UNIT-3

Part-I: Number Theory

(1) Prime Numbers:

Definition: Any positive integer greater than 1 is a prime number, if and only if it is divisible by only two integers, 1 and itself.

Ex: 2, 3, 5, 7, 11, 13, 17 are prime numbers.

4, 6, 8, 10 are non-prime numbers.

→ The integers other than the prime numbers are "composite numbers".

→ Prime numbers cannot be written as a product of other numbers.

→ **Prime Factorization:** To factor a number n is to write it as a product of other numbers $n = a \times b \times c$.

$$6 = 1 \times 2 \times 3.$$

→ **Importance of Prime Numbers:**

* Primes are important because the security of many encryption algorithms are based on the fact that it is very fast to multiply two large prime numbers.
* It is extremely computer intensive to do the reverse.
* It is hard to prime factorize a number.

→ The set of prime numbers is infinite.

→ The no. of primes $\pi(n)$, less than a given large integer n has the following limits

$$\frac{n}{\ln n} < \pi(n) < \frac{n}{\ln n - 1.08366}$$

(2) Relatively Prime Numbers:-

Definition: Two integers are relatively prime if they share no common positive factors except 1.

(or)

Two integers a and b are said to be relatively prime if $\gcd(a,b) = 1$

→ Relatively prime numbers are also called "Mutually Prime" or "Coprime".

Ex: 14, 15

Factors for 14 : 1, 2, 7, 14

Factors for 15 : 1, 3, 5, 15

Only 1 is the common factor for 14, 15

∴ 14, 15 are co-prime numbers.

→ Relative primality is not transitive

Ex: $\gcd(2,3) = 1$, $\gcd(3,4) = 1$

but $\gcd(2,4) = 2$.

(3) Modular Arithmetic:-

→ Modular arithmetic is a system of arithmetic for integers, where numbers "wrap around" upon reaching a certain value called modulus

→ Modulus Operator: Given any +ve or -ve integer 'a' from set $Z = \{\dots -2, -1, 0, 1, 2 \dots\}$ and a positive integer n, "a mod n" to be remainder "r" of division of a by n.

It is represented as $\boxed{a \bmod n = r}$

Ex: $a = 11, n = 7$

$11 \bmod 7 = 4$

→ The remainder r is called "residue" and must always be a +ve number. In case of negative integers, use a negative quotient when dividing "a by n" to get a +ve remainder.

Ex: $a = -35 \quad n = 13$

$-35 \bmod 13 = 4.$

→ Set of Residues:

Since the residue r can have integer values from 0 to n-1, the mod-n operator maps a positive or negative integer from set of integers Z to an element in the set $Z_n = \{0, 1, 2 \dots, n-1\}$

Z_n is called "set of Residues".

Ex: $n = 7$.

Z	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Z_n	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4

Congruence:- Two integers a, b from set $Z = \{\ldots -2, -1, 0, 1, 2 \ldots\}$ are said to be congruent (modulo n) if

$$a \bmod n = b \bmod n$$

(or)

If a, b maps to the same element of $Z_n = \{0, 1, 2 \ldots n-1\}$

→ The congruence is expressed using the symbol "≡"

$$a \equiv b \pmod n$$

Ex: $17 \bmod 13 = 4$
$56 \bmod 13 = 4$
$-9 \bmod 13 = 4.$
$\therefore 17 = -9 = 56 \pmod{13}.$

→ **Modular arithmetic Operations:**

$(14+11) \bmod 17 = 25 \bmod 17 = 8$
$(13-8) \bmod 17 = 5 \bmod 17 = 5.$
$(14 \times 2) \bmod 17 = 28 \bmod 17 = 11$

properties of Modular Arithmetic:

Commutative Law: $(a+b) \bmod n = (b+a) \bmod n$
$(a \times b) \bmod n = (b \times a) \bmod n$

Associative Law: $[(a+b)+c] \bmod n = [a+(b+c)] \bmod n.$
$[(a \times b) \times c] \bmod n = [a \times (b \times c)] \bmod n$

Distributive Law: $[(a \times (b+c))] \bmod n = [(a \times b) + (a \times c)] \bmod n.$

(4) Fermat's Theorem:-

→ Fermat's Theorem also called "Fermat's Litted Theorem".

Theorem: If 'p' is a prime number and 'a' is any positive integer, then $\boxed{a^p \equiv a \pmod p}$ [$a^p - a$ will always divisible by p

(or)

$\boxed{a^{p-1} \equiv 1 \pmod p}$ ['a' must be coprime to 'p' (or a is not divisible by p]

proof:

Consider a set of positive integers less than 'p': $\{1, 2, 3, \ldots (p-1)\}$ and multiply each element by 'a' and 'mod p' to get

$X = \{a \bmod p, 2a \bmod p, 3a \bmod p, \ldots, (p-1)a \bmod p\}$.

No element of X is zero or equal, since p doesn't divide a.

Multiply the numbers in both sets (P and X) and taking the result mod p yeilds.

$$a * 2a * 3a * \ldots * (p-1)a \equiv [1 * 2 * 3 * \ldots * (p-1)] \bmod p$$

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

On equating $(p-1)!$ from both sides,

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

(or)

$$\boxed{a^p \equiv a \pmod p}$$

→ Fermat's Theorem is used in public key (RSA) and Primality Testing.

Ex: $a = 5 \quad b = 3$.

i) $5^3 = 125 \equiv 2 \pmod 3$

$5 \equiv 2 \pmod 3$

ii) $5^{3-1} \equiv 25 \bmod 13 \equiv 1 \pmod 3$

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

→ Example application of Fermat's little theorem,

Find $4^{15} \bmod 13$

$= 4^2 \cdot 4^{13} \bmod 13$

$= 4^2 \times 4 \bmod 13$

$= 12$.

(5) Euler's Totient Function:

→ Euler's Totient Function is denoted as "$\phi(n)$".

$\phi(n)$: For a given positive integer n, $\phi(n)$ is the number of positive integers less than n that are coprime to n.

1. $\phi(n) = n-1$, if n is prime

2. If p and q are prime numbers with $p \neq q$ and $n = pq$, then

$$\phi(n) = \phi(pq) = \phi(p) \cdot \phi(q) = (p-1)(q-1).$$

Ex: $\phi(6) = \phi(2) \cdot \phi(3)$

$= (2-1) \cdot (3-1) = 1 \cdot 2 = 2$

[1 and 5 are the two integers less than 6 and coprime to 6]

$\phi(5) = 5-1 = 4$.

[1, 2, 3, 4 are four integers less than 5 and coprime to 5].

3. If p is a prime number, $\phi(p^k) = p^k - p^{k-1}$ Ex: $\phi(9) = \phi(3^2)$

$= 3^2 - 3 = 9 - 3 = 6$

(6) Euler's Theorem:-

Theorem: For every positive integer n and every 'a' that is coprime to n, the following must be true:

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

$\phi(n)$ is totient of n

Proof: $\phi(n) = n-1$ if n is prime and Fermat Theorem holds.

* $\phi(n)$ is the no of positive integers less than n that are relatively prime to n
* Consider set of such integers,

$$R = \{x_1, x_2, \ldots, x_{\phi(n)}\}.$$

* Each element x_i of R is a unique positive integer less than n with $\gcd(x_i, n) = 1$
* Multiply each element by 'a' and 'mod n'.

$$S = \{(ax_1 \bmod n), (ax_2 \bmod n) \ldots (ax_{\phi(n)} \bmod n)\}$$

* The set S is a permutation of R by the following reasoning:
  1. Because 'a' is relatively prime to n, x_i is relatively prime to n, ax_i must also be relatively prime to n. Thus all the members of S are integers that are less than n and are relatively prime to n.
  2. There are no duplicates in S. If $ax_i \bmod n \equiv ax_j \bmod n$ then $x_i = x_j$

$$\therefore \prod_{i=1}^{\phi(n)} (a x_i \bmod n) = \prod_{i=1}^{\phi(n)} x_i$$

$$\prod_{i=1}^{\phi(n)} a x_i \equiv \prod_{i=1}^{\phi(n)} x_i \pmod n$$

$$a^{\phi(n)} \prod_{i=1}^{\phi(n)} x_i \equiv \prod_{i=1}^{\phi(n)} x_i \pmod n$$

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

Ex①: $a = 3 \quad n = 10$.

$$\phi(n) = \phi(10) = \phi(2)\,\phi(5)$$

$$= \phi(2-1)(5-1) = 1 \times 4 = 4$$

$$\phi(10) = 4$$

$$a^{\phi(n)} = 3^4 = 81 \bmod 10 = 1$$

$$1 \bmod 10 = 1$$

$$a^{\phi(n)} = a\,3^4 \equiv 1 \pmod{10}$$

Ex②: $a = 2 \quad n = 11$

$$\phi(n) = \phi(11) = (11-1) = 10$$

$$a^{\phi(n)} = 2^{10} \bmod 11$$

$$= 1024 \bmod 11 = \frac{2^{11} \bmod 11}{2} = \frac{2}{2} = 1$$

$$= 1 \bmod 11 = 1$$

$$2^{10} \bmod 11 = 1 \bmod 11 \qquad 2^{10} \equiv 1 \pmod{11}$$

(7) Testing For Primality :-

→ Miller-Rabin Algorithm is used to test a large number for primality.

→ Miller-Rabin Algorithm is based on Fermat's and Square Roots test to determine, if the given number is a prime number.

→ It is a deterministic test, but gives the result with high probability

→ Miller-Rabin algorithm is based on following considerations:

1. The number n to be tested for primality is always odd because even numbers cannot be prime. Therefore n-1 is always even and can be written as product of an odd number m and power of 2.

$$n-1 = 2^k m.$$

2. If we choose a positive number 'a' such that $1<a<n-1$, we rewrite Fermat's test for primality of an integer 'n' using 'a' as the base

$$a^{n-1} \equiv 2^k m \equiv 1 \pmod{n}.$$

Algorithm:

a. Determine k and m from $n-1 = 2^k m$

b. Choose a, $1<a<n-1$

c. Compute $x = a^m \bmod n$. If x is 1 or n-1, declare n as prime with high probability.
Else move to next step.

d. Compute $x^{2i} \bmod n$ for $i = 1$ to $(k-1)$. If x^{2i} is $(n-1)$ at any stage, declare n as prime. Else n is declared as composite at the end.

Example: $n = 97$ with $a = 10$

$n - 1 = 96 = 2^5 \times 3$.

$m = 3 \quad k = 5$.

$10^3 \bmod 97 = 30$

$10^{3 \cdot 2} \bmod 97 = 27$

$10^{3 \cdot 4} \bmod 97 = 36$

$10^{3 \cdot 8} \bmod 97 = 75$.

$10^{3 \cdot 16} \bmod 97 = 96 = n - 1$

$\therefore$ 97 is a prime number with high probability.

Example ②.

(8) Chinese Remainder Theorem:-

→ The chinese Reminder Theorem says that if there is a set of Linear congruent equations of a variable 'x' with modulii $m_1, m_2, \ldots, m_k$ which are pairwise relatively prime, then there is a unique solution to the equations.

→ The CRT solution involves the following steps:

a. Determine $M = m_1 \times m_2 \times m_3 \ldots \times m_k$

b. Determine $M_1 = M/m_1$, $M_2 = M/m_2 \ldots$, $M_k = M/M_k$

c. Determine Multiplicative inverses M_i^{-1} modulo m_i.

d. Calculate x using,

$$x = (a_1 \times M_1 \times M_1^{-1} + a_2 \times M_2 \times M_2^{-1} + \ldots + a_1 \times M_k \times M_k^{-1}) \bmod M$$

Example: Solve the following congruent simultaneous equations:

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

a. $M = 3 \times 5 \times 7 = 105$

b. $M_1 = 105/3 = 35$, $M_2 = 105/5 = 21$, $M_3 = 105/7 = 15$

c. $M_1^{-1} \bmod 3 = 35^{-1} \bmod 3 = 2$

$M_2^{-1} \bmod 5 = 21^{-1} \bmod 5 = 1$

$M_3^{-1} \bmod 7 = 15^{-1} \bmod 7 = 1$

$$x = (2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1) \bmod 105$$

$$= 23.$$

$$x = 23$$

Applications of CRT:-

a) CRT is very useful for manipulating very large integers. (numbers potentially larger than 10^{150})

b) CRT is used for solving quadratic congruent equations of the form $a_2 x^2 + a_1 x + a_0 = 0 \pmod{n}$.

Discrete Logarithms:-

problem: For a prime number p. Let a, b be non-zero integers (mod p).

The problem of finding x such that $a^x \equiv b \pmod p$ is called "Discrete Logarithm" problem.

Ex: $p=11 \quad a=2 \quad b=9$

then $2^x \equiv 9 \pmod{11}$

$2^6 \equiv 9 \pmod{11}$

$x=6.$

→ It can be represented as $x = d\log_{a,p} b.$

Ex: $x = d\log_{2,11} 9$

Primitive Roots:

Definition: 'a' is a primitive root mod p if $\{a^k \mid 1 \le k \le p-1\} = \{1, 2, \cdots p-1\}$ [unique exponent].

Ex: $a^7 \bmod 7$

a^1	a^2	a^3	a^4	a^5	a^6	
1	1	1	1	1	1	
2	4	1	2	4	1	← ~~unique~~ 3, 5 are primitive
3	2	6	4	5	1	← unique roots mod 7.
4	2	1	4	2	1	
5	4	6	2	3	1	← unique
6	1	6	1	6	1	

UNIT-5

Part-II: Public key Cryptography

(1) Principles :-

→ Public key cryptography is also called "Asymmetric key Cryptogra

→ Public key cryptosystems rely on one key for encryption and diffe erent but related key for decryption. (2 keys)

→ In public key cryptography, it is computationally infeasible to determine the decryption from encryption key and encryption algorithm.

→ public key cryptosystem works with two keys.

1. A private key 'd' (known only to the owner)
2. A public key 'e' (known by possibly everyone)

→ A public key encryption scheme has 6 ingredients:

1. plaintext: The readable message
2. Ecryption Algorithm: Encryption algorithm performs various operati on plaintext.
3. Public and Private keys: The pair of keys that have been selected so that if one is used for encryption and the other is used for decryption.
4. Ciphertext: The scrambled message produced as output.
5. Decryption Algorithm: Algorithm accepts the ciphertext and matching key and produces original plaintext

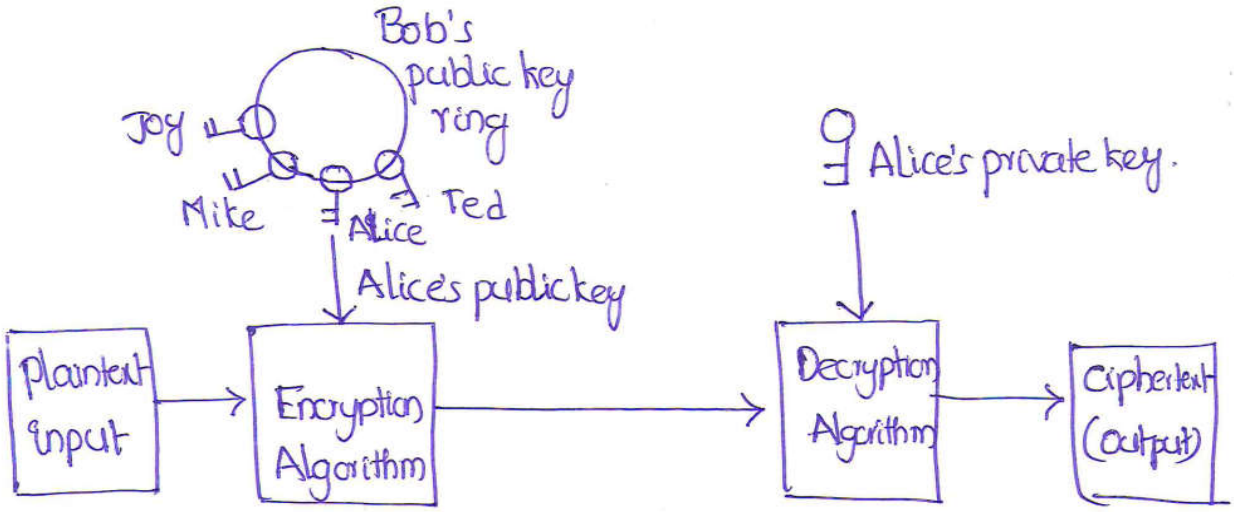

Fig: Public key cryptography encryption.

The essential steps in encryption are the following:

1. Each user generates a pair of keys to be used for encryption and decryption of the messages.
2. Each user places one of the two keys in public register or other accessible file. This is the public key.
3. The companion key is private key. Each user maintains a collection of public keys obtained from others.
4. If Bob wishes to send a confidential message to Alice, Bob encrypts the message with Alice's public key.
5. When Alice receives message she decrypts it using private key. No other recipient can decrypt the message because only Alice knows the private key.

→ The public key cryptosystems must have the following characteristics

1. It must be computationally easy to encipher or decipher a message given the appropriate key.
2. It must be computationally infeasible to derive private key from public key.
3. It must be computationally infeasible to determine private key from choosen plaintext attack.

→ Requirements for public key cryptosystems:

1. It is computationally easy for a party B to generate a pair (public key PU_b, Privatekey PR_b)
2. It is computationally easy for a sender A, knowing the public key and message to be encrypted M, to generate the corresponding ciphertext

$$C = E(PU_b, M)$$

3. It is computationally easy for the receiver B, to decrypt the resulting ciphertext using the private key to recover original message

$$M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$$

4. It is computationally infeasible for an adversary, knowing public key PU_b to determine private key PR_b.
5. It is computationally infeasible for an adversary, knowing publ key PU_b and a ciphertext C to recover original message M.
6. The two key can be applied in either order:

$$M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$$

→ Applications for Public key cryptosystems:

1. Encryption/Decryption: The sender encrypts a message with the receipient's public key.
2. Digital signature: The sender signs the message with private key. signing is achieved by a cryptographi algorithm applied to the message
3. key Exchange: Two sides cooperate to exchange session key.

→ Security in public key Algorithms:

The security of public key Algorithms is based on the dibficulty to factorize and compute discrete logarithms.

Factorising:

* Find the prime factors of a given number
* It is very time consuming.

Discrete Logarithm:

* problem to bind inverse to modular exponentiation.
* Not all discrete logarithms have solutions.
* very time consuming procas

② Public Key Cryptography Algorithms:-

i) RSA Algorithm:-

→ RSA Algorithm is named after Ron Rivest, Adi Shamir, Len Adleman, who invented it in 1977.

→ RSA is the most widely-used public key cryptography algorithm.

→ RSA can be used to encrypt a message without the need to exchange a secret key seperately.

→ RSA can be used for public key encryption and Digital Signatures. It's security depends on the dibficulty of factoring large integers.

→ RSA is a block cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n.

Key Generation Algorithm:-

1. Generate two large random prime numbers p and q
2. Compute $n = p \times q$ and $\phi(n) = (p-1)(q-1)$
3. Choose an integer e, $1 < e < \phi(n)$ such that $\gcd(e, \phi(n)) = 1$
4. Compute the secret exponent d, $1 < d < \phi(n)$, such that ~~$ed \equiv 1 \pmod{\phi(n)}$~~

$$ed \equiv 1 \pmod{\phi(n)}$$

5. public key is (n, e)
   private key is (d, p, q)

   n → modulus
   e → public exponent or encryption exponent
   d → private exponent or decryption exponent.

Example:

1. $p=3 \quad q=5$

2. $n = 3 \times 5 = 15 \qquad \phi(n) = (3-1) \times (5-1) = 8.$

3. Choose e, $1 < e < \phi(n)$

$1 < e < 8$

$e = 7.$

4. Choose d, $1 < d < \phi(n)$

$e \cdot d \equiv 1 \pmod{\phi(n)}$

$e \cdot d \bmod \phi(n) = 1$

$7 \cdot d \bmod 8 = 1$

$7 \cdot 7 \bmod 8 = 1$

$\therefore d = 7.$

Publickey : $PU = \{e, n\} = \{7, 15\}$

Private key : $PR = \{d, n\} = \{7, 15\}.$

Encryption Algorithm:

Sender A does the following:

1. Obtains the recipient's public key (n, e)
2. Represents the plaintext as a positive integer m, $1 \leq m < n$
3. Compute the ciphertext $C = m^e \bmod n$
4. Send the ciphertext to B.

Example: $e = PU_b = \{7, 15\}$.

$m = 2$

$C = 2^7 \bmod 15 = 128 \bmod 15$

$= 8.$

Decryption Algorithm:

Recipient B does the following,

1. Uses his private key (n, d) to compute $m = C^d \bmod n$
2. Extracts the plaintext from message representative m.

Example:

$d = PR_b = \{7, 15\}$

$C = 8.$

$m = 8^7 \bmod 15 = 8^2 . 8^5 \bmod 15$

$= (8^2 \bmod 15 \quad 8^5 \bmod 15) \bmod 15$

$= (4 . 8^2 \bmod 15 \quad 8^3 \bmod 15) \bmod 15$

$= (4 . 4 . 8^2 \bmod 15 \quad 8 \bmod 15) \bmod 15$

$= (4 . 4 . 4 . 8) \bmod 15$

$= [(64 \bmod 15)(8 \bmod 15)] \bmod 15$

$= 4 . 8 \bmod 15$

$= 32 \bmod 15$

$m = 2$

plaintext = 2

Security of RSA:- (Attacks on RSA)

Four possible approaches to attacking the RSA algorithm are as follows:

1. **Brute Force:** It involves trying all posible private keys.
2. **Mathematical Attacks:** There are several approaches, all equivalent in effort to factoring the product of two primes.
3. **Timing Attacks:** These depend on the running time of the decryption algorithm.
4. **Chosen Ciphertext Attacks:** This type of attack exploits properties of RSA algorithm.

Mathematical Attack on RSA:

→ Mathematical attacks focus on attacking the underlying structure of RSA function.

→ The first attack is the attempt to factor modulus N. Because, knowing the factorization of N, one may easily obtain $\phi(N)$ from which d can be determined by $\boxed{de \equiv 1 \pmod{N}}$

Timing Attacks on RSA:

→ Timing attacks target the implementation of RSA.

→ In the timing attack, the attacker can determine the private key by keeping track of how long a computer takes to decipher messages.

→ Repeated Squaring algorithm can be used for the timing attack.

→ Repeated Squaring algorithm depends on following equality:

$$M = c^d = c^{\sum_{i=0}^{k} 2^k d_i} = \prod_{i=0}^{k} c^{2^i d_i}$$

This leads to computing of modular exponentiation with atmost 2k modular multiplications.

→ Repeated Squaring Algorithm:

Initially set $z = c$ and $M = 1$. ~~For i~~

For $i = 0, 1 \ldots k-1$ do

1. If $d_i = 1$ then set $M = Mz \pmod N$
2. set $z = z^2 \bmod N$

At the end of this $M = c^d \pmod N$

→ Counter Measures for Timing Attacks:

1. Constant Exponentiation Time
2. Random Delay
3. Blending.

Diffie-Hellman Key Exchange:-

→ The purpose of Diffie-Hellman key exchange algorithm is to enable two users to securely exchange a key that can be used for symmetric encryption of messages.

→ It is the practical method for establishing a shared secret over an unsecured communication channel.

→ Steps in the Algorithm:

1. 'A' and 'B' agree on a prime number 'p' and a base 'g'.
2. 'A' choses a secret number 'a', and sends B, $(g^a \bmod p)$
3. 'B' choses a secret number 'b', and sends A, $(g^b \bmod p)$
4. 'A' computes $((g^b \bmod p)^a \bmod p)$
5. 'B' computes $((g^a \bmod p)^b \bmod p)$

B and A can used this number as their key.

→ 'p' and 'g' need not to be protected.

→ The effectiveness of Diffie-Hellman key exchange algorithm depends on the difficulty of computing discrete logarithms.

Ex: 'A' and 'B' agree on $p=23$ and $g=5$

'A' chooses $a=6$ and sends $5^6 \bmod 23$

$$5^6 \bmod 23 = 5^1 \cdot 5^2 \cdot 5^3 \bmod 23$$
$$= [(5 \bmod 23)(5^2 \bmod 23)(5^3 \bmod 23)] \bmod 23$$
$$= [5 \cdot 2 \cdot 10] \bmod 23 = 100 \bmod 23$$
$$= 8.$$

'B' choses $b=15$ and sends $5^{15} \bmod 23$

$$5^{15} \bmod 23 = [5^1 \cdot 5^2 \, 5^4 \cdot 5^8] \bmod 23$$

$$= [(5 \bmod 23)(5^2 \bmod 23)(5^4 \bmod 23)(5^8 \bmod 23)] \bmod 23$$

$$= [5 \cdot 2 \cdot 4 \cdot 16] \bmod 23$$

$$= [(10 \bmod 23) \cdot (16 \bmod 23)] \bmod 23$$

$$= [17 \cdot 16] \bmod 23$$

$$= 272 \bmod 23$$

$$= 19.$$

'A' computes $19^6 \bmod 23 = [19 \cdot 19^2 \, 19^3] \bmod 23$

$$= [(19 \bmod 23)(19^2 \bmod 23)(19^3 \bmod 23)] \bmod 23$$

$$= [19 \cdot 16 \cdot 19 \cdot 16] \bmod 23$$

$$= [(19 \cdot 16) \bmod 23 \; (19 \cdot 16) \bmod 23] \bmod 23$$

$$= [(5 \bmod 23)(5 \bmod 23)] \bmod 23$$

$$= 25 \bmod 23 = 2$$

'B' computes $8^{15} \bmod 23$

$$8^{15} \bmod 23 = \left[8^1 8^2 8^4 8^8\right] \bmod 23$$

$$= \left[(8 \bmod 23)(8^2 \bmod 23)(8^4 \bmod 23)(8^8 \bmod 23)\right] \bmod 23$$

$$= [8 \cdot 18 \cdot 2 \cdot 2 \cdot 2] \bmod 23$$

$$= [32 \cdot 36] \bmod 23$$

$$= [(32 \bmod 23)(36 \bmod 23)] \bmod 23$$

$$= [9 \cdot 13] \bmod 23$$

$$= 117 \bmod 23 = \underline{\underline{2}}$$

'2' is the shared secret key.

Elgamal Cryptographic System:-

→ The Elgamal algorithm provides an alternative to the RSA for public key encryption:

1. Security of RSA depends on difficulty of factoring large prime numbers.
2. Security of Elgamal algorithm depends on the difficulty of computing Discrete logarithms in a large prime modulus.

→ Elgamal algorithm uses Random Encryption.

Random Encryption: The same plaintext gives different ciphertext each time it is encrypted.

→ Applications of Elgamal cryptosystem:

1. Digital Signature Standard (DSS)
2. S/MIME
3. Establishing a secure channel for key sharing.

Key Generation Algorithm:

Participant 'A' generates the public/private key pair

1. Generate large prime number 'p' and generator 'g' of the multiplicative group Z_p^* of the integers modulo 'p'.
2. Select a random integer a, $1 < a < p-2$ and compute $g^a \bmod p$
3. A's public key is (p, g, g^a)

   A's private key is 'a'.

Encryption:

participant 'B' encrypts a message 'm' to 'A'

1. obtain A's authentic public key (p, g, g^a)
2. Represent the message as integers 'm' in the range $\{0, 1 \cdots p-1\}$
3. select a random integer k, $1 \le k \le p-2$
4. Compute $\gamma = g^k \bmod p$ and ~~$\delta = m * (g^{ak})^{mod}$~~ $\delta = [m \times g^k] \bmod p$.
5. Send Ciphertext $c = (\gamma, \delta)$ to A.

Decryption:

participant 'A' receives encrypted message m from B.

1. Use private key 'a' to compute $(\gamma^{p-1-a}) \bmod p$.

   Note: $\gamma^{p-1-a} = \gamma^{-a} = a^{-ak}$

2. Recover m' by computing $(\gamma^{-a}) * \delta \bmod p$.

Example:

key generation:

1. $p = 17$, $g = 6$
2. private key $a = 5$
3. $g^a \bmod p = 6^5 \bmod 17$

$= [6^1 6^2 6^2] \bmod 17$

$= [(6 \bmod 17)(6^2 \bmod 17)(6^2 \bmod 17)] \bmod 17$

$= [6 \times 2 \times 2] \bmod 17 = 24 \bmod 17.$

$= 7$

public key $(17, 6, 7)$

Encryption:

$m = 13$, public key $(17, 6, 7)$ $\quad k = 10$

$$\gamma = 6^{10} \bmod p = 6^{10} \bmod 17$$

$$= [6^5 \cdot 6^5] \bmod 17$$

$$= [(6^5 \bmod 17)(6^5 \bmod 17)] \bmod 17$$

$$= [7 \cdot 7] \bmod 17 = 49 \bmod 17$$

$$= 15$$

$$\delta = [m \times g^k] \bmod p = (13 \times 7^{10}) \bmod 17$$

$$= [(13 \bmod 17)(7^{10} \bmod 17)] \bmod 17$$

$$= [13 \times [7^2 \cdot 7^8] \bmod 17] \bmod 17$$

$$= [13 \times [15 \times 15 \times 15 \times 15] \bmod 17] \bmod 17$$

$$= [13 \times [4 \times 4] \bmod 17] \bmod 17$$

$$= [13 \times 16] \bmod 17$$

$$= [13 \times 4 \times 4 \times 15] \bmod 17 = [52 \times 60] \bmod 17$$

$$= [(52 \bmod 17)(60 \bmod 17)] \bmod 17 = [1 \times 9] \bmod 17$$

$$= 9$$

'B' sends the $\gamma = 15$, $\delta = 9$ to 'A'.

Decryption:

private key : $a = 5$

Decryption Factor:

$$r^{-a} * \delta \bmod p = 15^{-5} \bmod 17$$
$$= 15^{11} \bmod 17.$$
$$= 9.$$

Decryption:

$$(8 \times 9) \bmod p = (9 \times 9) \bmod 17$$
$$= 13.$$

Original message $m = 13$

Disadvantages:

1. The Ciphertext is twice as long as the plaintext.
2. Depends on intractability of DL and DH.