

Lab 2 - Documentation

February 22, 2020

1 Lab 2 - Secret Key Encryption Lab

1.1 ## By: Laura Pereda

1.1.1 Task 1 - Freq. Analysis against Monoalphabetic Substitution Cipher

By running a simple frequency analysis, the characters ‘R’ and ‘S’ were the top 2. Thus, it would make sense to map the character ‘E’ = ‘R’ and ‘T’ = ‘S’.

R	S	F	P	N	B	H	L	J	M	K	V	T	Y	U	A	I	C	E	Q	O	W	Z	D	G	X
153	137	123	111	104	101	101	101	83	57	52	44	37	37	33	30	25	20	20	12	10	8	3	2	2	2
10.9	9.7	8.7	7.9	7.4	7.2	7.2	7.2	5.9	4.0	3.7	3.1	2.6	2.6	2.3	2.1	1.8	1.4	1.4	0.9	0.7	0.6	0.2	0.1	0.1	

From the substitution, I was able to notice that there were a lot of ‘Tb’, which could indicate that ‘B’ = ‘O’. The word would then form to be ‘TO’:

```
[0mlpE kEpjlrvfTbh
Tufp j0mlpE lfwEp [REDACTED] fhlbknsTfbh [REDACTED] TuE anzb1. [REDACTED] fhlbknsTfbh nhk jbaamhfintfbh pEjnlftt nhk fhcolanTfbh nppmlnhje. TuE bezEjtFwE bc Tufp [REDACTED] vlbwfkE pTmKEhTp qftu n enpfj mhkElpInhkfh1 bc TuE vlbeyEp bc fhcolanTfbh nppmlnhje nhk TuE pbvnyTfbp [REDACTED] Egfp1 [REDACTED] DEjLE fhcolanTfbh bh jbavnTElp nhk hEtqblOp Tbvfjp fhjyNKE emt nLE hbt yfaTfek [REDACTED] umanh cnyfbp fh pEjnlftt vbyfjt enpfj nvyyfEk jltvblnVut vneyfj ott jltvblnVut ott nhk fKhtfTTT anhnleahTt_nHtHtjnfTbh njJEpB hEtqblOp Tbvfjp fh pEjnlftt knhnept pejnlftt kEfnyocpElwfje nTTnjO bveLnfTt ptpTea pejnlftt vlbilna pejnlftt nhk Kepfik vlfhjfryE anyrfjbnp pbcTqNle nhk c0lEhpfj vutpfjny pejnlftt ule nLE TuE KefnyfK ptyunep nhk TuE puknkyE
vIELEDmpfTfp
vblwfkE ptpfap bl knin jbaamhfintfbh bl knhnept blmlnfnsTfbh bl vlyEnpE jblTrnjt TuE fhpJlmjtTbI tbmpEc EyaEukqf fc tbm nLE fTHeLEpTpk fh Tufp jbmIpE nhk unwE n lEnphbneye pTlBh1 jp enjollbmhk fh pEjnlftt bl
hEtqblOpf1 bl ptpfap
jbmIpE bnTjaEap
vblwfkE nh fhlbknsTfbh bl TuE pejnlftt EhlfhEEfTfh1 kfjfjfyfhe
EgvbopE pTmKEhTp Tb jblTrnVut lfpap nhk nTnjO vlbjkmleP
vblwfkE pTmKEhTp qftu nh nvllejfnsTfbh bc TuE urpTbfjny vlePvEjtFwE fh fhcolanTfbh nppmlnhje lEpEnljp
kepjlfeE pEjnlftt EhlfhEEfTfh1 vbljEpEp vnlrfjnylyt Tbpf1 eEfhl mpEk fh fhkmpTl
pTmKEhTp qfyy ee cnafyfnl qftu cmhknahTny Ehjltvfbh nyblfTuap
pTmKEhTp qfyy ee neyE Tb Kepfth nh nljnftejTmLE Tb kecEhk n pveJfcfj ptpTea clba nTnjO
Tufp pTmKEhTp qfyy ee neyE Tb nvyyf pTmKnlk njjEvTEk EhlfhEEfTfh1 TEjuhdmpE Tb vlbvTEjt n ptpTea qftu lEpvEjt Tb n pveJfcfj bllnhfxnTfbhny pEjnlftt vbyfjt
lEnEne zbe intp]
```

A very common word within the English language is ‘THE’, which is similar to ‘TUE’. This would indicate that ‘U’ = ‘H’. Substituting these two letters results in the following:

```
[0mlpE kEpjlrvfTbh
Tufp j0mlpE lfwEp [REDACTED] fhlbknsTfbh [REDACTED] TuE anzb1. [REDACTED] fhlbknsTfbh nhk jbaamhfintfbh pEjnlftt nhk fhcolanTfbh nppmlnhje [REDACTED] DezejTfwe os Tufp j0mlpE fp to vlowfie pTmKEhTp qfTH [REDACTED] enpfj mhkElpInhkfh1 bc [REDACTED]
vlbwfkE ptpfap bl knin joamhfintfbh ol knhnept OllnhfxnTfbh ol vlyEnpE jblTrnjt [REDACTED] fhpJlmjtTbI tbmpEc EyaEukqf fc tbm nLE fTHeLEpTpk fh Tufp j0mlpE nhk HwE n lEnphbneye pTlBh1 jp enjollbmhk fh pEjnlftt bl
hEtqblOpf1 bl ptpfap
j0mlpE OnTjoAp
vblwfkE nh fhlbknsTfbh [REDACTED] pEjnlftt EhlfhEEfTfh1 kfjfjfyfhe
EgvbopE pTmKEhTp TO j0HtEawOlnt lfpap nhk nTnjO vlojkmleP
vblwfkE pTmKEhTp qftu nh nvllejfnsTfbh THE HfpTbfjny vlePvEjtFwE fh fhcolanTfbh nppmlnhje lEpEnljH
kepjlfeE pEjnlftt EhlfhEEfTfh1 vlojEpEp vnlrfjnylyt THOpE eEfhl mpEk fh fhkmpTl
pTmKEhTp qfyy ee cnafyfnl qfTH cmhknahTny Ehjltvfbh nyblfThap
pTmKEhTp qfyy ee neyE TO Kepfth nh nljnftejTmLE TO kecEhk [REDACTED] pveJfcfj ptpTea clba nTnjO
Tufp pTmKEhTp qfyy ee neyE TO nvyyf pTmKnlk njjEvTEk EhlfhEEfTfh1 TEjuhdmpE [REDACTED] vlojEjt [REDACTED] ptpTea qfTH lEpvEjt [REDACTED] pveJfcfj OllnhfxnTfbhny pEjnlftt vbyfjt
lEnEne zbe intp]
```

With this substitution, I noticed that the letter ‘N’ was always alone which could be a replacement

for ‘A’. Replacing that character led to a wild guess of replacing ‘FP’ with ‘IS’. This discovery lead to observing ‘IH’, so another possible replacement ‘H’ for ‘N’.

JOMISE KESJLIVITION
 THIS JOMISE LIVES A ELOAK **INTRODUCTION** TO THE AAZOL TOVIJS IN JOAVNTIEL ANK JOAMINATION SEJMLITTE ANK INCOLATION ASSMLANJE THE OZEJTIWE OC THIS JOMISE IS TO VLOWIKE STMKENTS QITH A EASIJ MNKESTANKINI OC THE VNEYIJ EET JLTVOILAVHT OET ANK IKENTTIE AANALEENT AUTHENTICATION AJESS JONTIOT NETQOLO SEJMLITTE KATAEASE SEJMLITTE KENIYOCSELWIE ATTJO OVELATINI SSTEAS SEJMLITTE ANK KESIN VLINJVYE AYIJIOMS SOCIALE ANK COLENSIJ VITSIJAY SEJMLITTE HELE ALE THE KETAJYK STYAAEUS ANK THE SJHEKHYE
VLEIDMISITES
 OVELATINI SSTEAS OL KATA JOAMINATION OL KATAKASE ORGANIZATION OL VYEASE JONTAJT THE INSTLMJTOV TOSEC EyaEHQI IC TOM ALE INTELESTER IN THIS JOMISE ANK HAWE A REASONAET STION JS EAJOLLONNK IN SEJMLITTE OL NETQOLOINL OLSSTEAS
JOMISE OMTHOES
 VLOWIKE AN **INTRODUCTION** TO THE SEJMLITTE ENLINEELINI KISJIVYNE
 EGVOSE STMKENTS QITH AN AVVEJATION OC THE HISTOLIJAY VELSVEJTIWE IN INCOLATION ASSMLANJE LESEAIJH
 KESJLIEE SEJMLITTE ENLINEELINI VLOJESSES VALTJYALY THOSE EINL MSEK IN INKINSTL
 STMKENTS QIY E CAAIYIAL QITH CMNKAENTAY ENJLTVTION AYIOLITHAS
 STMKENTS QIY E AEYE TO KESIN AN ALHITEJITLE TO KEENK A SVEJICIJ STSTEAS CLOA ATTJO
 THE STMKENTS QIY E AEYE TO AVVYT STANKAK AJEVTEK SEJMLITTE ENLINEELINI TEHNIMDES TO VLOTEJT A SSTEAS QITH LESVEJIT TO A SVEJICIJ ORGANIZATIONAY SEJMLITTE VUYIJT
 GREATE ZOE LUTS

I managed to complete one four-letter word, but I was more interested in the highlighted part of the picture. It seems to form the word ‘INTRODUCTION’, so ‘L’ = ‘R’, ‘K’ = ‘D’, ‘M’ = ‘U’, and ‘J’ = ‘C’:

COURSE DESCRIPTION
 THIS COURSE LIVES A BROAD INTRODUCTION TO THE MAJOR TOPICS IN COMPUTER AND COMMUNICATION SECURITY AND INFORMATION ASSURANCE. THE OBJECTIVE OF THIS COURSE IS TO PROVIDE STUDENTS WITH A BASIC UNDERSTANDING OF THE PROBLEMS OF INFORMATION ASSURANCE AND THE SOLUTIONS THAT EXIST TO SECURE INFORMATION ON COMPUTERS AND NETWORKS TOPICS INCLUDE BUT ARE NOT LIMITED TO HUMAN FACTORS IN SECURITY POLICY, BASIC APPLIED CRYPTOGRAPHY, SOFTWARE AND FORENSIC PHYSICAL SECURITY HERE ARE THE DETAILED SYLLABUS AND THE SCHEDULE
PREREQUISITES
 OPERATING SYSTEMS OR DATA COMMUNICATION OR DATABASE ORGANIZATION OR VYEASE CONTACT THE INSTRUCTOR YOUSEC EyaEHQI IC YOU ARE INTERESTED IN THIS COURSE AND HAVE A REASONAET STRONI CS EAGROUND IN SECURITY OR NETQOLOINL OLSSTEAS
COURSE OUTCOMES
 VROWIDE AN INTRODUCTION TO THE SECURITY ENLINEELINI DISCIVYNE
 EGVOSE STUDENTS TO CONTEAVORATE RISKS AND ATTACK PROCEDURES
 VROWIDE STUDENTS QITH AN APPRECIATION OC THE HISTORIC VERSVETIVE IN INFORMATION ASSURANCE RESEARCH
 DESCRIBE SECURITY ENLINEELINI VROCESSES VARTICUARY THOSE EINL USED IN INDUST
 STUDENTS QIY E CAAIYAR QITH CUNDAENTAY ENCRTVITION AYIORTHAS
 STUDENTS QIY E AEYE TO DESIN AN ARCHITECTURE TO DEFEND A SVECICIC STSTEAS CROA ATTAC
 THE STUDENT QIY E AEYE TO AVVYT STANDARD ACCEPTED SECURITY ENLINEELINI TECHNIQUES TO VROTECT A SSTEAS QITH RESPECT TO A SVECICIC ORGANIZATIONAY SECURITY VUYICT
 GREATE ZOE LUTS

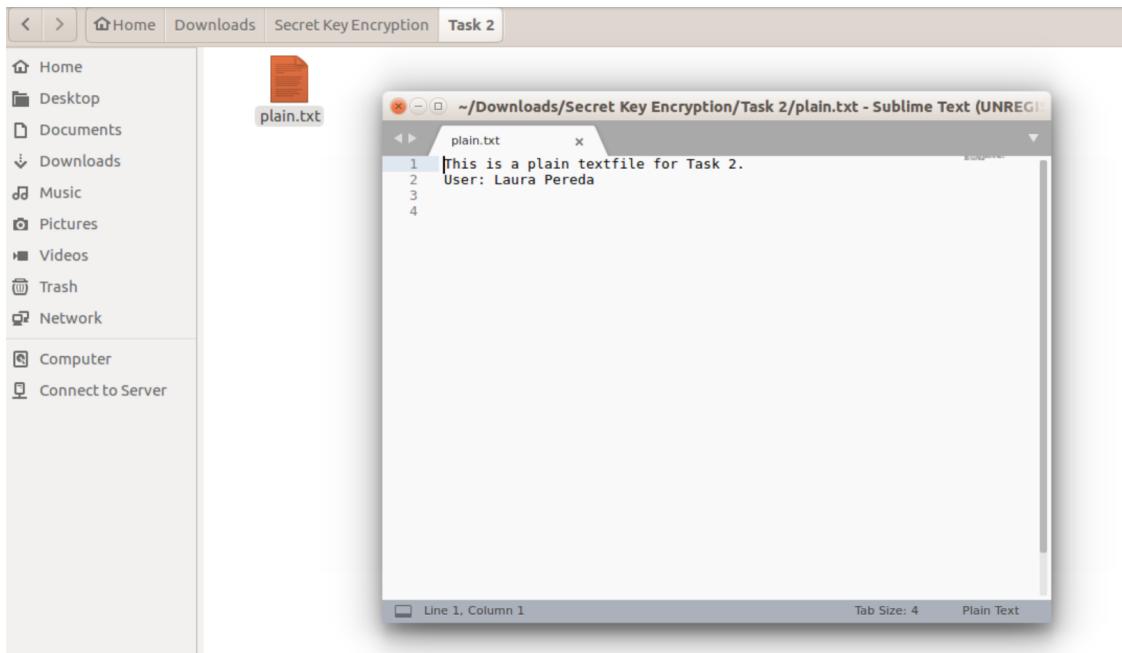
From there it was just finishing the last couple of words to finish the decryption. Final outcome:

COURSE DESCRIPTION
 THIS COURSE GIVES A BROAD INTRODUCTION TO THE MAJOR TOPICS IN COMPUTER AND COMMUNICATION SECURITY AND INFORMATION ASSURANCE. THE OBJECTIVE OF THIS COURSE IS TO PROVIDE STUDENTS WITH A BASIC UNDERSTANDING OF THE PROBLEMS OF INFORMATION ASSURANCE AND THE SOLUTIONS THAT EXIST TO SECURE INFORMATION ON COMPUTERS AND NETWORKS TOPICS INCLUDE BUT ARE NOT LIMITED TO HUMAN FACTORS IN SECURITY POLICY, BASIC APPLIED CRYPTOGRAPHY, PUBLIC KEY CRYPTOGRAPHY KEY AND IDENTITY MANAGEMENT AUTHENTICATION ACCESS CONTROL, NETWORK SECURITY, DATABASE SECURITY, DENIAL OF SERVICE ATTACK, OPERATING SYSTEM SECURITY, PROGRAM SECURITY AND DESIGN PRINCIPLE, MALICIOUS SOFTWARE AND FORENSIC PHYSICAL SECURITY. HERE ARE THE DETAILED SYLLABUS AND THE SCHEDULE
PREREQUISITES
 OPERATING SYSTEMS OR DATA COMMUNICATION OR DATABASE ORGANIZATION OR PLEASE CONTACT THE INSTRUCTOR YOUSEF ELMEHOWI IF YOU ARE INTERESTED IN THIS COURSE AND HAVE A REASONABLY STRONG CS BACKGROUND IN SECURITY OR NETWORKING OR SYSTEMS
COURSE OUTCOMES
 PROVIDE AN INTRODUCTION TO THE SECURITY ENGINEERING DISCIPLINE
 EXPOSE STUDENTS TO CONTEMPORARY RISKS AND ATTACK PROCEDURES
 PROVIDE STUDENTS WITH AN APPRECIATION OF THE HISTORICAL PERSPECTIVE IN INFORMATION ASSURANCE RESEARCH
 DESCRIBE SECURITY ENGINEERING PROCESSES PARTICULARLY THOSE BEING USED IN INDUSTRY
 STUDENTS WILL BE FAMILIAR WITH FUNDAMENTAL ENCRYPTION ALGORITHMS
 STUDENTS WILL BE ABLE TO DESIGN AN ARCHITECTURE TO DEFEND A SPECIFIC SYSTEM FROM ATTACK
 THE STUDENT WILL BE ABLE TO APPLY STANDARD ACCEPTED SECURITY ENGINEERING TECHNIQUES TO PROTECT A SYSTEM WITH RESPECT TO A SPECIFIC ORGANIZATIONAL SECURITY POLICY
 GREATE JOB GUYS

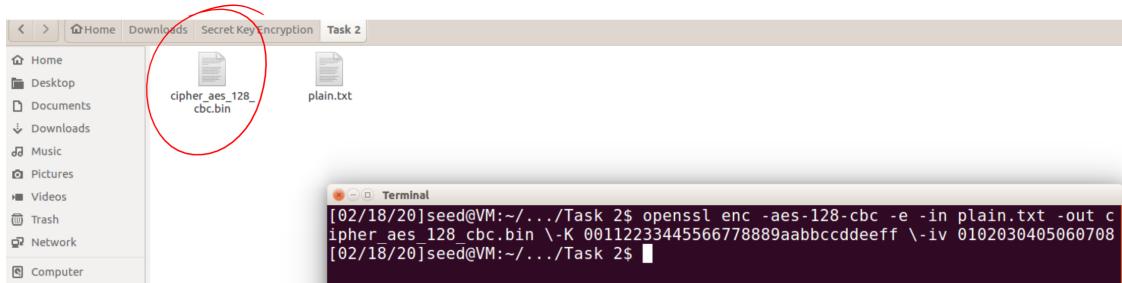
1.1.2 Task 2 - Encryption using Different Ciphers and Modes

This task indicates to try out different ciphertypes with different modes. The algorithm seems to follow a pattern of **AlgName-keySize-encryptionMode**. The familiar encryption mode for block cipher are Electronic Codebook (ECB) and Cipher Block Chaining (CBC). Because of this, I will be using the ciphertype examples to encrypt and decrypt, while increasing size if possible.

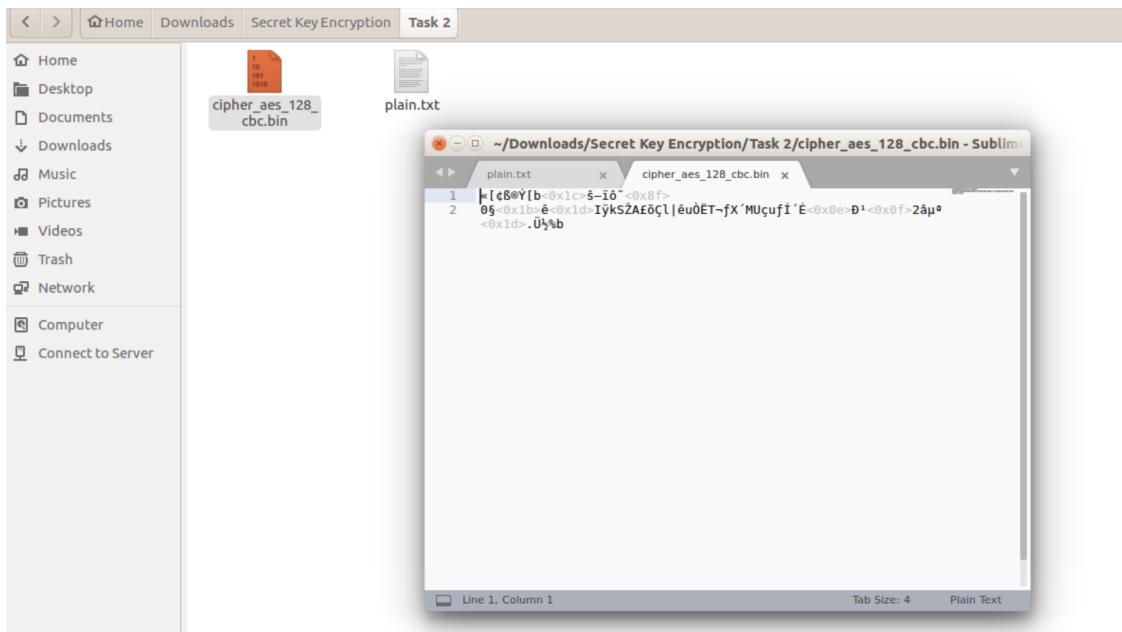
To implement the encryption methods, I created a plain textfile with the following content:



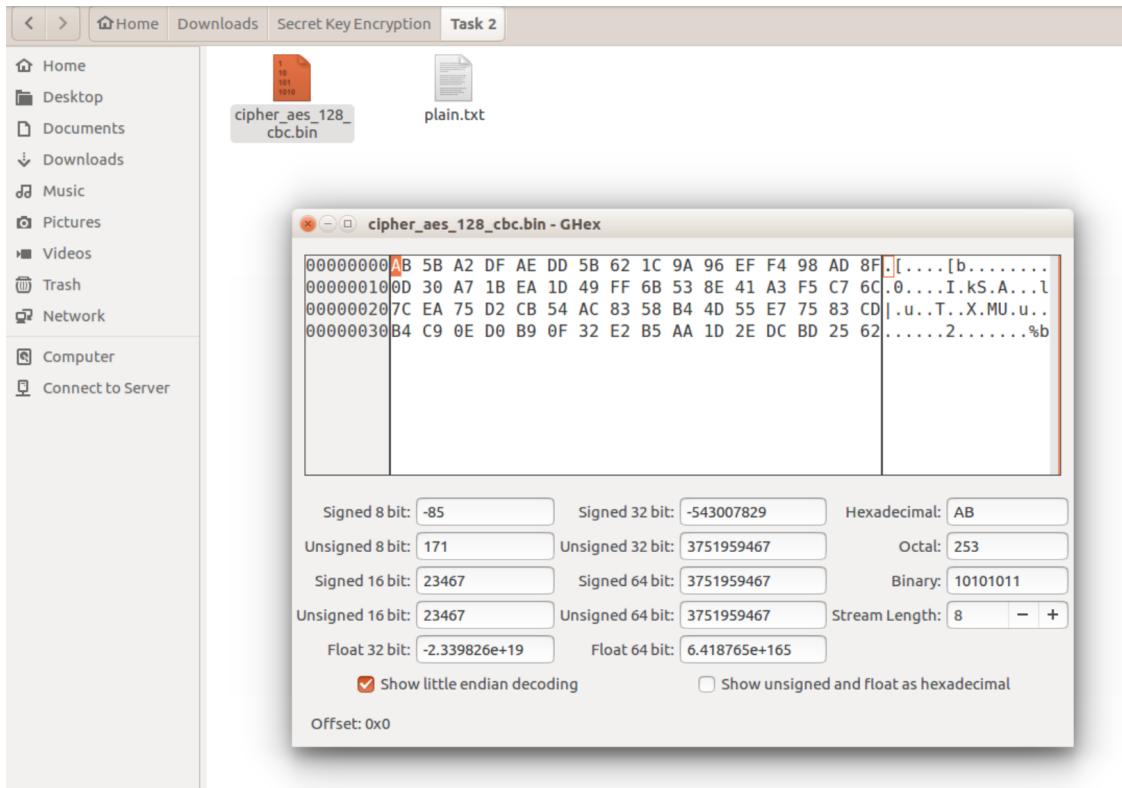
Using the cipher and mode: -aes-128-cbc, I encrypt the plain text file.



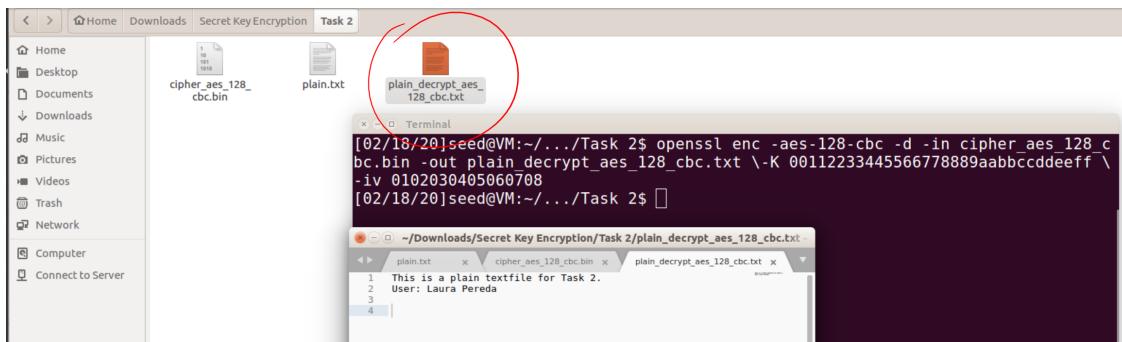
As shown below, using the cipher -aes-128-cbc resulted in an encrypted file that is not easily visible to understand when opening in a normal text editor.



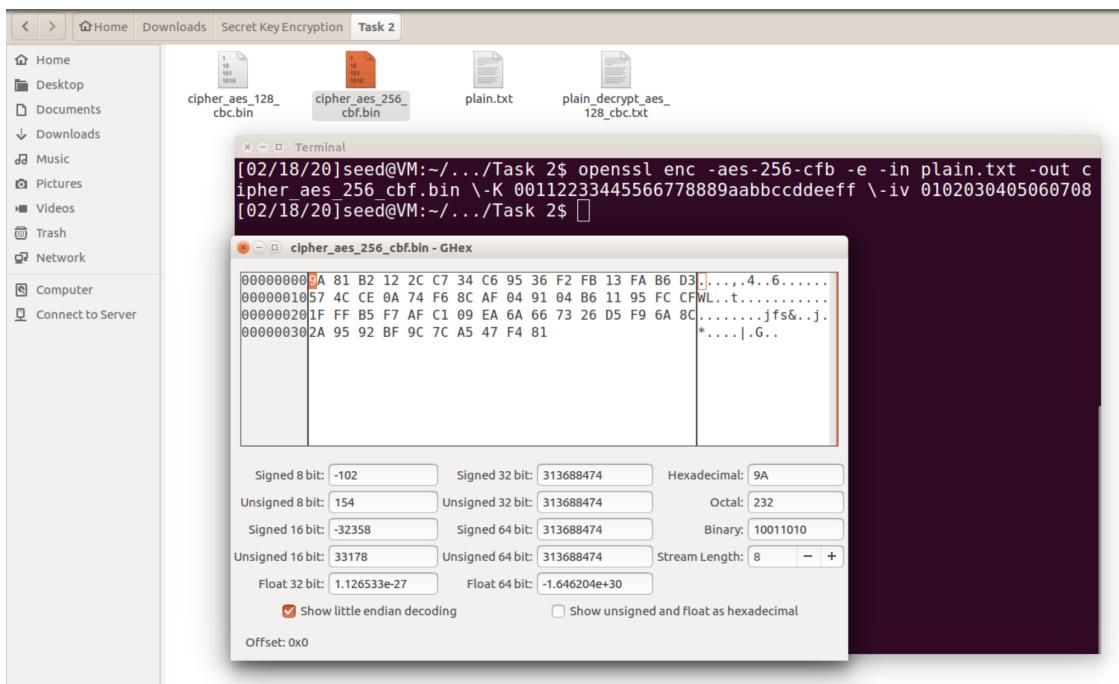
However, after doing some research, I realized GHEX was installed in the SeedUbuntu, which will let you view and modify files of binary format. This is just a simple demonstration for the first encryption.



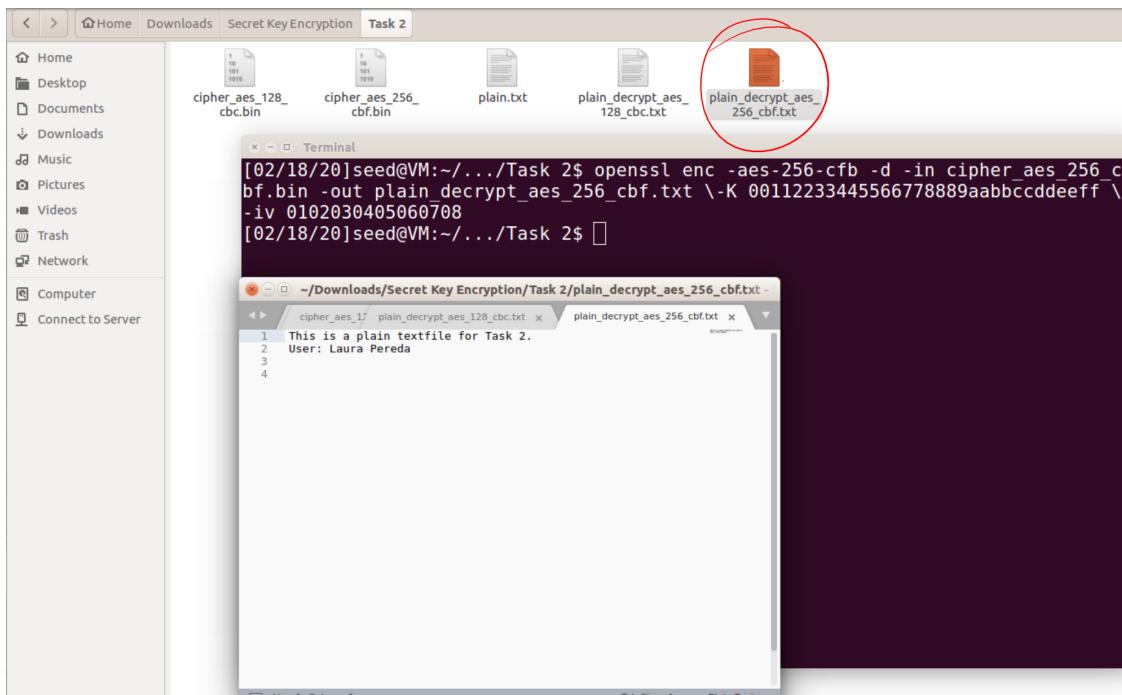
Here is the related decryption:



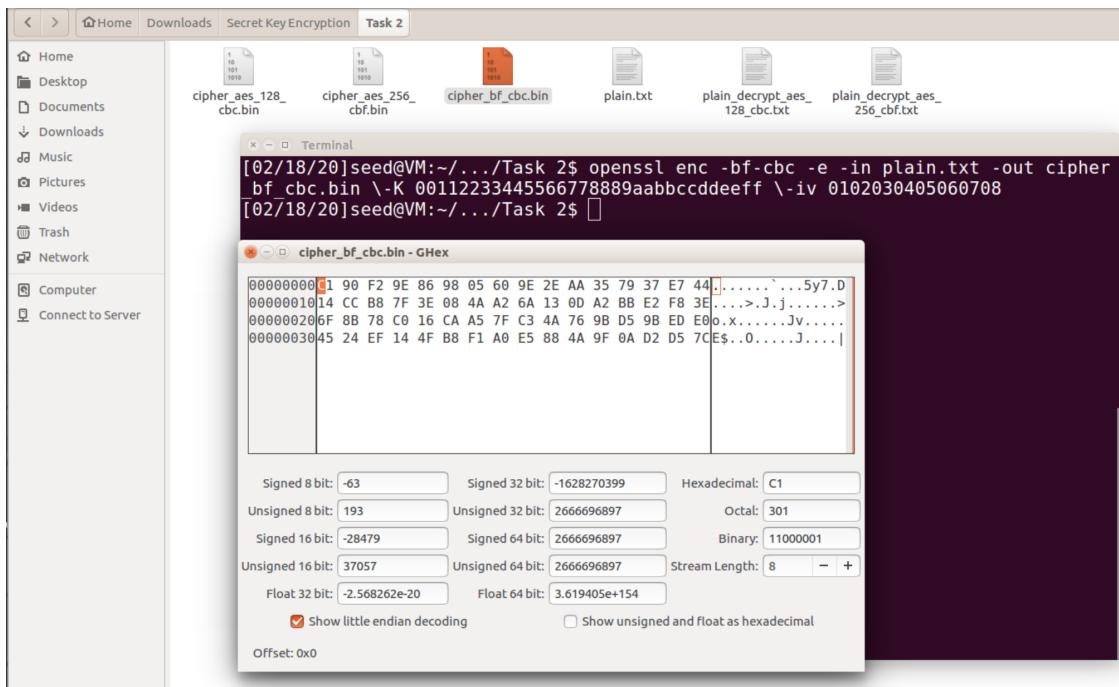
Using the cipher and mode: -aes-256-cfb (cbc with size 256 failed at decryption everytime for some reason) while using the same plain text file.



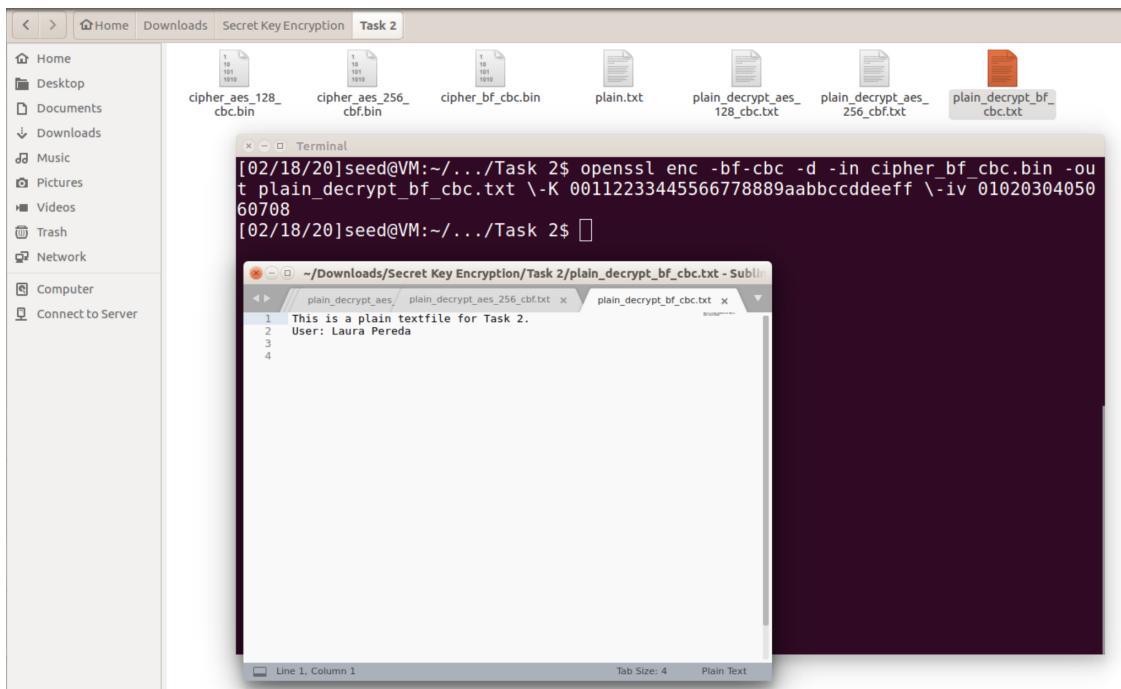
The decryption:



Using the cipher and mode: -bf-cbc while using same plain text file



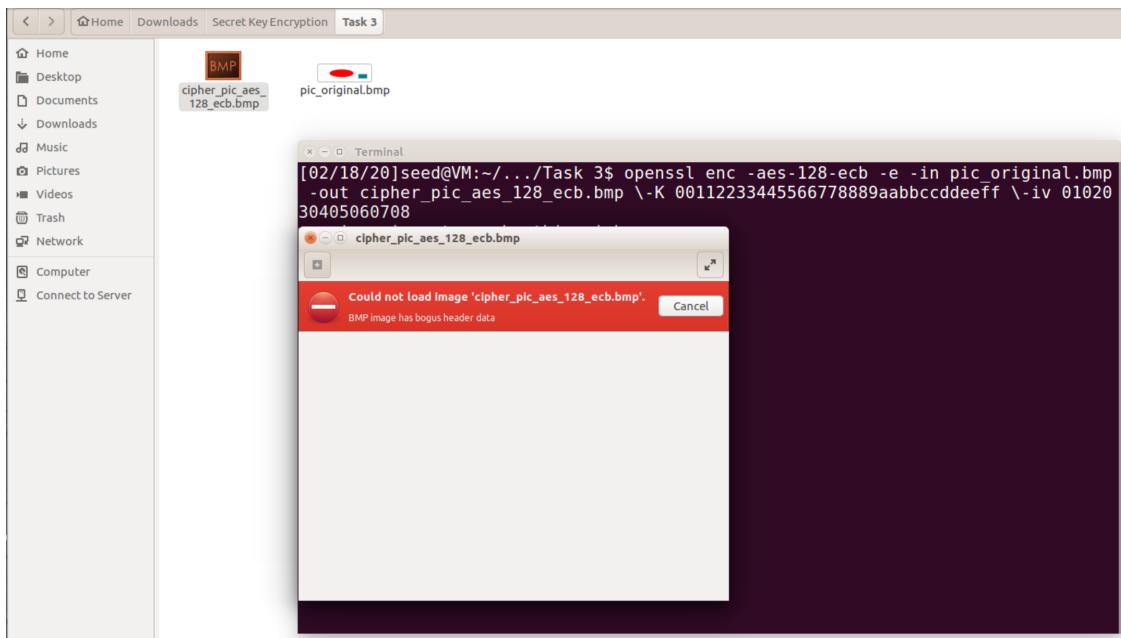
The decryption:



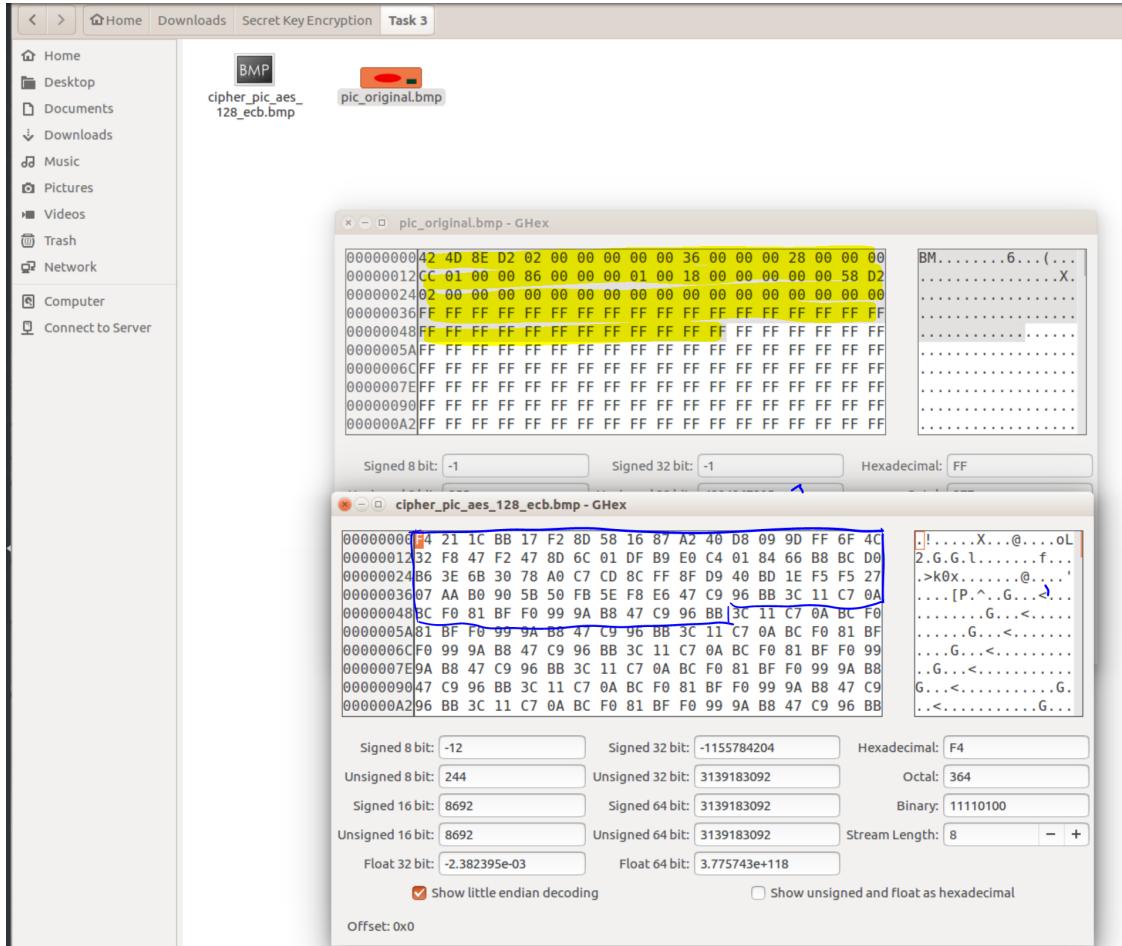
1.1.3 Task 3 - Encryption Mode: ECB vs CBC

Since the type of ECB or CBC mode is not defined, I decided to use the same ciphertype and mode above (but using ECB and CBC).

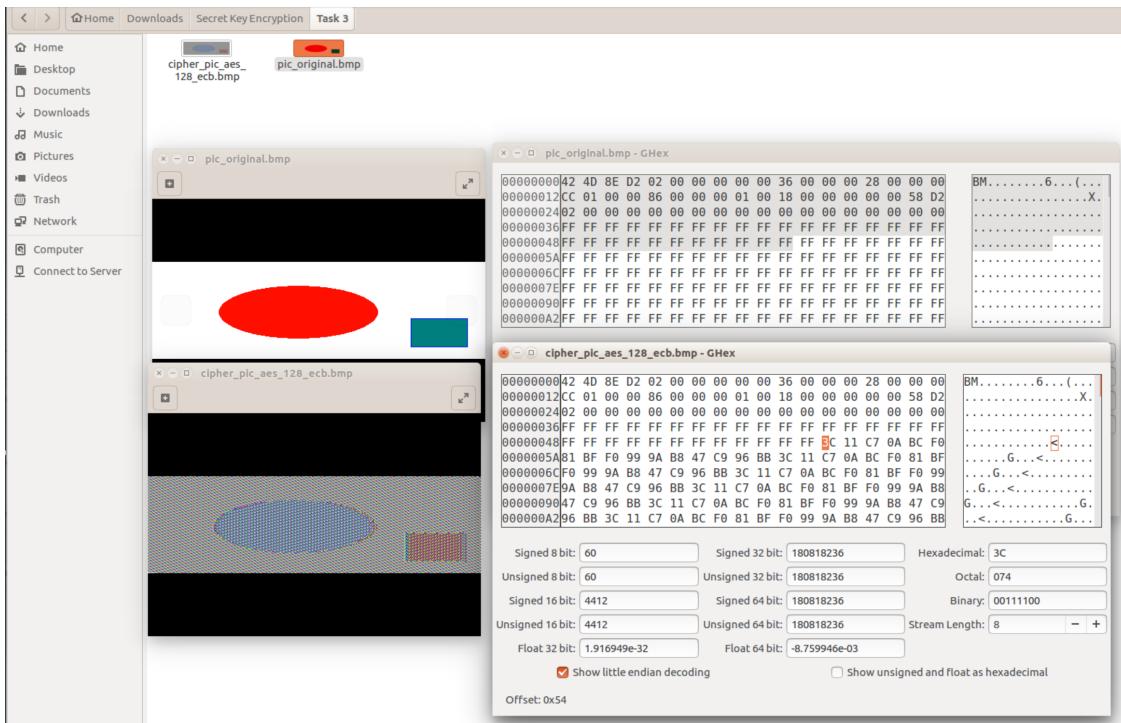
Using the image provided in Blackboard, I encrypted the image by using **-aes-128-cbc**. When I attempted to view the image through the ImageViewer, an error occurred stating that there was bogus header data.



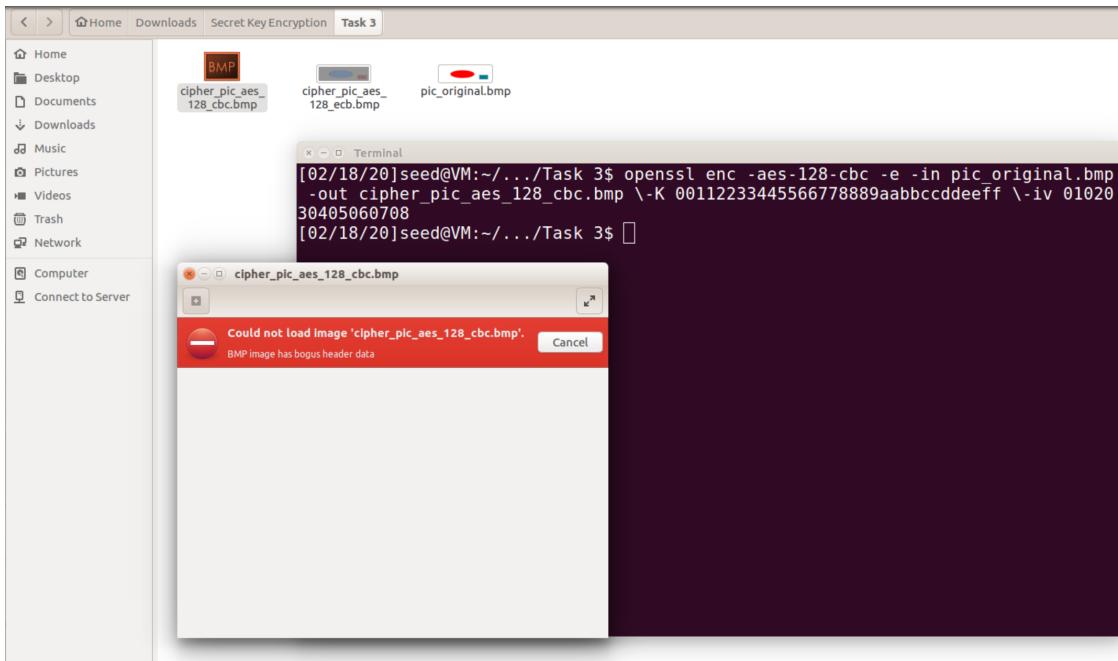
The task 3 description indicates that the first 54 bytes contain the header information about the picture, but we have to set it correctly in order for it to be treated as a legitimate .bmp file.



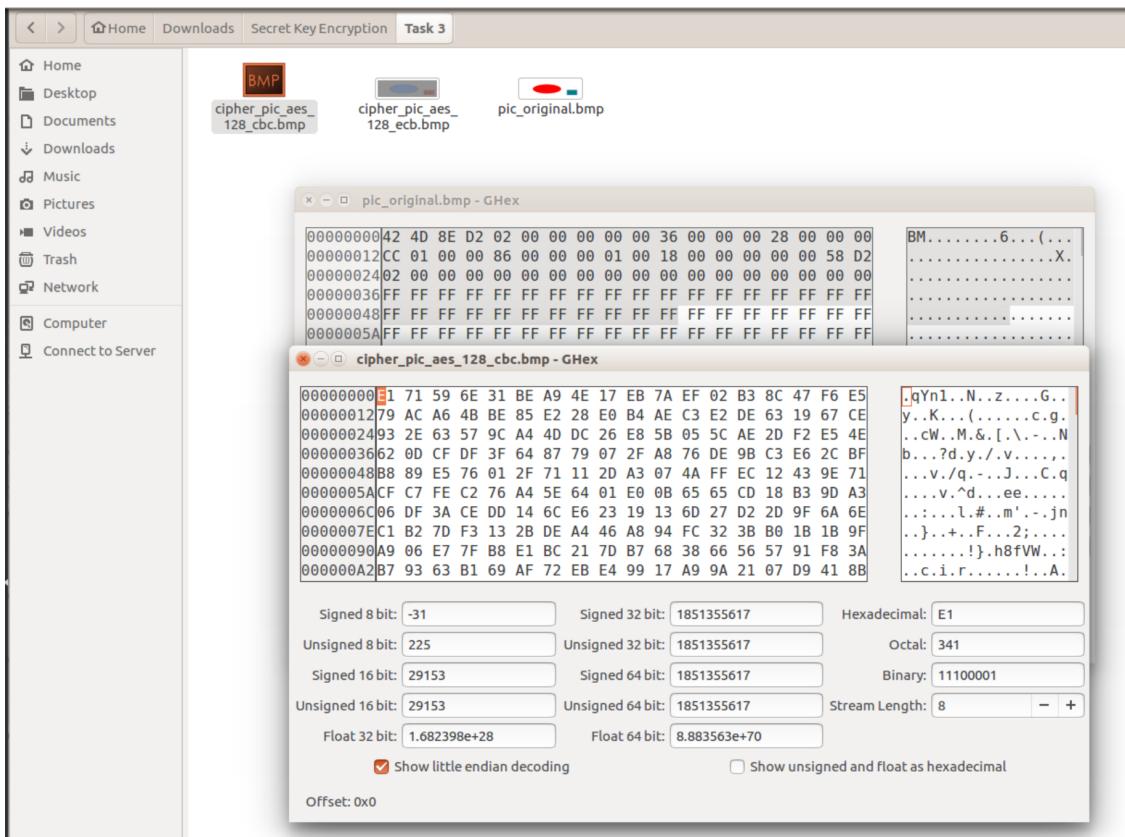
By replacing the first 54 bytes of cipher with the original, I was able to view the image.



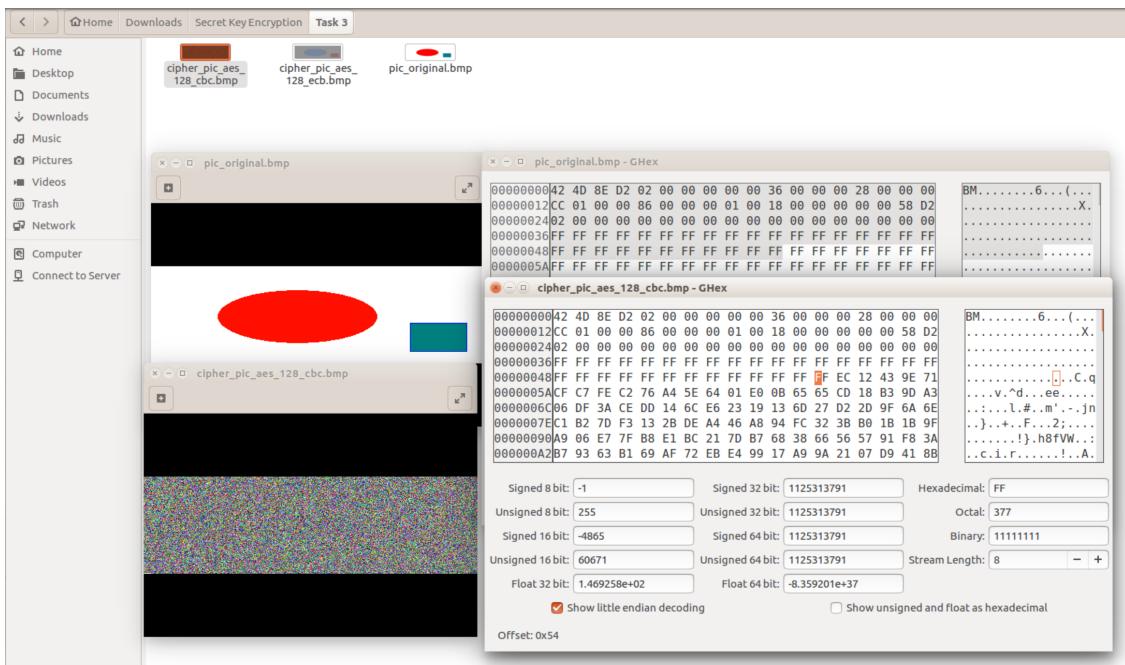
The same is done with the CBC encryption.



Identifying first 54 bytes to change:



Modifying to view image:



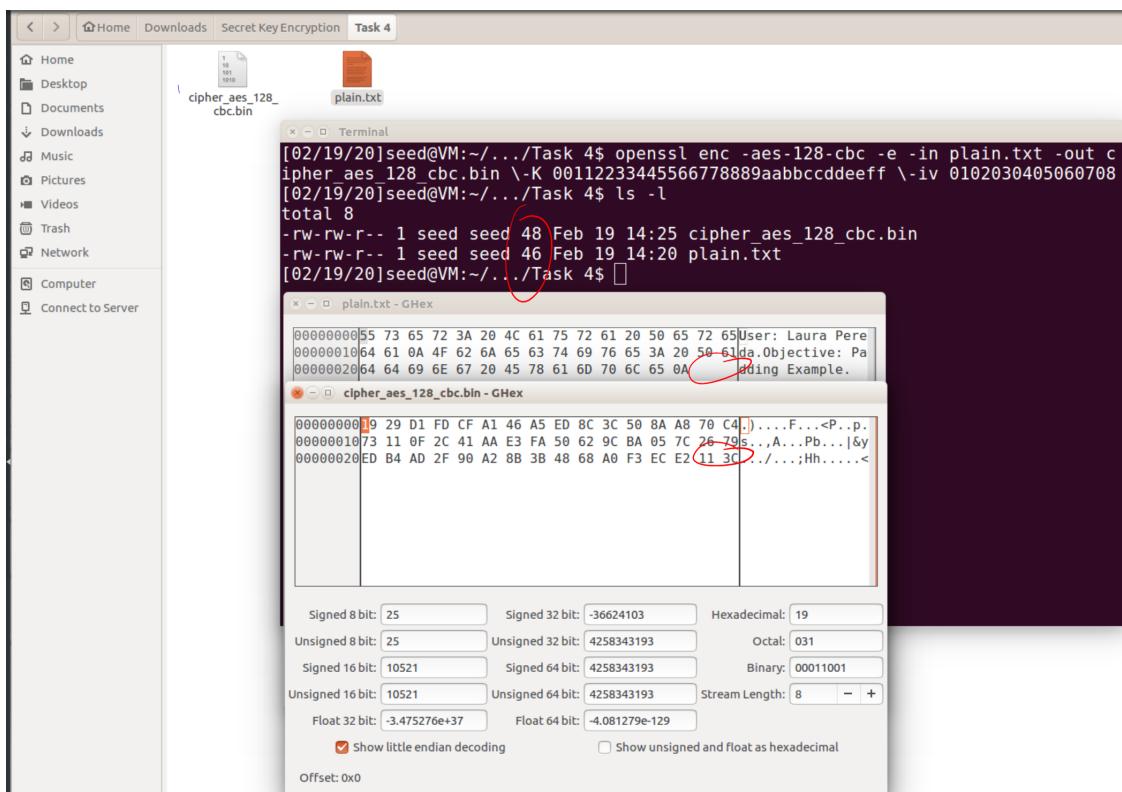
As shown above, the main disadvantage of ECB is when we are encrypting identical plain text

blocks into identical cipher text blocks. The patterns are still visible, which then leads to lack of confidentiality. CBC, on the otherhand, has a more secure system since it utilizes the XOR operation before encrypting each cipher text block.

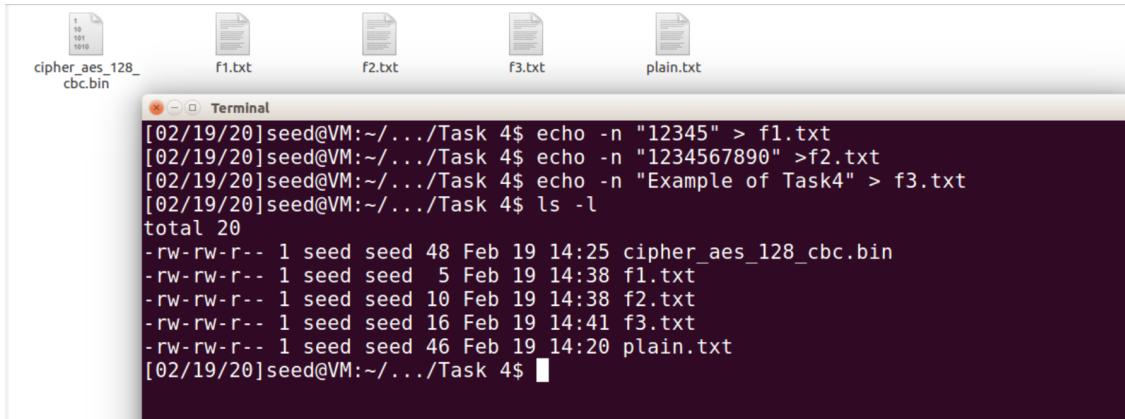
1.1.4 Task 4 - Padding

In this task, we are told to choose between using ECB, CBC, CFB, and OFB and indicating if any of these require padding, and if not, why they do not require padding. From the reference to “Block cipher mode of operation” in the pdf, cipher modes such as ECB and CBC require that the initial input be an exact multiple of the block size. IF the plain text that will be encrypted is not an exact multiple, then padding is required before encrypting (usually an additional string). When decrypting, the receiver will need to know how to remove the correct amount of padding in an unambiguous manner.

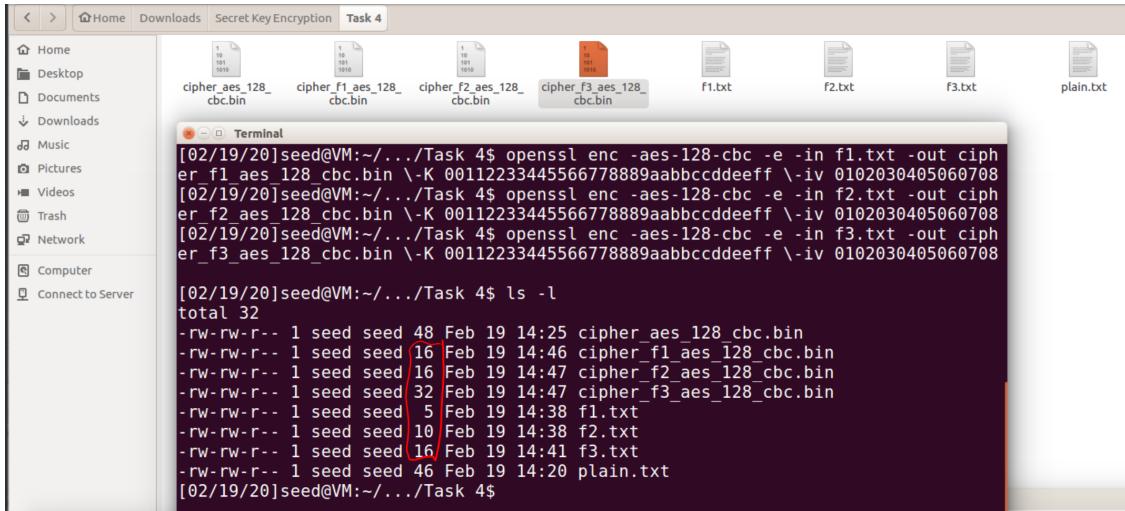
As an example of padding, I created a simple plain text file the size of 46. But after encrypting through -aes-128-cbc, the file grew to be the size of 48 bytes which is a multiple of 8 (standard size for block). The size of one block for 128/8 is 16 bytes. The 2 additional bytes were added to fulfill that requirement.



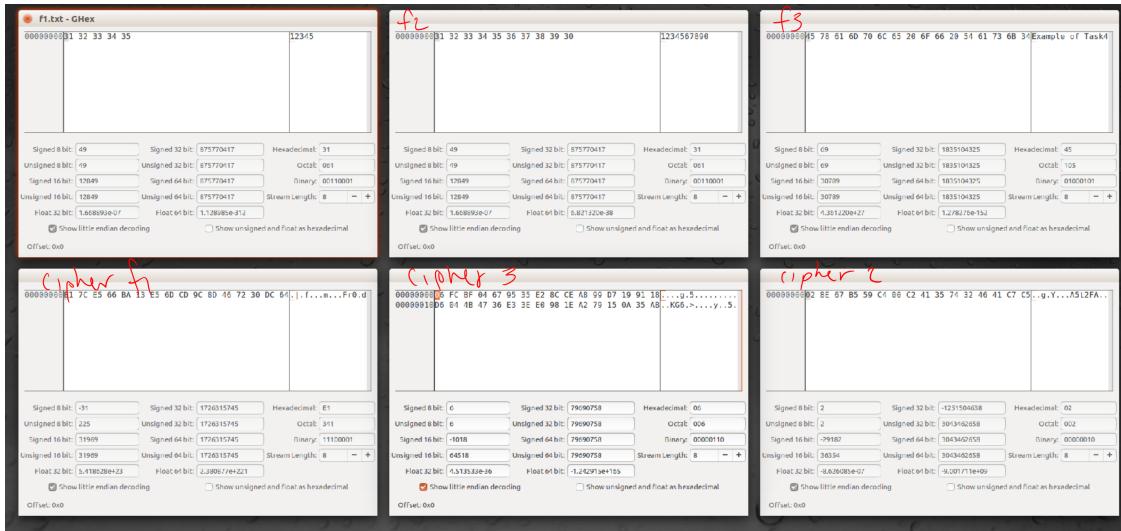
The next part of the task is to create 3 files the length of 5, 10 and 16. Which is shown below:



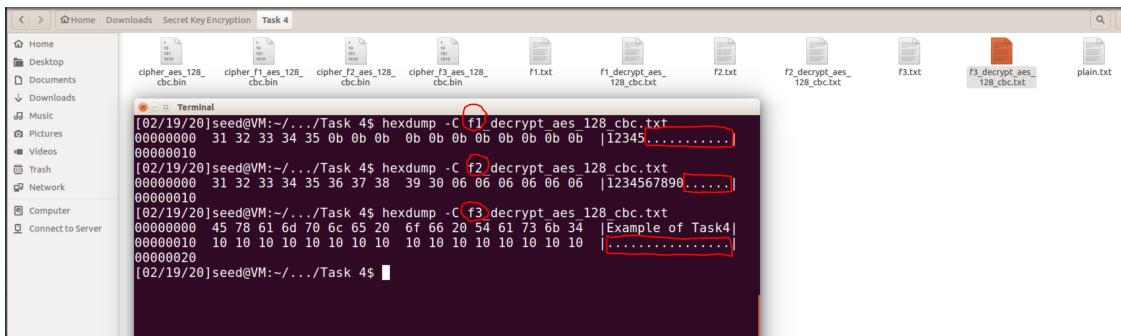
After encrypting them using -aes-128-cbc, we can see the additional padding incorporated into the files through GHEX. It's interesting to note that the original f3 files of 16 bytes was encrypted with an outcome of 32 bytes instead of 16 bytes like the other two.



Opening the original files and the encrypted files with GHEX, we can view the difference:



After decrypting the encrypted files, to view the actual content of the padding we can use a hex tool. Below is the following content of the padding for f1, f2 and f3 decrypted files:



1.1.5 Task 5 - Error Propagation – Corrupted Cipher Text

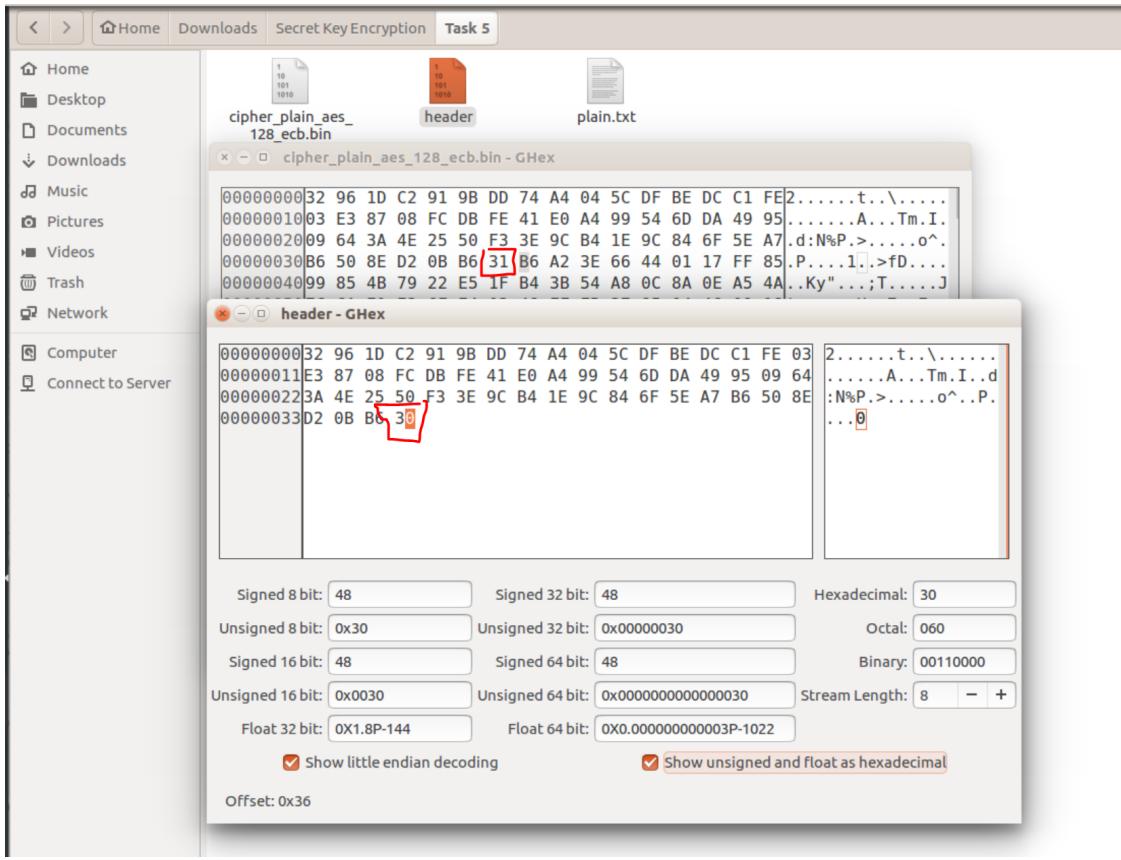
In this task, we want to answer the following question: How much information can you recover by decrypting the corrupted file, if the encryption mode is ECB, CBC, CFB, or OFB, respectively?

Since I must answer this question before finishing the task, I will base my answer on the results I have. I have compared ECB vs CBC and discovered that ECB fails in comparison to CBC when it comes to error propagation. This does not imply that CBC will also have the same results in recovering a cipher. Most likely the error propagation will be dependent on other blocks of encrypted blocks would most likely be the best at recovering information. Since I am not sure about the answer, I choose OFB.

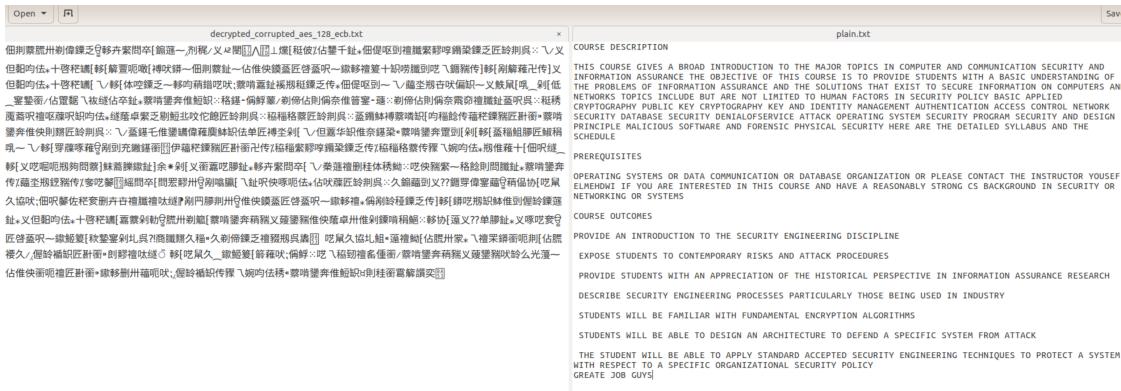
The task indicates to create a file at least 1000 bytes long, so I'll be using the original plain text from Task 1 since it is 1656 bytes long.

ECB Corrupted File

1. After encrypting the plain text, I modified the 55th byte of the encrypted text as shown below. I only changed the second bit from 0 to 1.

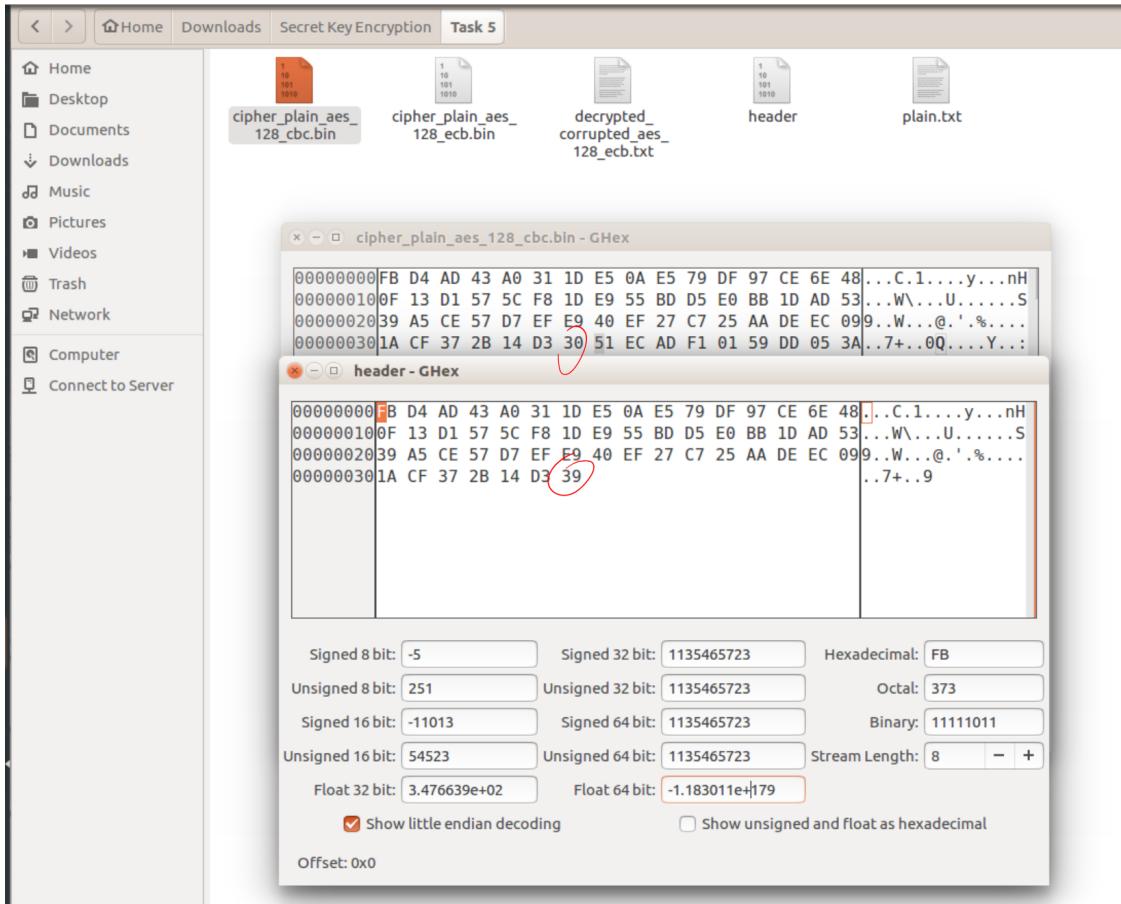


2. After corrupting, I decrypted the text and opened it within a text editor to compare the original plaintext to the corrupted decryption. As shown below, the one change resulted in being unable to view the original content at all.



CBC Corrupted File

1. Repeat by corrupting 55th byte of encrypted text. Changed second bit from 9 to 0.



- After corrupting, these are the results of decrypting corrupted file. Majority of the text file is still understandable with only that small portion being affected.

COURSE DESCRIPTION

THIS COURSE GIVES A BROAD INTRODUCTION TO THE MAJOR TOPICS IN COMPUTER AND COMMUNICATION SECURITY AND INFORMATION ASSURANCE THE OBJECTIVE OF THIS COURSE IS TO PROVIDE STUDENTS WITH A BASIC UNDERSTANDING OF THE PROBLEMS OF INFORMATION ASSURANCE AND THE SOLUTIONS THAT EXIST TO SECURE INFORMATION ON COMPUTERS AND NETWORKS. TOPICS INCLUDE BUT ARE NOT LIMITED TO HUMAN FACTORS IN SECURITY POLICY BASIC APPLIED CRYPTOGRAPHY PUBLIC KEY CRYPTOGRAPHY KEY AND IDENTITY MANAGEMENT AUTHENTICATION ACCESS CONTROL NETWORK SECURITY DATABASE SECURITY DENIALOFSERVICE ATTACK OPERATING SYSTEM SECURITY PROGRAM SECURITY AND DESIGN PRINCIPLE MALICIOUS SOFTWARE AND FORENSIC PHYSICAL SECURITY HERE ARE THE DETAILED SYLLABUS AND THE SCHEDULE.

PREREQUISITES

OPERATING SYSTEMS OR DATA COMMUNICATION OR DATABASE ORGANIZATION OR PLEASE CONTACT THE INSTRUCTOR YOUSEF ELMENHOT IF YOU ARE INTERESTED IN THIS COURSE AND HAVE A REASONABLY STRONG CS BACKGROUND IN SECURITY OR NETWORKING OR SYSTEMS

COURSE OUTCOMES

PROVIDE AN INTRODUCTION TO THE SECURITY ENGINEERING DISCIPLINE
EXPOSE STUDENTS TO CONTEMPORARY RISKS AND ATTACK PROCEDURES
PROVIDE STUDENTS WITH AN APPRECIATION OF THE HISTORICAL PERSPECTIVE IN INFORMATION ASSURANCE RESEARCH
DESCRIBE SECURITY ENGINEERING PROCESSES PARTICULARLY THOSE BEING USED IN INDUSTRY
STUDENTS WILL BE FAMILIAR WITH FUNDAMENTAL ENCRYPTION ALGORITHMS
STUDENTS WILL BE ABLE TO DESIGN AN ARCHITECTURE TO DEFEND A SPECIFIC SYSTEM FROM ATTACK
THE STUDENT WILL BE ABLE TO APPLY STANDARD ACCEPTED SECURITY ENGINEERING TECHNIQUES TO PROTECT A SYSTEM WITH RESPECT TO A SPECIFIC ORGANIZATIONAL SECURITY POLICY
CREATE JOB GUYS

COURSE DESCRIPTION

THIS COURSE GIVES A BROAD INTRODUCTION TO THE MAJOR TOPICS IN COMPUTER AND COMMUNICATION SECURITY AND INFORMATION ASSURANCE THE OBJECTIVE OF THIS COURSE IS TO PROVIDE STUDENTS WITH A BASIC UNDERSTANDING OF THE PROBLEMS OF INFORMATION ASSURANCE AND THE SOLUTIONS THAT EXIST TO SECURE INFORMATION ON COMPUTERS AND NETWORKS. TOPICS INCLUDE BUT ARE NOT LIMITED TO HUMAN FACTORS IN SECURITY POLICY BASIC APPLIED CRYPTOGRAPHY PUBLIC KEY CRYPTOGRAPHY KEY AND IDENTITY MANAGEMENT AUTHENTICATION ACCESS CONTROL NETWORK SECURITY DATABASE SECURITY DENIALOFSERVICE ATTACK OPERATING SYSTEM SECURITY PROGRAM SECURITY AND DESIGN PRINCIPLE MALICIOUS SOFTWARE AND FORENSIC PHYSICAL SECURITY HERE ARE THE DETAILED SYLLABUS AND THE SCHEDULE.

PREREQUISITES

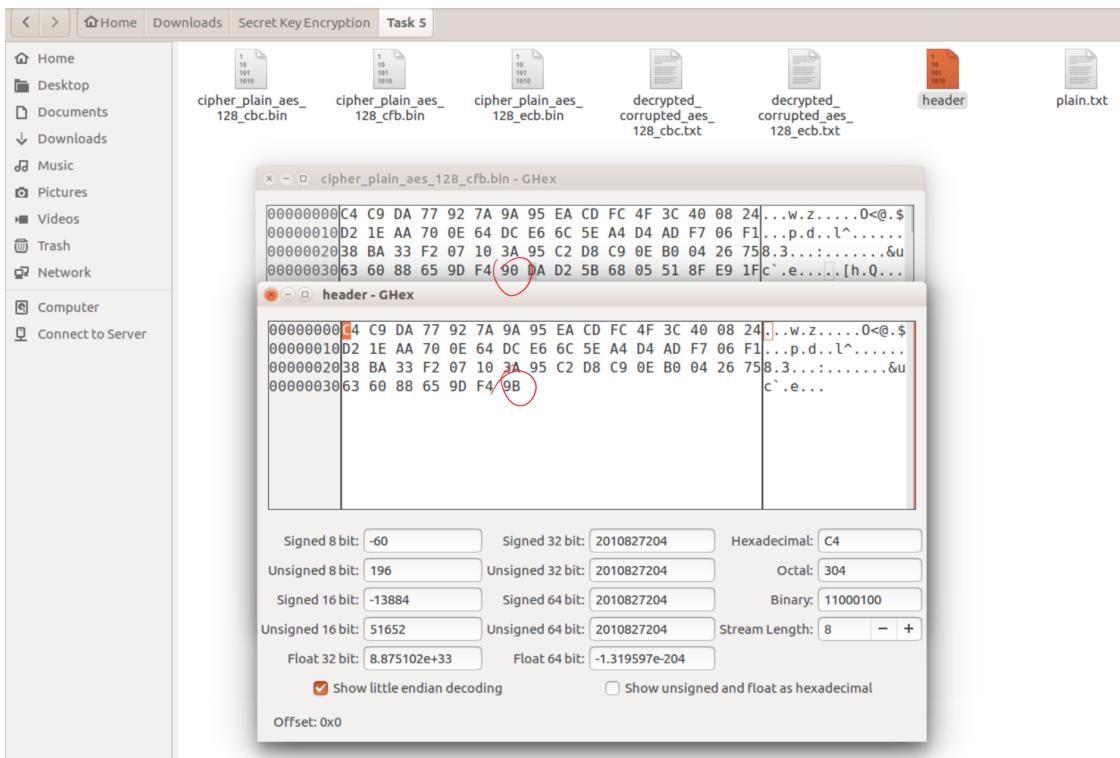
OPERATING SYSTEMS OR DATA COMMUNICATION OR DATABASE ORGANIZATION OR PLEASE CONTACT THE INSTRUCTOR YOUSEF ELMENHOT IF YOU ARE INTERESTED IN THIS COURSE AND HAVE A REASONABLY STRONG CS BACKGROUND IN SECURITY OR NETWORKING OR SYSTEMS

COURSE OUTCOMES

PROVIDE AN INTRODUCTION TO THE SECURITY ENGINEERING DISCIPLINE
EXPOSE STUDENTS TO CONTEMPORARY RISKS AND ATTACK PROCEDURES
PROVIDE STUDENTS WITH AN APPRECIATION OF THE HISTORICAL PERSPECTIVE IN INFORMATION ASSURANCE RESEARCH
DESCRIBE SECURITY ENGINEERING PROCESSES PARTICULARLY THOSE BEING USED IN INDUSTRY
STUDENTS WILL BE FAMILIAR WITH FUNDAMENTAL ENCRYPTION ALGORITHMS
STUDENTS WILL BE ABLE TO DESIGN AN ARCHITECTURE TO DEFEND A SPECIFIC SYSTEM FROM ATTACK
THE STUDENT WILL BE ABLE TO APPLY STANDARD ACCEPTED SECURITY ENGINEERING TECHNIQUES TO PROTECT A SYSTEM WITH RESPECT TO A SPECIFIC ORGANIZATIONAL SECURITY POLICY
CREATE JOB GUYS

CFB Corrupted File

- Repeat by corrupting 55th byte of encrypted text. Changed second bit from B to 0.

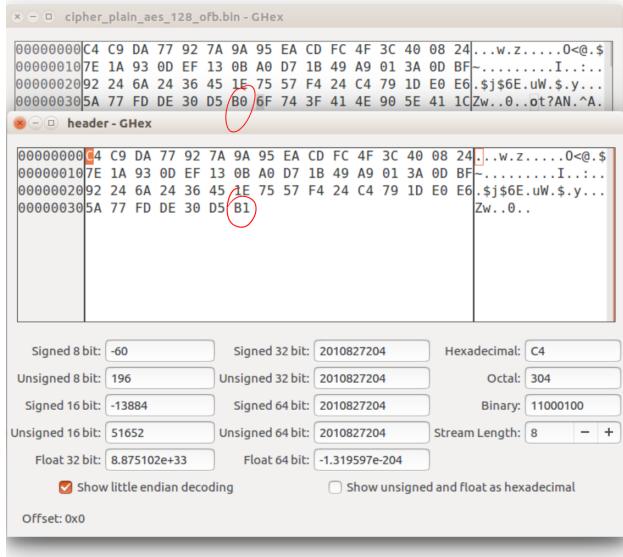


2. After corrupting, these are the results of decrypting corrupted file. Again, majority of text file is understandable, however, the area in which the corruption occurs is extended just a little more compared to CBC.

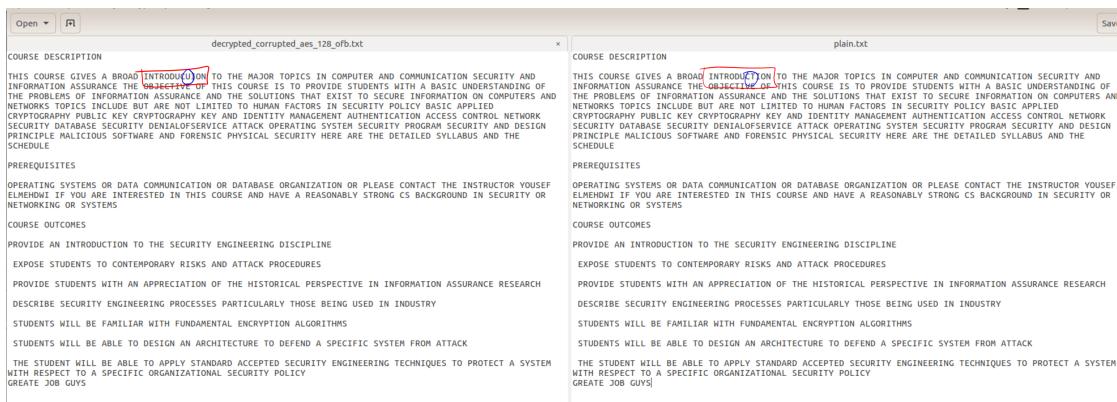
The screenshot shows two side-by-side text editors. The left editor is titled 'decrypted_corrupted_aes_128_cfb.txt' and the right is 'plain.txt'. The left editor contains corrupted text with many null characters (represented by question marks) and some readable text like 'COURSE DESCRIPTION'. The right editor contains the original readable text from 'plain.txt'. Both editors have identical content.

OFB Corrupted File

1. Repeat by corrupting 55th byte of encrypted text. Changed second bit from 1 to 0.



- After corrupting, these are the results of decrypting corrupted file. With this one, I am pretty amazed with the results. Only one letter within the word where the corruption occurred was affected.



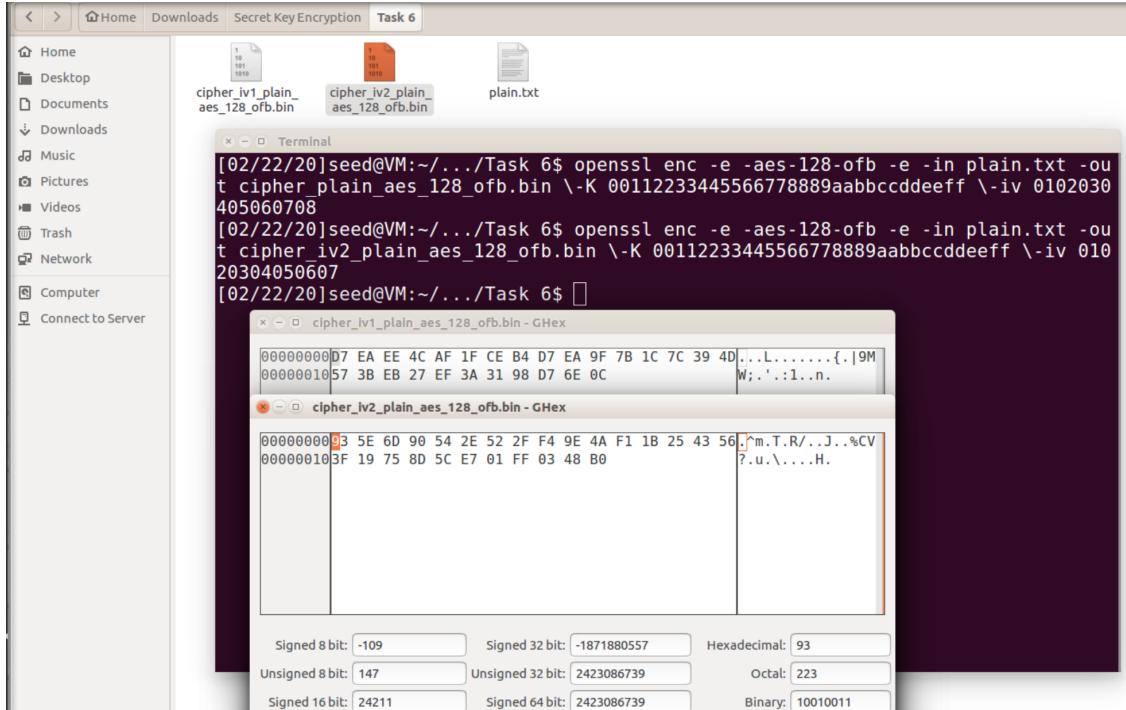
From these results, it is clear that OFB is able to recover or safe the most from a corruption.

1.1.6 Task 6 - Initial Vector (IV)

In this task, we are analyzing the usage of IV when encrypting. Specifically, the uniqueness.

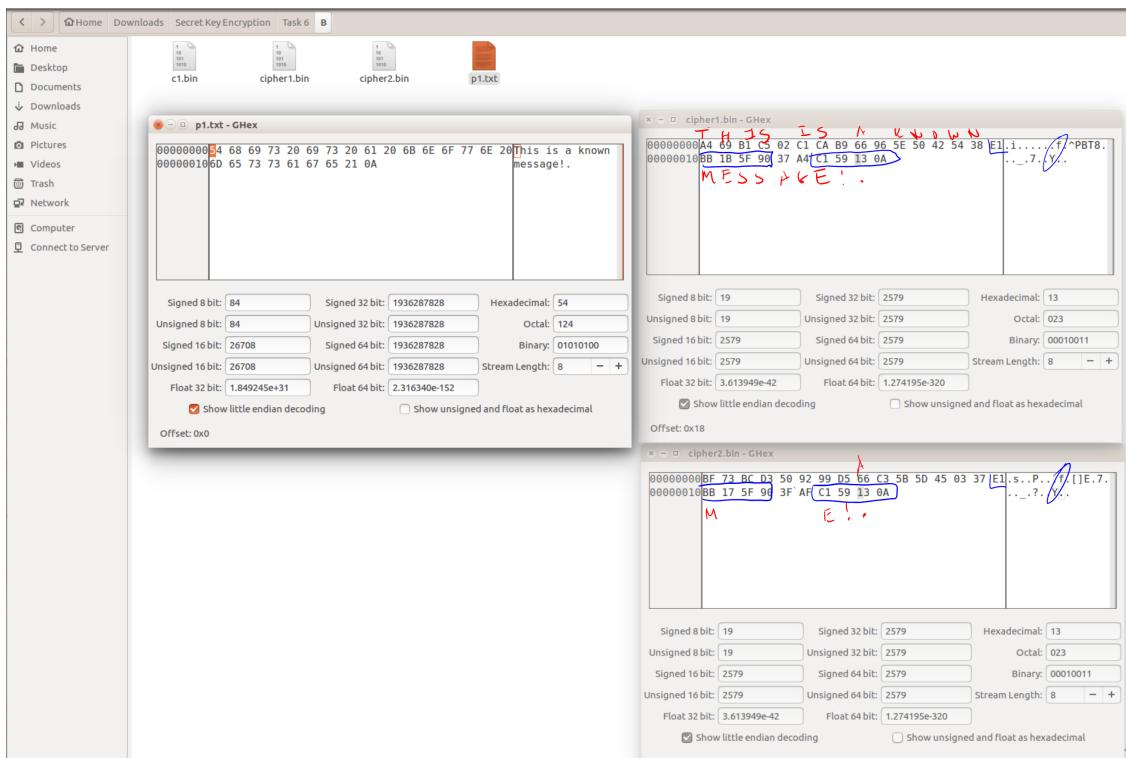
- First we compare the encryption of the same text file using two different IVs. For the first encryption, I used IV = 0102030405060708, but for the second encryption the IV = 01020304050607. I dropped the last two numbers. Both encryptions use the same plain text and key, just two different

keys. As shown below, the usage of unique IVs on the same plain text file produced two different encryptions.



The next picture shows that if you utilize the same IV, the encryption still remained the same for both. This is important because it indicates that the encryption is not secured. If we wish to encrypt the same text twice or more times, we do not want to make it obvious to the attackers.

B. Some argue that using a different plaintext with the same IV is safe. Thus, we are attempting to figure out plaintext 2 from the given information: plaintext 1 and ciphertext 1, and unknown plaintext 2 and cipher text 2. From the picture below, I tried examining the two cipher texts and realized there were a few similarities between them. For example, the ending trail of both ciphers are similar (outlined in blue). I tried mapping each character from known p1 to c1, but it doesn't seem to apply to c2. Either way, I'm sure a professional attack would be able to utilize this pair of (p1, c1) to figure out p2 from c2. OFB is suppose to a more safer form of encryption compared to CFB, which means that if I was able to see few similarities from using the same IV, then CFB would probably reveal even more.



[]: