

Self Introduction — Assignment 1

Anol Kurian Vadakkeparampil

August 2022

1 Information

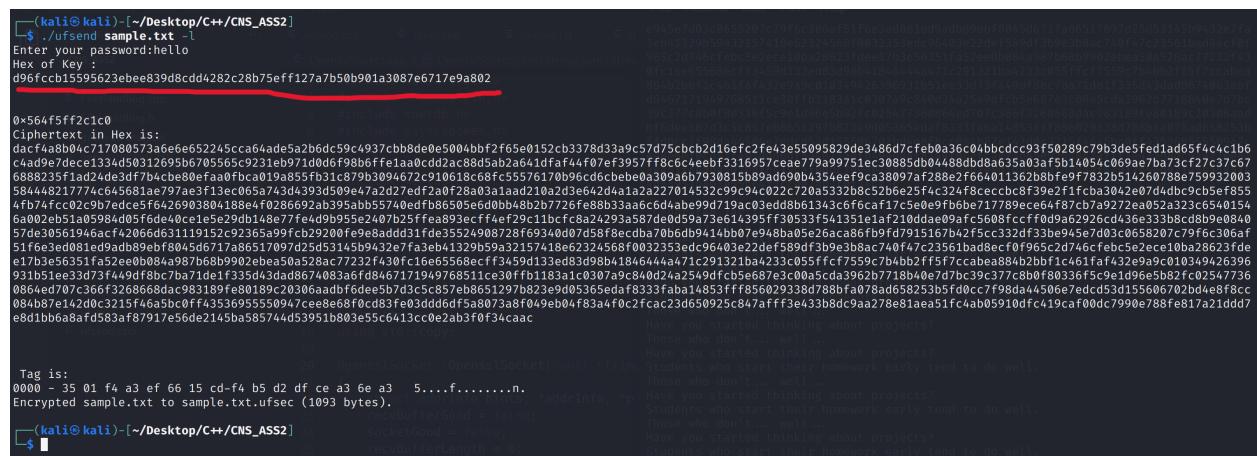
I used official documentation of Openssl and the openssl-wiki to navigate the libraries in openssl. I also used documentation and online resources to deal with socket programming with c++ libraries : arpa, netinet, socket.

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.



The terminal window shows the following output:

```
(kali㉿kali)-[~/Desktop/C++/CNS_ASS2]
$ ./ufsend sample.txt
Enter your password:hello
Hex Key :
d96fccb15595623ebee839d8cd4282c28b75eff127a7b50b901a3087e6717e9a802
0x564f5fff2c1c0
Ciphertext in Hex is:
dacf4a8bd04c717080573a6e6e652245cca64ade5a2b6dc59c4937cbb8de0e5004bf7f65e0152cb3378d33a9c57d75cbc2d16efc2fe43e55095829de3486d7cfeb0a36c04bbcddc93f50289c79b3de5fed1ad65f4c4c1b6
c4ad9e7dec133ad50312695b6705565c9231eb971d0d6f98bf6fela0@ccdd2ac88d5ab2a641dfaf4+f07e3f957ff8c6c4eeb779a99751ec30885db4488dbd8a635a03af5b14054c069ac7ba73cf27c37c67
6888235f1ad24de3df7b4ce80eefaa0fbc019a855fb31c879b3094672c910618c68Fc55576170b96c6dbebe03a0946b7930815b89ad6904354eeff9ca38097f288e2f664011362b8bFeF7832b514260788e75932003
584448217774c645681a797aae3f13ec065a743d4393d509e47a2d27edf2a0f28a03a1aad210a2d3e642d4a1a2a227014532c99c94c022c720a5332bcb52b6e25f4c324f8ecccbc8f39e2f1fcba3042e07d4dbc9cb5eF855
4f7b4fc029b7edce5f6426903804188eaf0286692aab395ab55740edfb86505ed0db48b2b7726fe8833aa6c6d4abe99d719a03edddbb1343c6f6cfa17c5e09fb8e717789ce64f87ch7a9272ea052a323c6540154
6a002e51a51a05984d05f5dede0ce1e5e29db143e77fe4dd9b55e2407h25f4ea8934ff4e4f29c11bfcfa24293a587d0e059a73a14395ff30533f5a1351e1af210dd499af5608fccfffd9a62926c4d36e3338b8db9e0840
57de3056194a6a4f420664611119152c92365a99fc29200fe9e8a2addd11fde35524796328f69340d07d58f8e7b70bd9d9414b07e948ba05e26ac8e8f97915167b72f5cc332d3f3b945e7d03c0658207c79f6c306af
51f6e3ed081e1e94d989e0f8045d6717a86517097d25d53145b9432e7fa1e841329595a32157418e62324568f0032353edc96e03e22deff589ff3b9e3b8a7c740f47c23561bad8eef0f965c2d746f6ebc5e2e10ba28623fde
e17b3e56351f452e00384a987b68b992ebe50a528a77232fr30fc16ee5568eefc459d133ed83d9841846444a71c291321ba233c0551f7f75597b4abbff5f7ccabae884b2bf1c461f1af32e9a9c010349426396
931b51ee33d73f44d9f78bc/ba71de1f335d43dad8674083aef846717194c7768511c303f0fb1183a1c0307a9c84d9425459d9c68673e300a5cda3962b7718b8d0e7d7bc39c37778b0f80336f59e1d96e5b82Fc02547736
086487d0712c366f32258668d8d981189e80189c20306ad9f6dee5b7d3c5c857e8b651297b782c9d05365eda8333Faba14853ff856029338d788bfa078ad658253bf0d0ccf798da445056e7edc153d155606702bd4e8f8cc
084b87e142d03215f46a5c0ff4353695555094/ceee8e68f0cd333f3d66ff5a8073a8f049e04f83a4fcac23d50925c847affff3e433b8d9aa278e81aea51fc4ab05910dfc419cfa00dc7990e788Fe817a21ddd7
e8d1bb6a8afdf583a8f87917e56de2145ba58544d53951b803e55c6413cc0e2ab3f0f34caac

Tag is:
0000 - 35 01 f4 a3 ef 66 15 cd-f4 b5 d2 df ce a3 6e a3 5.....f.....n.

Encrypted sample.txt to sample.txt.ufsec (1093 bytes).
```

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, attempt to decrypt it locally show messages

The encrypted file 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.

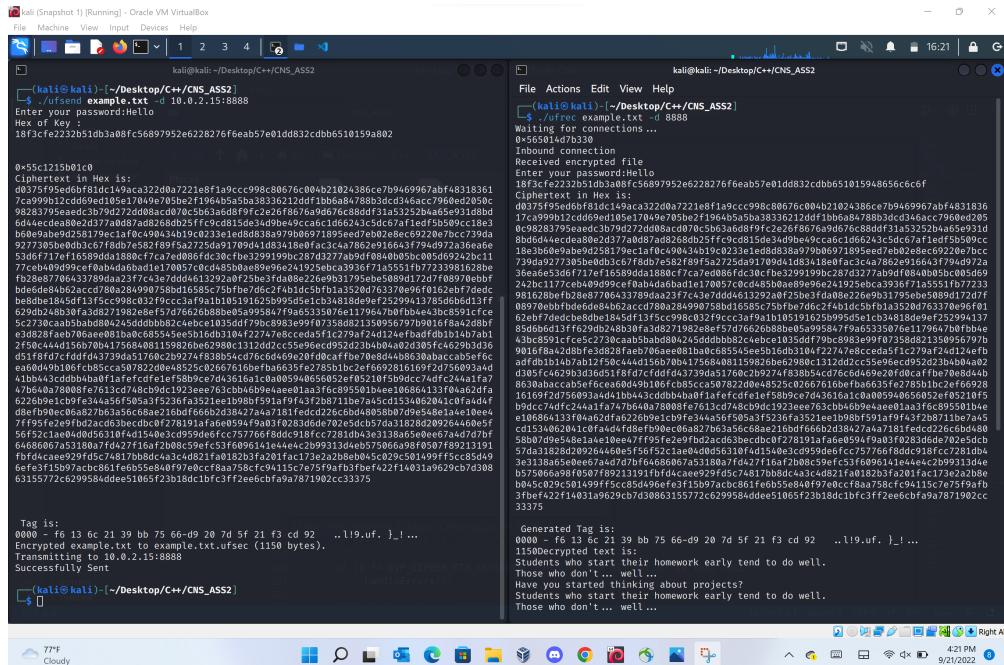


Figure 3: Send encrypted file to ufreco

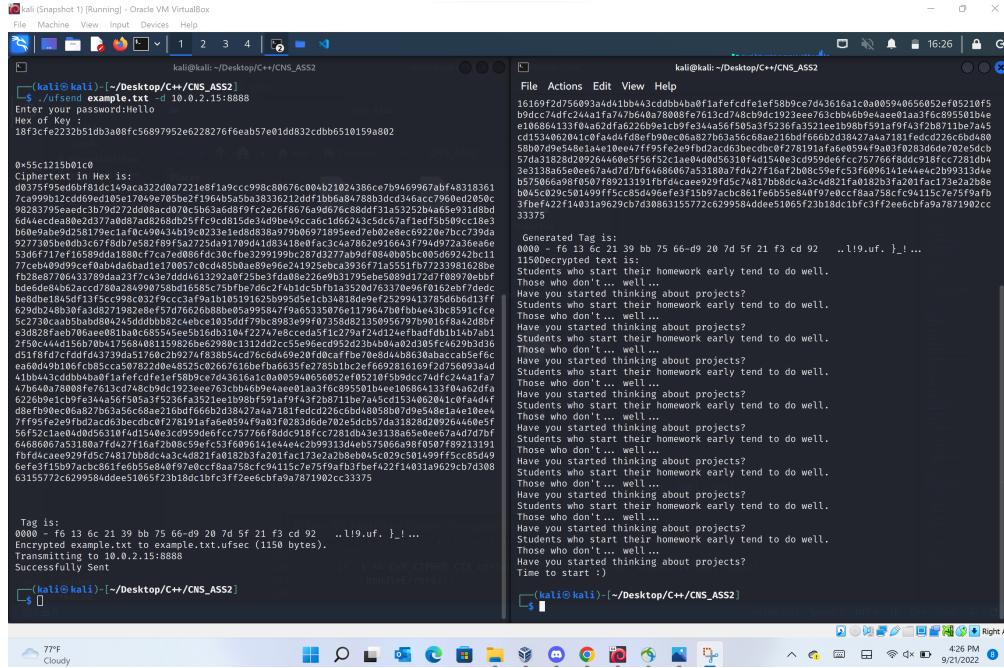


Figure 4: Decrypt at Ufred

Figure 5: Decrypt at Ufreedom

```
kali㉿kali:~/Desktop/C++/CNS_ASS2
```

```
File Actions Edit View Help
```

```
[-(kali㉿kali)-[~/Desktop/C++/CNS_ASS2]
```

```
└─$ ./ufsend sample.txt -l
```

```
Enter your password:hello
```

```
Hex of Key :
```

```
18f3cfef232b51bd3a0fffc56897952e6228276f6eab57e01dd832cd6b510159a802
```

```
Error: O/P file exists
```

```
[-(kali㉿kali)-[~/Desktop/C++/CNS_ASS2]
```

```
└─$
```

Figure 6: Decrypt at Ufreedom