IT8761 – Security Laboratory

Reshma Ramesh Babu

312217104129

Exercise 7

Aim: To implement the Diffie-Hellman Key Exchange algorithm.

Code:

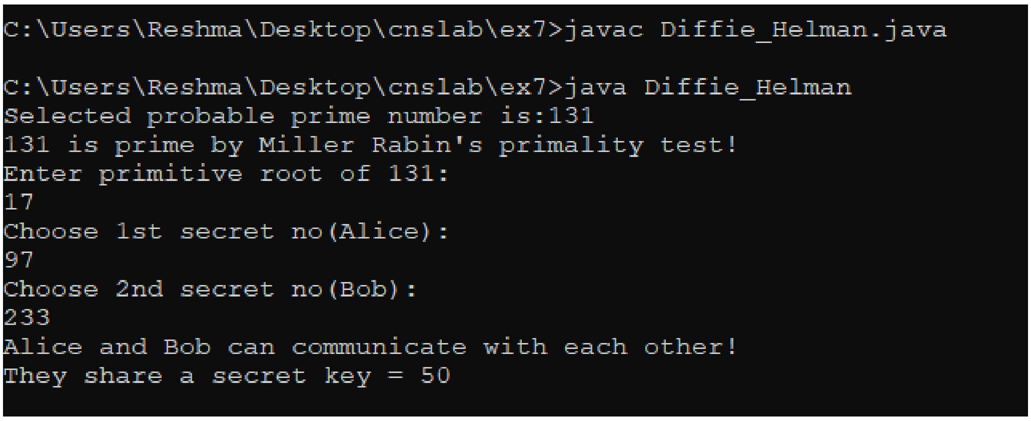
```
import java.util.*;
import java.math.BigInteger;
class Diffie_Helman
{
   public boolean isPrime(long n, int iteration)
   {
      /** base case **/
      if (n == 0 || n == 1)
         return false;
      /** base case - 2 is prime **/
      if (n == 2)
         return true;
      /** an even number other than 2 is composite **/
      if (n % 2 == 0)
         return false;

      long s = n - 1;
      while (s % 2 == 0)
         s /= 2;
      Random rand = new Random();
```

```
    for (int i = 0; i < iteration; i++)
    {
        long r = Math.abs(rand.nextLong());
        long a = r % (n - 1) + 1, temp = s;
        long mod = modPow(a, temp, n);
        while (temp != n - 1 && mod != 1 && mod != n - 1)
        {
            mod = mulMod(mod, mod, n);
            temp *= 2;
        }
        if (mod != n - 1 && temp % 2 == 0)
            return false;
    }
    return true;
}
/** Function to calculate (a ^ b) % c **/
public long modPow(long a, long b, long c)
{
    long res = 1;
    for (int i = 0; i < b; i++)
    {
        res *= a;
        res %= c;
    }
    return res % c;
}
```

```
    /** Function to calculate (a * b) % c **/
    public long mulMod(long a, long b, long mod)
    {
        return BigInteger.valueOf(a).multiply(BigInteger.valueOf(b)).mod(BigInteger.valueOf(mod)).longValue();
    }
    public static void main(String args[])
    {
        BigInteger q,g,xa,xb,ya,yb,k1,k2;
        Diffie_Helman dh = new Diffie_Helman();
        Scanner sc=new Scanner(System.in);
        int length = 8;
        Random random = new Random();
        //select random prime number
        q = BigInteger.probablePrime(length, random);
        System.out.println("Selected probable prime number is:"+q);
        long l;
        l = q.longValue();
        boolean prime = dh.isPrime(l, 3);
        if(prime){
            System.out.println(q+" is prime by Miller Rabin's primality test!");
            System.out.println("Enter a primitive root of "+q+":");
            g=sc.nextBigInteger();
            System.out.println("Choose 1st secret no(Alice):");
            xa=sc.nextBigInteger();
            System.out.println("Choose 2nd secret no(Bob):");
            xb=sc.nextBigInteger();
            ya = g.modPow(xa,q);
            yb = g.modPow(xb,q);
            k1 = yb.modPow(xa,q);
            k2 = ya.modPow(xb,q);
            if(k1.compareTo(k2) == 0){
                System.out.println("Alice and Bob can communicate with each other!");
                System.out.println("They share a secret key = "+k1);
            }
            else{
                System.out.println("ALice and Bob cannot communicate with each other!!!");
            }
        }
        else
            System.out.println(q+" is not prime");
    }
}
```

Output:



```
C:\Users\Reshma\Desktop\cnslab\ex7>javac Diffie_Helman.java

C:\Users\Reshma\Desktop\cnslab\ex7>java Diffie_Helman
Selected probable prime number is:131
131 is prime by Miller Rabin's primality test!
Enter primitive root of 131:
17
Choose 1st secret no(Alice):
97
Choose 2nd secret no(Bob):
233
Alice and Bob can communicate with each other!
They share a secret key = 50
```