EXPERIMENT NO. 4

***Aim:*** To study about Diffie Hellman Key sharing algorithm.

***Theory:***

Diffie–Hellman key exchange is a method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as originally conceptualized by Ralph Merkle and named after Whitfield Diffie and Martin Hellman. D–H is one of the earliest practical examples of public key exchange implemented within the field of cryptography.

Traditionally, secure encrypted communication between two parties required that they first exchange keys by some secure physical channel, such as paper key lists transported by a trusted courier. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

***Cryptographic explanation***

The simplest and the original implementation of the protocol uses the multiplicative group of integers modulo p, where p is prime, and g is a primitive root modulo p. These two values are chosen in this way to ensure that the resulting shared secret can take on any value from 1 to p–1. Here is an example of the protocol, with non-secret values in blue, and secret values in red.

Alice and Bob agree to use a modulus p = 23 and base g = 5 (which is a primitive root modulo 23).

* Alice chooses a secret integer a = 6, then sends Bob A = g^a mod p

A = 5^6 mod 23 = 8

* Bob chooses a secret integer b = 15, then sends Alice B = g^b mod p

B = 5^15 mod 23 = 19

* Alice computes s = B^a mod p

s = 19^6 mod 23 = 2

* Bob computes s = A^b mod p

s = 8^15 mod 23 = 2

* Alice and Bob now share a secret (the number 2).

Both Alice and Bob have arrived at the same value s, because, under mod p,

Diffie–Hellman key agreement is not limited to negotiating a key shared by only two participants. Any number of users can take part in an agreement by performing iterations of the agreement protocol and exchanging intermediate data (which does not itself need to be kept secret). For example, Alice, Bob, and Carol could participate in a Diffie–Hellman agreement as follows, with all operations taken to be modulo p:

The parties agree on the algorithm parameters p and g.

The parties generate their private keys, named a, b, and c.

* Alice computes g^a and sends it to Bob.

Bob computes (g^a)^b = g^ab and sends it to Carol.

Carol computes (g^ab)^c = g^abc and uses it as her secret.

* Bob computes g^b and sends it to Carol.

Carol computes (g^b)^c = g^bc and sends it to Alice.

Alice computes (g^bc)^a = g^bca = g^abc and uses it as her secret.

* Carol computes g^c and sends it to Alice.

Alice computes (g^c)^a = g^ca and sends it to Bob.

Bob computes (g^ca)^b = g^cab = g^abc and uses it as his secret.

An eavesdropper has been able to see g^a, g^b, g^c, g^ab, g^ac, and g^bc, but cannot use any combination of these to efficiently reproduce g^abc.

***Program:***

import java.util.\*;

class DH {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the value of p & q: " );

int p = sc.nextInt();

int q = sc.nextInt();

System.out.println("Enter a Prime Number a: ");

int a = sc.nextInt();

System.out.println("Enter Another prime number less than "+a+": ");

int b = sc.nextInt();

int r = (int)Math.pow(b,p)%a;

int s = (int)Math.pow(b,q)%a;

System.out.println("Public Key of Alice: "+r);

System.out.println("Public Key of Bob: "+s);

int rk = (int)Math.pow(s,p)%a;

int sk = (int)Math.pow(r,q)%a;

System.out.println("Secret Key of Alice: "+rk);

System.out.println("Secret Key of Bob: "+sk);

if (rk==sk) {

System.out.println("Transmission successful");

} else {

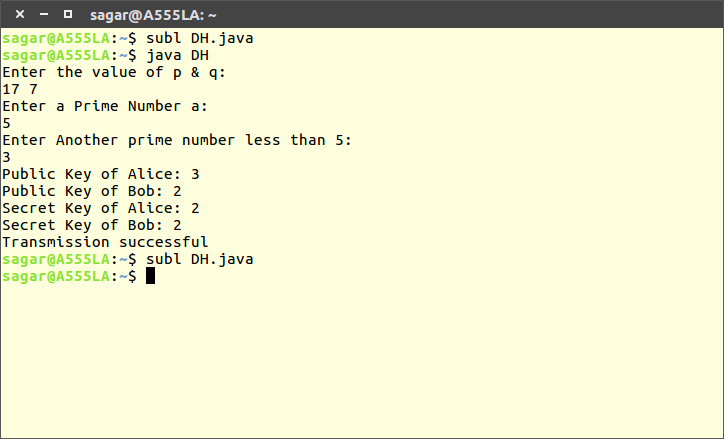
System.out.println("Transmission failed");

}

}

}

***Output:***

******

***Conclusion:*** Hence Diffie Hellman Key sharing algorithm is studied and implemented.