**Cryptography And Network Security Lab**

**Assignment submission**

**PRN No: 2019BTECS00017**

**Full name: Muskan Raju Attar**

**Batch: B5**

**Assignment: 11**

**Title of assignment: Implementation of Diffie – Hellman Key Exchange Method**

**Title:**

Implementation of Diffie – Hellman Key Exchange Method

**Aim:**

To develop and implement the Diffie – Hellman Key Exchange Method

**Theory:**

* Diffie–Hellman key exchange is a method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as conceived by Ralph Merkle and named after Whitfield Diffie and Martin Hellman.
* The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. This key can then be used to encrypt subsequent communications using a symmetric-key cipher.
* Although Diffie–Hellman key agreement itself is a non-authenticated key-agreement protocol, it provides the basis for a variety of authenticated protocols, and is used to provide forward secrecy in Transport Layer Security's ephemeral modes.

**Implementation of Diffie - Hellman Key Exchange Algorithm**

**Code:**

/\* This program calculates the Key for two persons

using the Diffie-Hellman Key exchange algorithm using C++ \*/

#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

P = 23; // A prime number P is taken

cout<<"Enter the Prime Number: ";

cin>>P;

G = 9; // A primitive root for P, G is taken'

cout<<"Enter the Primitive Root: ";

cin>>G;

// Alice will choose the private key a

a = 4; // a is the chosen private key

cout<<"Enter Alice Private Key: ";

cin>>a;

// Bob will choose the private key b

b = 3; // b is the chosen private key

cout<<"Enter Bob Private Key: ";

cin>>b;

cout<<"\n\tDiffie-Hellmen Key Exchnage Algorithm\t\n";

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

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

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

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

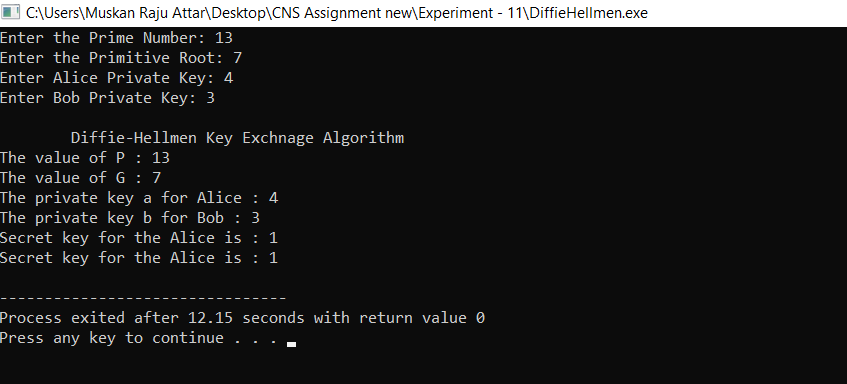
cout << "Secret key for the Alice is : " << ka << endl;

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

return 0;

}

**Output:**



**Conclusion:**.

The Diffie - Hellman theorem can be used to get the primitive number of the large Prime numbers