Experiment no.5

Learning Objective: Analyze and implement Diffie-Hellman Key Exchange Algorithm

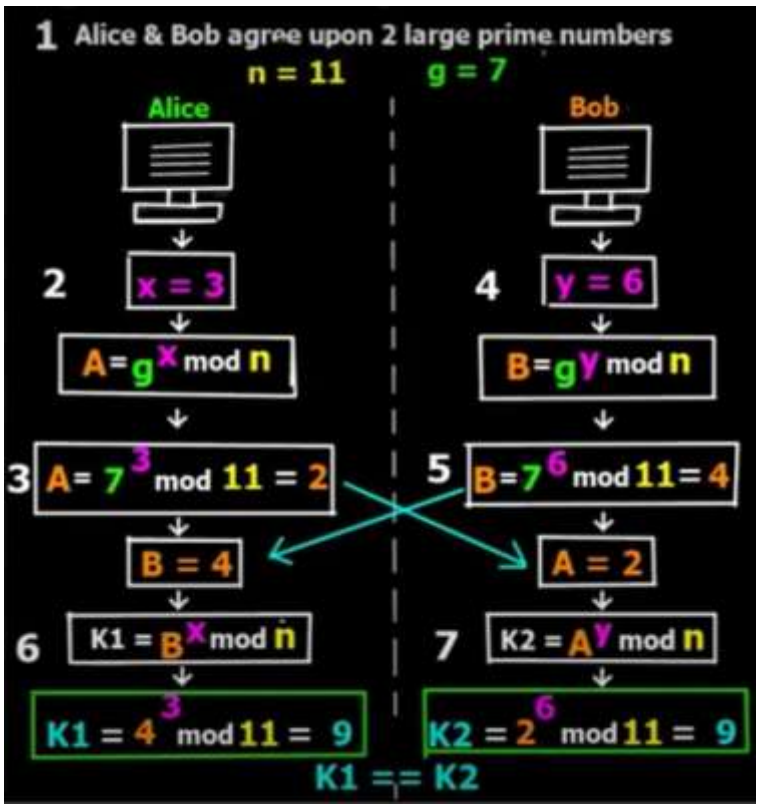
Tools: C/C++/Java/Python

Theory: DIFFIE–HELLMAN KEY EXCHANGE:

Diffie–Hellman key exchange (D–H) is a specific method of exchanging keys. It is one of the earliest practical examples of key exchange implemented within the field of cryptography. 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 communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

The Diffie–Hellman key agreement was invented in 1976 during a collaboration between Whitfield Diffie and Martin Hellman and was the first practical method for establishing a shared secret over an unprotected communication channel.

Diffie–Hellman establishes a shared secret that can be used for secret communications by exchanging data over a public network.



STEP 1: GLOBAL PUBLIC ELEMENTS:

Firstly, Alice and Bob agree on two large prime numbers, n and g. These two integers neednot be kept secret. Alice and Bob can use an insecure channel to agree on them.

STEP 2: ASYMMETRIC KEY GENERATION BY USER 'A':

Alice chooses another large random number X, and calculates, the public key, A, such that:

$$A = g^X \bmod n$$

STEP 3: Alice sends the number A to Bob.

STEP 4: KEY GENERATION BY USER 'B':

Bob independently chooses another large random number Y, and calculates, the public key,B, such that:

$$B = g^Y \bmod n$$

STEP 5: Bob sends the number B to Alice.

STEP 6: SYMMETRIC KEY (K) GENERATION BY USER 'A':

A now computes the secret key, K1 as follows:

$$K1 = B^X \bmod n$$

STEP 7: SYMMETRIC KEY (K) GENERATION BY USER 'B':

B now computes the secret key, K2 as follows:

$$K2 = A^Y \bmod n$$

NOTE:

It should be difficult for Alice to solve for Bob's private key or for Bob to solve for Alice's private key. If it is not difficult for Alice to solve for Bob's private key (or vice versa), Eve may simply substitute her own private / public key pair, plug Bob's public key into her private key, produce a fake shared secret key, and solve for Bob's private key (and use that to solve for the shared secret key. Eve may attempt to choose a public / private key pair that willmake it easy for her to solve for Bob's private key).

Program:

```
import random

p = int(input("Enter value of p:"))
g = int(input("Enter value of g:"))

# Alice will choose the private key a
a = random.randint(1, 100)
print("The value of private key of Alice:", a)
# Bob will choose the private key b
b = random.randint(1, 100)
print("The value of private key of Bob:",b)

 # gets the generated key
A = ((pow(g, a)) % p)
 # gets the generated key
B = ((pow(g, b)) % p)

  # Secret key for Alice
Ka = ((pow(B, a)) % p)

  # Secret key for Bob
Kb = ((pow(A, b)) % p)

print("Secret key at A = ", str(Ka))
print("Secret key at B = ", str(Kb))
```

Output:

```
ALS\exp_diffie.py"
Enter value of p:23
Enter value of g:9
The value of private key of Alice: 79
The value of private key of Bob: 28
Secret key at A =  9
Secret key at B =  9
```

Advantages :

- The sender and receiver don't need any prior knowledge of each other.
- Once the keys are exchanged, the communication of data can be done through an insecure channel.
- The sharing of the secret key is safe.

Disadvantages :

- The algorithm cannot be sued for any asymmetric key exchange.
- Similarly, it cannot be used for signing digital signatures.
- Since it doesn't authenticate any party in the transmission, the Diffie Hellman key exchange is susceptible to a man-in-the-middle attack.

Learning Outcomes: The student will be able to

LO1: Understand the Diffie-Hellman Key Exchange Algorithm

LO2: Analyze and implement the Diffie-Hellman Key Exchange Algorithm

Course Outcomes: Upon completion of the course students will be able to analyze and implement Diffie-Hellman Key Exchange Algorithm for generation of shared symmetric key

Conclusion: Successfully, analyzed and implemented Diffie-Hellman Exchange Algorithm.

For Faculty Use

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of Practical [40%]**	**Attendance / Learning Attitude [20%]**	
Marks Obtained				