Name: Ameya Jangam
UID:2019130025
Subject:CSS
TE COMPS

Experiment 2

Aim: To implement Diffie Hellman Key Exchange Cryptography Method in Python.

Code:

```
import numpy as np
import math
import copy

def diffieHellman():
    g=int(input("Enter Prime Number (g):"))
    p=int(input("Enter Second Prime Number (p):"))
    xa=int(input("Enter Secret (Xa):"))
    xb=int(input("Enter Secret (Xb):"))

    #Generated keys
    ya=int((pow(g,xa,p)))
    yb=int((pow(g,xb,p)))

    print(f"Ya:{ya}")
    print(f"Yb:{yb}")

    #Shared keys
    k1=int((pow(yb,xa,p)))
    k2=int((pow(ya,xb,p)))

    print(f"The secret key k1 shared with A is: {k1}")
    print(f"The secret key k2 shared with B is: {k2}")

    return
def exitter():
    print("Exiting...")
    exit()


def returnAlgoFunc(func):
```

```
    return func()

def takeInput(text):
    givenString=input(text)
    return givenString

print("Select one of the operations:")
switcher = {
    1: diffieHellman,
    2: exitter
}

while True:
    print("1. Diffie Hellman Algorithm")
    print("2. Exit")
    print("Enter your option:")
    option = int(input())
    returnAlgoFunc(switcher.get(option))
```

Output:

1. Diffie Hellman Algorithm

```
1. Substitution Algorithm
2. ROT 13 Algorithm
3. Transpose Algorithm
4. Double Transposition Algorithm
5. Vernam Cipher Algorithm
6. Diffie Hellman Algorithm
7. Exit
Enter your option:
6
Enter Prime Number (g):7
Enter Second Prime Number (p):11
Enter Secret (Xa):6
Enter Secret (Xb):9
Ya:4
Yb:8
The secret key k1 shared with A is: 3
The secret key k2 shared with B is: 3
```

Conclusion:

From this experiment, I got to learn about the Diffie Hellman Key Exchange cryptography algorithm and implementing it in python. Here are some points which I found out while working on this experiment:

1. **Diffie Hellman Method**

   From the Diffie Hellman Method, I learnt how we can send shared secret keys over an insecure channel with the help of simple mathematical operations.

   The Diffie Hellman algorithm makes use of primitive roots of any prime numbers that are given by the user. In this case, I have given the two prime numbers (G and N) as inputs in order to save the computation time of checking whether the number is prime or not.

   Since the exponentiation and modulo operation are used together in this algorithm, I used the pow(a,b,p) which took care of the exponentiation and then did a modular operation on it i.e replicating the expression $a^b \% p$ which is used in calculating the generating keys i.e Y_a and Y_b.

GitHub Link = https://github.com/ameyajangam22/CSS-LAB-2019130025