Name: Rahul Jha
Roll no: 09
Class: COMP-B

EXPERIMENT NO: 4

Aim: Implement program using Diffie Hellman Algorithm

Theory:

The Diffie–Hellman (DH) Algorithm is a key-exchange protocol that enables two parties communicating over public channel to establish a mutual secret without it being transmitted over the Internet. DH enables the two to use a public key to encrypt and decrypt their conversation or data using symmetric cryptography.

DH is generally explained by two sample parties, Alice and Bob, initiating a dialogue. Each has a piece of information they want to share, while preserving its secrecy. To do that they agree on a public piece of benign information that will be mixed with their privileged information as it travels over an insecure channel. Their secrets are mixed with the public information, or public key, and as the secrets are exchanged the information they want to share is commingled with the common secret. As they decipher the other's message, they can extract the public information and with knowledge of their own secret, deduce the new information that was carried along. While seemingly uncomplicated in this method's description, when long number strings are used for private and public keys, decryption by an outside party trying to eavesdrop is mathematically infeasible even with considerable resources.

The algorithm is based on Elliptic Curve Cryptography, a method of doing public-key cryptography based on the algebra structure of elliptic curves over finite fields. The DH also uses the trapdoor function, just like many other ways to do public-key cryptography. The simple idea of understanding to the DH Algorithm is the following.

1. The first party picks two prime numbers, g and p and tells them to the second party.

2. The second party then picks a secret number (let's call it a), and then it computes ga mod p and sends the result back to the first party; let's call the result A. Keep in mind that the secret number is not sent to anyone, only the result is.

3. Then the first party does the same; it selects a secret number b and calculates the result B similar to the step 2.

4.. Then, this result is sent to the second party.

5. The second party takes the received number B and calculates Ba mod p

6. The first party takes the received number A and calculates Ab mod p

This is where it gets interesting; the answer in step 5 is the same as the answer in step 4. This means both parties will get the same answer no matter the order of exponentiation.

(ga mod p)b mod p = gab mod p

(gb mod p)a mod p = gba mod p

The number we came within steps 4 and 5 will be taken as the shared secret key. This key can be used to do any encryption of data that will be transmitted, such as blowfish, AES, etc.

Diffie Hellman Algorithm:

1. key =(YA)XBmod q -> this is the same as calculated by B

2. Global Public Elements

q: q is a prime number

a: a < q and α is the primitive root of q

3. Key generation for user A

Select a Private key XA Here, XA <q

Now, Calculation of Public key YA YA = aXA mod q

4. Key generation for user B

Select a Private key XB Here, XB <q

Now, Calculation of Public key YB YB = aXb mod q

5. Calculation of Secret Key by A

key =(YB)XA mod q

6. Calculation of Secret Key by B

key =(YA)XB mod q

Program:

```
from random import randint

if__name__== '__main__':

    P = 11


    G = 5


    print('The Value of P is :%d'%(P))
    print('The Value of G is :%d'%(G))

    a = 4
    print('The Private Key a for Person1 is :%d'%(a))


    x = int(pow(G,a,P))


    b = 3
    print('The Private Key b for Person2 is :%d'%(b))


    y = int(pow(G,b,P))



    ka = int(pow(y,a,P))


    kb = int(pow(x,b,P))

    print('Secret key for the Person1 is : %d'%(ka))
    print('Secret Key for the Person2 is : %d'%(kb))
```

Output:

Conclusion: We have understood the concept Diffie Hellman Algorithm, which is a key-exchange protocol that enables two parties communicating over public channel to establish a mutual secret without it being transmitted over the Internet. We have successfully implemented Diffie Hellman Algorithm using python programming.