**Garvin Chanderia**

**2203132**

**LY-ISA1**

**EXPERIMENT 4**

**Aim:** Implement Diffie Hellman key exchange algorithm for secret key generation and distribution of public key.

**Theory:**

The Diffie-Hellman algorithm is being used to establish a shared secret that can be used for secret communications while exchanging data over a public network using the elliptic curve to generate points and get the secret key using the parameters.

* For the sake of simplicity and practical implementation of the algorithm, we will consider only 4 variables, one prime P and G (a primitive root of P) and two private values a and b.
* P and G are both publicly available numbers. Users (say Alice and Bob) pick private values a and b and they generate a key and exchange it publicly. The opposite person receives the key and that generates a secret key, after which they have the same secret key to encrypt.

**Program:**

#include <cmath>

#include <iostream>

using namespace std;

// Power function to return value of a ^ b mod P

long long int power(long long int a, long long int b,

long long int P)

{

if (b == 1)

return a;

else

return (((long long int)pow(a, b)) % P);

}

// Driver program

int main()

{

long long int P, G, x, a, y, b, ka, kb;

// Both the persons will be agreed upon the

// public keys G and P

cout<<"enter the prime no P :";

cin>>P;

// A prime number P is taken

cout << "The value of P : " << P << endl;

cout<<"enter the primitive root G:";

cin>>G;

// A primitive root for P, G is taken

cout << "The value of G : " << G << endl;

// Alice will choose the private key a

cout<<"enter the private key for a:";

cin>>a; // a is the chosen private key

cout << "The private key a for Alice : " << a << endl;

x = power(G, a, P); // gets the generated key

// Bob will choose the private key b

cout<<"enter the private key for b :";

cin>>b; // b is the chosen private key

cout << "The private key b for Bob : " << b << endl;

y = power(G, b, P); // gets the generated key

// Generating the secret key after the exchange

// of keys

ka = power(y, a, P); // Secret key for Alice

kb = power(x, b, P); // Secret key for Bob

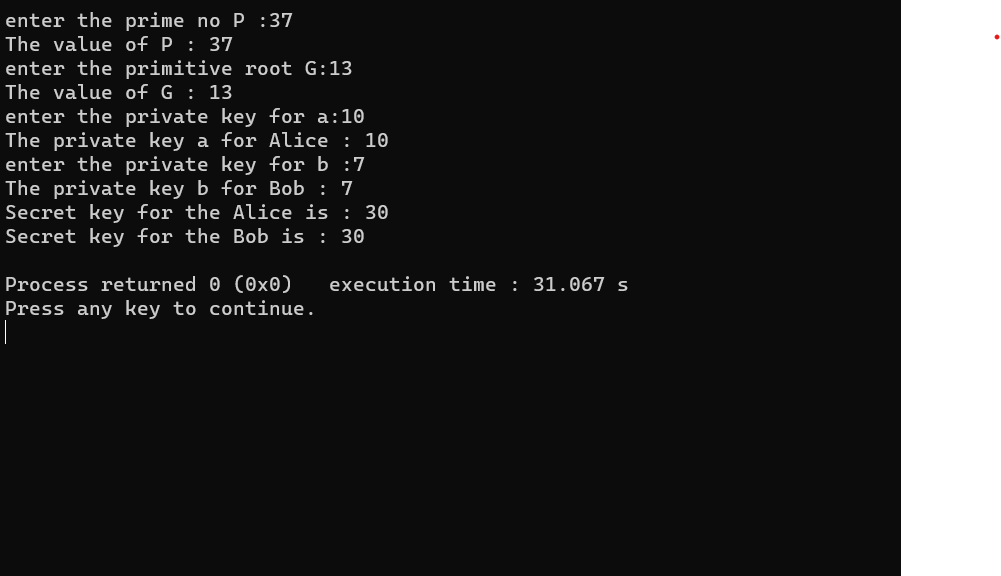
cout << "Secret key for the Alice is : " << ka << endl;

cout << "Secret key for the Bob is : " << kb << endl;

return 0;

}

**Output:**

****

**Conclusion:**

The common key exchange between both the people is 30.

**Application:**

Many protocols use Diffe-Hellman is used to enhance the security of the system. Some of the systems are – secure shell , public key infrastructure, internet key exchange, internet protocol security and many more.