Name: Faraz Pathan

Rollno: COTB70

‘’’ To implement the Diffie-Hellman Key Exchange algorithm’’’

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

// Function to calculate (base^exponent) % modulus

int power(int base, int exponent, int modulus) {

    int result = 1;

    while (exponent > 0) {

        if (exponent % 2 == 1)

            result = (result \* base) % modulus;

        base = (base \* base) % modulus;

        exponent = exponent / 2;

    }

    return result;

}

// Function to perform Diffie-Hellman key exchange

void diffieHellman() {

    int p, g, a, b; // Publicly known values: prime 'p', generator 'g'

    printf("Enter the value of prime (p) and generator (g): ");

    scanf("%d %d", &p, &g);

    printf("Enter the private key for Alice (a): ");

    scanf("%d", &a);

    printf("Enter the private key for Bob (b): ");

    scanf("%d", &b);

    // Calculate public keys

    int A = power(g, a, p);

    int B = power(g, b, p);

    // Calculate shared secret

    int secretKey\_Alice = power(B, a, p);

    int secretKey\_Bob = power(A, b, p);

    printf("Public Key for Alice: %d\n", A);

    printf("Public Key for Bob: %d\n", B);

    printf("Shared Secret Key for Alice: %d\n", secretKey\_Alice);

    printf("Shared Secret Key for Bob: %d\n", secretKey\_Bob);

}

int main() {

    diffieHellman();

    return 0;

}

**OUTPUT:**

Enter the value of prime (p) and generator (g): 97

17

Enter the private key for Alice (a): 37

Enter the private key for Bob (b): 19

Public Key for Alice: 74

Public Key for Bob: 26

Shared Secret Key for Alice: 15

Shared Secret Key for Bob: 15