

**FR. CONCEICAO RODRIGUES COLLEGE OF
ENGINEERING**
Department of Computer Engineering Course,

Subject & Experiment Details

Academic Year	2021-22	Estimated	
Course & Semester	T.E. (CMPN)- Sem VI	Subject Name & Code	CSS - (CSL604)
Module No.	02 – Mapped to CO-2	Chapter Title	Key Management Techniques
Practical No:	5		
Title:	Implementation of Diffie- Hellman Key exchange algorithm and Simulation of Man In the Middle attack		
Date of Performance:	15/02/2022		
Date of Submission:	02/03/2022		
Roll No:	8940		
Name of the Student:	Warren Fernandes		

Evaluation:

Sr. No	Rubric	Grade
1	On time submission Or completion (2)	
2	Preparedness(2)	
3	Skill (4)	
4	Output (2)	

Signature of the Teacher:

Date:

M NS

Title: Implementation of Diffie- Hellman Key exchange algorithm and Simulation of Man In the Middle attack.

Lab Objective:

This lab provides insight into:

- The working of Diffie – Hellman Key Exchange Protocol.

Reference: “Cryptography and Network Security” B. A. Forouzan
“Cryptography and Network Security” Atul Kahate

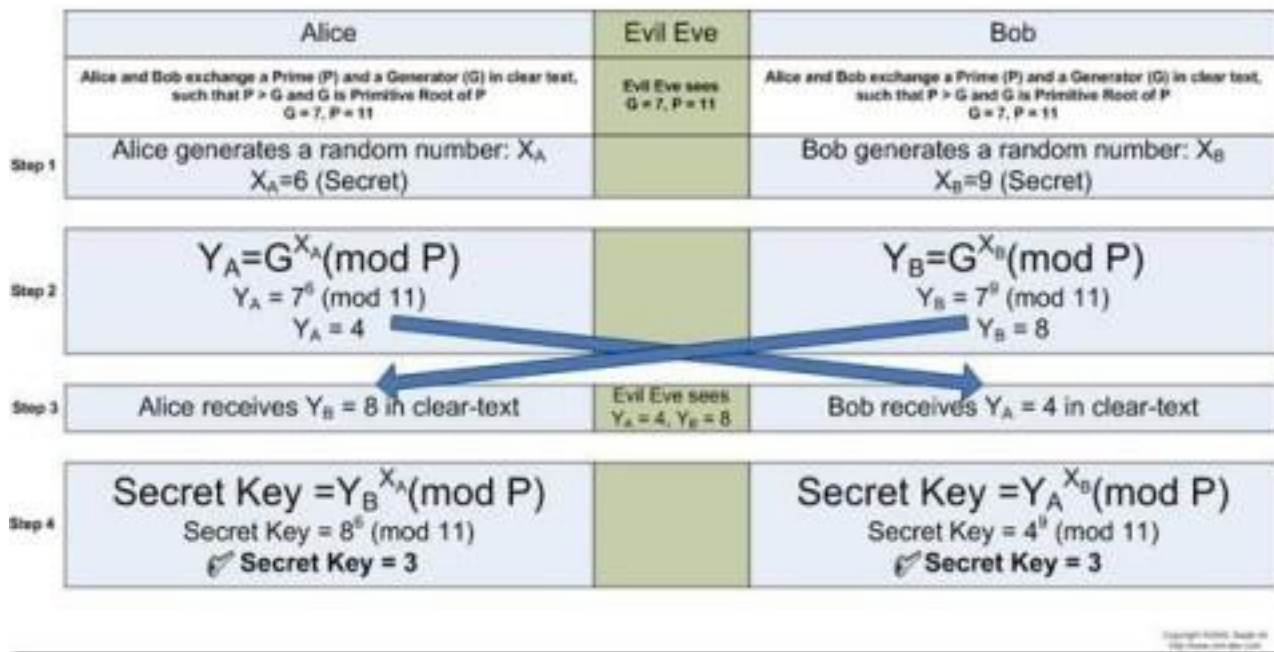
Prerequisite: Any programming Language and Knowledge of Symmetric Key cryptography.

Theory:

Diffie-Hellman is a way of *generating* a shared secret between two people in such a way that the secret can't be seen by observing the communication.

This is particularly useful because you can use this technique to create an encryption key with someone, and then start encrypting your traffic with that key. And even if the traffic is recorded and later analyzed, there's absolutely no way to figure out what the key was, even though the exchanges that created it may have been visible.

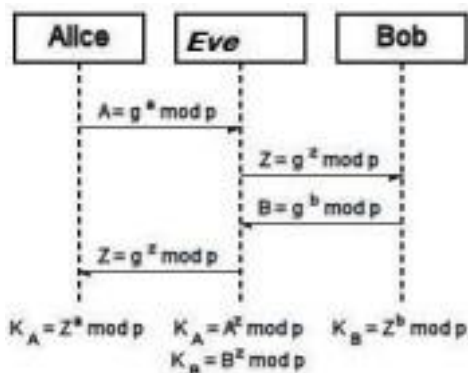
Diffie Hellman Key Exchange



M
NS

Man – In – The –Middle Attack


Let us take the example illustrated by Diffie-Hellman to discuss the Man-in-the-Middle Attack. Let us that Eve is in the middle of Alice and Bob. Eve does not need the value of x or y to attack the protocol. She can fool both Alice and Bob by the following process.



- 1 Alice choose a, calculate $A = g^a \pmod{p}$
- 2 Eve, the intruder, intercepts A, she chooses z, calculate $Z = g^z \pmod{p}$, and sends Z to both Alice and Bob.
- 3 Bob choose b, calculate $B = g^b \pmod{p}$, and sends B to Alice; B is interpreted by Eve and never reaches Alice.

- 4 Alice and Eve calculate the same key $g^{az} \bmod p$, which become a shared key between Alice and Eve. Alice however think that it is a key shared between Bob and herself.
- 5 Eve and Bob calculate the same key $g^{bz} \bmod p$, which become a shared key between Eve and Bob. Bob, however, thinks that it is a key shared between Alice and himself. This situation is called man-in-the-middle attack.

Practical and Real Time Applications

- Used as a method of exchanging cryptography keys for **use** in symmetric encryption algorithms like AES
- Public key encryption schemes based on DF – ElGamal encryption 
- Password-authenticated key agreement
- public key infrastructure - It is possible to use DF as part of PKI

Conclusion:

The program was tested for different sets of inputs.
 Program is working SATISFACTORY NOT SATISFACTORY (Tick appropriate outcome)

M
NS

CODE:

In [1]:

```
print("IMPLEMENTATION OF DIFFIE-HELLMAN KEY EXCHANGE ALGORITHM\n")
N = int(input("Enter the value of N :"))
G = int(input("Enter the value of G : "))

a = int(input("Enter the private key for Alice: "))
b = int(input("Enter the private key for Bob: "))
x = int(pow(G,a,N))

y = int(pow(G,b,N))
ka = int(pow(y,a,N))
kb = int(pow(x,b,N))
print('Secret key for the Alice is : %d'%(ka))
print('Secret Key for the Bob is : %d\n'%(kb))
```

OUTPUT:

IMPLEMENTATION OF DIFFIE-HELLMAN KEY EXCHANGE ALGORITHM

```
Enter the value of N :24
Enter the value of G : 8
Enter the private key for Alice: 5
Enter the private key for Bob: 4
Secret key for the Alice is : 16
Secret Key for the Bob is : 16
```

Post Lab Assignment:

- 1 In the Diffie- Hellman protocol, what happens if x and y have the same value,that is, Alice and Bob have accidentally chosen the same number? Are A and B (values exchanged by Alice and Bob to each other) the same? Do the session keys calculated by Alice and Bob have the same value? Use an example to prove your claims.**

In the Diffie-Hellman protocol even if x and y have the same value, there will not be any change and it remains as if they are different values.

```
*** IMPLEMENTATION OF DIFFIE-HELLMAN KEY EXCHANGE ALGORITHM ***

Enter the value of N :23
Enter the value of G : 7
-----

Enter the private key for Alice: 4
Enter the private key for Bob: 4
-----

Secret key for the Alice is : 6
Secret Key for the Bob is : 6
-----
```

From the above example, we can see that the session keys calculated by Alice and Bob have the same value that is 6.

2 How to secure Diffie-Hellman from Man-in –the –Middle attack?

A malicious Malory, that has a MitM (man in the middle) position, can manipulate the communications between Alice and Bob, and break the security of the key exchange.

Step by Step explanation of this process:

Step 1: Selected public numbers p and g, p is a prime number, called the “modulus” and g is called the base.

Step 2: Selecting private numbers.

Let Alice pick a private random number a and let Bob pick a private random number b, Malory picks 2 random numbers c and d.

Step 3: Intercepting public values,

Malory intercepts Alice's public value ($ga \pmod p$), block it from reaching Bob, and instead sends Bob her own public value ($gc \pmod p$) and Malory intercepts Bob's public

value ($gb \pmod p$), block it from reaching Alice, and instead sends Alice her own public

value ($gd \pmod p$)

Step 4: Computing secret key

Alice will compute a key $S1 = gda \pmod p$, and Bob will compute a different key, $S2 = gcb \pmod p$

Step 5: If Alice uses $S1$ as a key to encrypt a later message to Bob, Malory can decrypt it, re-encrypt it using $S2$, and send it to Bob. Bob and Alice won't notice any problem and may assume their communication is encrypted, but Malory can decrypt, read, modify, and then re-encrypt all their conversations.

M
NS