

## LABORATORY

CEL62: Cryptography and System Security  
Winter 2021

<b>Experiment 2:</b>	<b>Implementing Diffie Helman</b>
----------------------	-----------------------------------

Note: Students are advised to read through this lab sheet before doing experiment. On-the-spot evaluation may be carried out during or at the end of the experiment. Your performance, teamwork/Personal effort, and learning attitude will count towards the marks.

NAME : SHREYAS PATEL

ROLL NO : 42

## Experiment 2 : Traditional Crypto Methods and Key Exchange

### OBJECTIVE :

Implement Diffie Hellman key exchange algorithm in Scilab/C/Python/R.

#### Diffie –Hellman Key exchange algorithm:

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. Although Diffie–Hellman key agreement itself is an anonymous (non-authenticated) key-agreement protocol, it provides the basis for a variety of authenticated protocols, and is used to provide perfect forward secrecy in Transport Layer Security's ephemeral modes (referred to as EDH or DHE depending on the cipher suite).

#### Diffie Hellman Key Exchange

	Alice	Evil Eve	Bob
	Alice and Bob exchange a Prime (P) and a Generator (G) in clear text, such that $P > G$ and G is Primitive Root of P $G = 7, P = 11$		Alice and Bob exchange a Prime (P) and a Generator (G) in clear text, such that $P > G$ and G is Primitive Root of P $G = 7, P = 11$
Step 1	Alice generates a random number: $X_A$ $X_A = 6$ (Secret)	Evil Eve sees $G = 7, P = 11$	Bob generates a random number: $X_B$ $X_B = 9$ (Secret)
Step 2	$Y_A = G^{X_A} \pmod{P}$ $Y_A = 7^6 \pmod{11}$ $Y_A = 4$		$Y_B = G^{X_B} \pmod{P}$ $Y_B = 7^9 \pmod{11}$ $Y_B = 8$
Step 3	Alice receives $Y_B = 8$ in clear-text	Evil Eve sees $Y_A = 4, Y_B = 8$	Bob receives $Y_A = 4$ in clear-text
Step 4	Secret Key $= Y_B^{X_A} \pmod{P}$ Secret Key $= 8^6 \pmod{11}$ ✔ Secret Key = 3		Secret Key $= Y_A^{X_B} \pmod{P}$ Secret Key $= 4^9 \pmod{11}$ ✔ Secret Key = 3

#### a. Diffie Hellman

- Enter the Prime Number g:
- Enter second Prime Number n:
- Enter the Secret x:
- Enter the Secret y
- $K_1$ :
- $K_2$ :

### CODE :

```
import math
```

```

def diffie_hiemann(g,n,x,y):
    a = pow(n,x,g)
    b = pow(n,y,g)
    temp = a
    a = b
    b = temp
    k1 = pow(a,x,g)
    k2 = pow(b,y,g)
    print("The keys are " + str(k1) + " and " + str(k2))
if a == 6:
    print("Diffie-Hiemann Method:")
    print("Enter g and n:")
    g = int(raw_input())
    n = int(raw_input())
    print("Enter x and y:")
    x = int(raw_input())
    y = int(raw_input())
    diffie_hiemann(g,n,x,y)

```

## OUTPUT :

Enter your choice of method to be used:

6

Diffie-Hiemann Method:

Enter g and n:

23

45

Enter x and y:

3

8

The keys are 1 and 1

**OBSERVATIONS** : This algorithm makes algebraic calculations to turn the keys of both the users to one key which they actually want to share.

**CONCLUSION** : Through this experiment I came to know the use of these algorithms in cryptography.