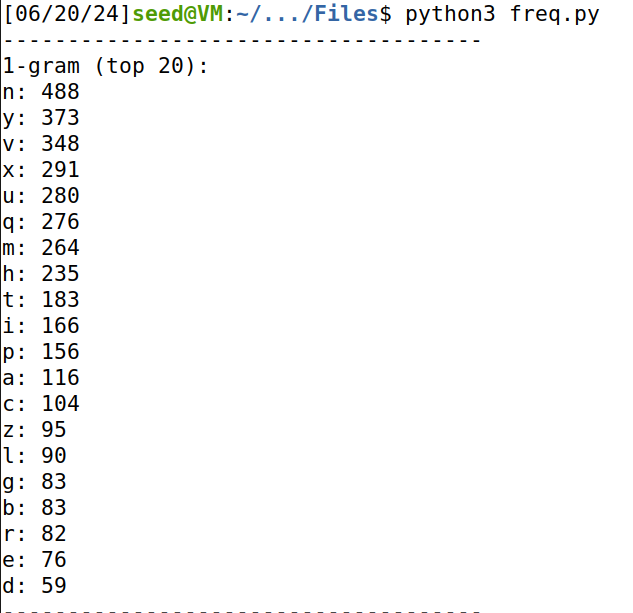
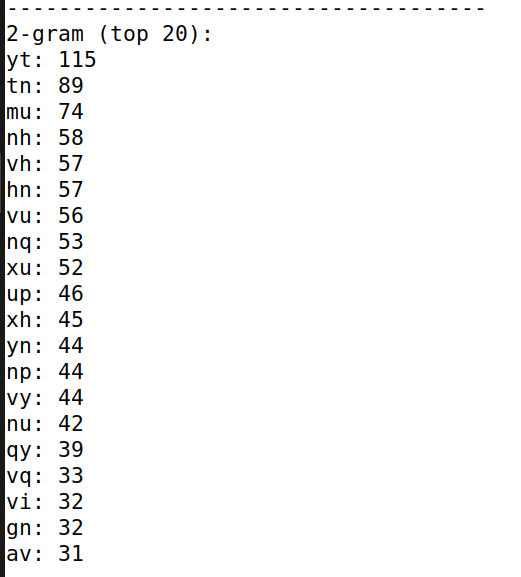
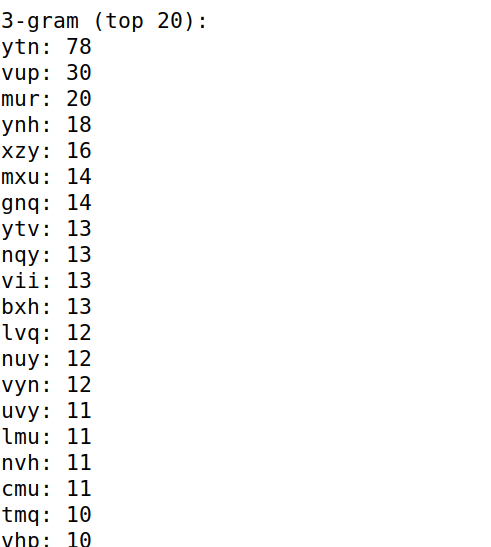
**Task 1: Frequency Analysis**

We obtained the following analysis of the one-gram, bi-gram, and tri-gram frequencies in the provided ciphertext by running the freq.py script.





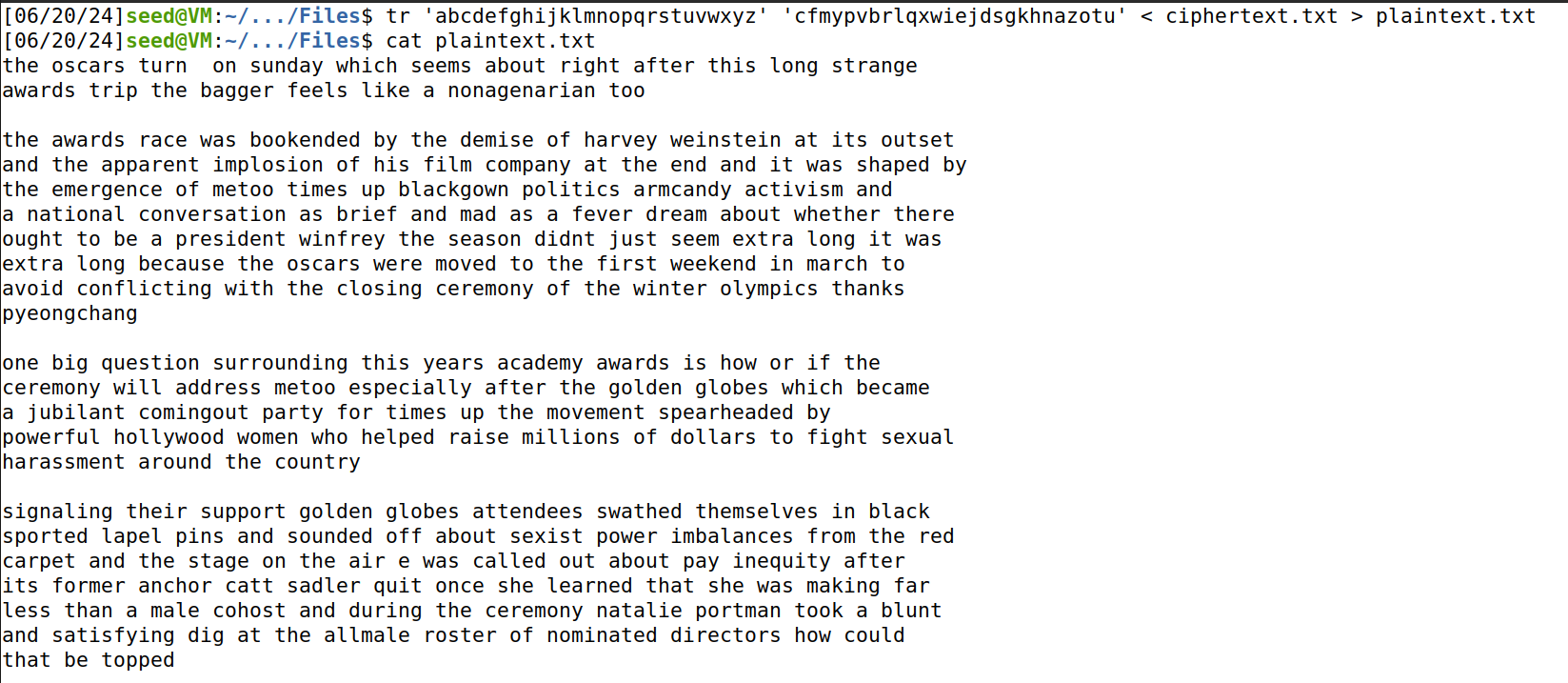


Consider 'y', 'yt', and 'ytn' sequences. These are commonly found, with 'ytn' frequently beginning paragraphs, according to the analysis above. Since paragraphs in English usually start with "the," we can deduce:   
Y as T, T as H, and N as E

In similar way, after considering and analyzing all the 1-gram, 2-gram and 3-gram we get the key as

c f m y p v b r l q x w i e j d s g k h n a z o t u

Now, let us use the key and decrypt the cipher text. To decrypt the cipher text the command is   
>tr ‘abcdefghijklmnopqrstuvwxyz’ ‘cfmypvbrlqxwiejdsgkhnazotu’ < ciphertext.txt > plaintext.txt



The cipher text is decrypted and saved in the file named plaintext.txt

**Task 2 : Encryption using Different Cipher and Modes**

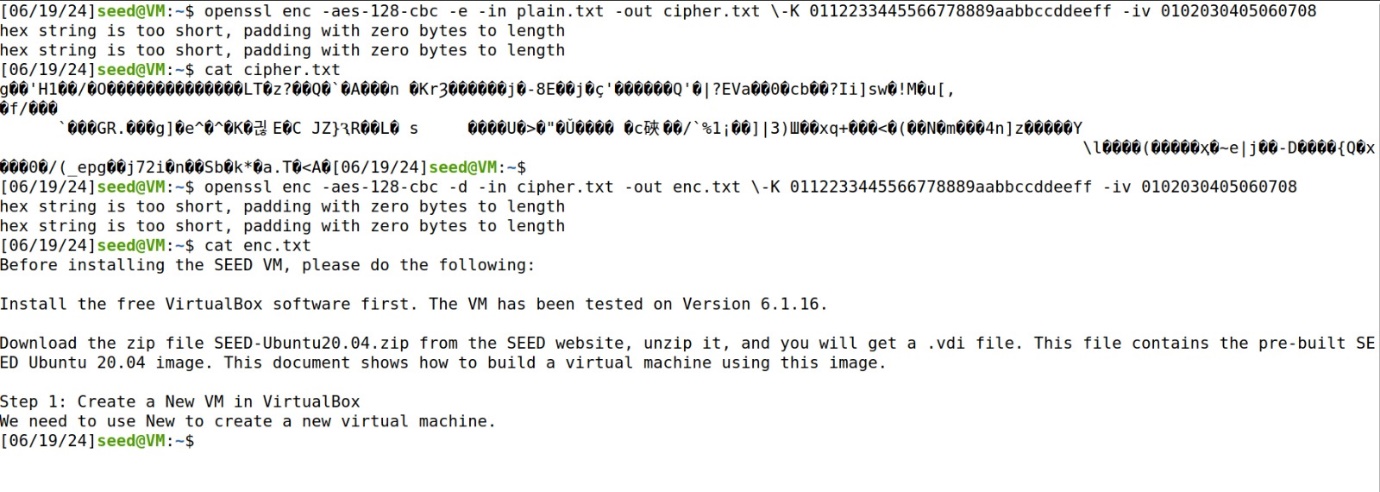
In the command below, we use different cipher modes

openssl enc -ciphertype -e -in plain.txt -out cipher.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708

-e for encryption and -d for decryption  
  
They are

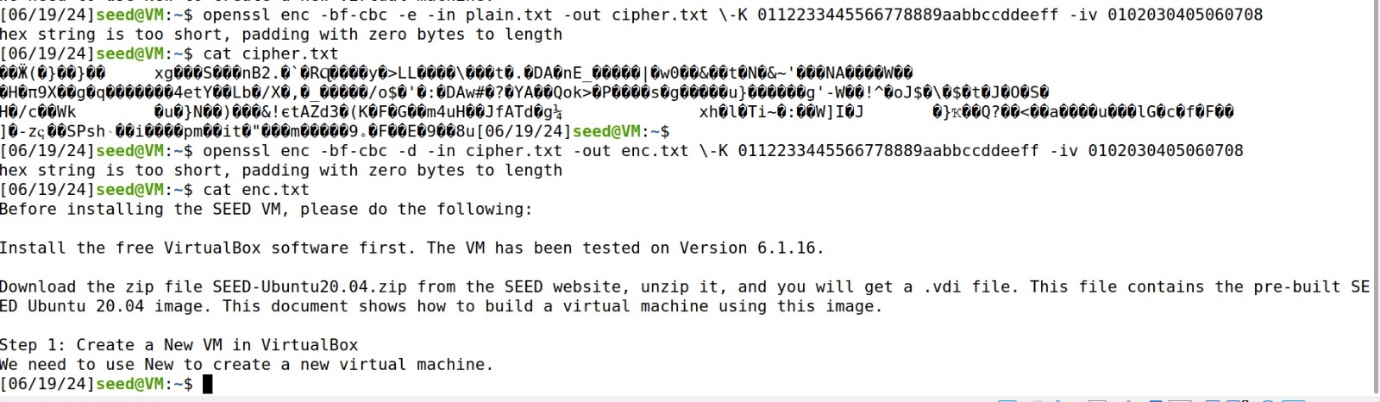
1) -aes-128-cbc

Replacing -ciphertype with “-aes-128-cbc”



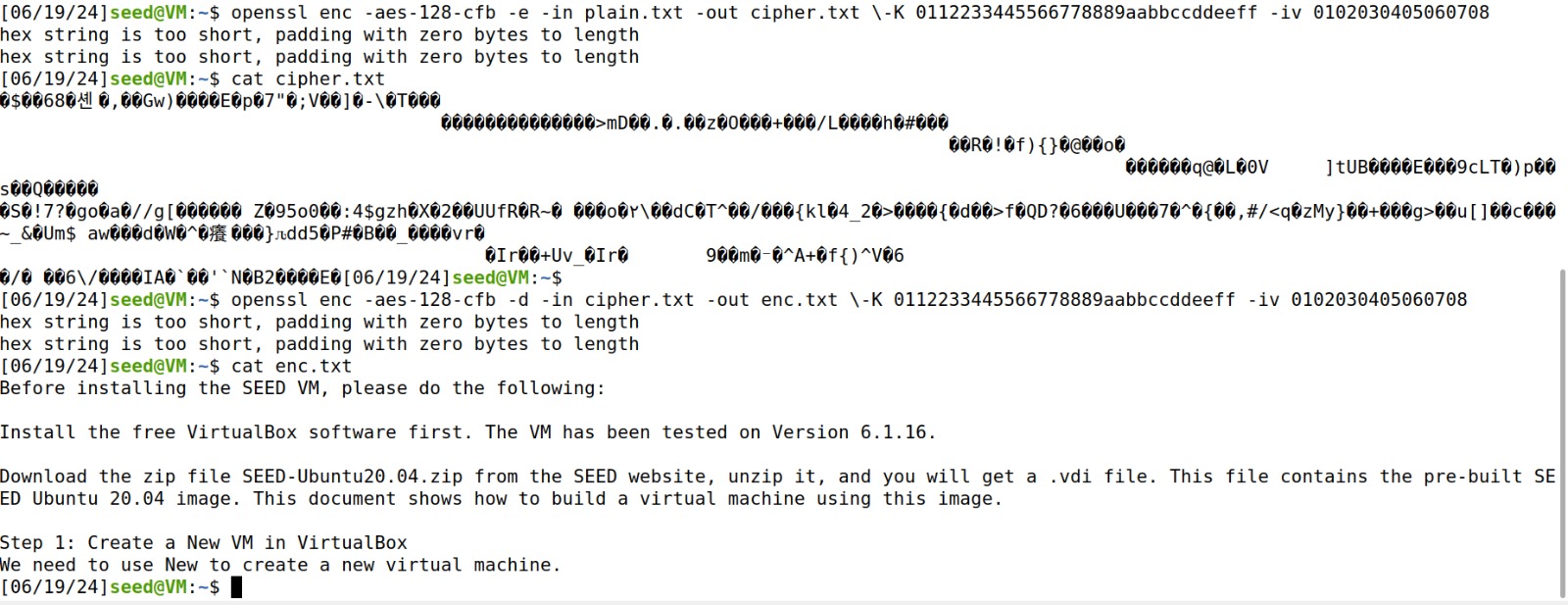
2) -bf-cbc

Replacing -ciphertype with “-bf-cbc”



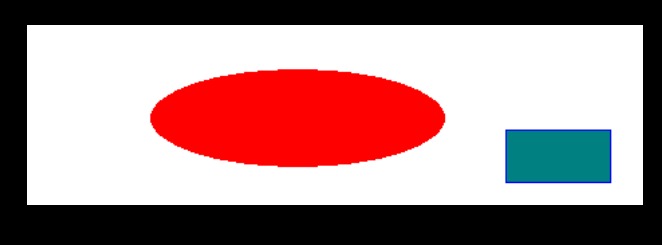
3) -aes-128-cfb

Replacing -ciphertype with -aes-128-cfb



**Task 3 : Encryption Mode – ECB vs CBC**

This is the image(pic\_original.bmp) provided in the Labsetup.zip file



First we perform **CBC** (Cipher Block Chaining)

We will use the command mentioned below to encrypt the above image

openssl enc -aes-128-cbc -e -in pic\_original.bmp -out cipher.bmp -K 00112233445566778889aabbccddeeff -iv 0102030405060708

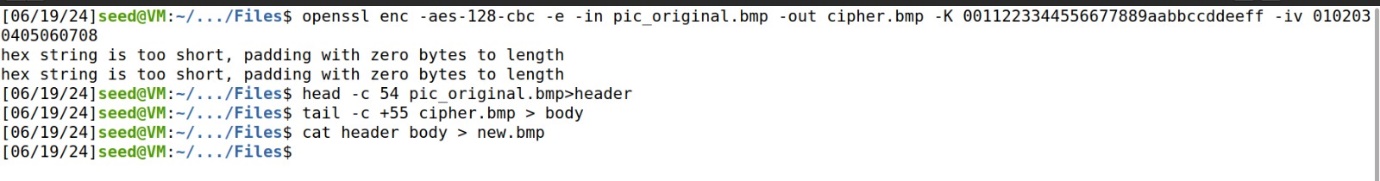
and we need to generate header and body and combine them together to get a new file

commands used to generate header, footer and new file are

head -c 54 pic\_original.bmp > header

tail -c +55 cipher.bmp > body

cat header body > new.bmp



This is the encrypted image which is saved as new.bmp



Now we perform **ECB** (Electronic Code Block)

We will use the command mentioned below to encrypt pic\_original.bmp

openssl enc -aes-128-ecb -e -in pic\_original.bmp -out cipher2.bmp -K 00112233445566778889aabbccddeeff

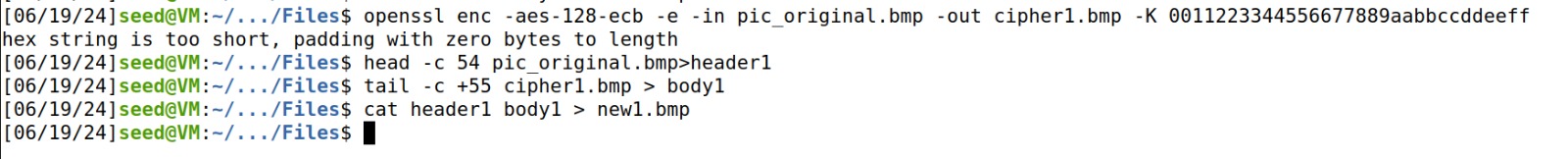
and we need to generate header and body and combine them together to get a new file

commands used to generate header, footer and new file are

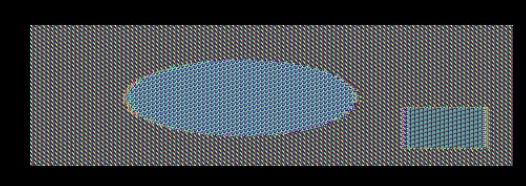
head -c 54 pic\_original.bmp > header1

tail -c +55 cipher1.bmp > body1

cat header1 body1 > new1.bmp



This is the encrypted image which is saved as new1.bmp



**CBC Mode Encryption**

Each block of plaintext is XORed with the previous ciphertext block before being encrypted. An initialization vector (IV) is used for the first block.

The encrypted image appears as completely random noise with no visible patterns. This is expected behaviour for CBC mode, as it effectively hides any patterns in the plaintext data, making it highly secure against visual attacks.

The random noise indicates that CBC mode is effective in ensuring that the encrypted data does not reveal any patterns from the original image.

**ECB Mode Encryption**

Each block of plaintext is encrypted independently using the same key.

The encrypted image shows some pattern and structure related to the original image. Identical plaintext blocks are encrypted into identical ciphertext blocks, resulting in visible artifacts that reveal the structure of the original image.

This indicates that ECB mode is not secure for encrypting images or data with repeating patterns, as it fails to hide the structure of the original data effectively.

Let us select a picture named ab.bmp and lets encrypt it



First we perform **CBC** (Cipher Block Chaining)

We will use the command mentioned below to encrypt the above image

openssl enc -aes-128-cbc -e -in ab.bmp -out ab1.bmp -K 00112233445566778889aabbccddeeff -iv 0102030405060708

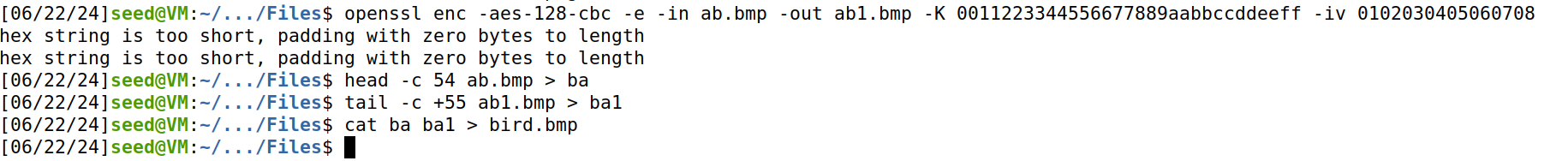
and we need to generate header and body and combine them together to get a new file

commands used to generate header, footer and new file are

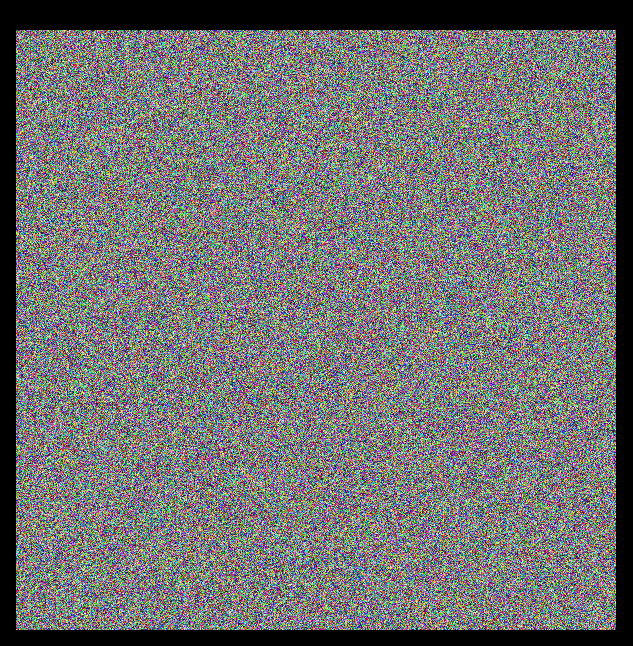
head -c 54 ab.bmp > ba

tail -c +55 ab1.bmp > ba1

cat ba ba1 > bird.bmp



This is the encrypted image which is saved as bird.bmp



Now we perform **ECB** (Electronic Code Block)

We will use the command mentioned below to encrypt ab.bmp

openssl enc -aes-128-ecb -e -in ab.bmp -out ab2.bmp -K 00112233445566778889aabbccddeeff

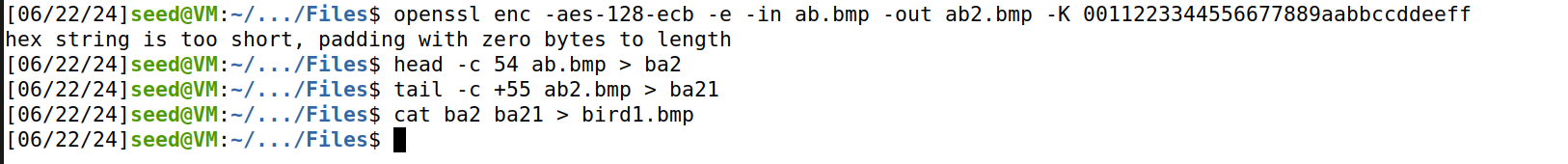
and we need to generate header and body and combine them together to get a new file

commands used to generate header, footer and new file are

head -c 54 ab.bmp > ba2

tail -c +55 ab2.bmp > ba21

cat ba2 ba21 > bird1.bmp



This is the encrypted image which is saved as bird1.bmp

