Cryptography & Network Security Lab

PRN/ Roll No: 2019BTECS00090

Full name: Udaykumar Gadikar

Assignment No. 11

Title: Diffie-Hellman Key Exchange

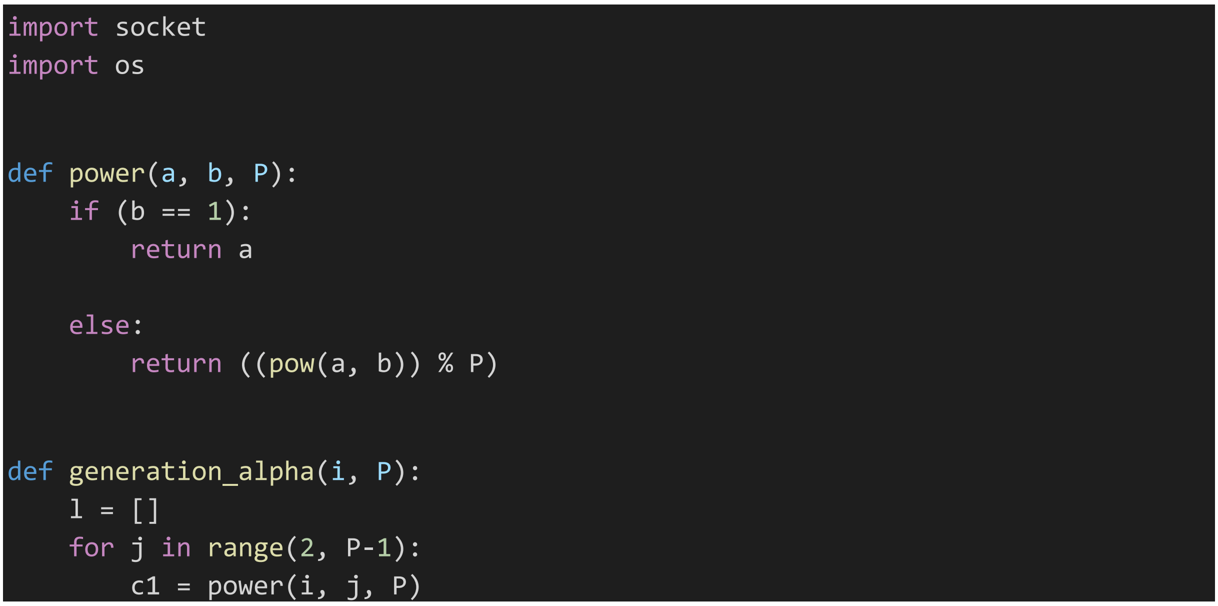
Aim: To Demonstrate Diffie-Hellman Key Exchange

Theory:

Diffie–Hellman key exchange is a method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as conceived by Ralph Merkle and named after Whitfield Diffie and Martin Hellman.

Code:

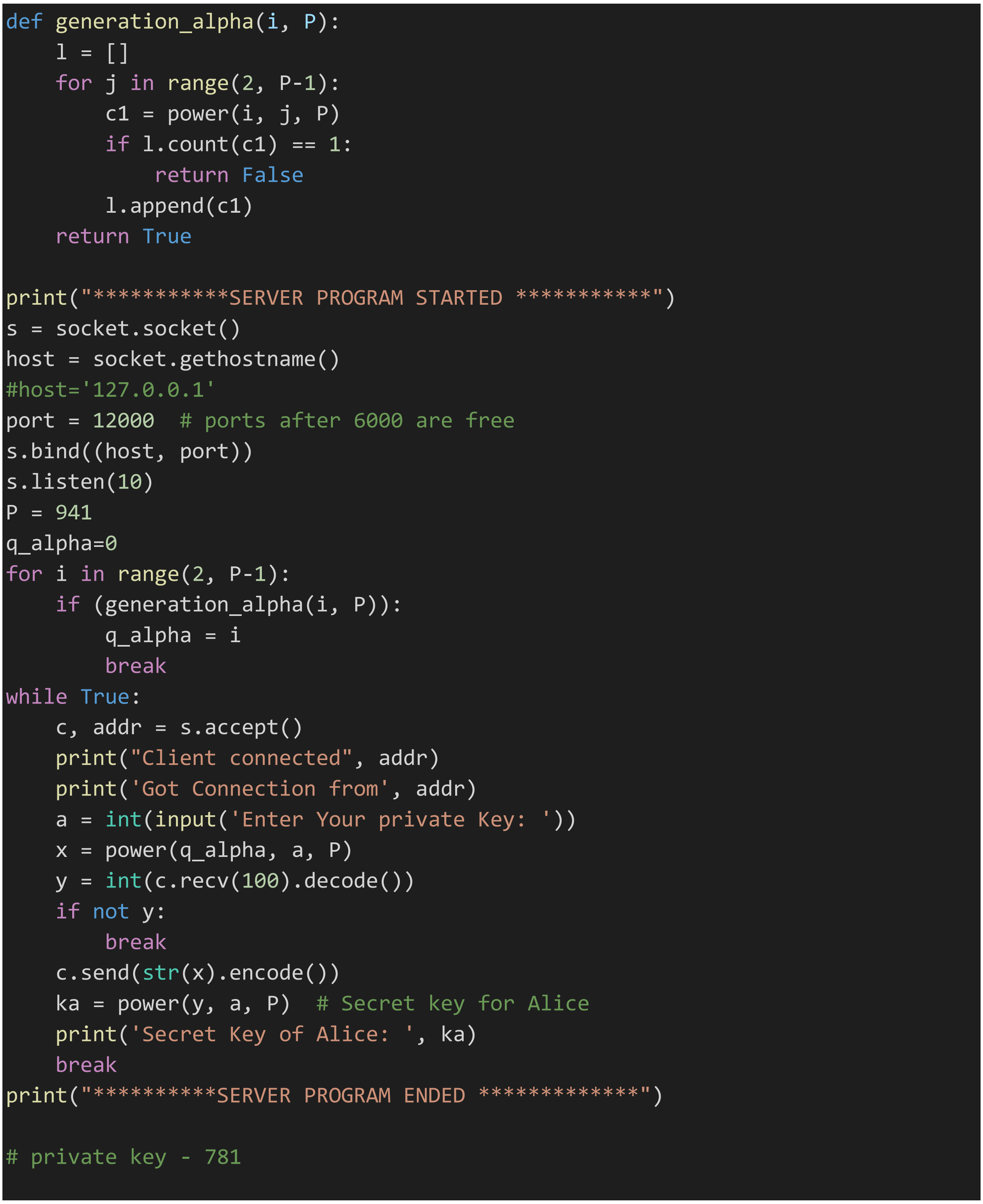
Client side Code: -



```
import socket
import os


def power(a, b, P):
    if (b == 1):
        return a

    else:
        return ((pow(a, b)) % P)


def generation_alpha(i, P):
    l = []
    for j in range(2, P-1):
        c1 = power(i, j, P)
        if l.count(c1) == 1:
            return False
        l.append(c1)
    return True


print("************CLIENT PROGRAM STARTED ******************")
s = socket.socket()
host = socket.gethostname()  # server hostname
#host='127.0.0.1'
port = 12000  # same as server
s.connect((host, port))
print("Connected to : ", host, port)
# fileToSend = open("ToSend.txt","r")
# content = fileToSend.read()
P = 941
q_alpha=0
for i in range(2, P-1):
    if (generation_alpha(i, P)):
        q_alpha = i
        break
b = int(input('Enter Your private Key: '))
y = power(q_alpha, b, P)
s.send(str(y).encode())
x = int(s.recv(100).decode())
kb = power(x, b, P)
print('Secret Key of Bob: ', kb)
print("************CLIENT PROGRAM ENDED ******************")

# private key - 347
```

Server side Code : -

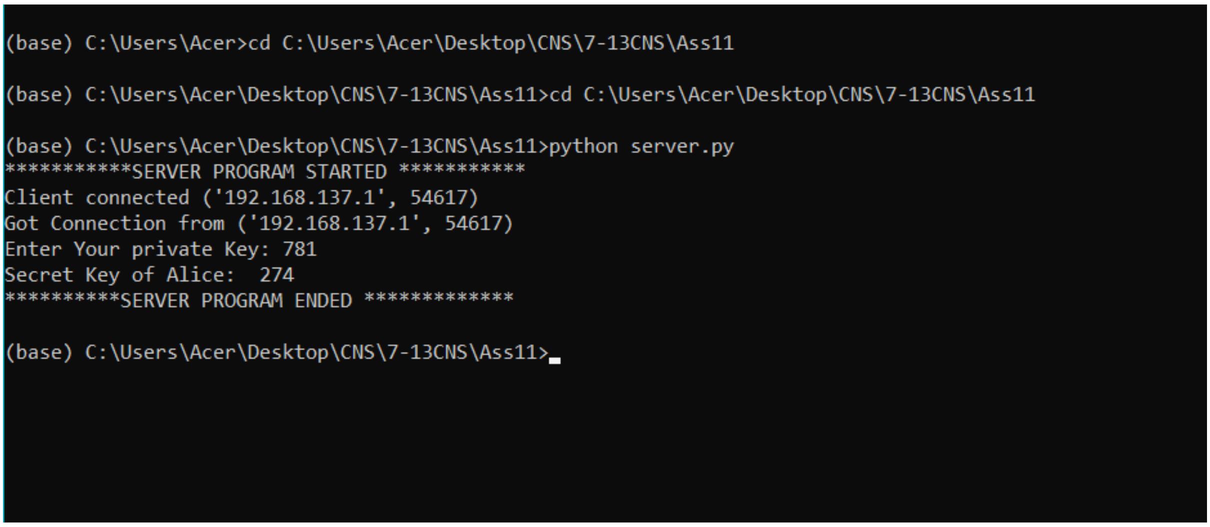
```
import socket
import os
import sys


def power(a, b, P):
    if (b == 1):
        return a

    else:
        return ((pow(a, b)) % P)
```



```python
def generation_alpha(i, P):
    l = []
    for j in range(2, P-1):
        c1 = power(i, j, P)
        if l.count(c1) == 1:
            return False
        l.append(c1)
    return True

print("***********SERVER PROGRAM STARTED ***********")
s = socket.socket()
host = socket.gethostname()
#host='127.0.0.1'
port = 12000  # ports after 6000 are free
s.bind((host, port))
s.listen(10)
P = 941
q_alpha=0
for i in range(2, P-1):
    if (generation_alpha(i, P)):
        q_alpha = i
        break
while True:
    c, addr = s.accept()
    print("Client connected", addr)
    print('Got Connection from', addr)
    a = int(input('Enter Your private Key: '))
    x = power(q_alpha, a, P)
    y = int(c.recv(100).decode())
    if not y:
        break
    c.send(str(x).encode())
    ka = power(y, a, P)  # Secret key for Alice
    print('Secret Key of Alice: ', ka)
    break
print("**********SERVER PROGRAM ENDED *************")

# private key - 781
```

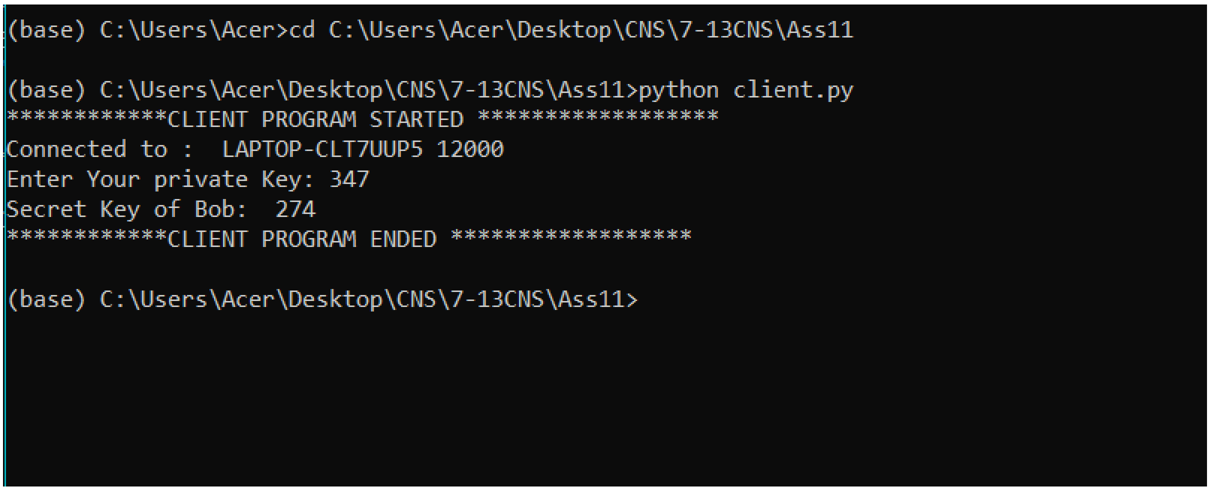
Output:

Server side Output: -



```
(base) C:\Users\Acer>cd C:\Users\Acer\Desktop\CNS\7-13CNS\Ass11

(base) C:\Users\Acer\Desktop\CNS\7-13CNS\Ass11>cd C:\Users\Acer\Desktop\CNS\7-13CNS\Ass11

(base) C:\Users\Acer\Desktop\CNS\7-13CNS\Ass11>python server.py
***********SERVER PROGRAM STARTED ***********
Client connected ('192.168.137.1', 54617)
Got Connection from ('192.168.137.1', 54617)
Enter Your private Key: 781
Secret Key of Alice:  274
**********SERVER PROGRAM ENDED *************

(base) C:\Users\Acer\Desktop\CNS\7-13CNS\Ass11>
```

Client side Output :-



```
(base) C:\Users\Acer>cd C:\Users\Acer\Desktop\CNS\7-13CNS\Ass11

(base) C:\Users\Acer\Desktop\CNS\7-13CNS\Ass11>python client.py
************CLIENT PROGRAM STARTED ******************
Connected to :  LAPTOP-CLT7UUP5 12000
Enter Your private Key: 347
Secret Key of Bob:  274
************CLIENT PROGRAM ENDED ******************

(base) C:\Users\Acer\Desktop\CNS\7-13CNS\Ass11>
```

Conclusion:

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 channel. This key can then be used to encrypt subsequent communications using a symmetric-key cipher.