An-Najah National University
Faculty of Engineering and IT

جامعة النجاح الوطنية
كلية الهندسة وتكنولوجيا المعلومات

Computer Networks 2 (10636455)

OpenSSL Assignment

Assignment 3 – Fall 2023

Dr. Raed Alqadi

Description:

You are asked to study and use the OpenSSL Command Line Tools to encrypt & decrypt and apply the SSL for Confidentiality, Authentication and Message Integrity that we learned in this course.
Before you start solving the assignment you must install **OpenSSL** as shown here and also explained in class.

OpenSSL Installation and Practice

Download and learn about the OpenSSL Command Line Tools.

1. Download the OpenSSL for Windows 10 64-bit or 32-bit depending on your machine. It is preferred to use the 64-bit from the website:
   http://slproweb.com/products/Win32OpenSSL.html
   Note: The site has both 64 bit and 32-bit versions even though it has the name of Win32OpenSSL.html

   Simply execute *sites.bat* and it will open the download and tutorial locations

2. Read about the commands in the following websites, you can also search for other information. Mainly the following site:
   https://www.keycdn.com/blog/openssl-tutorial
   //there is one example here
   https://www.cloudinsidr.com/content/how-to-install-the-most-recent-version-of-openssl-on-windows-10-in-64-b

3. I have provided a good starting point for you in the class lecture about OpenSSL. See the second half of the recorded lecture Lec75min_Security_Part8_Dec18_Fall2023

Part 1: Practice

In this part, first read the tutorial https://www.keycdn.com/blog/openssl-tutorial
and follow the steps there, then put all the commands and write the batch file ***part1.bat*** that does the followings: (Do not submit anything in this part, this is just practice)

1. Encrypt the file **test.txt** by using Symmetric Shared Key Encryption. Use the command:
   openssl enc with the options **-aes-256-cbc -pbkdf2 , and** -pass option.
   use YourName_YourPartnerName
   Name the Encrypted File ***encrypted.bin***
2. Decrypt the encrypted file and make sure it is the same as the original file.
   To compare two files, use can use the ***FC.exe /B file1 file2***
3. Generate a public Key by using the ***openssl genrsa*** command .This is actually a private and public key pair
4. Extract the public key from the one generated in 3.
5. Encrypt the small file ***abc.txt*** by using the public key. Make sure the -pubkin option
   openssl pkeyutl -encrypt -in abc.txt -pubin …. continue the command
6. Decrypt the encrypt file in 5 by using the private key
   pkeyutl -decrypt -in abc.enc …. continue the command
7. Generate the hash for the file test.txt by using sha256
   openssl dgst -sha256 -binary ….. continue the commands
8. Sign the hash by encrypting it with the private key
   openssl pkeyutl -sign …. continue the commands9.
9. Verify the hash. What happens here is that the hash is first decoded and compared to original hash.
   openssl pkeyutl -verify -sigfile ..continue the commands

   If everything is correct you should get **verified successfuly**
   Note: you need to use the -binary for 7, 8, 9 to work correctly

Part 2:

You are required to write and execute OpenSSL Commands to encrypt/decrypt any file, you will also generate the Hash, signed Hash. We will use both public key and shared key encryption similar to the Session Encryption Method explained in class. To Make it easier I divided the assignment into three Tasks .

Task 1: (Generate Self Signed Certificte and Private/Public Key Pair):

You are asked to write the Batch file "***gen_cpp.bat***"to do the followings

gen_cpp <yourname>
The batch file gen-cpp should take one argument which is your name.
For example, if your name is Sami
> gen_cpp Sami

1. Generate Private key and Self-signed Certificate for you.
   You need to use the command ***openssl req*** with correct options to generate the private key, see the example in 2 above. You must use your actual Name, put also in the company your University Registration Number, also use Your Father and Surname in the company and Branch ...etc. You must put Registration number also.

Generate certificate with the -x509 option in the ***openssl req*** command
Store your Certificate and Private Key (Actually Private/pulick pair) in the Files
<YourName>Certificate.pem and <YourName>PrivateKey.pem , respectively

2. Extract your Public Key from the certificate and store it in the file
*<YourName>*PublicKey.pem

3. Launch Notepad++ or notepad to display the three generated files in 1 and 2.

Your partner must execute this task also and should generate this/her certificate also. Exchange the certificates. So each Both partners should have both certificates

Task 2: (Encrypt, hash, sign, using Public and Shared keys)

You are asked to write the Batch file "***enc_file.bat***"to do the followings

The *Sender* is You(*YourName*), and *Receiver* is the *PartnerName*

> enc_file <Receiver> <filenme> <Sender>

The batch file enc_file should take two arguments which is your partner name and the name of the file to be encrypted. For example, to encrypt TestFile.txt if your partner Name is ***Jamal*** *and you are* *Ahmad*

> enc_file ***Jamal*** *TestFile.txt* ***Ahmad***

Here **You** are **Ahmad** the **Sender** and Your ***Partne*r Jamal** is the ***Receiver***

1. Extract your partner public key from *<Receiver>Certificate.pem*
and name it *<Receiver>*PublicKey.pem

2. Generate a shared key, Use the command *openssl rand.* Store the shared key in a file and name it *Task2sharedkey.bin.* Do not exceed 116 bytes. You can use 64 bytes length. You will use this key to encrypt any file you want.
3. Now use your partner's public (*Receiver*) key to encrypt the *Task2sharedkey.bin* and call it *Task2sharedkey.bin.enc. This should be sent to your partner(Receiver). This is the encrypted shared key.*

4. Now encrypt the Test File given to you which is the assignment pdf file (hw3f23.pdf) with the shared key that you generated.
Use the ***Openssl enc*** command and the aes-192-cbc algorithm
Your *enc_file.bat* should be able to encrypt any file as this is a parameter passed as shown
Call the encrypted file by the name <filename>.enc

5. Compute the Hash for the file. Use the openSSL dgst … // here use sha512. Generate the sha in Binary format and name it *Task2.sha512*
6. Sign the *Task2.sha512* with your private key(sender(and name *Task2sign.sha512*

This is why the third parameter is used so that you use <Sender>PrivateKey.pem. Recall Sender is YourName

7. Verify the signed hash file with your public key. If verified your code should print Signature Verified Successfully
8. **Send the encrypt file ,the encrypted share key and the signed hash to your partner. Your partner should decrypt the file and verify the hash.**

Task 3: (Decryt the file and verify the Hash)

You are asked to write the Batch file "***dec_file.bat***"to do the followings
So in Task 2 *Ahmad* was the *Sender* and Jamal is the *Receiver*.
Here you are the *Receiver(Jamal)* and you want to decrypt the file sent by the *Sender(Ahmad)*

dec_file <Receiver> <Sender> <file name>

The batch file dec_file should take 3 arguments which is your partner name(Sender) and the name of the file to be encrypted. For example, to derypt TestFile.txt.enc *if your partner Name is Ahmad and you are Jamal*
> dec_file Jamal Ahmad TestFile.txt
// Here it is assumed that the encrypted file name is TestFile.txt.enc and the orignal file is TestFile.txt. This is done so that we compare the decrypted file and the original file.

1. Extract your partner's public key from his/her certificate. This is needed to decrypt the signed hash
2. Decrypt the *Task2sharedkey.bin.enc* and name it *dec_sharedkey.bin* . Compare it to the orignal key. They should be the same. You should decrypt by using your private key
3. Use the decrypted shared key to decrypt the encrypted file and name it *<filename>.dec*
   Compare the original key to the decrypted one they should be the same
4. Compute the hash of the decrypted file.
5. Compare the hash with signed one. You can use the *openssl pkeyutl -verify -sigfile …In your work you will need to compare files, this may help:*

   Important Note:
   Use the windows command line FC to do compare the two files
   Example: To compare file1 to file 2 use the command line: ***FC.exe /B file1 file2***
   You can use **FC** or **FC.exe**. They are the same thing, both of them invoke FC.exe.

Submit:

Create a Folder and put the following in it.

- The 3 batch files described in Tasks 1, 2, and 3
- The Encrypted Test File is the hw3f23.pdf
- The signed Hash file. (Must be signed)
- Your Certificate and Your Partner certificate
- The shared keys, signed hash ..etc. Everything you generated.
- Put all of these files in a Folder and compress them rar file.