Ex.No. : 2(b) DIFFIE-HELLMAN KEY EXCHANGE ALGORITHM
Date :

AIM:

To implement a Diffie-Hellman Key Exchange algorithm.

ALGORITHM:

1. Sender and receiver publicly agree to use a modulus p and base g which is a primitive root modulo p.
2. Sender chooses a secret integer x then sends Bob R1 = g^x mod p
3. Receiver chooses a secret integer y, then sends Alice R2 = g^y mod p
4. Sender computes k1 = B^x mod p
5. Receiver computes k2 = A^y mod p
6. Sender and Receiver now share a secret key.

PROGRAM:

```
import java.io.*;
import java.math.BigInteger;
class dh
{
public static void main(String[]args)throws IOException
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter prime number:");
BigInteger p=new BigInteger(br.readLine());

System.out.print("Enter primitive root of "+p+":");
BigInteger g=new BigInteger(br.readLine());

System.out.println("Enter value for x less than "+p+":");
BigInteger  x=new BigInteger(br.readLine());
BigInteger R1=g.modPow(x,p);
System.out.println("R1="+R1);

System.out.print("Enter value for y less than "+p+":");
BigInteger y=new BigInteger(br.readLine());
BigInteger R2=g.modPow(y,p);
System.out.println("R2="+R2);

BigInteger k1=R2.modPow(x,p);
System.out.println("Key calculated at Sender's side:"+k1);
BigInteger k2=R1.modPow(y,p);
System.out.println("Key calculated at Receiver's side:"+k2);
System.out.println("Diffie-Hellman secret key was calculated.");
}
}
```

OUTPUT

```
C:\Security Lab New\programs>javac dh.java

C:\Security Lab New\programs>java dh
Enter prime number:
11
Enter primitive root of 11:7
Enter value for x less than 11:
3
R1=2
Enter value for y less than 11:6
R2=4
Key calculated at Sender's side:9
Key calculated at Receiver's side:9
Diffie-Hellman secret key was calculated.
```

RESULT:

Thus the Diffie-Hellman key exchange algorithm was implemented and executed successfully.