Experiment no.: 07

Title: Perform key exchange using DH algorithm
Course Outcome: Use cryptography algorithms and protocols to achieve Computer Security
Theory:

What is Diffie-Hellman Key Exchange (exponential key exchange)?
The Diffie-Hellman key exchange (also known as exponential key exchange) is a method for securely exchanging cryptographic keys over an insecure channel. It is a fundamental building block of many secure communication protocols, including SSL/TLS and SSH.

The Diffie-Hellman key exchange works by allowing two parties (Alice and Bob) to agree on a shared secret key over an insecure channel, without any other party being able to intercept the key or learn anything about it. The key exchange involves the following steps ?

- Alice and Bob agree on two large prime numbers, p and g, and a public key exchange algorithm.

- Alice chooses a secret integer, a, and computes A = g^a mod p. She sends A to Bob.

- Bob chooses a secret integer, b, and computes B = g^b mod p. He sends B to Alice.

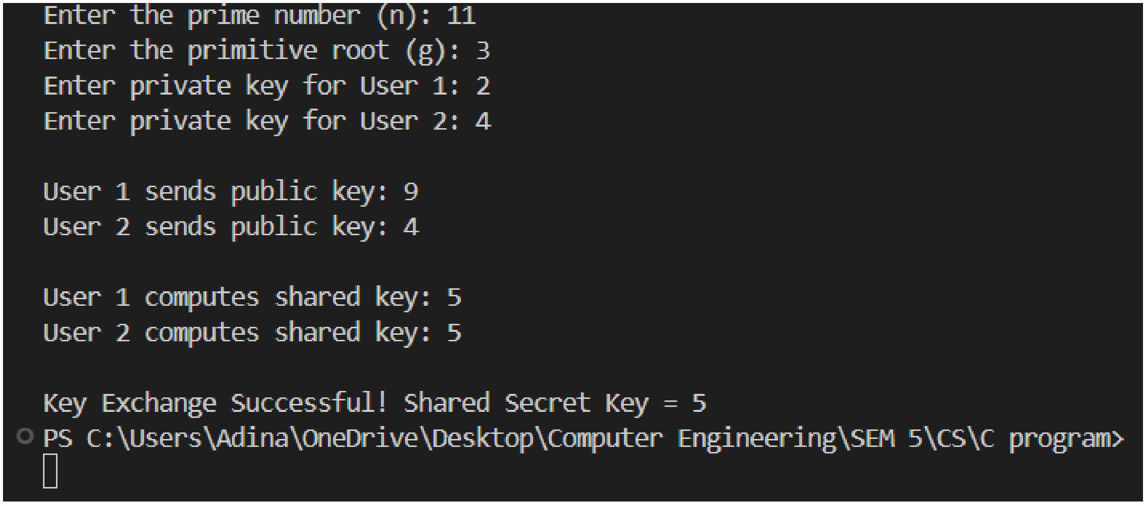
- Alice computes s = B^a mod p. Bob computes s = A^b mod p.

- Alice and Bob now both have shared secret keys, which they can use to establish a secure communication channel.

The security of the Diffie-Hellman key exchange relies on the fact that it is computationally infeasible for an attacker to determine the shared secret keys from the public values of p, g, A, and B. This allows Alice and Bob to exchange the key securely, even over an insecure channel.
Code:

```
#include <stdio.h>
#include <math.h>
int main() {
    int prime, base, privA, privB;
    printf("Enter the prime number (n): ");
    scanf("%d", &prime);
    printf("Enter the primitive root (g): ");
    scanf("%d", &base);
    printf("Enter private key for User 1: ");
    scanf("%d", &privA);
    printf("Enter private key for User 2: ");
    scanf("%d", &privB);
    double pubA = fmod(pow(base, privA), prime);
    double pubB = fmod(pow(base, privB), prime);
    double sharedX = fmod(pow(pubB, privA), prime);
    double sharedY = fmod(pow(pubA, privB), prime);
    printf("\nUser 1 sends public key: %d\n", (int)pubA);
    printf("User 2 sends public key: %d\n", (int)pubB);
    printf("\nUser 1 computes shared key: %d\n", (int)sharedX);
    printf("User 2 computes shared key: %d\n", (int)sharedY);
    if ((int)sharedX == (int)sharedY)
        printf("\nKey Exchange Successful! Shared Secret Key = %d\n", (int)sharedX);
    else
        printf("\nKey Exchange Failed.\n");
    return 0;}
```

Output:



```
Enter the prime number (n): 11
Enter the primitive root (g): 3
Enter private key for User 1: 2
Enter private key for User 2: 4

User 1 sends public key: 9
User 2 sends public key: 4

User 1 computes shared key: 5
User 2 computes shared key: 5

Key Exchange Successful! Shared Secret Key = 5
PS C:\Users\Adina\OneDrive\Desktop\Computer Engineering\SEM 5\CS\C program>
```

Conclusion:

I have successfully written C program to implement DH key exchange algorithm.