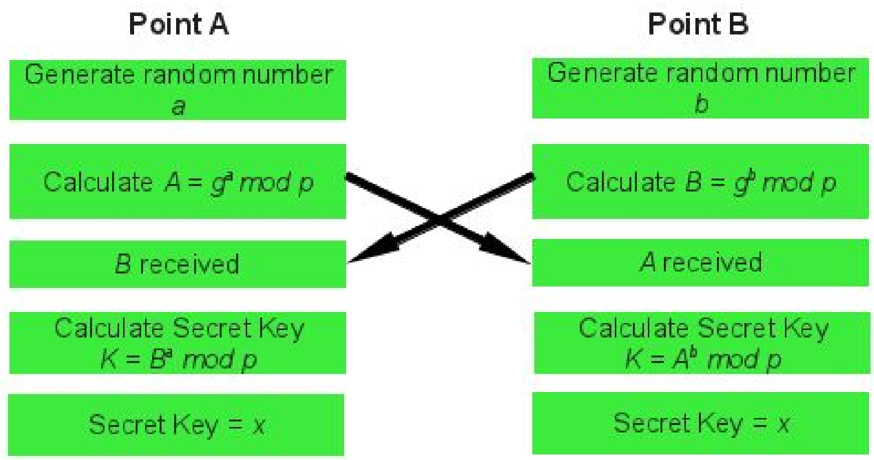
Experiment 7

Aim: Write a program to implement key exchange using Diffie-Hallman key exchange.

Theory: Diffie-Hellman key exchange, also called exponential key exchange, is a method of digital encryption that uses numbers raised to specific powers to produce decryption keys on the basis of components that are never directly transmitted, making the task of a would-be code breaker mathematically overwhelming. To implement Diffie-Hellman, the two end users Alice and Bob, while communicating over a channel they know to be private, mutually agree on positive whole numbers p and q, such that p is a prime number and q is a generator of p. The generator q is a number that, when raised to positive whole-number powers less than p, never produces the same result for any two such whole numbers.



p = prime number, g = generator

The value of p may be large but the value of q is usually small. Once Alice and Bob have agreed on p and q in private, they choose positive whole-number personal keys a and b, both less than the prime-number modulus p. Neither user divulges their personal key to anyone; ideally they memorize these numbers and do not write them down or store them anywhere. Next, Alice and Bob compute public keys a* and b* based on their personal keys according to the formulas: -

a* = [q^a] mod p

b* = [q^b] mod p

The two users can share their public keys a* and b* over a communications medium assumed to be insecure, such as the Internet or a corporate wide area network (WAN). From these public keys, x and y can be generated by either user on the basis of their own personal keys.

Alice computes x using the formula

x = [(b*)^a] mod p

Bob computes x using the formula

y = [(a*)^b] mod p

The value of x and y turns out to be the same according to either of the above two formulas. The two users can therefore, in theory, communicate privately over a public medium with an encryption method of their choice using the decryption key x.

Source Code:

```
#include <iostream>
#include<math.h>
using namespace std;
long long int calc(long long int a, long long int b, long long int N)
{
    if (b == 1)
        return a;
    else
        return (((long long int)pow(a, b)) % N);
}

int main() {
    long long int N, G, x, a, y, b, ka, kb;
    cout<<"\nEnter the values for N and G\n";
    cin>>N>>G;
    cout<<"\nEnter the private key for A ";
    cin>>a;
    cout<<"Enter the private key for B ";
    cin>>b;
    cout<<"\nThe private key of A: "<<a;
    cout<<"\nThe private key of B: "<<b;

    x = calc(G, a, N);
    y = calc(G, b, N);
    cout<<"\n\nAfter exchange of keys";
    cout<<"\nkey recieved by A is (y):"<<y;
    cout<<"\nkey recieved by B is (x):"<<x;

    ka = calc(y, a, N);
```

```
    kb = calc(x, b, N);
    cout<<"\n\nActual key for the A is : "<<ka;
    cout<<"\nActual Key for the B is : "<<kb;
    if(ka= =kb)
            cout<<"\n\nBoth users have matching keys, thus successful";
    else
            cout<<"Key Generation failed";
    return 0;
}
```

Output:

```
kunal@DESKTOP-AITAEP7:/mnt/c/Users/Admin/Desktop/college/7th Semester/Information and network
kunal@DESKTOP-AITAEP7:/mnt/c/Users/Admin/Desktop/college/7th Semester/Information and network

Enter the values for N and G
23 8

Enter the private key for A 3
Enter the private key for B 6

The private key of A: 3
The private key of B: 6

After exchange of keys
key recieved by A is (y):13
key recieved by B is (x):6

Actual key for the A is : 12
Actual Key for the B is : 12

Both users have matching keys, thus successful
```

Learning Outcomes:

The personal keys a and b, which are critical in the calculation of x, have not been transmitted over a public medium. Because it is a large and apparently random number, a potential hacker has almost no chance of correctly guessing x, even with the help of a powerful computer to conduct millions of trials.