Self Introduction — Assignment 1

Anol Kurian Vadakkeparampil

August 2022

1 Information

I used official documentation of Openssl and the openssl-wiki to navigate the libraries in openssl. I also used documentation and online resources to deal with socket programming with c++ libraries : arpa, netinet, socket.

I was able to achieve most functionality required by the assignment. Unfortunately I was not able to figure out how to send IV, Additional data and Tag along with the ciphertext. My approach was to append these data with the ciphertext and somehow provide a way to authenticate before passing this data. Hence for this assignment I have used static IV and static Additional data, and am reading the tag from a file stored locally, which gets updated with the new tag after every encryption call.

2 Rubrics Points

2.1 With Password ”Hello”, Print the Hexadecimal value of the symmetric key derived by PBKDF2

I Have used OpenSSL library function PKCS5-PBKDF2-HMAC with ”Hello” as the password and SodiumChloride as the salt. The resultant Hex value is printed on the terminal as demonstrated in figure 1.

Figure 1: Hex Value of Key

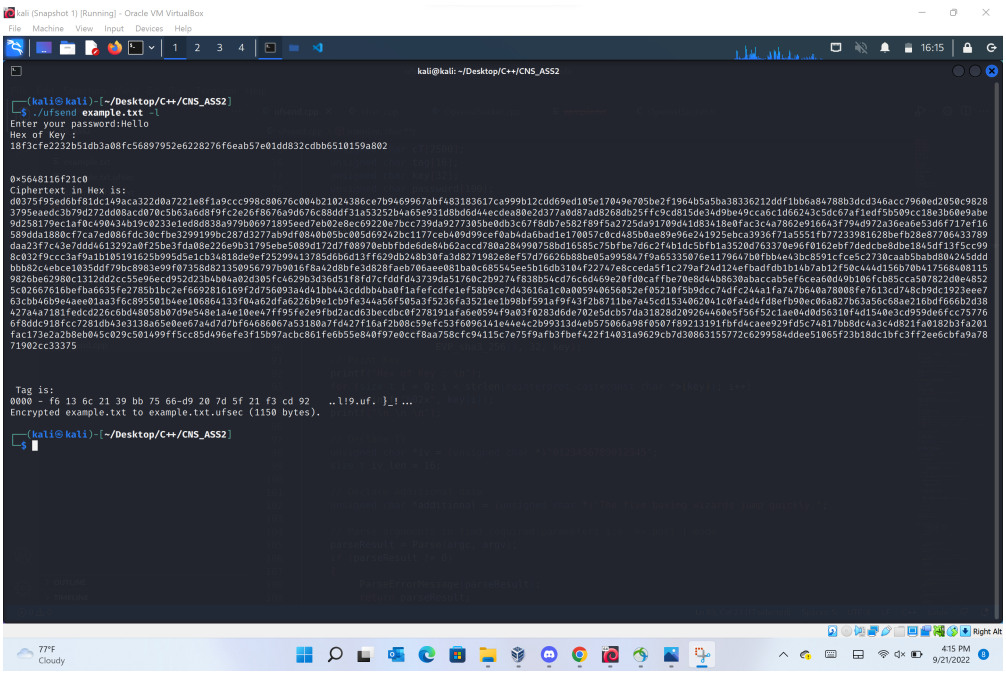

Figure 2: Hex Value of Encrypted File

2.2 Encrypt Example.txt, print Hexadecimal value of encrypted file on screen

The key generated was fed to gcm-encrypt function which used openssl libraries to encrypt plain text and return a cipher text, which was displayed on the terminal as seen in figure 2.

2.3 Send encrypted file to ufrec, display receipt of file and hexadecimal value of encrypted file

The file was sent to ufrec via -d mode and hexadecimal value of received file is printed as shown in figure 3

2.4 Decrypt File, display contents on screen

The encrypted file was decrypted at the receiver end and decrypted hex value is printed on the terminal as seen in figure 4

2.5 Manually modify ciphertext generated in local mode, attemot to decrypt it locally show messages

The encrypted file was was manipulated and a decryption failed message was shown, as seen in figure 5.

2.6 Show graceful exit codes

Exit codes are being handled gracefully, as shown in the example in figure 6.

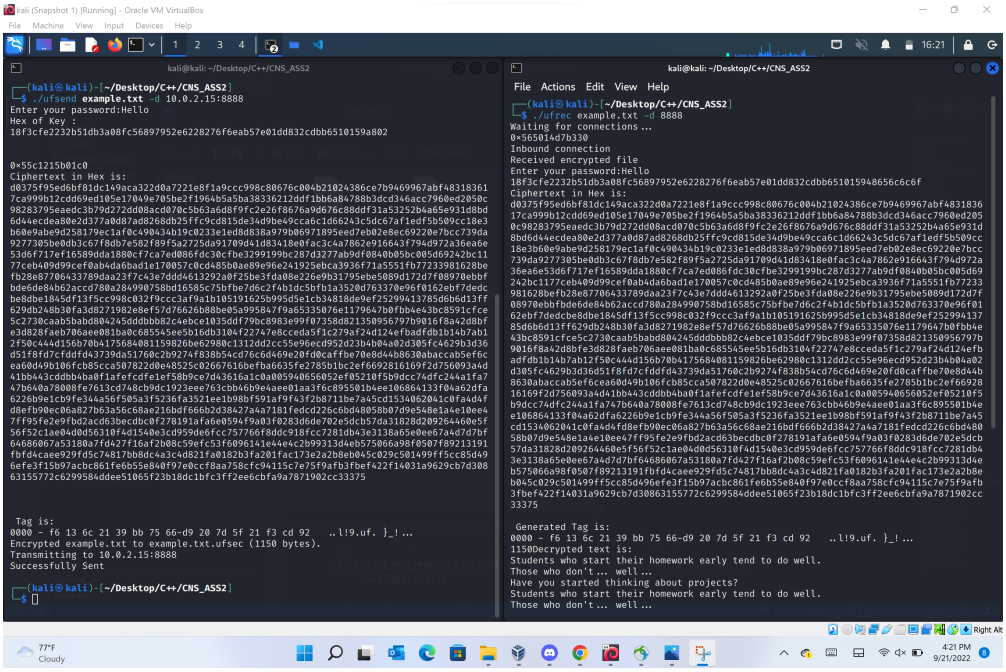

Figure 3: Send encrypted file to ufrec

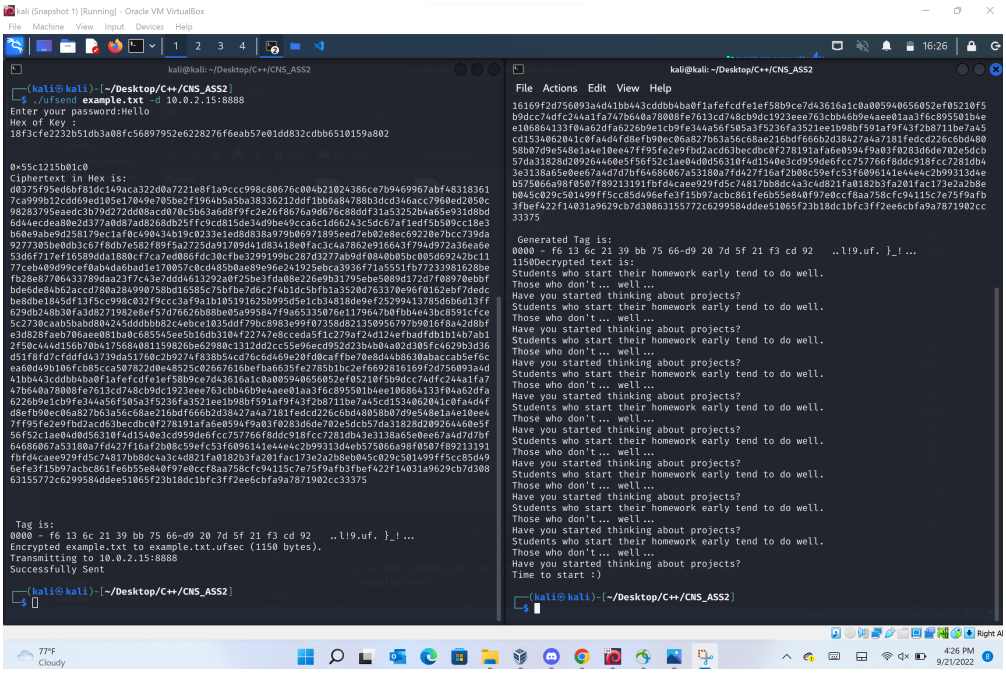

Figure 4: Decrypt at Ufrec

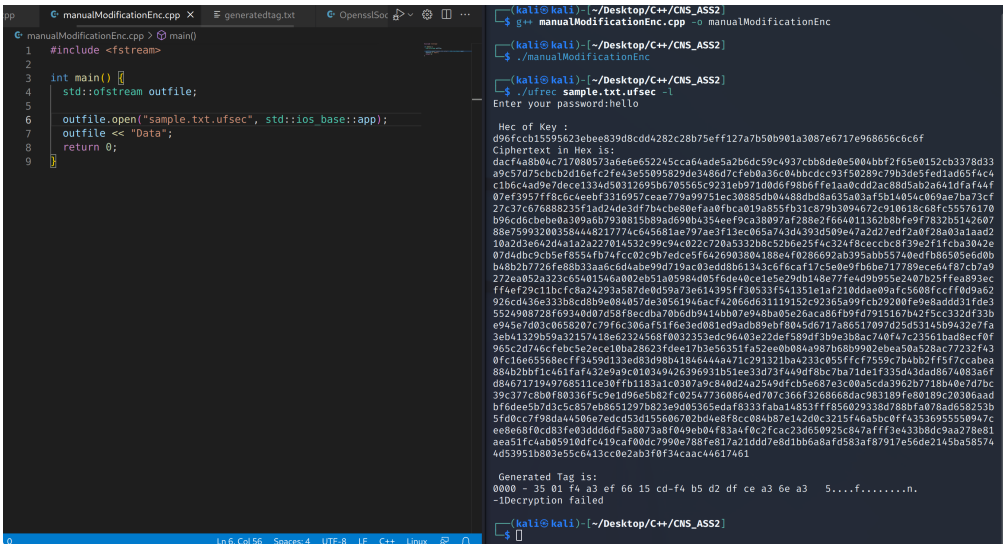

Figure 5: Decrypt at Ufrec

Figure 6: Decrypt at Ufrec