Liliane Owens
U00846594
Undergraduate
Meilin Liu
2/27/2025

Lab/Project 1
Using Cryptographic Tools (60 Points)

- **I did the project by myself.**

1. The objective

   The objective of this lab is to familiarize students with basic cryptographic tools. Cryptographic capabilities and the tools that implement these capabilities are some of the most important tools we can use to secure data. Students will gain hands on experience creating and using encryption, hash, and generating the public/private keys.

2. Submission
   A team can have up to 3 students. All students in the same team will receive the same grade.

   a. Each team only submits one report or a copy of reports/answers/files by the representative of the team. (See the section of tasks).
   b. **Each team member needs to submit the list of names of all team members**.

3. Tools

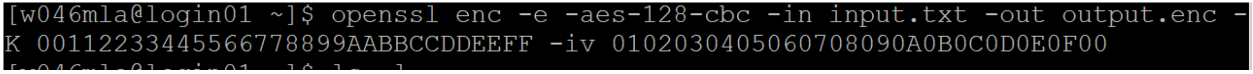
We will use the linux built-in tools, OpenSSL, for this project. In order to use the OpenSSL tool installed under the cs Unix server, fry.cs.wright.edu, you need to connect to this Unix server remotely using a secure shell client, such as putty.

If you want to connect to this server remotely off campus, you need to install VPN on your computer first (You can download the VPN from WSU, https://www.wright.edu/information-technology/virtual-private-network-vpn.)

4. Tasks (Each Task for 10 Points)

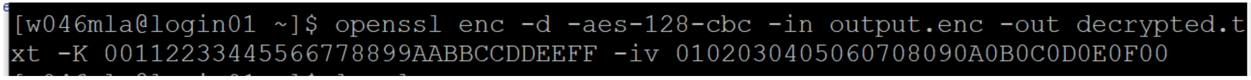
   Create a simple txt file, named as “input.txt”. This file contains a few letters such as” hello world!”.

a. Task-1: Encrypt input.txt using AES in cbc mode with 128 bits. The output file is "output.enc". Please specify the encryption key and the IV. Please include the complete screenshot of the execution result of the command in your report. The screenshots should include your username, current directory, the openssl command, and the complete execution results.



```
[w046mla@login01 ~]$ openssl enc -e -aes-128-cbc -in input.txt -out output.enc -
K 00112233445566778899AABBCCDDEEFF -iv 0102030405060708090A0B0C0D0E0F00
```

b. Task-2: Decrypt "output.enc" and name the output file as "decrypted.txt". Please include the complete screenshots of the execution results of the commands in your report. The screenshot should include your username, current directory, the openssl commands, and the complete execution results.



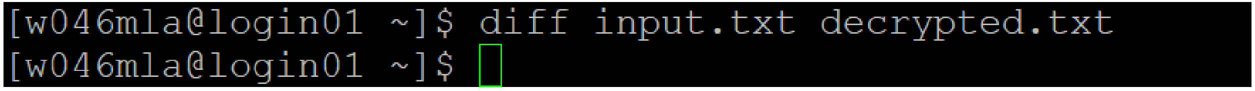
```
[w046mla@login01 ~]$ openssl enc -d -aes-128-cbc -in output.enc -out decrypted.t
xt -K 00112233445566778899AABBCCDDEEFF -iv 0102030405060708090A0B0C0D0E0F00
```

c. Task-3 use "diff" to verify the decrypted file is identical as the original plaintext file under the linux environment. The command format is as follows:

diff input.txt decrypted.txt

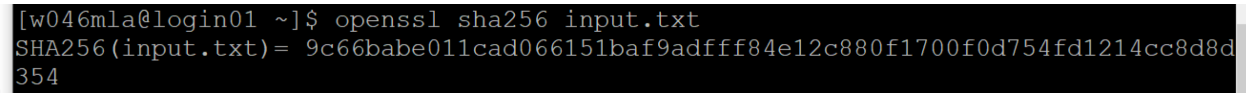
The output should be empty when the two files are identical. Again, please include the complete screenshots of the execution result of the command in your report.

Please include the complete screenshot of the execution result of the command in your report. The screenshot should include your username, current directory, the command, and the complete execution result.



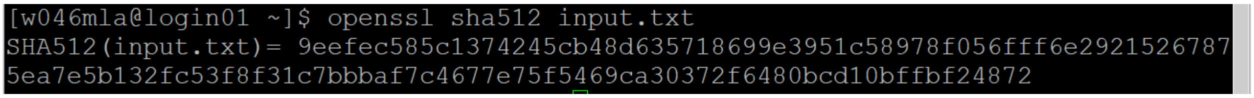
```
[w046mla@login01 ~]$ diff input.txt decrypted.txt
[w046mla@login01 ~]$
```

d. Task-4: Using openssl to generate the cryptographic hash of the input file using SHA256. Please include the complete screenshot of the execution result of the command in your report. The screenshots should include your username, current directory, the command, and the complete execution results.



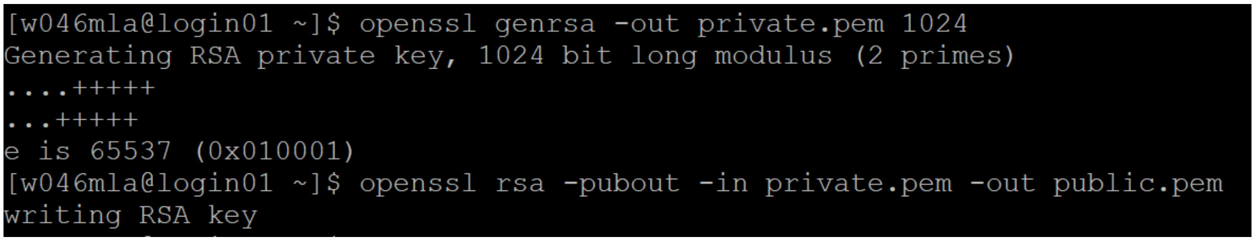
```
[w046mla@login01 ~]$ openssl sha256 input.txt
SHA256(input.txt)= 9c66babe011cad066151baf9adfff84e12c880f1700f0d754fd1214cc8d8d
354
```

e. Task-5: Using openssl to generate the cryptographic hash of the input file using SHA512. Please include the complete screenshot of the execution result of the command in your report. The screenshot should include your username, current directory, the command, and the complete execution results.



```
[w046mla@login01 ~]$ openssl sha512 input.txt
SHA512(input.txt)= 9eefec585c1374245cb48d635718699e3951c58978f056fff6e2921526787
5ea7e5b132fc53f8f31c7bbbaf7c4677e75f5469ca30372f6480bcd10bffbf24872
```

f. Task-6: Using openssl to generate a pair of public and private key pair with 1024 bits. Please include the complete screenshot of the execution result of the command in your report. The screenshots should include your username, current directory, and the complete execution results.



```
[w046mla@login01 ~]$ openssl genrsa -out private.pem 1024
Generating RSA private key, 1024 bit long modulus (2 primes)
....+++++
...+++++
e is 65537 (0x010001)
[w046mla@login01 ~]$ openssl rsa -pubout -in private.pem -out public.pem
writing RSA key
```

How to use Openssl on fry.cs.wright.edu

1. Connect to fry.cs.wright.edu using a VPN. In order to use the openssl environment installed under the cs unix server, fry.cs.wright.edu. If you want to connect to this server remotely off campus, you need to install VPN on your computer first.

2. Use putty or other secure shell clients to connect to fry.cs.wright.edu using your campus id (for example, w123abc) and password.

3. Using the secure file transfer client (WinSCP, you can download it online, and install it on your personal computer) to transfer your files between your local computer and fry.cs.wright.edu.