruption until the shift register flushes the error (depends on feedback size).
  - Impact: Corruption affects current and a few subsequent blocks.

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

Answer:

For encryption large files with parallel processing the best choice is AES in CTR (Counter) mode.

Justification

Sheta Saha
IT-21009

1. Parallel Processing

- CTR: Each block is encrypted by XORing plaintext with an encrypted counter value. counters can be generated independently, so all blocks can be encrypted/decrypted in parallel.
- CBC: Encryption is sequential (each block depends on the previous ciphertext), so no parallelism in encryption; only decryption can be partially parallelized.

• ECB : Fully parallelizable but insecure (patterns in plaintext remain visible).

2. Performance

• CTR : No feedback loop → faster for large files and well-suited for hardware acceleration.

• CBC : Slower for encryption of large data due to chaining dependency.

• ECB : Fast, but insecure for structured data.

Shetu Saha
IT-21009

3. Security

• CTR : Hides plaintext patterns, assuming the counter is unique for each block

• CBC : Secure but slower for large parallel tasks.

• ECB : Insecure for most practical uses (reveals patterns)

Recommendation : AES-CTR mode - it's secure

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

Shelu Saha
IT-21009

Answer:

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.)

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

Answer:

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

$= 5^3 \bmod 33$

$= 125 \bmod 33$

$= 26$

Signature = 26

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

Shelu Saha
IT-21009

Answer:

- Aleya public $A = g^a \bmod p$
  $= 3^4 \bmod 17$
  $= 81 \bmod 17$
  $= 13$

- Badal 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, Badal = 5

Shared secret = 13 (Ans:)

(30) Hash $H(x) = (\text{Sum ASCII chars}) \bmod 100$. Compute $H(\text{"A"})$ and $H(\text{"BA"})$. What does this imply about collision resistor?

Answer:

- ASCII('A') = 65, ASCII('B') = 66, Sum = 66+65 = 131
- $H(\text{"A"}) = 131 \bmod 100 = 31$
- $H(\text{"BA"}) = 66+65 = 131 \bmod 100 = 31$

Both produce same hash (collision).

Implication: The function is not collision resistor many different output. A cryptographic hash must make finding collisions computationally infeasible.

Shetu Saha
IT-21009

(31) Mac = (Message + SecretKey) mod 17.
Given message = 15, Key = 7, compute MAC
If attacker changes messages to 10
without key, can they forge mac?

Shetu Saha
IT-21009

Answer:

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.

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

Shatu Saha
IT-21009

Answer:

TSL Handshake:

1. Client Hello - Client sends supported TLS version, cipher suites, and a random number.
2. Server Hello - Server picks the TLS settings, sends its random number and digital certificate (with public key)
3. Key Exchange -
   - In RSA: Client generates a random session key encrypts it with server's public key, and sends it.
   - In Diffie: Hellman/ ECDHE: Both sides exchange parameters to derive the same session key.
4. Session Key Creation: Both client and server use the exchanged info to create

the same symmetric key.

b. Finished Messages – Both send encrypted "finished" messages to confirm keys work.

How symmetric keys are secured: The key exchange uses asymmetric encryption (public/private keys)

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

Answer: SSH Architecture is given below:

Shetu Saha
IT-21009

1. Transport Layer

- Establishes a secure, encrypted channel.
- Handles server authentication, confidentiality and integrity.

2. Authentication Layer

- Varifies the client's identity (password, public key, etc.)
- Ensures only authonized user access the server.

3. Connection Layer:

- Manages multiple logical channels over the secure connection.
- Supports services like remote shell, file transfer (SCP/SFTP) and port forwarding.

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

Shetu Saha
IT-21009

Answer:

TLS Handshake:

1. Client Hello:- Client sends supported TLS versions, cipher options, and a random number.
2. Server Hello - Server picks settings, sends its random number and digital certificate.

3. Key exchange – Client and server use RSA or Diffie-Hellman to securely share a pre-Master secret.

4. Session Key Creation – Both generate the same symmetric keys from the shared secret + random number.

5. Finished Messages – Both send encrypted "finished" messages to confirm the secure channel is ready

Shetu Saha
IT-21009

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

Answer:

General form:

$$y^2 = x^3 + ax + b \quad (\text{over a finite field } \mathbb{F}_p)$$

where $4a^3 + 27b^2 \neq 0$ to avoid singular curve

Why used in cryptography:

→ Provides high security with smaller keys.

→ Based on the Elliptic curve Discrete Logarithm Problem (ECDLP), which is hard to solve.

→ More efficient than RSA for the same security level.

Shetu Saha
IT-21009

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

Shetu Saha
IT-21009

Answer:

ECC achieves the same security as RSA with a smaller key size because solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) is much harder than factoring large integers (RSA's basis)

- Example: A 256-bit ECC key offers roughly the same security as a 3072-bit RSA key.
- Smaller key mean faster computations, less storage, and lower bandwidth use, while keeping the same strength agaist attacks.

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

Answer:

Compute both sides modulo 97:

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

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

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

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

Shetu Saha
IT-21009

Answer:

- $c_1 = g^K \bmod P$
- $c_2 = m \cdot h^K \bmod P$

Compute:

- $c_1 = 5^6 \bmod 23 = 15625 \bmod 23 = 8$
- $h^K = 8^6 \bmod 23 =$ compute $\equiv$ (gives)?
  → $8^2 = 64 \equiv 18$, $8^4 \equiv 18^2 = 324 = 2$

Then $8^2 = 2 \cdot 18 = 36 \equiv 13$. So, $h^6 = 13$

$\cdot c_2 = 10 \cdot 13 \bmod 23 = 130 \bmod 23 = 13$.

$\therefore$ ciphertext : $(c_1, c_2) = (8, 13)$

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

Shetu Saha
IT-21009

Answer:

Importance of lightweight cryptography in IOT:

IoT devices – such as smart sensor, wearable devices, and home automation gadgets - are resource-constrained: they have limited CPU power, memory, and battery life. Traditional cryptographic algorithms like standard

Sheta Saha
IT- 21009

AES or RSA can be too heavy for these devices.

Lightweight cryptography is designed to:

1. Minimize computational load.
2. Reduce memory usage
3. Conserve energy
4. Maintain security

Example:

- Speck or Simon (designed by NSA)
  → Uses simple bitwise operations (XOR, shifts, addition) for encryption.
  → Very fast and lightweight, suitable for microcontrollers and IoT sensors
- AES-128 in CTR mode is also commonly used in IoT when some moderate resources are available, providing a balance of security and efficiency.

(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?

Shetu Saha
IT-21009

Answer:

The three common IoT Attacks are given below:

1. Firmware hijacking

- What is it: Attackers replace or modify device firmware to gain control or implant malware.
- Mitigation: Use signed firmware updates, secure boot, and regular patching.

2. Physical Tampering

- What it is: Direct physical access to the device allows attackers to extract data or bypass security.

Shetu Saha
IT-21009

- Miltgation : Tamper-resistant hardware, secure enclosures and hardware-based encryption.

3. Botnets (e.g. Mirai)

- What it is: Compromised & IoT devices are controlled remotely to perform large-scale attacks like DDoS.
- Mitigation: Change default credentials, network segmentation, and continuous monitoring for ~~unusual~~ unusual activity.

Shetu Saha
IT-21009