**Secret-Key Encryption**

Chawin Metah

Department of Computer Science, Purdue University Fort Wayne

ACS 54500: Cryptography and Network Security

Lab 7

Dr. Zesheng Chen

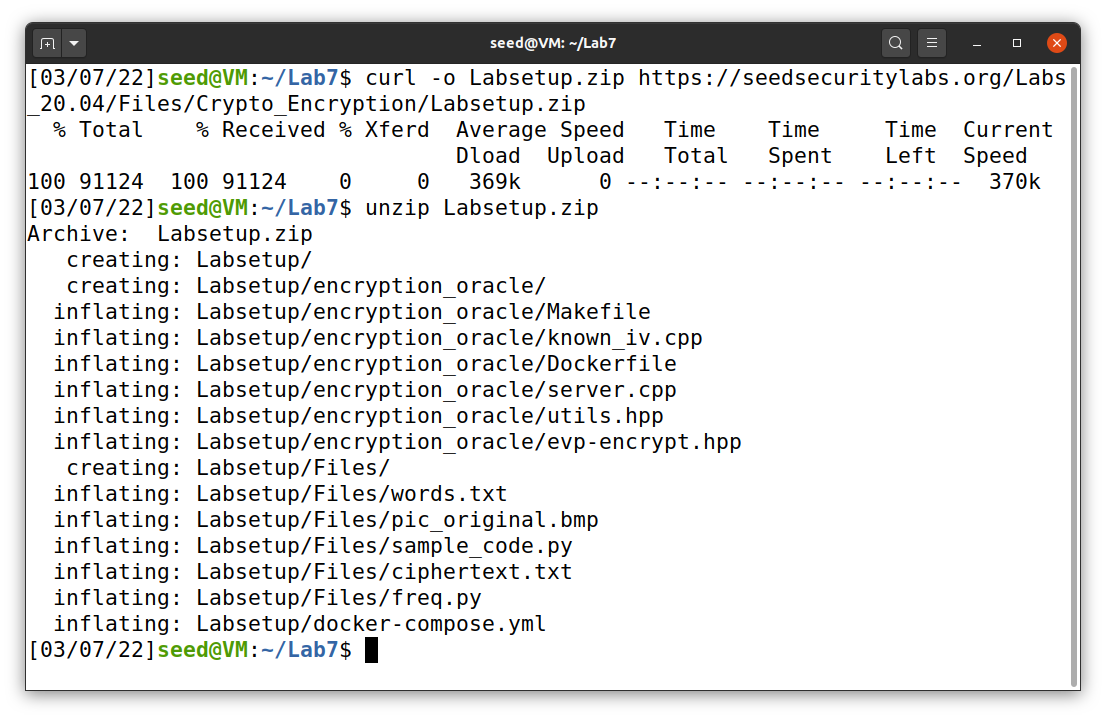
March 18, 2022

**Task 1: Frequency Analysis**

Similar to the previous labs, we download and unzip Labsetup.zip, which contains docker-compose.yml and other directories (Figure 1). It can be downloaded with the following command: curl -o Labsetup.zip https://seedsecuritylabs.org/Labs\_20.04/‌Files/Crypto\_Encryption/Labsetup.zip. I used the unzip command to extract the file.

**Figure 1**

*Labsetup.zip*

**

There are two files that I used in this task: ciphertext.txt and freq.py. Both files are provided in Labsetup under “Files” directory. The ciphertext has its content as follow:

“ytn xqavhq yzhu xu qzupvd ltmat qnncq vgxzy hmrty vbynh ytmq ixur qyhvurn vlvhpq yhme ytn gvrrnh bnniq imsn v uxuvrnuvhmvu yxx

ytn vlvhpq hvan lvq gxxsnupnp gd ytn pncmqn xb tvhfnd lnmuqynmu vy myq xzyqny vup ytn veevhnuy mceixqmxu xb tmq bmic axcevud vy ytn nup vup my lvq qtvenp gd ytn ncnhrnuan xb cnyxx ymcnq ze givasrxlu eximymaq vhcavupd vaymfmqc vup v uvymxuvi axufnhqvymxu vq ghmnb vup cvp vq v bnfnh phnvc vgxzy ltnytnh ytnhn xzrty yx gn v ehnqmpnuy lmubhnd ytn qnvqxu pmpuy ozqy qnnc nkyhv ixur my lvq nkyhv ixur gnavzqn ytn xqavhq lnhn cxfnp yx ytn bmhqy lnnsnup mu cvhat yx vfxmp axubimaymur lmyt ytn aixqmur anhncxud xb ytn lmuynh xidcemaq ytvusq ednxuratvur

xun gmr jznqymxu qzhhxzupmur ytmq dnvhq vavpncd vlvhpq mq txl xh mb ytn

anhncxud lmii vpphnqq cnyxx nqenamviid vbynh ytn rxipnu rixgnq ltmat gnavcn

v ozgmivuy axcmurxzy evhyd bxh ymcnq ze ytn cxfncnuy qenvhtnvpnp gd

exlnhbzi txiidlxxp lxcnu ltx tnienp hvmqn cmiimxuq xb pxiivhq yx bmrty qnkzvi

tvhvqqcnuy vhxzup ytn axzuyhd

qmruvimur ytnmh qzeexhy rxipnu rixgnq vyynupnnq qlvytnp ytncqnifnq mu givas

qexhynp iveni emuq vup qxzupnp xbb vgxzy qnkmqy exlnh mcgvivuanq bhxc ytn hnp

avheny vup ytn qyvrn xu ytn vmh n lvq aviinp xzy vgxzy evd munjzmyd vbynh

myq bxhcnh vuatxh avyy qvpinh jzmy xuan qtn invhunp ytvy qtn lvq cvsmur bvh

inqq ytvu v cvin axtxqy vup pzhmur ytn anhncxud uvyvimn exhycvu yxxs v gizuy

vup qvymqbdmur pmr vy ytn viicvin hxqynh xb uxcmuvynp pmhnayxhq txl axzip

ytvy gn yxeenp

vq my yzhuq xzy vy invqy mu ynhcq xb ytn xqavhq my ehxgvgid lxuy gn

lxcnu mufxifnp mu ymcnq ze qvmp ytvy viytxzrt ytn rixgnq qmrumbmnp ytn mumymvymfnq ivzuat ytnd unfnh muynupnp my yx gn ozqy vu vlvhpq qnvqxu avcevmru xh xun ytvy gnavcn vqqxamvynp xuid lmyt hnpavheny vaymxuq muqynvp v qexsnqlxcvu qvmp ytn rhxze mq lxhsmur gntmup aixqnp pxxhq vup tvq qmuan vcvqqnp cmiimxu bxh myq inrvi pnbnuqn bzup ltmat vbynh ytn rixgnq lvq bixxpnp lmyt ytxzqvupq xb pxuvymxuq xb xh inqq bhxc enxein mu qxcn axzuyhmnq

ux avii yx lnvh givas rxluq lnuy xzy mu vpfvuan xb ytn xqavhq ytxzrt ytn cxfncnuy lmii vicxqy anhyvmuid gn hnbnhnuanp gnbxhn vup pzhmur ytn anhncxud nqenamviid qmuan fxavi cnyxx qzeexhynhq imsn vqtind ozpp ivzhv pnhu vup umaxin smpcvu vhn qatnpzinp ehnqnuynhq

vuxytnh bnvyzhn xb ytmq qnvqxu ux xun hnviid suxlq ltx mq rxmur yx lmu gnqy emayzhn vhrzvgid ytmq tveenuq v ixy xb ytn ymcn muvhrzvgid ytn uvmigmynh uvhhvymfn xuid qnhfnq ytn vlvhpq tden cvatmun gzy xbynu ytn enxein bxhnavqymur ytn hvan qxaviinp xqavhxixrmqyq avu cvsn xuid npzavynp rznqqnq

ytn lvd ytn vavpncd yvgzivynq ytn gmr lmuunh pxnquy tnie mu nfnhd xytnh avynrxhd ytn uxcmunn lmyt ytn cxqy fxynq lmuq gzy mu ytn gnqy emayzhn avynrxhd fxynhq vhn vqsnp yx imqy ytnmh yxe cxfmnq mu ehnbnhnuymvi xhpnh mb v cxfmn rnyq cxhn ytvu enhanuy xb ytn bmhqyeivan fxynq my lmuq ltnu ux cxfmn cvuvrnq ytvy ytn xun lmyt ytn bnlnqy bmhqyeivan fxynq mq nimcmuvynp vup

myq fxynq vhn hnpmqyhmgzynp yx ytn cxfmnq ytvy rvhunhnp ytn nimcmuvynp gviixyq

qnaxupeivan fxynq vup ytmq axuymuznq zuymi v lmuunh ncnhrnq

my mq vii ynhhmgid axubzqmur gzy veevhnuyid ytn axuqnuqzq bvfxhmyn axcnq xzy vtnvp mu ytn nup ytmq cnvuq ytvy nupxbqnvqxu vlvhpq atvyynh mufvhmvgid mufxifnq yxhyzhnp qenazivymxu vgxzy ltmat bmic lxzip cxqy imsnid gn fxynhq qnaxup xh ytmhp bvfxhmyn vup ytnu njzviid yxhyzhnp axuaizqmxuq vgxzy ltmat bmic cmrty ehnfvmi

mu my lvq v yxqqze gnylnnu gxdtxxp vup ytn nfnuyzvi lmuunh gmhpcvu mu lmyt ixyq xb nkenhyq gnyymur xu ytn hnfnuvuy xh ytn gmr qtxhy ytn ehmwn lnuy yx qexyimrty ivqy dnvh unvhid vii ytn bxhnavqynhq pnaivhnp iv iv ivup ytn ehnqzceymfn lmuunh vup bxh ylx vup v tvib cmuzynq ytnd lnhn axhhnay gnbxhn vu nufnixen quvbz lvq hnfnvinp vup ytn hmrtybzi lmuunh cxxuimrty lvq ahxlunp

ytmq dnvh vlvhpq lvyatnhq vhn zunjzviid pmfmpnp gnylnnu ythnn gmiigxvhpq xzyqmpn nggmur cmqqxzhm ytn bvfxhmyn vup ytn qtven xb lvynh ltmat mq ytn gvrrnhq ehnpmaymxu lmyt v bnl bxhnavqymur v tvmi cvhd lmu bxh rny xzy

gzy vii xb ytxqn bmicq tvfn tmqyxhmavi xqavhfxymur evyynhuq vrvmuqy ytnc ytn qtven xb lvynh tvq uxcmuvymxuq cxhn ytvu vud xytnh bmic vup lvq viqx uvcnp ytn dnvhq gnqy gd ytn ehxpzanhq vup pmhnayxhq rzmipq dny my lvq uxy uxcmuvynp bxh v qahnnu vayxhq rzmip vlvhp bxh gnqy nuqncgin vup ux bmic tvq lxu gnqy emayzhn lmytxzy ehnfmxzqid ivupmur vy invqy ytn vayxhq uxcmuvymxu

qmuan ghvfntnvhy mu ytmq dnvh ytn gnqy nuqncgin qvr nupnp ze rxmur yx ythnn gmiigxvhpq ltmat mq qmrumbmavuy gnavzqn vayxhq cvsn ze ytn vavpncdq ivhrnqy ghvuat ytvy bmic ltmin pmfmqmfn viqx lxu ytn gnqy phvcv rxipnu rixgn vup ytn gvbyv gzy myq bmiccvsnh cvhymu capxuvrt lvq uxy uxcmuvynp bxh gnqy pmhnayxh vup vevhy bhxc vhrx cxfmnq ytvy ivup gnqy emayzhn lmytxzy viqx nvhumur gnqy pmhnayxh uxcmuvymxuq vhn bnl vup bvh gnylnnu”

Obviously, the text file is unreadable. So, we have to decipher it by using the below command. Also, in order to know the key, I use freq.py to find the frequency of encrypted characters and compare them with the most frequently used character in English, which can be found on Wikipedia. The outputs of freq.py are shown in Figure 2. I tried to decrypt with Trigram frequency, Bigram frequency. Then, I deciphered the character one by one.

tr ’abcdefg’ ’ABCDEFG’ < ciphertext.txt > out.txt

**Figure 2**

*freq.py*

**

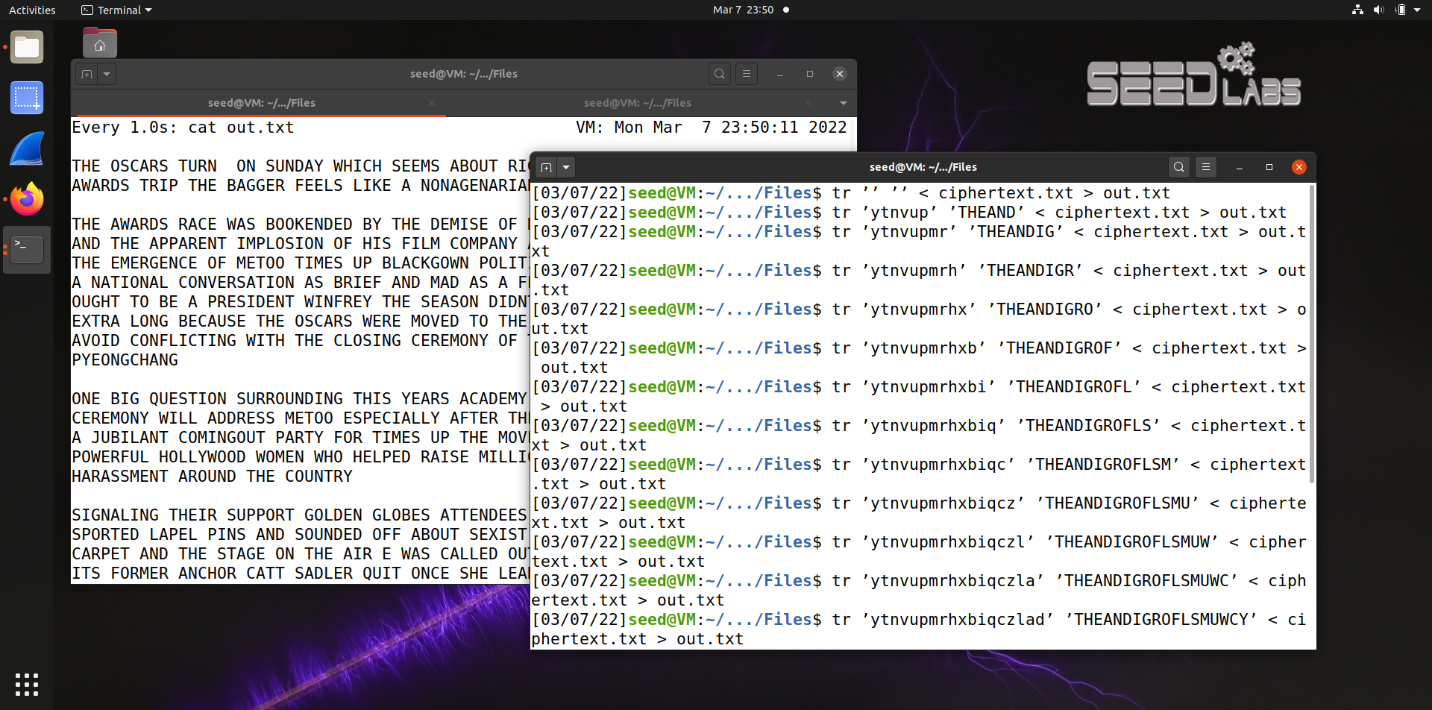
To view the result of the encryption, I used the command below.

watch -n 1 cat out.txt

The final command that I got is tr ’ytnvupmrhxbiqczladgesfjkow’ ’THEANDIGROFLSMUWCYBPKVQXJZ’ < ciphertext.txt > out.txt. The results are shown in Figure 3.

**Figure 3**

*The result of task 1*

**

Finally, the decrypted text is as follow:

THE OSCARS TURN ON SUNDAY WHICH SEEMS ABOUT RIGHT AFTER THIS LONG STRANGE AWARDS TRIP THE BAGGER FEELS LIKE A NONAGENARIAN TOO

THE AWARDS RACE WAS BOOKENDED BY THE DEMISE OF HARVEY WEINSTEIN AT ITS OUTSET AND THE APPARENT IMPLOSION OF HIS FILM COMPANY AT THE END AND IT WAS SHAPED BY THE EMERGENCE OF METOO TIMES UP BLACKGOWN POLITICS ARMCANDY ACTIVISM AND A NATIONAL CONVERSATION AS BRIEF AND MAD AS A FEVER DREAM ABOUT WHETHER THERE OUGHT TO BE A PRESIDENT WINFREY THE SEASON DIDNT JUST SEEM EXTRA LONG IT WAS EXTRA LONG BECAUSE THE OSCARS WERE MOVED TO THE FIRST WEEKEND IN MARCH TO AVOID CONFLICTING WITH THE CLOSING CEREMONY OF THE WINTER OLYMPICS THANKS PYEONGCHANG

ONE BIG QUESTION SURROUNDING THIS YEARS ACADEMY AWARDS IS HOW OR IF THE CEREMONY WILL ADDRESS METOO ESPECIALLY AFTER THE GOLDEN GLOBES WHICH BECAME A JUBILANT COMINGOUT PARTY FOR TIMES UP THE MOVEMENT SPEARHEADED BY POWERFUL HOLLYWOOD WOMEN WHO HELPED RAISE MILLIONS OF DOLLARS TO FIGHT SEXUAL HARASSMENT AROUND THE COUNTRY

SIGNALING THEIR SUPPORT GOLDEN GLOBES ATTENDEES SWATHED THEMSELVES IN BLACK SPORTED LAPEL PINS AND SOUNDED OFF ABOUT SEXIST POWER IMBALANCES FROM THE RED CARPET AND THE STAGE ON THE AIR E WAS CALLED OUT ABOUT PAY INEQUITY AFTER ITS FORMER ANCHOR CATT SADLER QUIT ONCE SHE LEARNED THAT SHE WAS MAKING FAR LESS THAN A MALE COHOST AND DURING THE CEREMONY NATALIE PORTMAN TOOK A BLUNT AND SATISFYING DIG AT THE ALLMALE ROSTER OF NOMINATED DIRECTORS HOW COULD THAT BE TOPPED

AS IT TURNS OUT AT LEAST IN TERMS OF THE OSCARS IT PROBABLY WONT BE

WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SIGNIFIED THE INITIATIVES LAUNCH THEY NEVER INTENDED IT TO BE JUST AN AWARDS SEASON CAMPAIGN OR ONE THAT BECAME ASSOCIATED ONLY WITH REDCARPET ACTIONS INSTEAD A SPOKESWOMAN SAID THE GROUP IS WORKING BEHIND CLOSED DOORS AND HAS SINCE AMASSED MILLION FOR ITS LEGAL DEFENSE FUND WHICH AFTER THE GLOBES WAS FLOODED WITH THOUSANDS OF DONATIONS OF OR LESS FROM PEOPLE IN SOME COUNTRIES

NO CALL TO WEAR BLACK GOWNS WENT OUT IN ADVANCE OF THE OSCARS THOUGH THE MOVEMENT WILL ALMOST CERTAINLY BE REFERENCED BEFORE AND DURING THE CEREMONY ESPECIALLY SINCE VOCAL METOO SUPPORTERS LIKE ASHLEY JUDD LAURA DERN AND NICOLE KIDMAN ARE SCHEDULED PRESENTERS

ANOTHER FEATURE OF THIS SEASON NO ONE REALLY KNOWS WHO IS GOING TO WIN BEST PICTURE ARGUABLY THIS HAPPENS A LOT OF THE TIME INARGUABLY THE NAILBITER NARRATIVE ONLY SERVES THE AWARDS HYPE MACHINE BUT OFTEN THE PEOPLE FORECASTING THE RACE SOCALLED OSCAROLOGISTS CAN MAKE ONLY EDUCATED GUESSES

THE WAY THE ACADEMY TABULATES THE BIG WINNER DOESNT HELP IN EVERY OTHER CATEGORY THE NOMINEE WITH THE MOST VOTES WINS BUT IN THE BEST PICTURE

CATEGORY VOTERS ARE ASKED TO LIST THEIR TOP MOVIES IN PREFERENTIAL ORDER IF A MOVIE GETS MORE THAN PERCENT OF THE FIRSTPLACE VOTES IT WINS WHEN NO MOVIE MANAGES THAT THE ONE WITH THE FEWEST FIRSTPLACE VOTES IS ELIMINATED AND ITS VOTES ARE REDISTRIBUTED TO THE MOVIES THAT GARNERED THE ELIMINATED BALLOTS SECONDPLACE VOTES AND THIS CONTINUES UNTIL A WINNER EMERGES

IT IS ALL TERRIBLY CONFUSING BUT APPARENTLY THE CONSENSUS FAVORITE COMES OUT AHEAD IN THE END THIS MEANS THAT ENDOFSEASON AWARDS CHATTER INVARIABLY INVOLVES TORTURED SPECULATION ABOUT WHICH FILM WOULD MOST LIKELY BE VOTERS SECOND OR THIRD FAVORITE AND THEN EQUALLY TORTURED CONCLUSIONS ABOUT WHICH FILM MIGHT PREVAIL

IN IT WAS A TOSSUP BETWEEN BOYHOOD AND THE EVENTUAL WINNER BIRDMAN IN WITH LOTS OF EXPERTS BETTING ON THE REVENANT OR THE BIG SHORT THE PRIZE WENT TO SPOTLIGHT LAST YEAR NEARLY ALL THE FORECASTERS DECLARED LA LA LAND THE PRESUMPTIVE WINNER AND FOR TWO AND A HALF MINUTES THEY WERE CORRECT BEFORE AN ENVELOPE SNAFU WAS REVEALED AND THE RIGHTFUL WINNER MOONLIGHT WAS CROWNED

THIS YEAR AWARDS WATCHERS ARE UNEQUALLY DIVIDED BETWEEN THREE BILLBOARDS

OUTSIDE EBBING MISSOURI THE FAVORITE AND THE SHAPE OF WATER WHICH IS THE BAGGERS PREDICTION WITH A FEW FORECASTING A HAIL MARY WIN FOR GET OUT

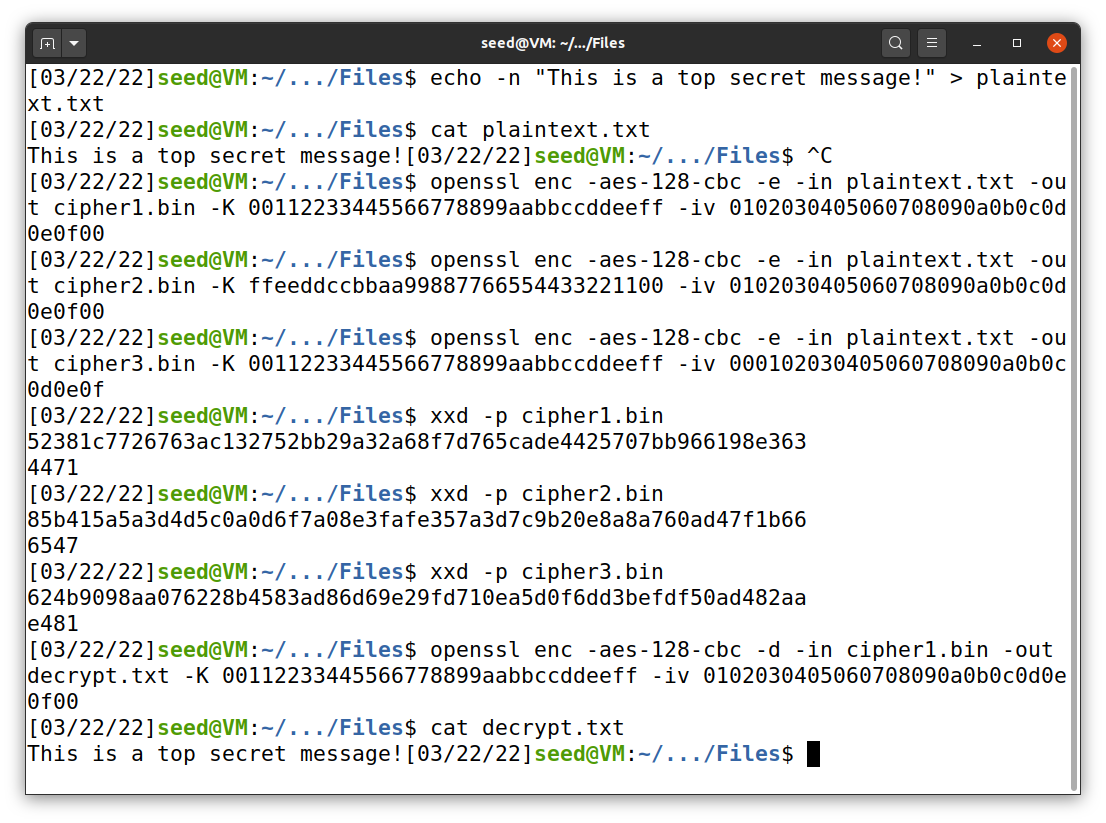
BUT ALL OF THOSE FILMS HAVE HISTORICAL OSCARVOTING PATTERNS AGAINST THEM THE SHAPE OF WATER HAS NOMINATIONS MORE THAN ANY OTHER FILM AND WAS ALSO NAMED THE YEARS BEST BY THE PRODUCERS AND DIRECTORS GUILDS YET IT WAS NOT NOMINATED FOR A SCREEN ACTORS GUILD AWARD FOR BEST ENSEMBLE AND NO FILM HAS WON BEST PICTURE WITHOUT PREVIOUSLY LANDING AT LEAST THE ACTORS NOMINATION SINCE BRAVEHEART IN THIS YEAR THE BEST ENSEMBLE SAG ENDED UP GOING TO THREE BILLBOARDS WHICH IS SIGNIFICANT BECAUSE ACTORS MAKE UP THE ACADEMYS LARGEST BRANCH THAT FILM WHILE DIVISIVE ALSO WON THE BEST DRAMA GOLDEN GLOBE AND THE BAFTA BUT ITS FILMMAKER MARTIN MCDONAGH WAS NOT NOMINATED FOR BEST DIRECTOR AND APART FROM ARGO MOVIES THAT LAND BEST PICTURE WITHOUT ALSO EARNING BEST DIRECTOR NOMINATIONS ARE FEW AND FAR BETWEEN”

All files and output results are also available in my repository.

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

In this task, I used a lot of openssl enc commands. So, I will not post these commands but please refer to the Figures below. First, I used aes-128-cbc encryption mode with different keys and IVs. The output of cipher1.bin, cipher2.bin, and cipher2.bin are totally different (Figure 4). Moreover, I also use one decryption command to get the original message just to test it out.

**Figure 4**

*aes-128-cbc encryption *

Next, I used aes-192-cfb and aes-192-ofb encryption mode to encrypt the same message, as shown in Figure 5. Because of the structure of CFB and OFB encryption, the output files have the same (32) hexadecimal digits at the start but different ones at the end. The original message is too short, so the program will automatically pad the data for us.

**Figure 5**

*aes-192-cfb and aes-192-ofb encryption*



Finally, I tried to use other encryption modes, which are aria-128-ecb and bf-cbc. The result is shown in Figure 6.

**Figure 6**

*aria-128-ecb and bf-cbc encryption*



**Task 6: Initial Vector (IV) and Common Mistakes**

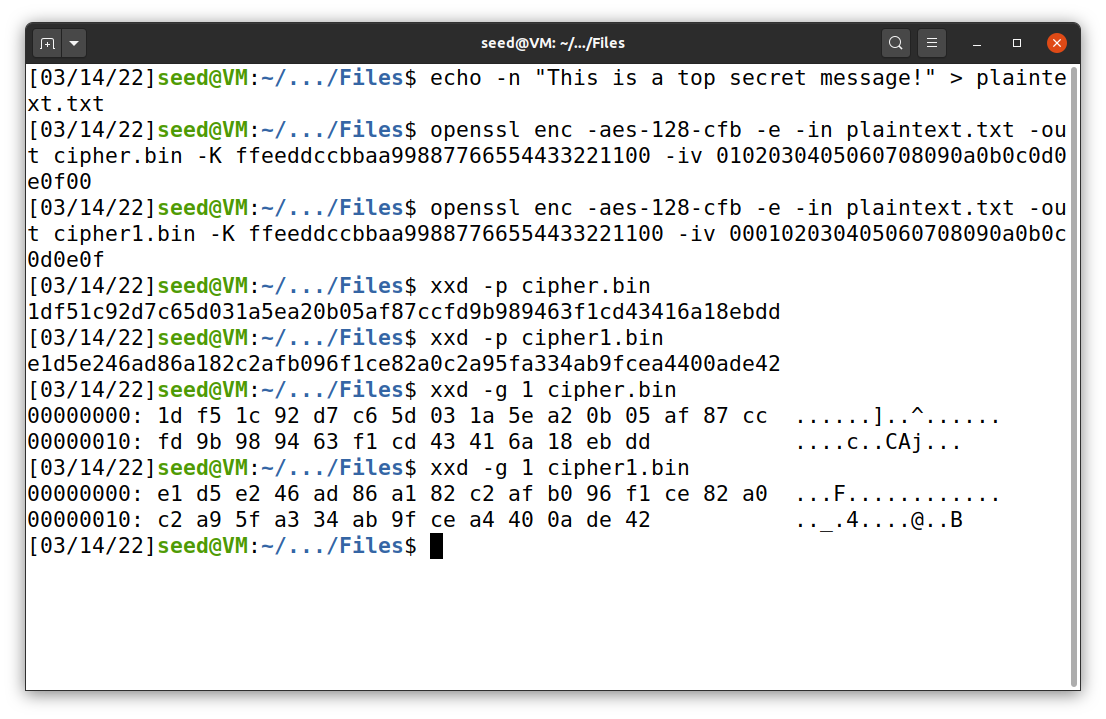
***Task 6.1: IV Experiment***

In this task, I used the example command below to encrypt the plain text file (.txt). After the encryption, I used hexdump (xxd) to show the output results. First, I select the aes-128-cfb encryption mode and the key of ffeeddccbbaa99887766554433221100, which will be used throuout task 6.1. Then, I encrypted the file by using different IV. It produced distinct output as shown in Figure 7.

openssl enc -aes-128-cfb -e -in plain.txt -out cipher.bin -K <Key> -iv <IV>

**Figure 7**

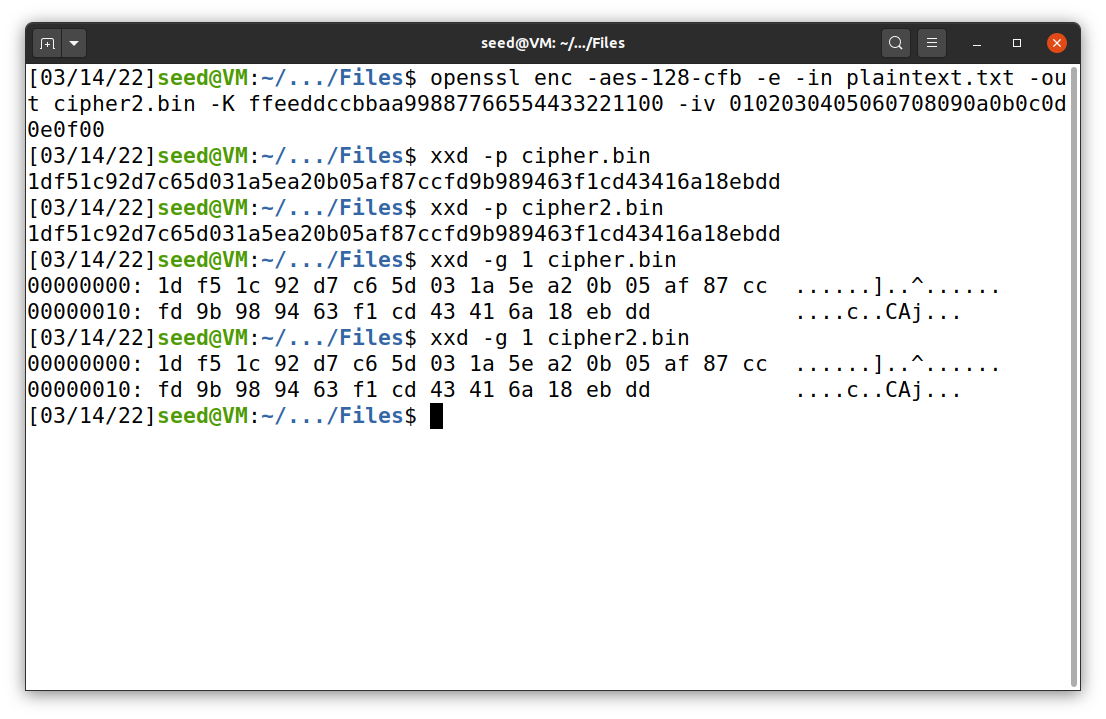
*The result when different IV was used*



On the other hand, I encrypted the file by using the same IV. It produced similar output as shown in Figure 8.

**Figure 8**

*The result when similar IV was used*



What happen here is that cipher feedback mode uses single initialization vector to compare with the key. If we reuse that IV and key, it will produce the same block cipher encryption. So, if the plaintexts are the same, the ciphertext will also be the same. To avoid this problem, one should use a new and unique IV every time. This may also happen with other encryption methods.

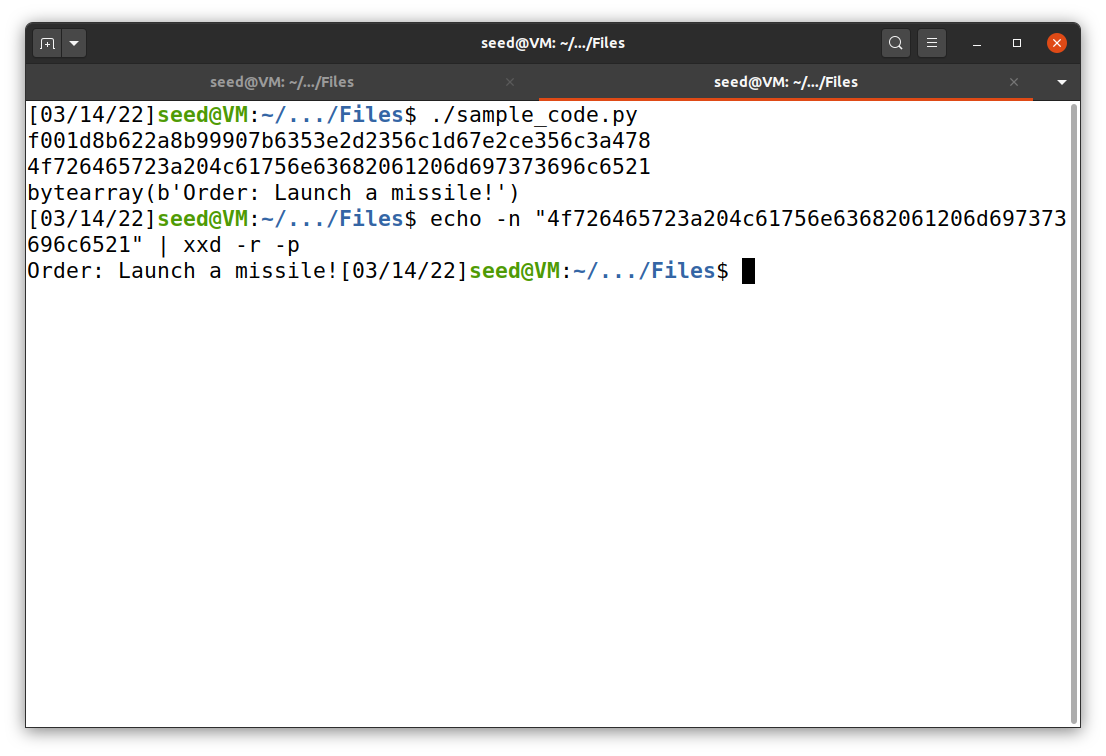
***Task 6.2: Common Mistake: Use the Same IV***

I reconfigured the program sample\_code.py, provided in the Files directory. The file is available in my GitHub repository. First, I input P1, C1, and C2. The program will make XOR operation between text P1 and ciphertext C1. Then, it will do XOR again between the previous XOR result and C2. The secret plain text can be printed out by the python program or using the command below (Figure 9).

echo -n "4f726465723a204c61756e63682061206d697373696c6521" | xxd -r -p

**Figure 9**

*The result when similar IV was used*



Unlike OFB, CFB uses the XOR result between plaintext and key to encrypt the consequence section of plaintext. If we use this known-plaintext attack, only the first part of the plaintext (probably the first 16 bytes) can be revealed.

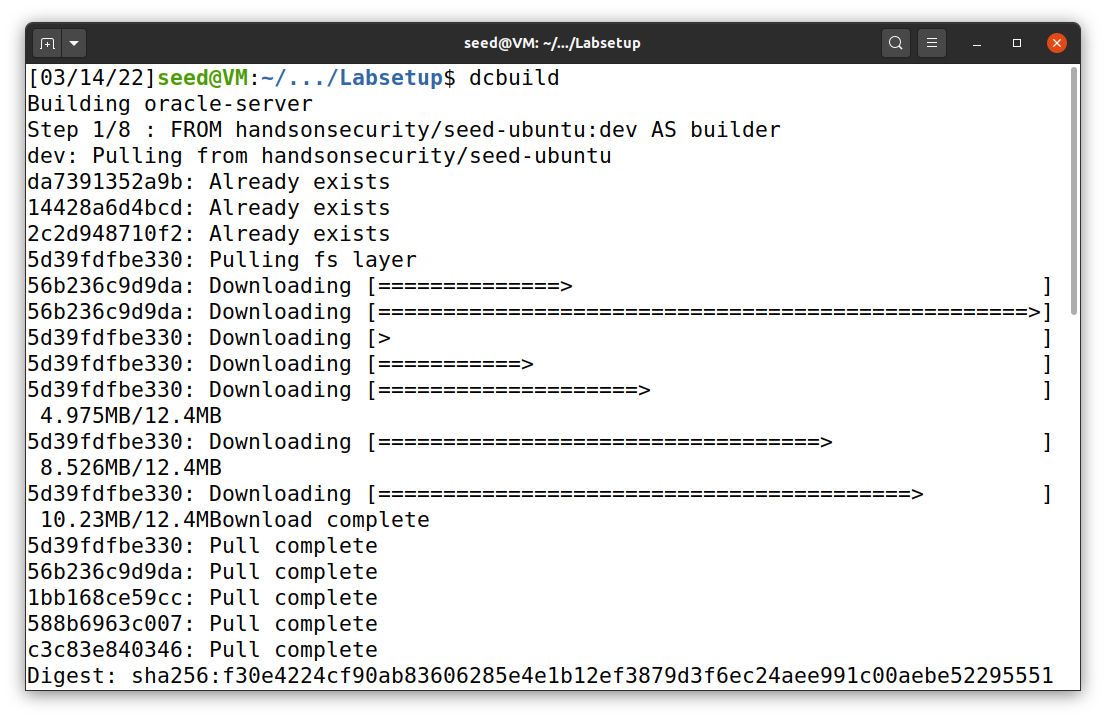
***Task 6.3: Common Mistake: Use a Predictable IV***

In this task, I used the commands below to initiate oracle server, as shown in Figure 10.

dcbuild, dcup

**Figure 10**

*Oracle server initiation*



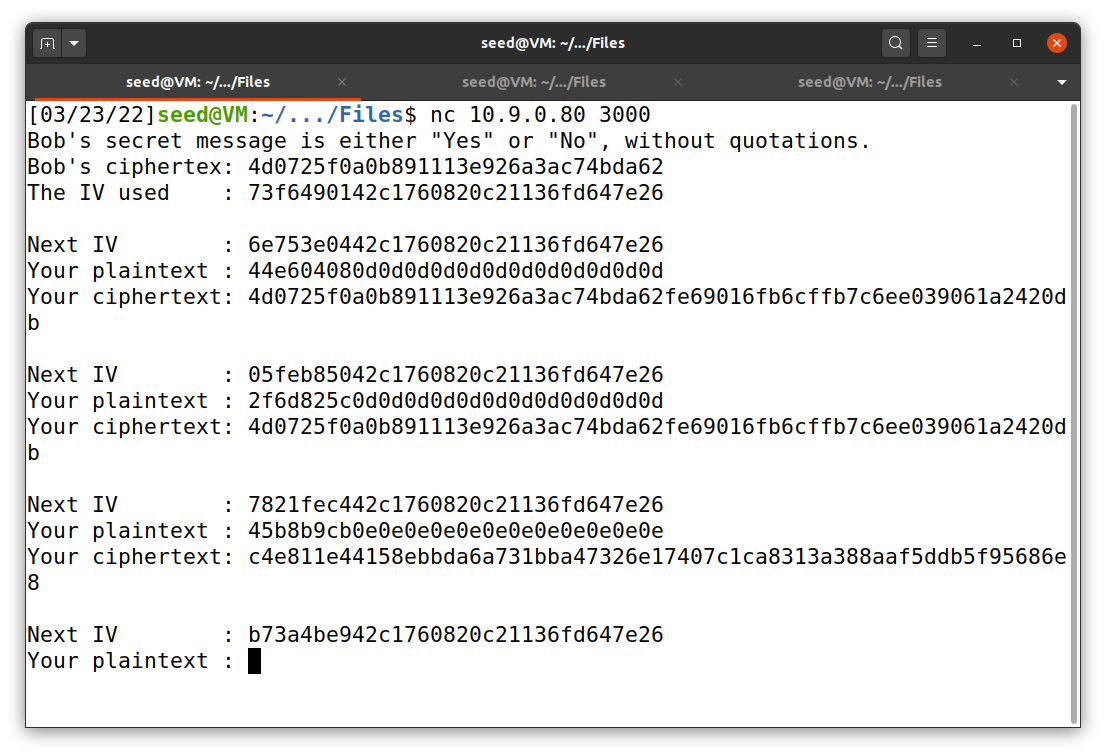
To connect to Oracle server, I used the netcat command below. When the connection is initiated, the server will show Bob’s cipher text, his IV, and the next IV (Figure 11).

nc 10.9.0.80 3000

To figure out if Bob’s message is Yes or No, I reconfigured the sample\_code.py program. The program can be found in my GitHub repository (not shown in this report). The program takes two values of input: Bob’s original IV and next IV. First, the program converts the message (either Yes or No) into a hexadecimal and concatenate with padding data. Then, it executes XOR operation between the padded message and Bob’s original IV. The result goes through XOR operation again with the next IV. Finally, it prints out the result. The result can be copied to Oracle sever to compare with Bob’s cipher text.

**Figure 11**

*Oracle server*



No

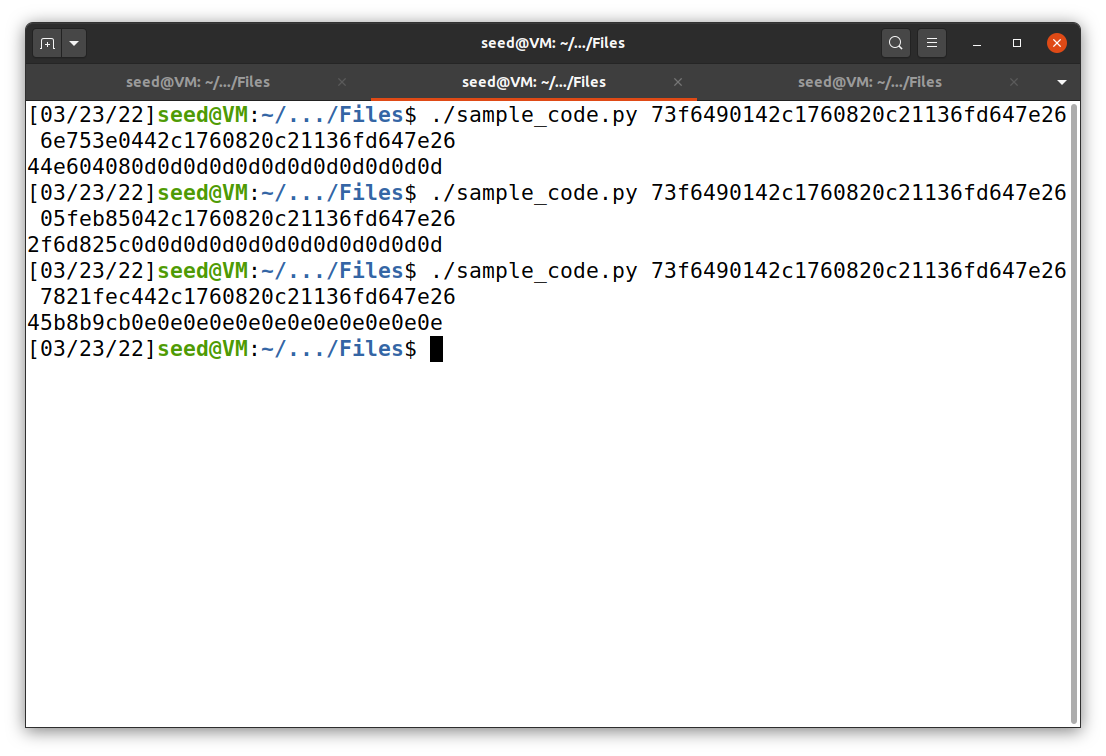
Yes

Yes

If our message match with the original message, the first 32 hexadecimal digits of “Your ciphertext” will be similar to Bob’s cipher text. I tried to run using both Yes and No as a message. The result shown in Figure 11 clearly indicates that Bob’s original message is Yes.

**Figure 12**

*Sample\_code.py for task 6.3*



No

Yes

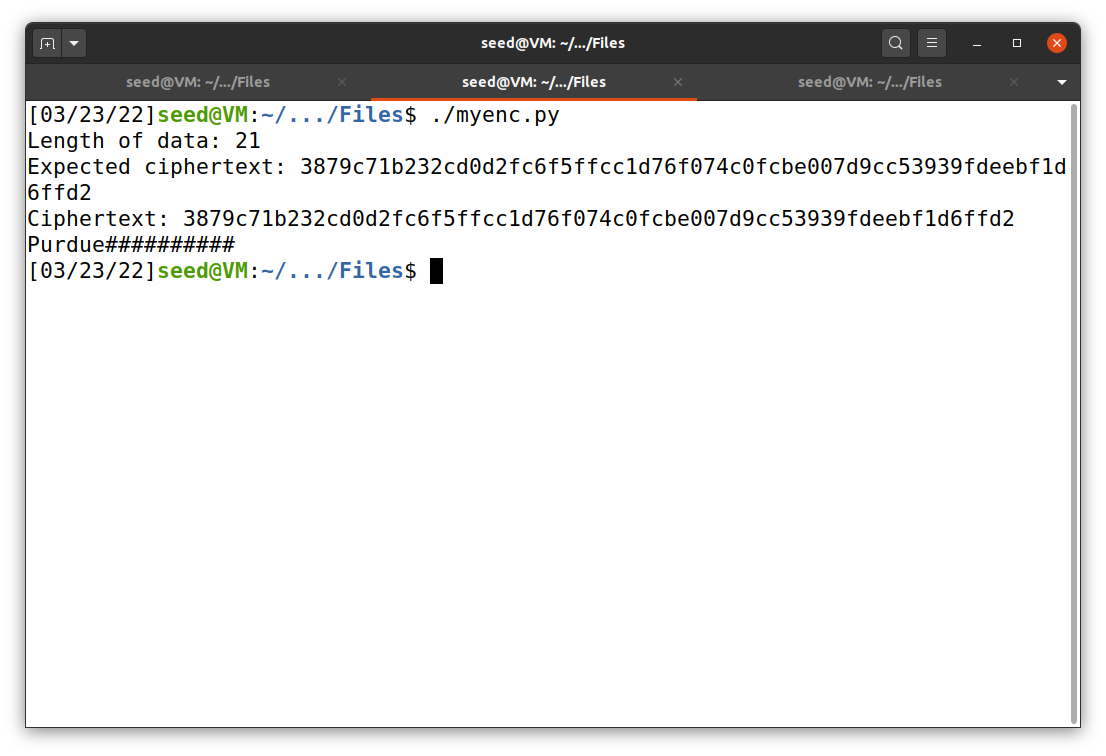
Yes

**Task 7: Programming using the Crypto Library**

This task’s file has been uploaded to my GitHub repository as myenc.py. The IV, expected cipher text, and a list of possible keys (English words) are given. The code is not shown in this report but the output. The program grabs a line of word from words.txt. If the word is longer than 16 characters, we can ignore it. If it is shorter, the program will add pound signs (#) to make it 16 characters. Then, it turns the pounded word into hexadecimal and encrypt it using aes-128-cbc method. After that, it compares the output with expected ciphertext. If both shares the same hexadecimal values, the word used to encrypt ciphertext is the key, which in this case is Purdue (Figure 13).

**Figure 13**

*myenc.py*

**

**References**

Du, W. (2019). Computer & internet security: A hands-on approach (2nd ed.). Independently published.

Du, W. (n.d.). Crypto Lab -- Secret-Key Encryption. SeedLabs 2.0. https://seedsecuritylabs.org/‌Labs\_20.04/Crypto/Crypto\_Encryption/