CNS Lab 05

Implement a client and a server on different computers using python. Perform the communication between these two entities by using RSA cryptosystem.

UDP - Server

In [1]:
```
import random
import hashlib
import sys
```

In [2]:
```
def gcd(a,b):
    if b==0:
        return a
    else:
        return gcd(b,a%b)
```

In [3]:
```
def isPrime(n) :
    # Corner cases
    if (n <= 1) :
        return False
    if (n <= 3) :
        return True

    # This is checked so that we can skip
    # middle five numbers in below loop
    if (n % 2 == 0 or n % 3 == 0) :
        return False

    i = 5
    while(i * i <= n) :
        if (n % i == 0 or n % (i + 2) == 0) :
            return False
        i = i + 6

    return True
```

In [4]:
```
# Get a prime number
def generatePrime(num = 100):
    L1 = []
    for i in range(60, num + 1):
        if isPrime(i):
            L1.append(i)

    p = random.choice(L1)
    L1.pop(L1.index(p))
    q = random.choice(L1)

    t = (p-1)*(q-1)
    n = p*q

    for e in range(2,t):
        if gcd(e,t)== 1:
            break

    for i in range(1,10):
        x = 1 + i*t
        if x % e == 0:
```

```
            d = int(x/e)
            break

    return e,d,n
```

In [5]:
```
Alphabet_List = {'A': '01', 'B': '02', 'C': '03', 'D': '04', 'E': '05', 'F': '06',
                 'K': '11', 'L': '12', 'M': '13', 'N': '14', 'O': '15', 'P': '16',
                 'U': '21', 'V': '22', 'W': '23', 'X': '24', 'Y': '25', 'Z': '26',
                 '4':'31', '5':'32', '6':'33', '7':'34', '8':'35', '9':'36', '0':'3

key_list = list(Alphabet_List.keys())
val_list = list(Alphabet_List.values())


def convertText(msg,Ekey, N):
    li = list(msg)
    lii = [Alphabet_List[i] for i in li]
    if(len(lii)%2 != 0):
        lii.append('27')
    l1 = [int(lii[i]+lii[i+1]) for i in range(0,len(lii),2)]
    ctt = [str(pow(no,Ekey)%N).zfill(4) for no in l1]
    ct = ''.join(ctt)
    return ct

def decrypt(cipherText, Dkey, N):
    text = [str(pow(int(cipherText[i:i+4]),Dkey)%N).zfill(4) for i in range(0, len(

    L1 = []
    for i in text:
        L1.append(i[0:2])
        L1.append(i[2:4])

    L2 = []
    for i in L1:
        L2.append(key_list[val_list.index(i)])

    msg = ''.join(L2)
    return msg
```

In [ ]:
```
import socket,threading
localIP     = "192.168.1.113"
localPort   = 2000
bufferSize  = 1024

e,d,n = generatePrime()

msgFromServer = "Hello UDP Client!. My Pulbic key is: "+str(e)+" "+str(n)
flag=1
bytesToSend = str.encode(msgFromServer)

# Create a datagram socket
UDPServerSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)

# Bind to address and ip
UDPServerSocket.bind((localIP, localPort))
print("UDP server up and listening")

def recv(E, N):
    global flag
    while True:
        recieve= UDPServerSocket.recvfrom(bufferSize)
        msg = recieve[0].decode('utf-8')
        token = msg[-64:]
```

```
        cipherText = msg[:-64]
        plainText = decrypt(cipherText, d, n)

        tk = hashlib.md5(cipherText.encode())
        tk = tk.hexdigest().upper()
        tk1 = decrypt(token, E, N)
        if(tk.upper() == tk1.upper()):
            if plainText=='bye':
                flag=0
                break
            print (plainText)

def Send(a):
    global flag
    while True:
        if flag==0:
            print("Connection closed")
            break
        message=input("Enter your reply  ").upper()
        try:
            L1 = list(message.split())
            if(len(L1) != 3):
                raise Exception('Error')
        except:
            print('You must provide all three input in this sequence: Message , eKe
            sys.exit(0)

        cipherText = convertText(L1[0], int(L1[1]), int(L1[2]))
        m = hashlib.md5(cipherText.encode()).hexdigest().upper()
        N1 = convertText(m, d, n)
        ct = cipherText+N1
        UDPServerSocket.sendto(ct.encode('utf-8'),a)


E = 0
N = 0

# Listen for incoming datagrams
while(True):
    flag=1
    bytesAddressPair = UDPServerSocket.recvfrom(bufferSize)
    message = bytesAddressPair[0].decode('utf-8')
    l1 = list(message.split())
    E = int(l1[-2])
    N = int(l1[-1])
    print(E,N)
    address = bytesAddressPair[1]

    clientMsg = "Message from Client:{}".format(message)
    clientIP  = "Client IP Address:{}".format(address)

    print(clientMsg)
    print(clientIP)

    UDPServerSocket.sendto(bytesToSend, address)
    threading.Thread(target=recv, args=(E, N)).start()
    threading.Thread(target=Send(address)).start()
```

```
UDP server up and listening
7 4087
Message from Client:Hello UDP Server!. My public Key is: 7 4087
Client IP Address:('192.168.1.113', 56811)
HELLO
Enter your reply  nice 7 4087
```