Assignment 4

Network Security

Reshan Faraz
PhD19006

Problem Statement (Project 0) : Upload the document to the server (or perhaps some version of it) and expect to receive the same but with the current date and time stamped onto the document. And establish document existed at the date/time stamped, and that the document has not been modified .

1 - How and where do you get the correct GMT date and time? And how often?
Answer : I got the GMT date and time from local server machine using python library datetime and it easy to get
`datetime.datetime.now(datetime.timezone.utc).strftime("%Y-%m-%d %H:%M:%S")`

2 -Is the source reliable and the GMT date and time obtained in a secure manner?
Answer : Yes the source is reliable and GMT date and time obtained in a secure manner.

3- How do you ensure privacy, in that the server does not see/keep the original document?
Answer : After adding the time stamp to the document server delete the temporary file it used for signing the document , hence the privacy will be maintained . Server did not keep any copy of the document.

4- How do you share the document with others in a secure manner with the date/time preserved,and its integrity un-disturbed?
Answer : I hashed the document and attached the hashed to the end and encrypt the document with my private key and send it ,in that way receiver will verify whether the integrity of the document is maintained or not by decrypting and hashing the document and matching those hashed with the attached hashed.

5-Also ensure that the user (both the owner and anyone verifying the date/time) uses the correct “public-key” of the “GMT date/time stamping server”.
Answer : Both the parties have prior knowledge about the key pairs . For implementing this I used openssl to generate 2048 RSA key pairs.

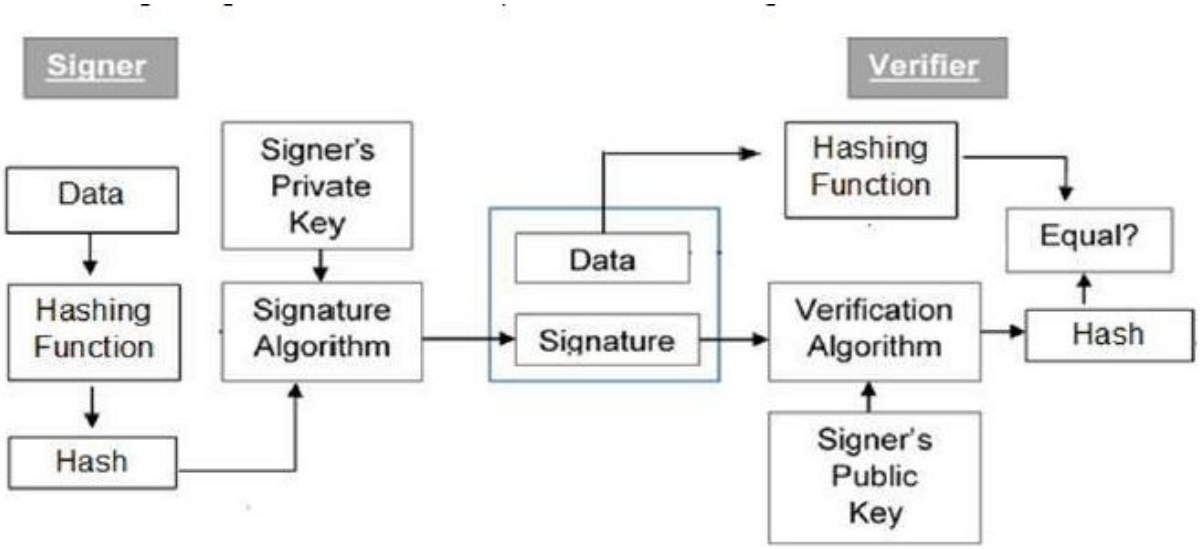
System Description and Implementation:

I program two files *client.py* and *server.py* . The client.py will contain the implementation of the user who wants to time stamp the document . while the server.py contains the implementation of the server who time stamps the document and performs digital

signature on the document. For connecting client and server I used socket programming. Following are the libraries used in python socket, os, item ,datetime ,haslib .

Following are the step wise implementation details with sample outputs from the running code.

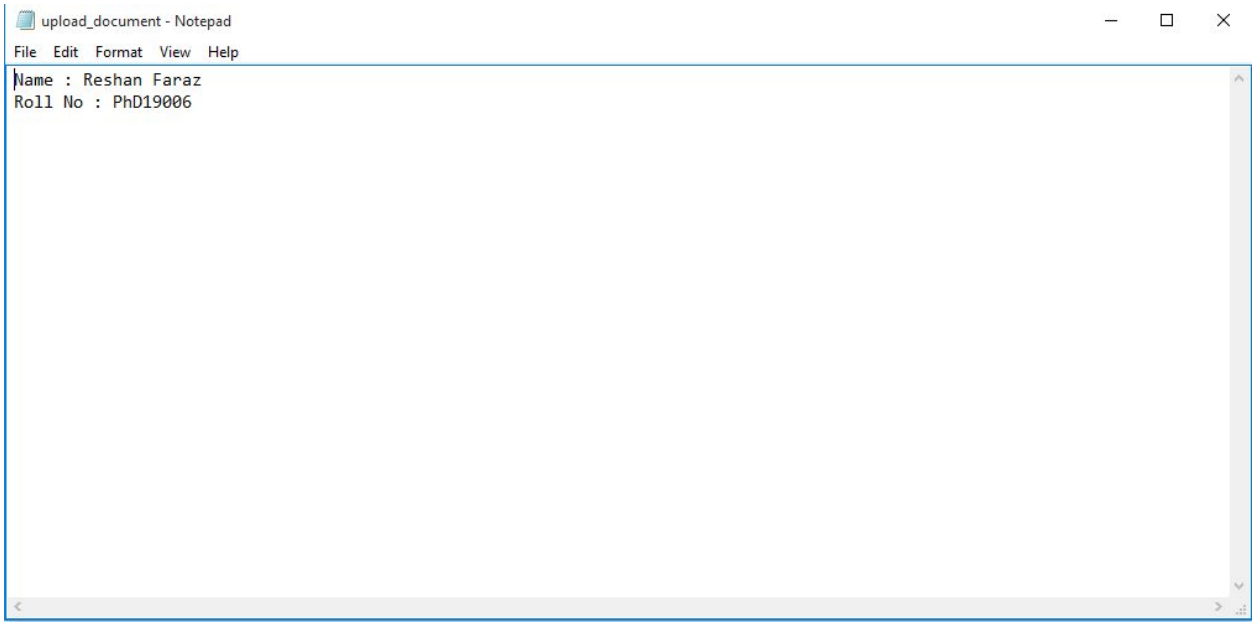
For the digital signature I used the following approach :



1- I generate the RSA key pair using openssl library , here is the output for the same :

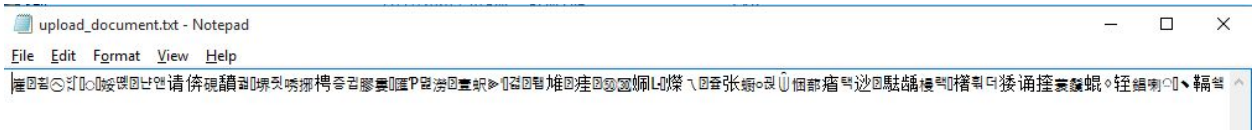
```
C:\Windows\system32\cmd.exe

C:\Users\reshanriq\Desktop>openssl genrsa -out key.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................................+++++
.......+++++
e is 65537 (0x010001)

C:\Users\reshanriq\Desktop>openssl rsa -in key.pem -outform PEM -pubout -out public.pem
writing RSA key

C:\Users\reshanriq\Desktop>
```

2- After that Client will send the document which he/she wants a timestamp. I used a simple .txt file *upload_document.txt* which is sent to the server. Following is the file :

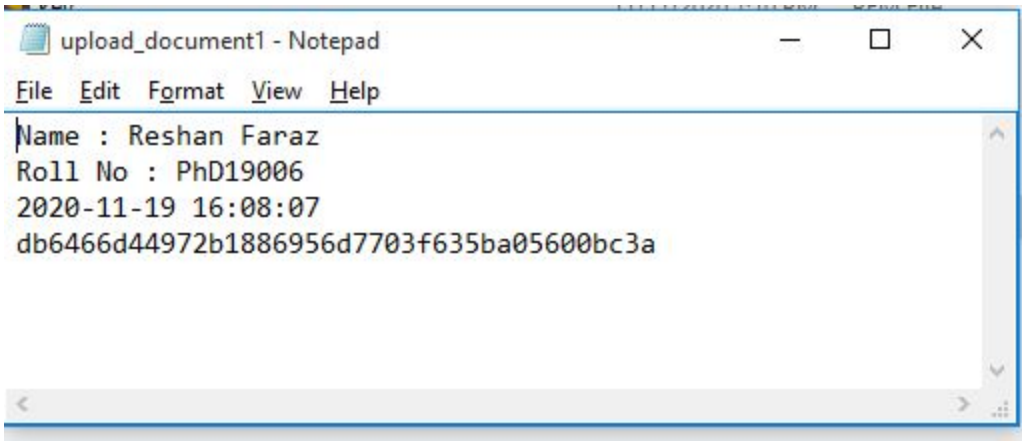


3 - After Receiving the file from the client ,server will attach the GMT date/time and hashed the document and attached hashed digest to the message and encrypt the document using openssl and sent back to the client.

4 - After receiving the document, the client first decrypt the file. Before decrypting the file looks like :

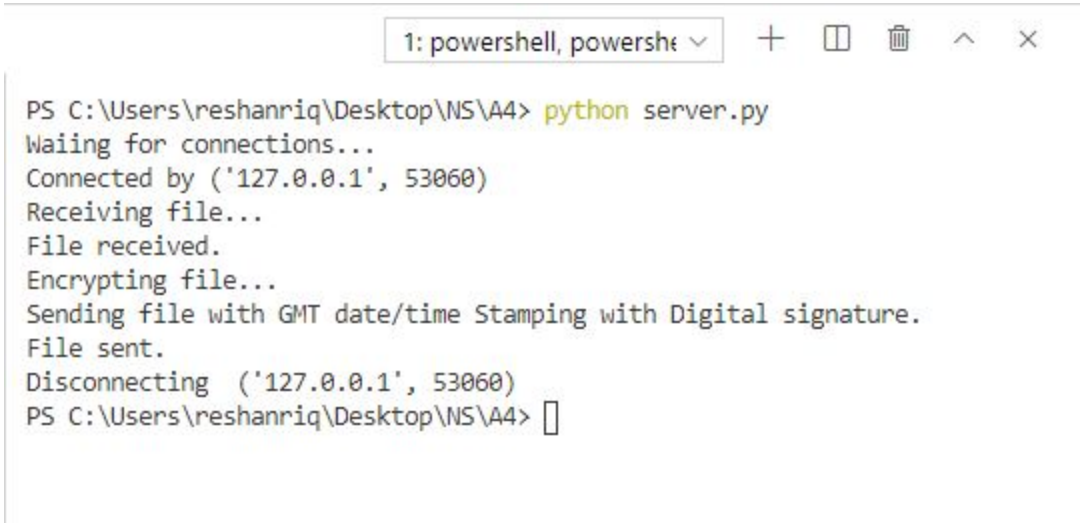


5 - After Decryption(I used openssl to decrypt using the key pair which was created in first step) the file looks like :

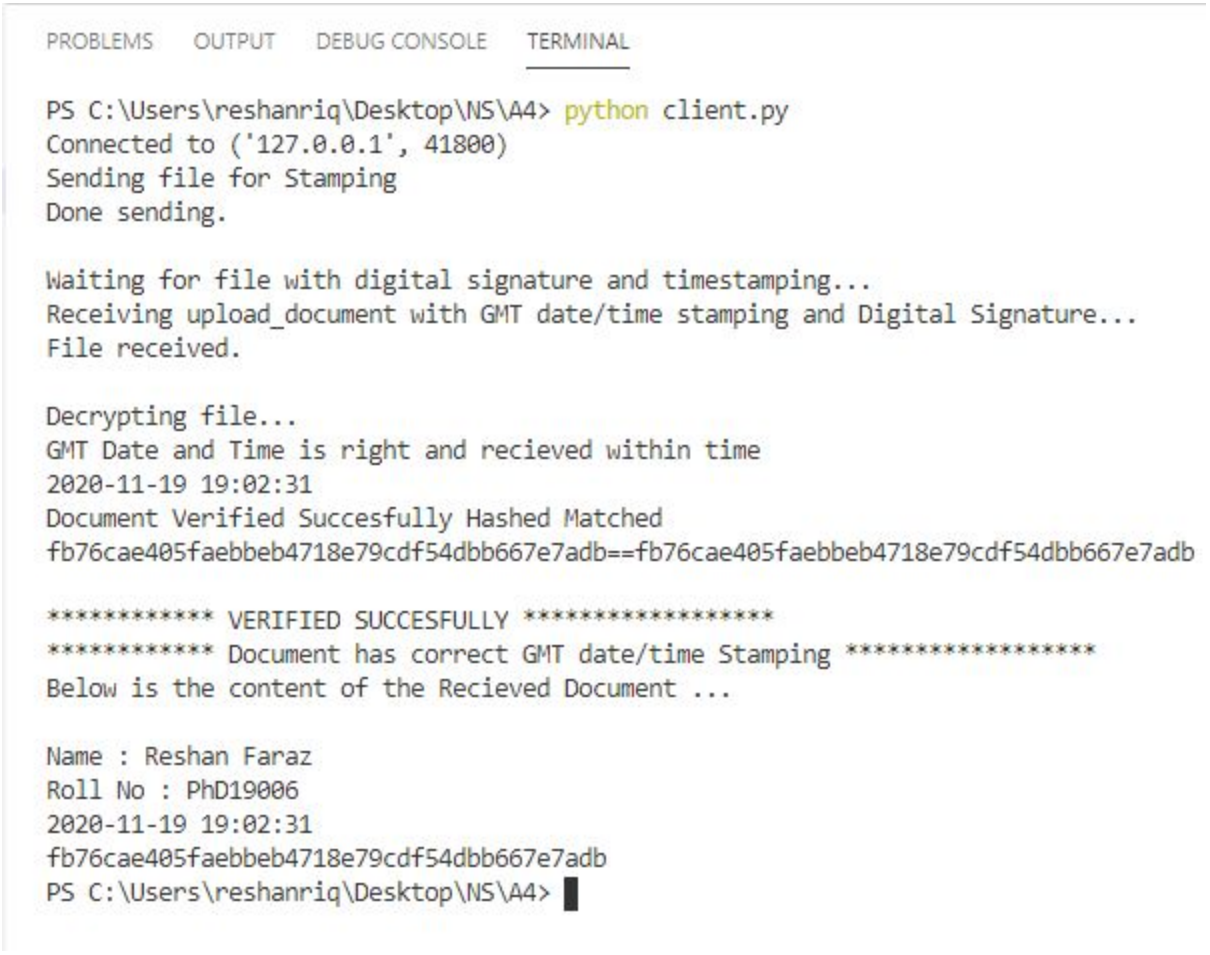


6- After that client gets the GMT date/time and verifies the digital signature. By matching the hashed attached and creating its own hashed digest. For the date and time if the current GMT matches with the attached GMT with tolerance of 4 minute then the GMT date and time is good otherwise it is outdated.

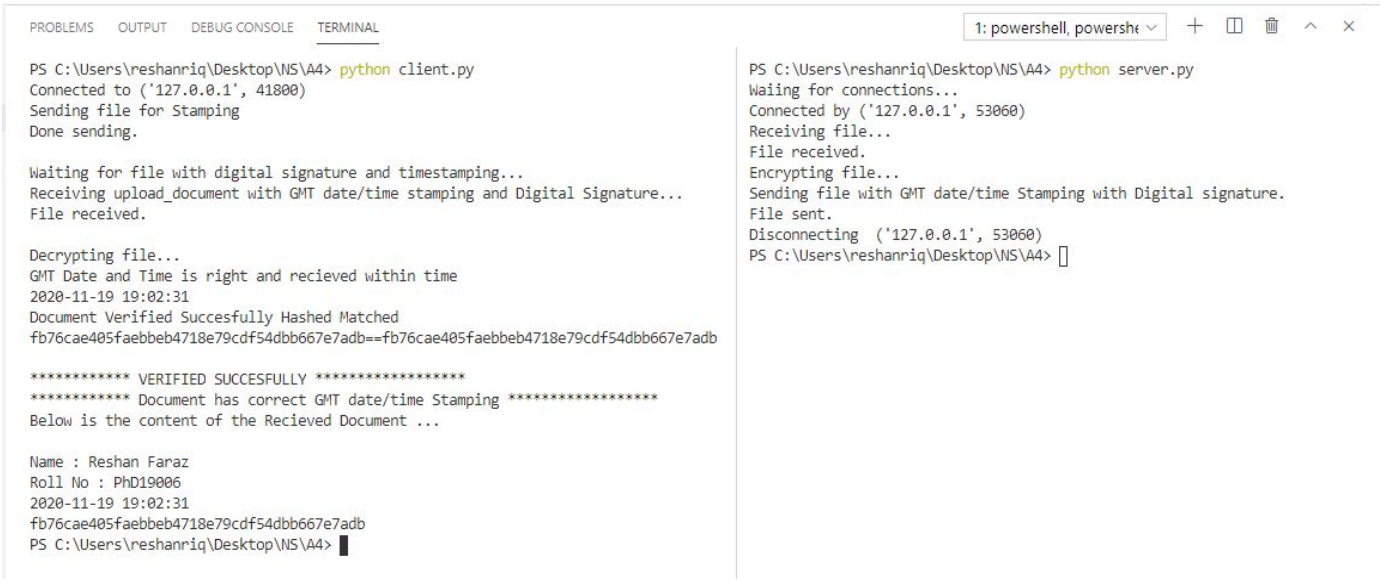
Here is the terminal output of *server.py*



Here is the terminal output of *client.py*



Below is the output for both :


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: powershell, powershe
PS C:\Users\reshanriq\Desktop\NS\A4> python client.py
Connected to ('127.0.0.1', 41800)
Sending file for Stamping
Done sending.
Waiting for file with digital signature and timestamping...
Receiving upload_document with GMT date/time stamping and Digital Signature...
File received.
Decrypting file...
GMT Date and Time is right and recieved within time
2020-11-19 19:02:31
Document Verified Succesfully Hashed Matched
fb76cae405faebbeb4718e79cdf54dbb667e7adb==fb76cae405faebbeb4718e79cdf54dbb667e7adb
********** VERIFIED SUCCESFULLY **********
********** Document has correct GMT date/time Stamping **********
Below is the content of the Recieved Document ...
Name : Reshan Faraz
Roll No : PhD19006
2020-11-19 19:02:31
fb76cae405faebbeb4718e79cdf54dbb667e7adb
PS C:\Users\reshanriq\Desktop\NS\A4>
PS C:\Users\reshanriq\Desktop\NS\A4> python server.py
Waiing for connections...
Connected by ('127.0.0.1', 53060)
Receiving file...
File received.
Encrypting file...
Sending file with GMT date/time Stamping with Digital signature.
File sent.
Disconnecting ('127.0.0.1', 53060)
PS C:\Users\reshanriq\Desktop\NS\A4>