Experiment No. 2

Implementation of Diffie Hellman Key Exchange Algorithm

Course Outcome [CSL602.2]: Implement symmetric and asymmetric key cryptography

Aim: Write a program to implement Diffie-Hellman Algorithm.

Objectives:

- To understand the principles of symmetric key cryptography.
- To understand the Diffie-Hellman Key exchange algorithm.
- To understand the possible attacks on Diffie-Hellman.

Outcomes: The learner will be able to

Apply the cryptosystem to ensure secure key exchange between sender and receiver.

Hardware / Software Required: C/C++/JAVA/Python

Theory:

Diffie-Hellman key exchange is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

The Diffie–Hellman key exchange algorithm solves the following dilemma. Alice and Bob want to share a secret key for use in a symmetric cipher, but their only means of communication is insecure.Every piece of information that they exchange is observed by their adversary Eve. How is it possible for Alice and Bob to share a key without making it available to Eve? At first glance it appears that Alice and Bob face an impossible task. It was a brilliant insight of Diffie and Hellman that the difficulty of the discrete logarithm problem for F* p provides a possible solution. The simplest, and original, implementation of the protocol uses the Multiplicative group of integers modulo p, where p is prime and g is primitive root mod p. Here is an example of the protocol:

Step 1: Alice and Bob get public numbers P = 23, G = 9

Step 2: Alice selected a private key a = 4 and Bob selected a private key b = 3

Step 3: Alice and Bob compute public values Alice: x =(9^4 mod 23) = (6561 mod 23) = 6

Bob: y = (9^3 mod 23) = (729 mod 23) = 16

Step 4: Alice and Bob exchange public numbers

Step 5: Alice receives public key y =16 and Bob receives public key x = 6

Step 6: Alice and Bob compute symmetric keys Alice: ka = y^a mod p = 65536 mod 23 = 9

Bob: kb = x^b mod p = 216 mod 23 = 9

Step 7: 9 is the shared secret.

In the original description, the Diffie-Hellman exchange by itself does not provide authentication of the communicating parties and is thus vulnerable to a man-in-the-middle attack. A person in the middle may establish two distinct Diffie-Hellman key exchanges, one with Alice and the other with Bob, effectively masquerading as Alice to Bob, and vice versa, allowing the attacker to decrypt (and read or store) then re-encrypt the messages passed between them. A method to authenticate the communicating parties to each other is generally needed to prevent this type of attack.

Algorithm:

Alice and Bob, two users who wish to establish secure communications. We can assume that Alice and Bob know nothing about each other but are in contact.

1. Communicating in the clear, Alice and Bob agree on two large positive integers, p and g, where p is a prime number and g is a primitive root mod p.
2. Alice randomly chooses another large positive integer, XA, which is smaller than p. XA will serve as Alice's private key.
3. Bob similarly chooses his own private key, XB.
4. Alice computes her public key, YA, using the formula YA = (g^XA) mod p.
5. Bob similarly computes his public key, YB, using the formula YB = (g^XB) mod p.
6. Alice and Bob exchange public keys over the insecure circuit.
7. Alice computes the shared secret key, k, using the formula k = (YB ^XA) mod p.
8. Bob computes the same shared secret key, k, using the formula k = (YA ^XB) mod p.
9. Alice and Bob communicate using the symmetric algorithm of their choice and the shared secret key, k, which was never transmitted over the insecure circuit.

Implementation:

```
from random import randint
if __name__ == '__main__':
P=int(input("The Value of P is :"))
G=int(input("The Value of G is :"))
a = int(input("The Private Key a for Alice is :"))
```

```
x = int(pow(G,a,P))
b = int(input("The Private Key b for Bob is :"))
y = int(pow(G,b,P))
ka = int(pow(y,a,P))
kb = int(pow(x,b,P))
print('Secret key for the Alice is : %d'%(ka))
print('Secret Key for the Bob is : %d'%(kb))
```

Output:

```
The Value of P is :23
The Value of G is :9
The Private Key a for Alice is :4
The Private Key b for Bob is :3
Secret key for the Alice is : 9
Secret Key for the Bob is : 9
```

Conclusion: The Diffie-Hellman key exchange algorithm is used to make secure channel to share secret key between sender and receiver.