CRYPTOGRAPHY LABORATORY FILE
CS-511

Submitted to:
Dr. Samayveer Singh
Assistant Professor
Department of Computer Science

Submitted by:
Shashi Shekhar Azad
Roll No.: 23203029
M. Tech. CSE 1st Semester

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DR. B. R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY
JALANDHAR

Assignment 10

Write a program to implement Diffie-Hellman Key Exchange Algorithm and verify that same key value is generated at both sides.

Diffie-Hellman code:

```
import java.util.Scanner;
public class DiffieHellman {
   public static void main(String args[]) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter two large prime number: ");
      System.out.print("n: ");
      int n = sc.nextInt();
      while (!isPrime(n)) {
         System.out.print(n + " is not a prime number, Enter prime number: ");
         n = sc.nextInt();
      }
      System.out.print("g: ");
      int g = sc.nextInt();
      while (!isPrime(g)) {
         System.out.print(g + " is not a prime number, Enter prime number: ");
         g = sc.nextInt();
      }
      System.out.print("Alice, Enter a random number x: ");
      int x = sc.nextInt();
      System.out.print("Bob, Enter a random number y: ");
      int y = sc.nextInt();
      long A = powerMod(g, x, n);
      long B = powerMod(g, y, n);
      long K1 = powerMod(B, x, n);
      long K2 = powerMod(A, y, n);
      System.out.println("Alice and Bob's shared public key is");
      System.out.println("Alice's calculated key K1: " + K1);
      System.out.println("Bob's calculated key K2: " + K2);
      sc.close();
   }
   public static boolean isPrime(int n) {
      if (n <= 1) {
         return false;
```

```
        }
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }

    public static long powerMod(long base, long exponent, long modulus) {
        long result = 1;
        while (exponent > 0) {
            if (exponent % 2 == 1) {
                result = (result * base) % modulus;
            }
            base = (base * base) % modulus;
            exponent /= 2;
        }
        return result;
    }
}
```

Output:

```
PS D:\DATAs\NITJ\CryptoLab\java> java DiffieHellman
Enter two large prime number:
n: 173
g: 113
Alice, Enter a random number x: 36
Bob, Enter a random number y: 41
Alice and Bob's shared public key is
Alice's calculated key K1: 109
Bob's calculated key K2: 109
PS D:\DATAs\NITJ\CryptoLab\java>
```