Somaya Jannat
ID: IT-21034

Q1: Fermat's Little Theorem :-

Theorem Statements:

Fermat's Little Theorem statements:

If p is a prime number and $a \in \mathbb{Z}$ such that $a \not\equiv 0 \mod p$, then,

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

Proof of Fermat's Little Theorem:

Let a be any integer such that $a \not\equiv 0 \mod p$, and p is prime.

Consider the set $S = \{1, 2, 3, \ldots, p-1\}$
Since $\gcd(a, p) = 1$, multiplying each element in S by $a \mod p$ gives a permutation of S. So:

$$a \cdot 2a \cdot 3a \ldots (p-1)a \equiv 1 \cdot 2 \cdot 3 \ldots (p-1) \mod p$$

$= a^{P-1} (P-1)! \equiv (P-1)! \mod P$ ∴

Now, since $(P-1)!$ is not divisible by P it has an inverse modulo P. ~~Cancelli~~ Concelling it gives:

$$a^{P-1} \equiv 1 \mod P.$$

[Proven]

Apply the Theorem:

Given,

→ $a = 7$, $P = 13$

$a^{P-1} \mod P = 7^{12} \mod 13$

By Fermat's Theorem:

$$7^{12} \equiv 1 \mod 13$$

Ans: $7^{12} \mod 13 = 1$.

Application in Cryptography (RSA):

Fermat's Theorem forms the basis for modular exponentiation used in RSA encryption and decryption.

RSA Algorithm Steps:

1. Choose two large primes P, q.
2. Compute $n = Pq$ and $\phi(n) = (P-1)(q-1)$
3. Choose e such that $\gcd(e, \phi(n)) = 1$
4. Compute d such that $ed \equiv 1 \bmod \phi(n)$
5. Encryption : $c = m^e \bmod n$
6. Decryption : $m = c^d \bmod n$

Using Euler's theorem (generalization of Fermat's):

$$m^{\phi(n)} \equiv 1 \bmod n \Rightarrow m^{ed} \equiv m \bmod$$

Q2: Euler's Totient function and Euler's Theorem:

$\phi(n)$:

$\phi(n)$ for given values:

→ $35 = 5 \times 7$

$$\phi(35) = \phi(5) \cdot \phi(7) = (5-1)(7-1) = 4 \cdot 6 = 24$$

→ $45 = 3^2 \cdot 5$

$$\phi(45) = \phi(3^2) \cdot \phi(5) = (9-3)(5-1) = 6 \cdot 4 = 24$$

→ $100 = 2^2 \cdot 5^2$

$$\phi(100) = \phi(2^2) \cdot \phi(5^2) = (4-2)(25-5) = 2 \cdot 20 = 40$$

Euler's Theorem Statement:

If $\gcd(a,n) = 1$, then:

$$a^{\phi(n)} \equiv 1 \mod n$$

Proof of Euler's Theorem:

Let, $R = \{r_1, r_2, \ldots, r_{\phi(n)}\}$ be the reduced residue system modulo n.

Multiply each element by a (since $\gcd(a,n) = 1$), then:

$$\{ar_1, ar_2, \ldots, ar_{\phi(n)}\} \equiv \text{permutation of } R \mod n.$$

So,

$$a^{\phi(n)} \cdot r_1 r_2 \cdots r_{\phi(n)} \equiv r_1 r_2 \ldots r_{\phi(n)} \mod n$$

Cancelling the product on both sides:

$$a^{\phi(n)} \equiv 1 \mod n.$$

Q3: :

Given System of congruences:

$$x \equiv 2 \mod 3$$
$$x \equiv 3 \mod 4$$
$$x \equiv 1 \mod 5$$

Let, $N = 3 \cdot 4 \cdot 5 = 60$

Step1: Define Components:

→ $N_1 = 60/3 = 20$

→ $N_2 = 60/4 = 15$

→ $N_3 = 60/5 = 12$

Step 2: find Inverses:

We need ~~find~~ y_i such that $N_i \cdot y_i \equiv 1 \mod n_i$

→ $20y_1 \equiv 1 \mod 3 \Rightarrow y_1 = 2$

→ $15y_2 \equiv 1 \mod 4 \Rightarrow y_2 = 3$

→ $12y_3 \equiv 1 \mod 5 \Rightarrow y_3 = 3$

Step 3: Apply CRT formula:

$$x = a_1 N_i y_1 + a_2 N_2 y_2 + a_3 N_3 y_3 \mod N$$

$$x = 2 \cdot 20 \cdot 20 + 3 \cdot 15 \cdot 3 + 1 \cdot 12 \cdot 3 = 80 + 135 + 36 = 251$$

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

Q4:

To determine if 561 is a carmichael number we follow these steps:

1. Check if 561 is composite:

$561 = 3 \times 11 \times 17 \Rightarrow$ Yes, ~~it's~~ it's composite

2. Apply Fermat's Little Theorem Test:

For multiple values a where gcd(a,561) = 1, we test whether:

$$a^{560} \equiv 1 \mod 561$$

For, a = 2, 3, 5 we find that fermat's test passes:

$$a^{560} \mod 561 = 1$$

3. Use Korselt's Criterion (a test for carmichael numbers):

→ 561 is square-free.

→ Prime divisors of 561 are: 3, 11, 17

→ 3 - 1 = 2; 2 | 560

→ 11 - 1 = 10; 10 | 560

→ 17 - 1 = 16; 16 | 560

All conditions are satisfied.
So, 561 is a carmichael number.

Q4:

Primitive root modulo 17

We are to find a primitive root g such that its powers generate all elements of the multiplicative group modulo 17.

→ 17 is a prime number, so the multiplicative group modulo 17 has $\phi(17) = 16$ elements.

→ A primitive root g satisfies:

$$g^k \not\equiv 1 \bmod 17 \text{ for all } k < 16$$

We test $g = 3$

→ $3^2 = 9 \bmod 17 \neq 1$

→ $3^4 = 81 \bmod 17 = 13 \neq 1$

→ $3^8 = 6561 \mod 17 = 16 \neq 1$

→ $3^{16} \mod 17 = 1$

Since no lower powers return 1, $\text{ord}_{17}^{(3)} = 16$

So, 3 is a primitive root modulo 17.

Q6:

Find the smallest positive integer x such that:

$$3^x \equiv 13 \mod 17$$

→ We compute successive power of 3 modulo 17:

$3^1 = 3 \mod 17 = 3$

$3^2 = 9 \mod 17 = 9$

$3^3 = 27 \mod 17 = 10$

$3^1 = 81 \mod 17 = 13$

thus $x = 1$

Q7:

The Diffie - Hellman key Exchange is a cornerstone of modern cryptography. It allows two parties of generate a shared secret key over an insecure channel.

Steps:

1. Public parameters: a large prime p, and a primitive root g.

2. Alice selects a secret a, computes $A = g^a \mod P$.

3. Bob selects a secret b, computes $B = g^b \mod p$.

4. Shared secret:

→ Alice computes $s = B^a \bmod P$

→ Bob computes $s = A^b \bmod P$

Even if an eavesdropper knows g, P, A, B computing the shared key requires solving the Discrete Logarithm Problem (DLP): given g and $A = g^a \bmod P$, find a.

This is computationally hard for large primes, ensuring security.

Q8: Comparison of Substitution, Transposition and Playfair Ciphers.

1. Encription Mechanism:

Substitution Cipher:

Each letter in the plaintext is replaced by another letter based on a fixed key.

Example: In Caesar Cipher, A becomes D, B becomes E, and so on.

Transposition Cipher:

Letters of the plaintext are not changed but rearranged in a specific pattern or order using a key.

Playfair Cipher:

Encrypts two letters at a time using a 5×5 matrix ~~but~~ built from a key word. It replaces each pair with another pair using specific rules.

2. Key Space:

Substitution Cipher:

Very large key space – 26! possible ways to rearrange the alphabet.

Transposition Cipher:

Key space depends on block size. For a block of n letters, there are n! permutations.

Playfair Cipher:

Uses a 5×5 matrix, so, ~~but~~ total arrangements are 25!

secured but less than full digraph substitution.

3. Vulnerability to Frequency Analysis:

- Substitution Cipher:

Weak. Frequency of letters remains visible in ciphertext.

- Transposition Cipher:

Medium vulnerability. Letter frequency is not hidden, but the order is changed.

- Playfair Cipher:

More secure. Encripting digraphs helps break frequency patterns and adds complexity.

Example: Encrypt the word "HELLO"

1. Substitution Cipher (Caesar + 3):

H → K, E → H, L → O, L → O, O → R

→ Ciphertext: KHOOR

2. Transposition Cipher (3-columns):

HEL
LO

Read Column-wise: HLEOL

→ Ciphertext: HLEOL

3. Playfair Cipher (key = ~~Mon~~ MONARCHY):

Prepare matrix (I = J):

Plaintext: HELLO → Digraphs: HELLOX (x added)

→ HE → CF

→ LL → becomes LX → SU

→ OX → VZ

→ Ciphertext : CFSUVZ

- Substitution Cipher is easy but weakest.
- Transposition Cipher improves security by suffling letters.
- Playfair Cipher is stronger, works on letters pairs and resists basic all attack

Q9: Affine Cipher Encryption and Decryption

→ We are given the encryption function:

$$E(x) = (a \cdot x + b) \bmod 26$$

where,

→ $a = 5$

→ $b = 8$

→ x is the numeric equivalent of the plaintext character (A = 0, B = 1, ... Z = 25

We need to:

1. Encrypt the plaintext: "Dept of ICT, MBSTU"

2. Derive the decryption function and decrypt the ciphertext.

a) Encryption:

Step1:

Ignore punctuation and convert all letters to uppercase:

"Dept of ICT, MBSTU" → DEPTOFICTMBSTU

step 2: Convert letters to numbers (A=0 to Z=25

Letter	D	E	P	T	O	F	I	C	M	B	S	U
Value	3	4	15	19	14	5	8	2	12	1	18	20

Step 3: Apply the encryption function:

$$E(x) = (5x+8) \bmod 26$$

Let's compute:

→ $E(3) = (5\times3+8) \bmod 26 = 23 \rightarrow X$

→ $E(4) = (5\times4+8) \bmod 26 = 2 \rightarrow C$

→ $E(15) = (5\times15+8) \bmod 26 = 83 \bmod 26 = 5 \rightarrow F$

→ $E(19) = (5\times19+8) \bmod 26 = 103 \bmod 26 = 25 \rightarrow Z$

→ $E(14) = (5\times14+8) \bmod 26 = 78 \bmod 26 = 0 \rightarrow A$

→ $E(5) = (5\times5+8) \bmod 26 = 33 \bmod 26 = 7 \rightarrow H$

→ $E(8) = (5\times8+8) \bmod 26 = 48 \bmod 26 = 22 \rightarrow W$

→ $E(2) = (5\times2+8) \bmod 26 = 18 \rightarrow S$

→ $E(12) = (5\times12+8) \bmod 26 = 68 \bmod 26 = 16 \rightarrow Q$

→ $E(1) = (5\times1+8) \bmod 26 = 13 \rightarrow N$

→ $E(18) = (5\times18+8) \bmod 26 = 98 \bmod 26 = 20 \rightarrow U$

$E(20) = (5\times20+8) \bmod 26 = 108 \bmod 26 = 4 \rightarrow E$

Final ciphertext letters (in order):

X C F Z A H W S Z Q N U Z E

So, Encrypted Ciphertext:

XCFZAHWSZQNUZE

b) Decryption:

The decryption function:

$$D(y) = a^{-1} \cdot (y - b) \bmod 26$$

Step 1: Find ~~the~~ $a^{-1} \bmod 26$

modulen inverse of $a = 5 \bmod 26$

$$5 \cdot a^{-1} \equiv 1 \bmod 26$$

→ $5 \times 1 = 5$

→ $5 \times 21 = 105 \bmod 26 = 1$

so,

$$a^{-1} = 21$$

Step 2: Apply decryption formula:

$$D(y) = 21 \cdot (y - 8) \bmod 26$$

Use ciphertext: XCFZAHWSZQ

Convert to numbers:

Letter	X	C	F	Z	A	H	W	S	Z	Q	N	U	Z	E
value	23	2	5	25	0	7	22	18	25	16	13	20	25	4

Now, decrypt each:

→ $D(23) = 21 \times (23 - 8) \bmod 26 = 21 \times 15 = 315 \bmod 26 = 3 \rightarrow D$

→ $D(2) = 21 \times (2 - 8) \bmod 26 = -126 \bmod 26 = 4 \rightarrow E$

→ $D(5) = 21 \times (-3) \bmod 26 = -63 \bmod 26 = 15 \rightarrow P$

→ $D(25) = 21 \times (17) = 357 \bmod 26 = 19 \rightarrow T$

→ $D(0) = 21 \times (-8) \bmod 26 = -168 \bmod 26 = 14 \rightarrow O$

→ $D(7) = 21 \times (7 - 8) \bmod 26 = -21 \bmod 26 = 5 \rightarrow F$

→ D(22) = 21 × (22-8) mod 26 = 294 mod 26 = 8 → I

→ D(18) = 21 × (18-8) mod 26 = 210 mod 26 = 2 → C

→ D(16) = 21 × (16-8) mod 26 = 168 mod 26 = 12 → M

→ D(13) = 21 × (13-8) mod 26 = 105 mod 26 = 1 → B

→ D(20) = 21 × (20-8) mod 26 = 252 mod 26 = 18 → S

→ D(4) = 21 × (4-8) mod 26 = -84 mod 26 = 20 → U

~~Deprij~~

Decrypted Text: DEPTOFICT

Q 10:

Here's a simple ~~m~~ novel cipher that uses a combination of substitution and permutation techniques. It also uses a custom pseudo-random number generator (PRNG) for added complexity

Cipher name: Sub-Perm Cipher (~~SPC~~ SPC)

Overview:

→ Substitution: Each character is substituted using a keyed caesar shift.

→ Permutation: Blocks of text are permuted using a PRNG-based shuffle.

→ PRNG: Custom linear congruential Generator (LCG).

Key:

K_1: Integer (Used for Caesar shift)

K_2: Seed value for PRNG.

Block size: fixed block size.

Encryption Process:

Step 1: Substitution:

Each character C in plaintext is shifted forward using a Caesar-like method with a ~~vari~~ varying shift based on the PRNG.

PRNG: $X_{n+1} = (aX_n + c) \bmod m$

Parameters: $a = 17, c = 43, m = 256$

For each character Ci, Compute:

Shift i = PRNG (K_2) mod 26

Ci = (Ci + Shift i + Ki) mod 26

Step 2: Permutation:

Split the substituted ciphertext into blocks of size N.

For each block:

1. Use PRNG (same seed K_2) to generate a permutation of indices.

2. Permute the block accordingly.

Decryption Process:

Step 1: Reverse Permutation:

using the same PRNG and Block siz
rearranginge the permutation pattern
and reverse the it for each block.

Step 2: Reverse Substitution:

For each character C_i in the block
n

$Shift_i = PRNG(K_2) \bmod 26$

$C_i = (C_i' - Shift_i - K_1 + 26) \bmod 26$

Example:

Inputs:

→ Plaintext: 'HELLO'

→ $K_1 = 3$, $K_2 = 7$, Block Size = 2

Step 1: Substitution:

Let's say PRNG gives shift = [5, 12, 7, 19, 2]

H → H(7) + 5 + 3 = 15 → P

E → E(4) + 12 + 3 = 19 → T

L → L(11) + 7 + 3 = 21 → V

L → L(11) + 19 + 3 = 33 → H (mod 26)

O → O(14) + 2 + 3 = 19 → T

Substituted: "PT VHT"

Step 2: Permutation (Block size 2):

Split : [PT][VH][T-]

Permutation generated : [1,0]

Apply permutation to each block:

→ [PT] → [TP]
→ [VH] → [HV]
→ [T-] → [-T] (Padding with -)

Final Ciphertext : "TPHV_T".