**CS5460 Assignment #1 Crypto Lab – Secret Key Encryption**

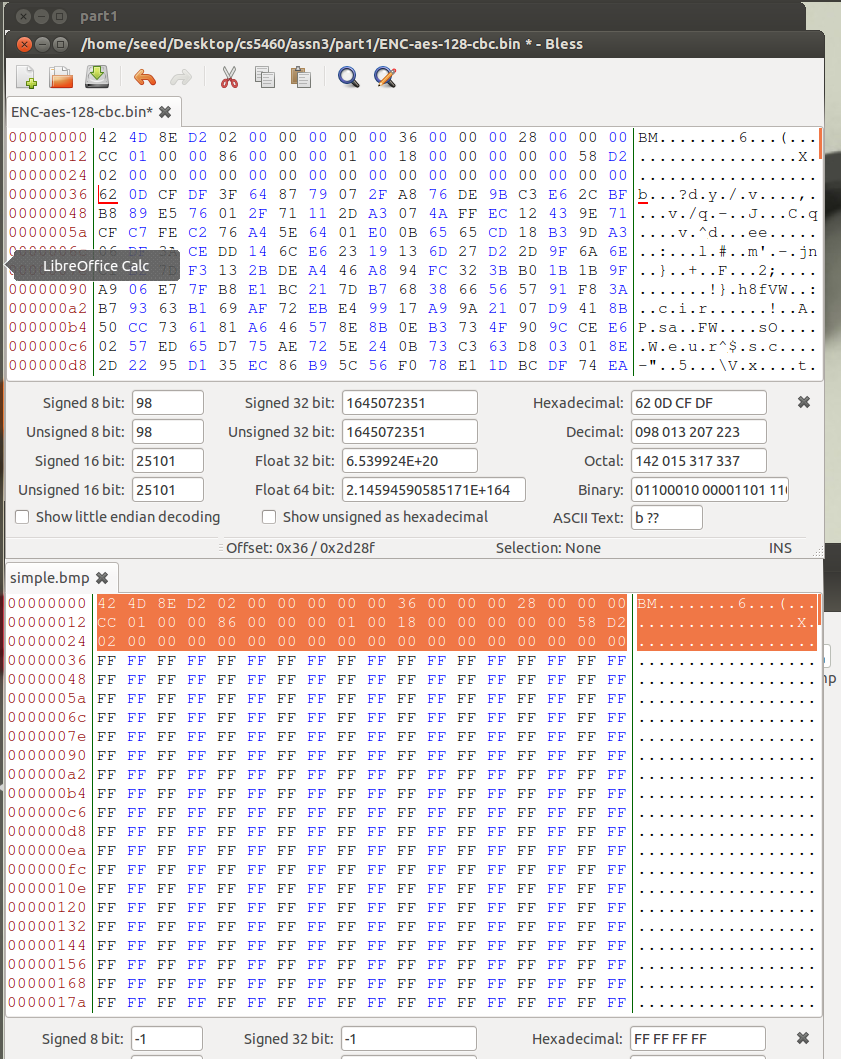
Austin Derbique A01967241 Fall 2017

https://github.com/aderbique/cs5460/tree/master/assn3

**3.1 Task 1: Encryption using different ciphers and modes**

During this task, I spent time getting familiarized with openssl and how to encrypt and decrypt files. I practiced on the README file that is found in the openssl directory on the seed image. These are a couple of the commands I used for this task. The files can be found in the Task 1 directory of the github repo.

seed@ubuntu:~/openssl-1.0.1$ sudo openssl enc -camellia-128-ecb -e -in README -out README\_rc2-128-cfb -K 00112233445566778889aabbccddeeff -iv 0102030405060708

seed@ubuntu:~/openssl-1.0.1$ sudo openssl enc -camellia-128-ecb -e -in README -out README\_camellia-128-ecb -K 00112233445566778889aabbccddeeff -iv 0102030405060708

**3.2 Task 2: Encryption Mode – ECB vs. CBC**

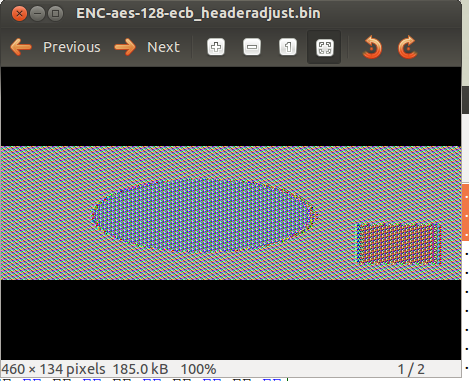
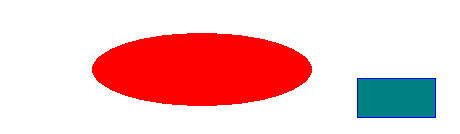
1. This task consisted of encrypting a given simple.bmp image file using ECB and CBC ciphers. To do this, the following commands are used:

seed@ubuntu:~/Desktop/cs5460/assn3/part1$ sudo openssl enc -aes-128-ecb -in simple.bmp -out ENC-aes-128-ecb.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708

seed@ubuntu:~/Desktop/cs5460/assn3/part1$ sudo openssl enc -aes-128-cbc -in simple.bmp -out ENC-aes-128-ccb.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708

The resulting files are encrypted binary files that must have the headers changed for them to be recognized as an image file. To do this, the first 54 bytes of the files are replaced with the first 54 bytes of the original image file using Bless. These files are available in the Task 2 section of the github repo.

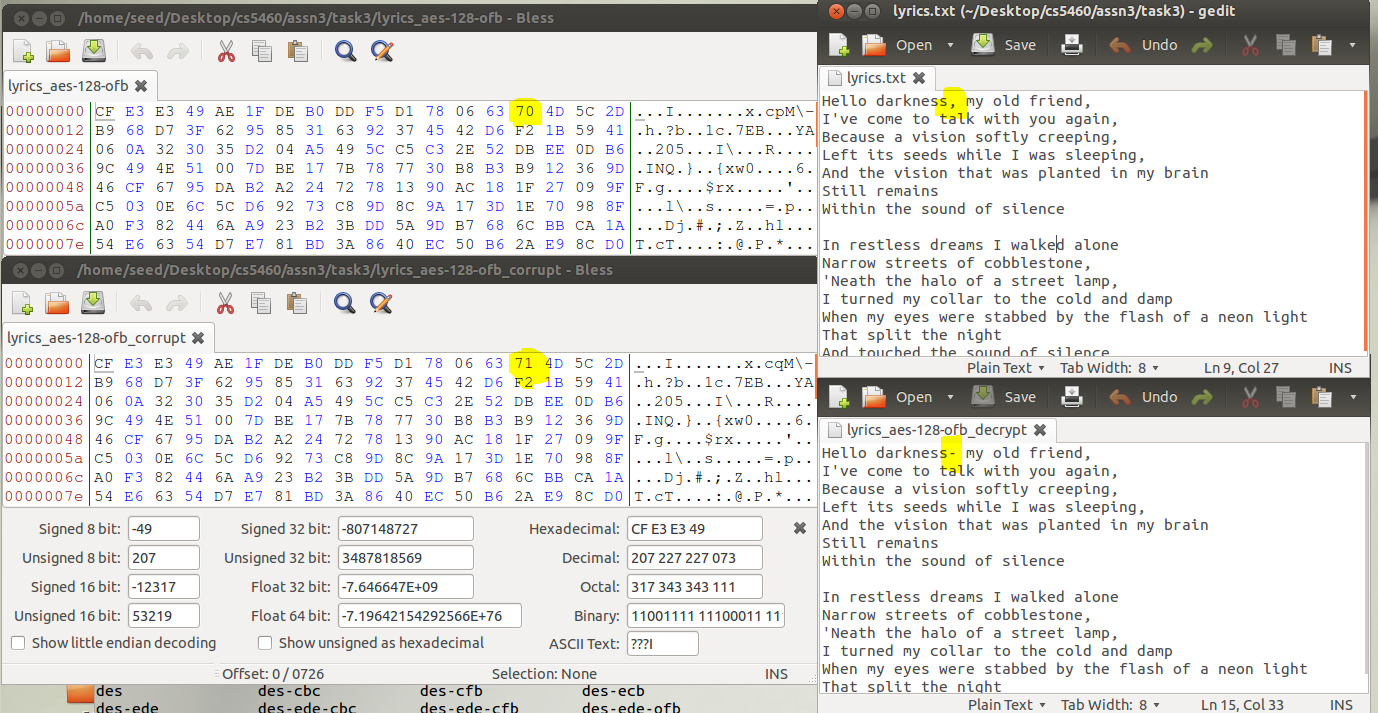
2. The CBC encryption completely scatters the image into a way that gives no information as to what the original image may have been. On the other hand, the ECB encryption shows a vague relation to the original image.



3. The task3.txt encrypted bmp image file is an encrypted version of the simple.bmp image using rc2-cbc cipher and a password of “cs5460”. This is available in the task 3 github repo.

**3.3 Task 3: Encryption Mode – Corrupted Cipher Text**

For this task, lyrics from a song were used and the input filename is lyrics.txt. The file was encrypted using AES-128-CBC cipher and saved as lyrics\_aes-128-cbc with the corrupted file as lyrics\_aes-128-cbc\_corrupt. The 30th bit the corrupted file had its value changed from 1 to 2. This corrupted file was then decrypted to a file named lyrics\_decrypt. A comparison of the before and after are shown below.



To answer the questions asked in the lab document, a majority of the information is recoverable. The only information that is changed is what came before the corrupted bit. I predict that this will change depending on the cipher used. For example, I predict that the CFB cipher will make this completely unrecoverable. To my surprise, the information in every document is mostly recoverable. There are slight differences between the files, but ultimately all are recoverable.

**3.4 Task 4 : Padding**