ASSIGNMENT NO.1

TITLE: Chinese Remainder Theorem

PROBLEM STATEMENT: Implement a number theory such as Chinese remainder Theorem

OBJECTIVE:

To study & implement Chinese Reminder Theorem

THEORY:

Chinese Reminder Theorem is used to solve set of congruent equations with one variable but different modulus, which are relatively prime

$$x \equiv a1 \bmod m1$$
$$x \equiv a2 \bmod m2$$
$$x \equiv a3 \bmod m3$$
$$\ldots\ldots$$
$$x \equiv ak \bmod mk$$

The Chinese Reminder Theorem states that the above equations have unique solution if the moduli are relatively prime. Below are the steps needed to follow to solve set of congruent equations using Chinese Reminder Theorem

Step I: Find M = m1 x m2 x m3…mk where M is common modulus
Step II: Find M1 = M/m1, M2 = M/m2 and so on
Step III: Find multiplicative inverses for M1, M2 and so on
Step IV: Put the values in the below equation to solve for X

$$X = (a1 \times M1 \times M1^{-1} + a2 \times M2 \times M2^{-1} + a3 \times M3 \times M3^{-1}) \bmod M$$

Example

$$X = 4 \bmod 5$$
$$X = 6 \bmod 8$$
$$X = 8 \bmod 9$$

Step I: M = 5 * 8* 9 = 360

Step II: M1 = M/m1 = 360 / 5 = 72

M2 = M/m2 = 360 / 8 = 45
M3 = M/m3 = 360 / 9 = 40

Step III:
To find M1 inverse, Solve for GCD (m1, M1) using Extended Euclidean Algorithm. GCD (5, 72)

q	*r1*	*r2*	*r*	*t1*	*t2*	*t*
0	5	72	5	0	1	0
14	72	5	2	1	0	1
2	5	2	1	0	1	-2

The inverse value cannot be negative, so add modulus into it to make it positive.

M1 inverse = -2 + 5 = 3

To find M2 inverse, Solve for GCD (m2, M2) using Extended Euclidean Algorithm.
GCD (8, 45)

q	*r1*	*r2*	*r*	*t1*	*t2*	*t*
0	8	45	8	0	1	0
5	45	8	5	1	0	1
1	8	5	3	0	1	-1
1	5	3	2	1	-1	2
1	3	2	1	-1	2	-3

M2 inverse = -3 + 8 = 5

To find M3 inverse, Solve for GCD (m3, M3) using Extended Euclidean Algorithm.
GCD (9, 40)

q	*r1*	*r2*	*r*	*t1*	*t2*	*t*
0	9	40	9	0	1	0
4	40	9	4	1	0	1
2	9	4	1	0	1	-2

M3 inverse = -2 + 9 = 7

Step IV: Put the values in the below equation to solve for X

$X = (a1 \times M1 \times M1^{-1} + a2 \times M2 \times M2^{-1} + a3 \times M3 \times M3^{-1}) \bmod M$

X = (4*72*3 + 6*45*5 + 8*40*7) mod 360

X = (864 + 1350 + 2240) mod 360

X = 4454 mod 360

X = 134

Applications

The Chinese Reminder Theorem has several applications in cryptography. One is to solve the quadratic congruence and the other is to represent very large number in terms of list of small integers.

CONCLUSION:

We have studied & implemented Chinese Reminder Theorem.

QUESTIONS:

Batch A:

1. Use the Chinese remainder theorem to find a solution to each of the following linear systems:

   $x \equiv 0 \bmod 2$

   $x \equiv 0 \bmod 3$

   $x \equiv 1 \bmod 5$

   $x \equiv 6 \bmod 7$
2. In a Chinese Reminder theorem, Let N=210 & let n1=5, n2=6, n3=7. Compute f-1(3,5,2). i.e x1=3, x2=5, x3=2. Compute x.

Batch B:

1. Jessica breeds rabbits. She is not sure exactly how many she has today, but as she was moving them about this morning she noticed some things. When she fed them in group of 5, she had 4 left over. When she baths them in group of 8, she had group of 6 left over. She took them outside to wonder in groups of 9, but then last group consisted of only 8. Find out how many rabbits she has?
2. Find the integer that has a reminder of 3 when divided by 7 & 13, but is divisible by 12.

Batch C:

1. Determine the value of x using Chinese remainder theorem

   $X = 1 \pmod 5$

   $X = 6 \pmod 7$

   $X = 8 \pmod{11}$
2. Determine the value of x using Chinese remainder theorem

   $X = 1 \pmod{13}$

   $X = 11 \pmod{23}$

ASSIGNMENT NO.2

TITLE: Extended Euclidian Algorithm

PROBLEM STATEMENT: Implement Euclidean and Extended Euclidean algorithm to find out GCD and solve the inverse mod problem.

OBJECTIVES:

To study Euclidian & Extended Euclidian algorithm

THEORY:

The extended Euclidean algorithm is an extension to the Euclidean algorithm. Besides finding the greatest common divisor of integers *a* and *b*, as the Euclidean algorithm does, it also finds integers *x* and *y* (one of which is typically negative).

$$ax + by = \gcd(a, b) \text{ or } sa + tb = \gcd(a, b)$$

The extended Euclidean algorithm is particularly useful when *a* and *b* are coprime, since *x* is the multiplicative inverse of *a* modulo *b*, and *y* is the multiplicative inverse of *b* modulo *a*.

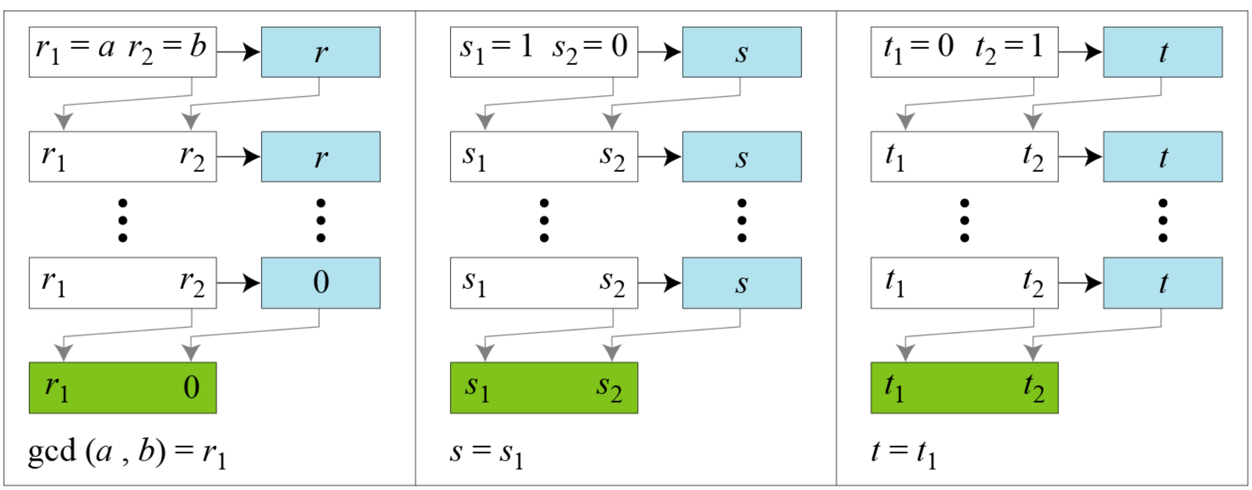

Figure: Extended Euclid's Algorithm Process

Algorithm:

Extend the algorithm to compute the integer coefficients x and y such that
gcd (a, b) = ax + by

Extended-Euclid (a, b)

$r_1 \leftarrow a; \quad r_2 \leftarrow b;$
$s_1 \leftarrow 1; \quad s_2 \leftarrow 0;$ (Initialization)
$t_1 \leftarrow 0; \quad t_2 \leftarrow 1;$

while ($r_2 > 0$)

{

$q \leftarrow r_1 / r_2;$

$r \leftarrow r_1 - q \times r_2;$
$r_1 \leftarrow r_2; \; r_2 \leftarrow r;$ (Updating r's)

$s \leftarrow s_1 - q \times s_2;$
$s_1 \leftarrow s_2; \; s_2 \leftarrow s;$ (Updating s's)

$t \leftarrow t_1 - q \times t_2;$
$t_1 \leftarrow t_2; \; t_2 \leftarrow$ t; (Updating t's)

}

gcd (a , b) $\leftarrow r_1; \; s \leftarrow s_1; \; t \leftarrow t_1$

Example:

GCD (161, 28)

q	*r1*	*r2*	*r*	*s1*	*s2*	*s*	*t1*	*t2*	*t*
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-23
	7	0		-1	4		6	-23	

Here GCD value we are getting as 7. Value of s is -1 and value of t is 6. If we put these values into equation,

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

$$161 * (-1) + 28 * (6) = 7$$

This satisfies the equation for Extended Euclidean Algorithm

CONCLUSION:

We have studied and implemented the Extended Euclidian algorithm.

QUESTIONS:

Batch A:

1. Determine integers x and y such that gcd(421, 111) = 421x + 111y.
2. Find x and y such that 51x + 100y = 1.
3. How to compute multiplicative inverse using Extended Euclid's algorithm. Illustrate.

Batch B:

1. Using Euclidian algorithm calculate GCD(48,30) & GCD(105,80).
2. What is the significance of Extended Euclidian algorithm with reference to RSA Algorithm?
3. Find the multiplicative inverse of 8 mod 11, using the Euclidean Algorithm.

Batch C:

1. Using Euclidean algorithm calculate gcd (16,20) and gcd (50,60).
2. Find the multiplicative inverse of 43 mod 64, using the Euclidean Algorithm?
3. Find x and y such that 97x + 20y = 1.

ASSIGNMENT NO.3

TITLE: RSA Algorithm

PROBLEM STATEMENT: Implement RSA public key cryptosystem for key generation and cipher verification.

OBJECTIVES:

To understand,

1. Public key algorithm.
2. RSA algorithm
3. Concept of Public key and Private Key

THEORY:

Public Key Algorithm:

Asymmetric algorithms rely on one key for encryption and a different but related key for decryption. These algorithms have the following important characteristics:

- It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key.

In addition, some algorithms, such as RSA, also exhibit the following characteristics:

- Either of the two related keys can be used for encryption, with the other used for decryption.

A public key encryption scheme has six ingredients:

- **Plaintext**: This is readable message or data that is fed into the algorithm as input.
- **Encryption algorithm**: The encryption algorithm performs various transformations on the plaintext.
- **Public and private key**: This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input.
- **Cipher text**: This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different cipher texts.
- **Decryption algorithm**: This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

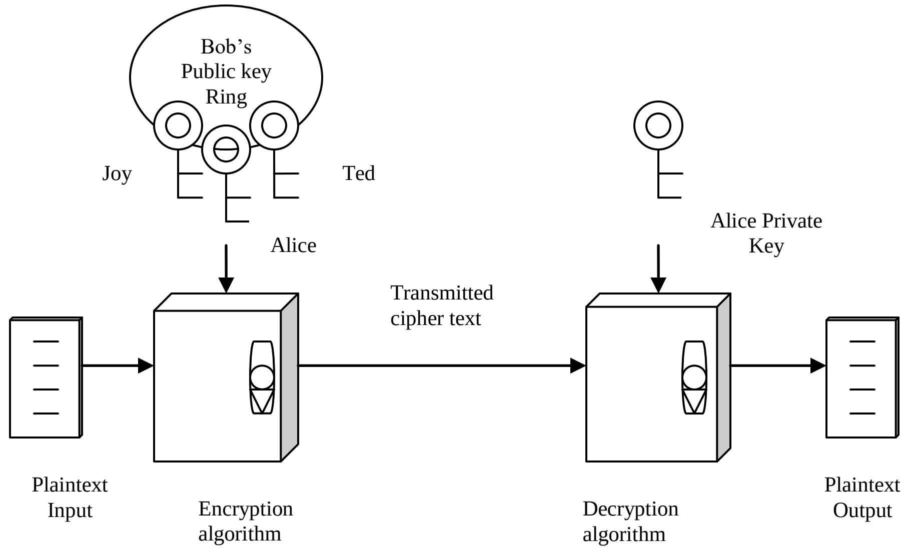

Figure: Public key cryptography

The essential steps are as the following:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or the other accessible file. This is the public key. The companion key is kept private. As figure suggests, each user maintains a collection of public keys obtained from others.
3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.

When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

The RSA Algorithm:

The scheme developed by Rivest, Shamir and Adleman makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary value less than some number n. That is the block size must be less than or equal to log2 (n); in practice the block size is I bits, where 2i<n<=2i+1. Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C:

$C = M^e \bmod n$

$M = C^d \bmod n$

Both sender and receiver must know the value of n. The sender knows the value of e, and only the receiver knows the value of d. Thus, this is a public-key encryption algorithm with a public key of PU = {e, n} and a private key of PR = {d, n}. For this algorithm to be satisfactory for public key encryption, the following requirements must meet:

1. It is possible to find values of e, d, n.
2. It is relatively easy to calculate $M^e \bmod n$ and $C^d \bmod n$ for all values of $M<n$.
3. It is feasible to determine d given e and n.

Key Generation	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p * q$	
Calculate $Ø(n) = (p-1)(q-1)$	
Select integer e	$\gcd(Ø(n),e) = 1;\ 1<e< Ø(n)$
Calculate d	$d = e^{(-1)} \bmod Ø(n)$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption	
Plaintext	$M<n$
Ciphertext	$C=M^e \bmod n$

Decryption	
Ciphertext	C
Plaintext	$M = C^d \bmod n$

Figure: The RSA Algorithm

Example 1:

1. Select two prime numbers, p = 17 and q = 11.
2. Calculate n = pq = 17*11 = 187.
3. Calculate $Ø(n) = (p-1)(q-1) = 16*10 = 160$.
4. Select e such that relatively prime to $Ø(n)=160$ & less than $Ø(n)$; we choose e = 7.

5. Determine d such that de ≡ 1 (mod 160) and d < 160. The correct value is d = 23; d can be calculated using the extended Euclid's algorithm.

The resulting keys are public key PU = {7, 187} and private key PR = {23, 187}. The example shows the use of these keys for plaintext input of M=88.

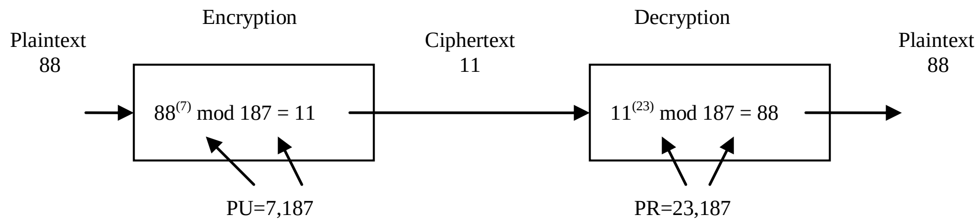

Figure: Example of RSA.

Example 2:

P = 7, Q = 13, M = 10.

Step I: n = 3 * 11 = 33
Step II: Ø(n) = 2 * 10 = 20
Step III: Select e, such that gcd (Ø(n),e) = 1, gcd(20,3) = 1, So we can select e =3
Step IV: To calculate d, solve for gcd (Ø(n),e) using extended Euclid's algorithm and pick up the value of t.

q	*r1*	*r2*	*r*	*t1*	*t2*	*t*
6	20	3	2	0	1	-6
1	3	2	1	1	-6	7

d = 7

Step V: For Encryption,

$C = M^e \bmod n$
= 2^3 mod 33
= 8 mod 33
= 8

Step VI: For Decryption,

$M = C^d \bmod n$
= 8^7 mod 33
= ((8^2 mod 33) (8^2 mod 33) (8^2 mod 33) (8^1 mod 33)) mod 33
= (31* 31 *31 *8) mod 33
= (961 mod 33) (248 mod 33) mod 33
= 68 mod 33
= 2

Advantages:

1. Easy to implement.

Disadvantages:

1. Anyone can announce the public key.

Algorithm:

1. Start
2. Input two prime numbers p and q.
3. Calculate n = pq.
4. Calculate Ø(n) = (p-1)(q-1).
5. Input value of e.
6. Determine d.
7. Determine PU and PR.
8. Take input plaintext.
9. Encrypt the plaintext and show the output.
10. Stop.

CONCLUSION:

We have studied and implemented the public key algorithm that is RSA algorithm.

QUESTIONS:

Batch A:

1. What are the principle elements of a public key cryptosystem?
2. Differentiate public key and conventional encryption?
3. Perform encryption and decryption using RSA Alg. for the following. P=7; q=11; e=17; M=8.

Batch B:

1. Specify the applications of the public key cryptosystem?
2. Define Euler's totient function or phi function and their applications?
3. Perform encryption and decryption using RSA Alg. for the following. P=3; q=11; e=17; M=12.

Batch C:

1. Give the significance of Extended Euclid's Algorithm with respect to RSA.
2. How to calculate private key d in RSA algorithm?
3. Calculate cipher text using RSA algorithm. Given data is as follows: Prime numbers P, Q as 13, 17 & the plain text to be sent is 12. Assume public key as 19.

ASSIGNMENT NO.4

TITLE: Diffie Hellman Key Exchange

PROBLEM STATEMENT: Implement Diffie Hellman key exchange algorithm for secret key generation and distribution of public key

OBJECTIVE:

1. To learn the basics of key management.
2. To study & implement Diffie Hellman key exchange algorithm.

THEORY:

The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of secret values.

Algorithm:

There are two publicly known numbers: a prime number q and an integer α that is a primitive root of q.

Suppose the users A and B wish to exchange a key.

User A selects a random integer $X_A < q$ and computes $Y_A = \alpha^{X_A} \bmod q$. Similarly User B independently selects a random integer $X_B < q$ and computes $Y_B = \alpha^{X_B} \bmod q$.

Each side keeps the value X private and makes the value Y available publicly to the other side.

User A computes the key K as $K = Y_B^{xA} \bmod q$

User B computes the key K as $K = Y_A^{xB} \bmod q$

These two calculations produce identical results.

Eg.

1. Users Alice & Bob who wish to swap keys:
2. Agree on prime q=353 and α=3
3. Select random secret keys:
   - A chooses X_A=97, B chooses X_B=233
4. Compute public keys:
   - $Y_A = 3^{97} \bmod 353 = 40$ (Alice)
   - $Y_B = 3^{233} \bmod 353 = 248$ (Bob)

5. Compute shared session key as:

   $K_{AB} = Y_B^{xA} \bmod 353 = 248^{97} = 160$ (Alice)

   $K_{AB} = Y_A^{xB} \bmod 353 = 40^{233} = 160$ (Bob)

Global Public Elements	
q	prime number
α	$\alpha < q$ and α a primitive root of q

User A Key Generation	
Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{XA} \bmod q$

User B Key Generation	
Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{XB} \bmod q$

Calculation of Secret Key by User A
$K = (Y_B)^{XA} \bmod q$

Calculation of Secret Key by User B
$K = (Y_A)^{XB} \bmod q$

Figure: Diffie Hellman Process

CONCLUSION:

We have studied & implemented Diffie Hellman key exchange algorithm.

QUESTIONS:

Batch A:

1. What is primitive root of a number? Explain with suitable example.
2. Explain Key management with respect to key generation.

3. Let p = 37 and g = 13. Let Alice pick a = 10. Let Bob pick b = 7. Find of the secret key K.

Batch B:

1. What is Man in the middle attack? Explain with respect to Diffie-Hellman Algorithm.
2. Given the values of p=11 and g=2. Alice chooses a secret integer whose value is 9 and Bob chooses a secret integer whose value is 4. Find of the secret key K
3. Explain Key management with respect to key distribution.

Batch C:

1. Do we share a key in Diffie-Hellman algorithm or do we create it?
2. Show that 2 in primitive root of 11.
3. Explain Key management with respect to key storage.