Name: Mohammad Fakhruddin Babar

ID: M564K763

**Task 1: Encryption using Different Ciphers and Modes**

Cipher mode that I chose in this task:

1. -aes-128-cbc
2. -aes-128-ecb
3. -des-cbc
4. -des-ecb

Reason of Choosing:

Those cipher modes are discussed in class, and I wanted to familiarize myself with them by using the OpenSSL library.

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

1(a)Encrypted picture using ECB mode for both picture:



Fig: Encrypted picture for given image 1 in ECB mode

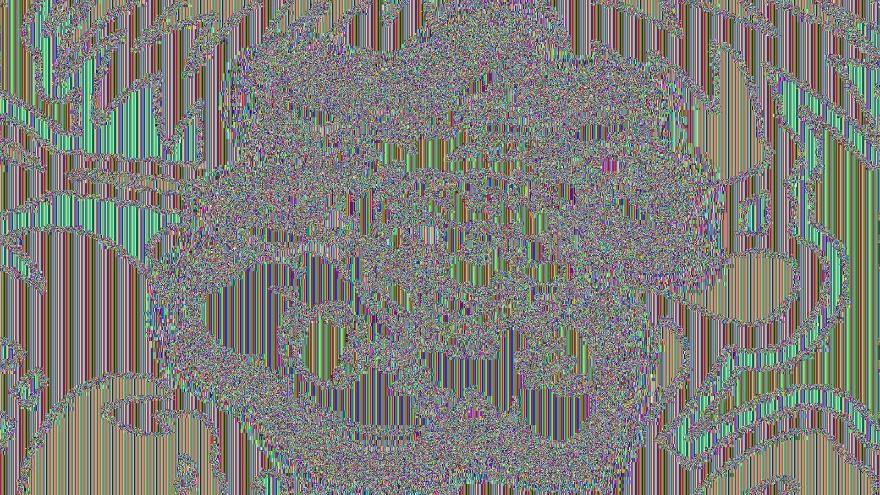


Fig: Encrypted picture for given image 2 in ECB mode

1(b): We can get a rough idea of the original picture's shape from the encrypted image, but not the color. Using ECB mode, however, an attacker can extract significant information from those encrypted images. As a result, this can't possibly be a good encryption mode.

2(a): Encrypted picture using CBC mode for both picture:



Fig: Encrypted picture for given image 1 in CBC mode



Fig: Encrypted picture for given image 2 in CBC mode

2(b) We can't tell what the original picture looked like or what color it was from the encrypted image. We have no idea what the photograph is about. As a result, an attacker won't be able to extract any useful information from the CBC-encrypted images. As a result, this could be a useful encryption mode.

**Task 3: Error Propagation – Corrupted Cipher Text**

**ECB Mode:**

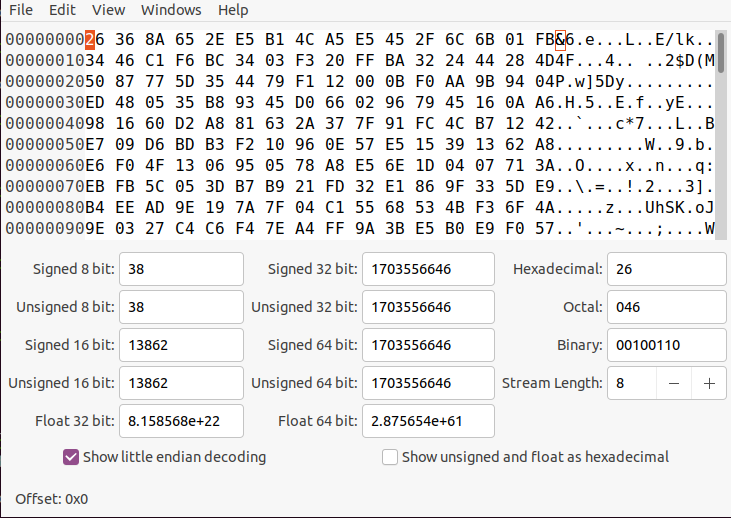
****

Fig: Original encrypted picture in ECB mode

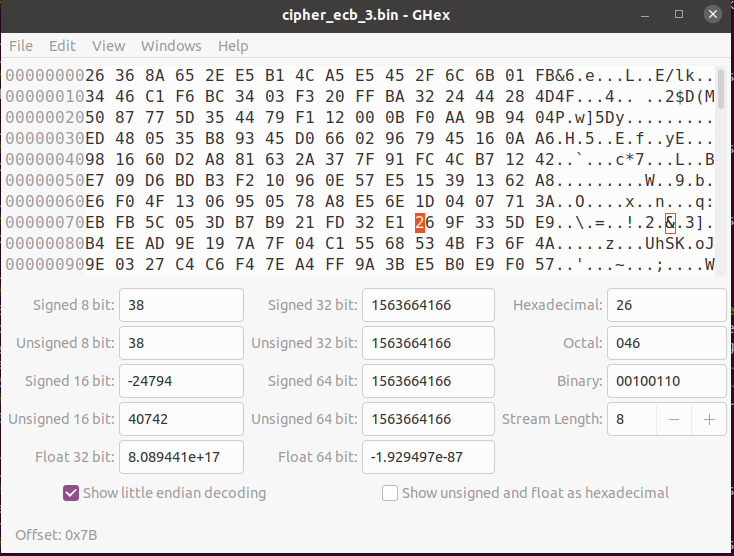


Fig: Corrupted file in ECB mode

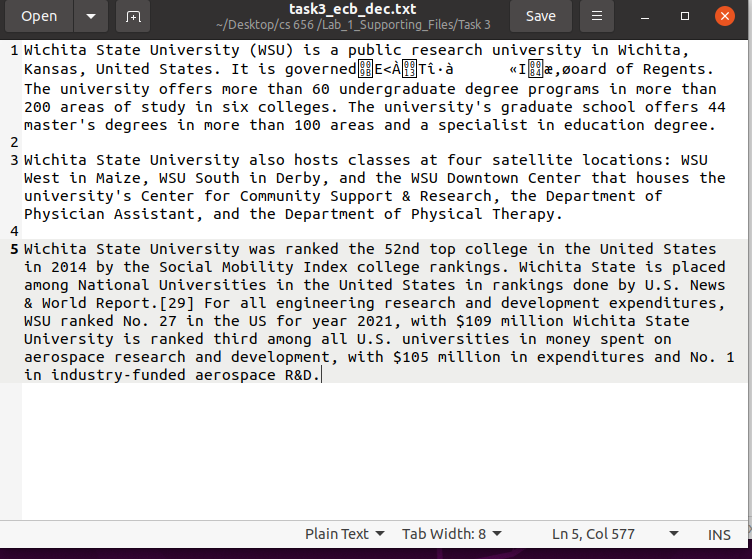


Fig: Decrypted file of corrupted encrypted file in ECB mode

**CBC Mode:**

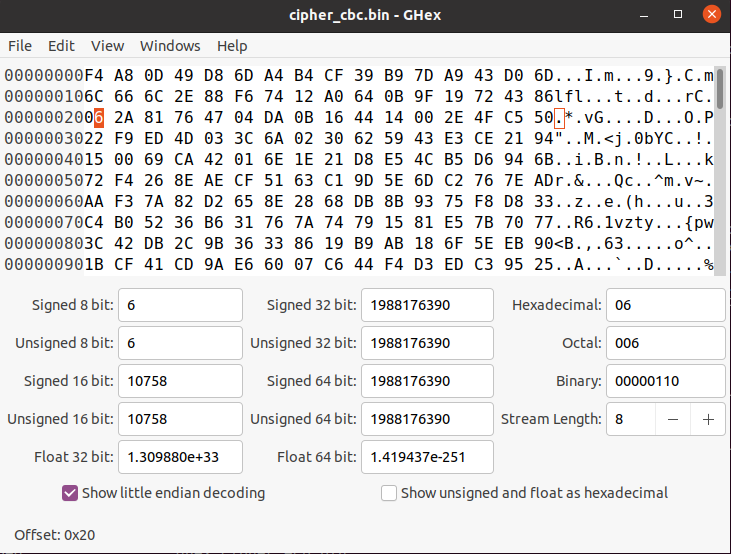
****

Fig: Original encrypted picture in CBC mode

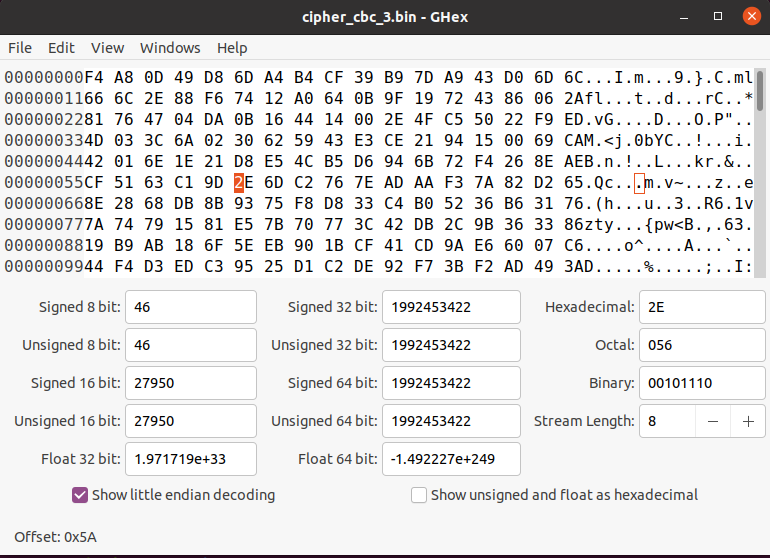


Fig: Corrupted file in CBC mode

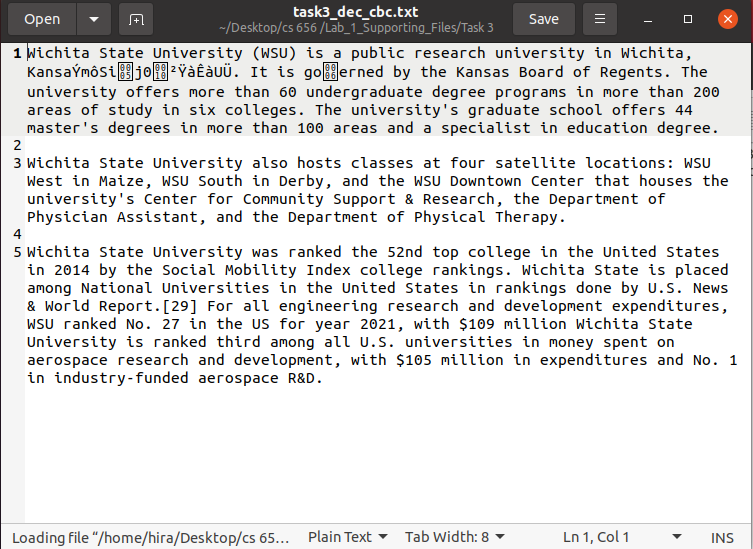


Fig: Decrypted file of corrupted encrypted file in CBC mode

CBC and ECB’s are block ciphers, and if we change a single bit in CBC or ECB, it will only affect the next one or two blocks. Before beginning this task, I reasoned that changing a single bit in the encrypted file would result in a minor data loss. And after completing this task, I've concluded that changing a single bit does not affect a lot and it changes the information a little bit. Which is matched with my previous assumption. I'm able to extract almost all of the data from the original file.

**Task 4: Brute-force Attack using the Crypto Library**

For this task I have used python crypto library to get the keys from the given English word list. And key is **Kansas**.

Plaintext, ciphertext, IV, and a wordlist are all provided. To get the key, I used the Python crypto library. To recover the key, I used AES-128-CBC. I padded each key in the given worldlist with '#' until it was 16 bytes(128 bits) long. Then I used that key to encrypt the given text, and then compared the new cipher text to the given cipher text. We stop checking and get the key if both the new cipher and the given cipher match.

Here is the code:

#Mohammad Fakhruddin Babar  
#M564K763  
#dictionary Attack with common word  
#CS656, Lab-1, Task-4  
  
from Crypto.Cipher import AES  
from Crypto.Util import Padding  
  
#given plaintext cipher text and iv  
plaintext=b"I will crack the key."  
#print(plaintext)  
cipher\_text="673ff1376c5b9a2e7cdbd7daf1faeb180bfa2b7ab150e25b7915525500a045fa"  
iv='0f0e0d0c0b0a00908070605040302010'  
  
#converting iv and ciphertext   
iv=bytes.fromhex(iv)  
cipher\_text=bytes.fromhex(cipher\_text)  
  
# Reading the wordlist file   
fp=open("task4\_wordlist.txt","r")  
#accessing the word line-wise  
wordlist\_linewise=fp.readlines()  
fp.close()  
  
for word in wordlist\_linewise:  
 #print(len(word))  
 #replacing new line with space ; then padding with #  
 word=word.replace("\n","")  
 if len(word) <= 16:  
 n=16-len(word)  
 key=word+'#'\*n  
 # print(key)  
 key=key.encode("ascii")  
 #print(key)  
 # creating the AES cipher  
 mode=AES.MODE\_CBC  
 cipher=AES.new(key,mode,iv)  
 # encrypt the plaintext with padding   
 cipher\_text\_new=cipher.encrypt(Padding.pad(plaintext,16))  
 l=0;  
 if cipher\_text ==cipher\_text\_new:  
 key=word  
 l=1  
 print("Key is found and key is:",word)  
 break  
  
#if key is not in the given list  
if(l==0):  
 print("Key is not found in given wordlist")