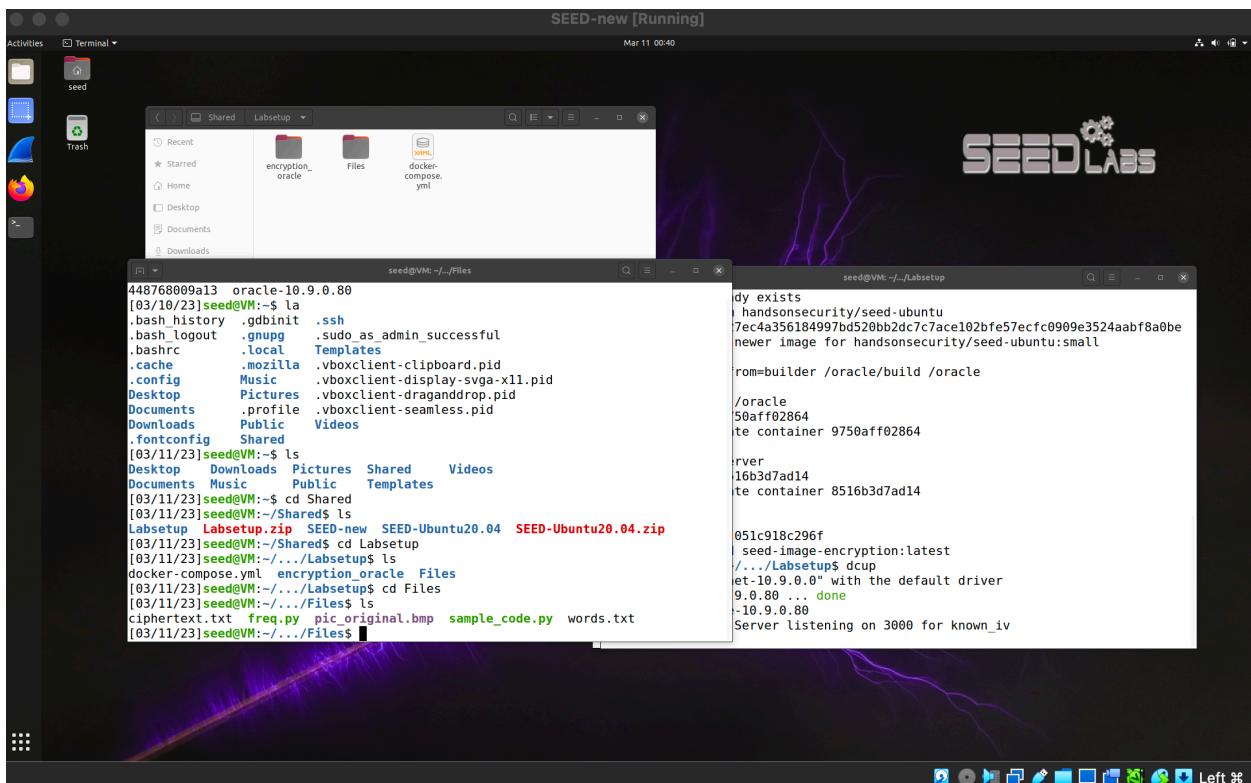


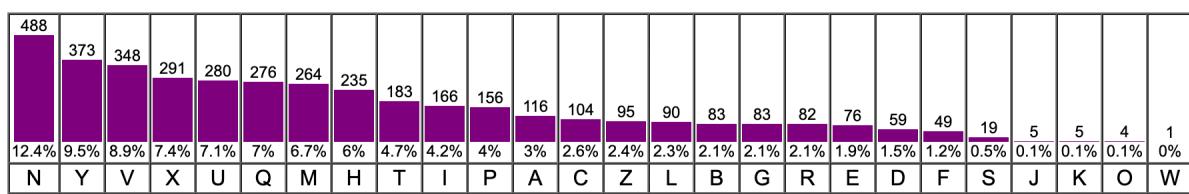
Christian Kuhn
Lab 1
March 12th, 2023

Lab Environment Setup: Here is my SEEDLabs VM setup and running



Task 1 Frequency Analysis:

Single Letter Frequencies:



```
[03/11/23]seed@VM:~/.../Files$ freq.py
-----
1-gram (top 20):
n: 488
y: 373
v: 348
x: 291
u: 280
q: 276
m: 264
h: 235
t: 183
i: 166
p: 156
a: 116
c: 104
z: 95
l: 90
g: 83
b: 83
r: 82
e: 76
d: 59
-----
2-gram (top 20):
yt: 115
tn: 89
mu: 74
nh: 58
vh: 57
hn: 57
vu: 56
nq: 53
xu: 52
up: 46
xh: 45
yn: 44
np: 44
vy: 44
nu: 42
qy: 39
~~
```

Starting
to decrypt the message:

I and MnR at leaQt the aAtorQ noCMnatMon QMnAe braFeheart Mn thMQ Dear the beQt enQeCbIe QaR ended ZE RoMnR to three bMIIboardQ whMAh MQ QMRnMfMAant beAaZQe aAtorQ CaSe ZE the aAadeCDQ IarReQt branAh that fMIC whMIE dMFMQMFe aIQo won the beQt draCa RoIden RIobe and the bafta bZt MtQ fMICCaSer CartMn CADonaRh waQ not noCMnated for beQt dMreAtor and aEart froC arRo CoFMeQ that I and beQt EMATZre wMthoZt aIQo earnMnR beQt dMreAtor noCMnatMonQ are few and far between	N - e Y - t T - h V - a S - k H - r B - f
---	---

The last line gave me a very good starting point.
The following is the output text once I finished the frequency analysis:

"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”

Task 2 Encryption using Different Ciphers and Modes:

Created three ciphers using the following:

-aes-128-cbc, -camellia-128-cfb, -cast5-cfb

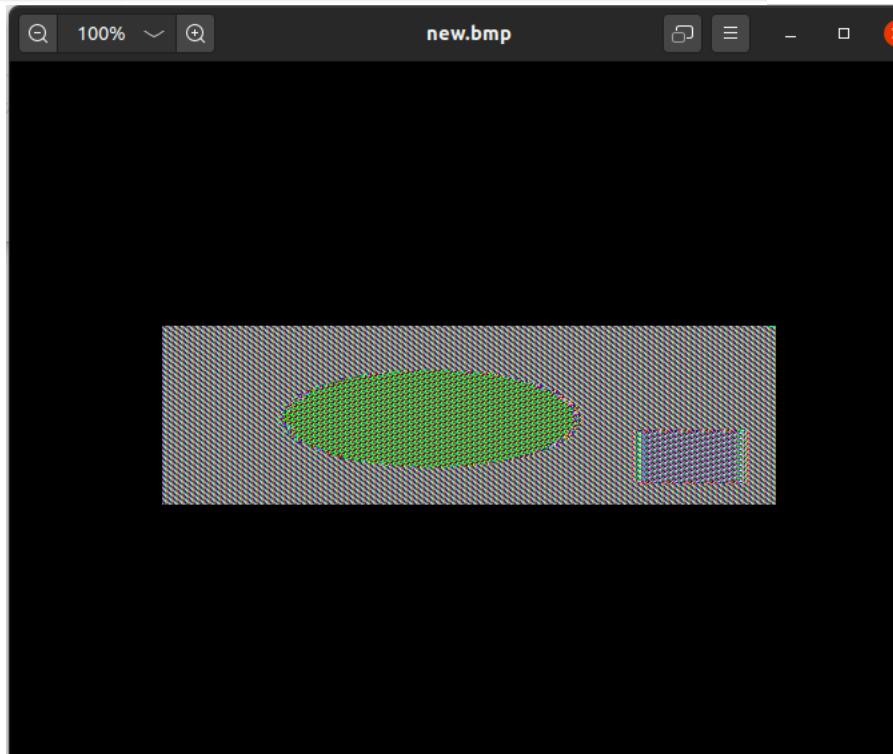
```
[03/11/23]seed@VM:~/.../Labsetup$ sudo openssl enc -aes-128-cbc -e -in plaintext.txt -out cipher1.bin -K 00112233445566778889aabbccddeeff -iv 00112233445566778889aabbccddeeff
[03/11/23]seed@VM:~/.../Labsetup$ ls
cipher1.bin  docker-compose.yml  encryption_oracle  Files  plaintext.txt
[03/11/23]seed@VM:~/.../Labsetup$ sudo openssl enc -camelia-128-cfb -e -in plaintext.txt -out cipher2.bin -K 00112233445566778889aabbccddeeff -iv 00112233445566778889aabbccddeeff
enc: Unrecognized flag camelia-128-cfb
enc: Use -help for summary.
[03/11/23]seed@VM:~/.../Labsetup$ sudo openssl enc -camellia-128-cfb -e -in plaintext.txt -out cipher2.bin -K 00112233445566778889aabbccddeeff -iv 00112233445566778889aabbccddeeff
[03/11/23]seed@VM:~/.../Labsetup$ ls
cipher1.bin  cipher2.bin  docker-compose.yml  encryption_oracle  Files  plaintext.txt
[03/11/23]seed@VM:~/.../Labsetup$ sudo openssl enc -cast5-fb -e -in plaintext.txt -out cipher3.bin -K 00112233445566778889aabbccddeeff -iv 00112233445566778889aabbccddeeff
enc: Unrecognized flag cast5-fb
enc: Use -help for summary.
[03/11/23]seed@VM:~/.../Labsetup$ sudo openssl enc -cast5-cfb -e -in plaintext.txt -out cipher3.bin -K 00112233445566778889aabbccddeeff -iv 00112233445566778889aabbccddeeff
hex string is too long, ignoring excess
[03/11/23]seed@VM:~/.../Labsetup$ ls
cipher1.bin  cipher2.bin  cipher3.bin  docker-compose.yml  encryption_oracle  Files  plaintext.txt
```

Task 3 Encryption Mode ECB vs. CBC:

ECB & CBC encryption:

```
[03/11/23]seed@VM:~/.../Files$ sudo openssl enc -aes-128-ecb -e -in pic_original.bmp -out ECBpic.bmp -K 1001011 -iv 0010011
warning: iv not used by this cipher
hex string is too short, padding with zero bytes to length
[03/11/23]seed@VM:~/.../Files$ sudo openssl enc -aes-128-cbc -e -in pic_original.bmp -out CBCpic.bmp -K 1001011 -iv 0010011
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[03/11/23]seed@VM:~/.../Files$
```

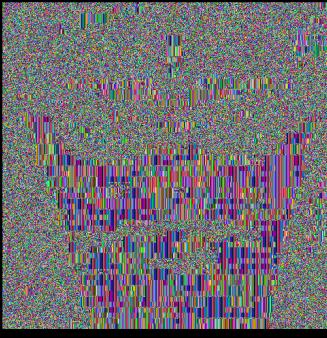
```
$ head -c 54 p1.bmp > header
$ tail -c +55 p2.bmp > body
$ cat header body > new.bmp
```



From this image, we can make out the general size and shape of objects in the image.

My own picture

```
seed@VM: .../files
[03/12/23]seed@VM:.../files$ sudo openssl enc -aes-128-ecb -e -in shake.bmp -out newshake.bmp -iv 0010011
warning: iv not used by this cipher
hex string is too short, padding with zero bytes to length
[03/12/23]seed@VM:.../files$ head -c 54 shake.bmp > header
[03/12/23]seed@VM:.../files$ tail -c +55 shake2.bmp > body
[03/12/23]seed@VM:.../files$ cat header body > newShake.bmp
[03/12/23]seed@VM:.../files$
```



Task 4 Padding:

Create three files which contain 5 bytes, 10 bytes, 16 bytes, respectively

```
seed@VM: .../files
[03/11/23]seed@VM:.../files$ echo -n "12345" > f1.txt
[03/11/23]seed@VM:.../files$ echo -n "1234567891" > f2.txt
[03/11/23]seed@VM:.../files$ echo -n "1234567891011121" > f3.txt
[03/11/23]seed@VM:.../files$
```

OFB:

```
seed@VM: .../files
[03/12/23]seed@VM:.../files$ openssl enc -aes-128-ofb -e -in f1.txt -out ofb.out
enter aes-128-ofb encryption password:
Verifying - enter aes-128-ofb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[03/12/23]seed@VM:.../files$ ls
CBCpic.bmp    ECBpic.bmp   f2.txt   freq.py   pic_original.bmp  words.txt
ciphertext.txt f1.txt     f3.txt   ofb.out   sample_code.py
[03/12/23]seed@VM:.../files$
```

ECB:

```
seed@VM: .../files
[03/12/23] seed@VM: .../files$ openssl enc -aes-128-ecb -e -in f1.txt -out ecb.out
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
*** WARNING : deprecated key derivation used.
```

CFB:

```
seed@VM: .../files
[03/12/23] seed@VM: .../files$ openssl enc -aes-128-cfb -e -in f1.txt -out cfb.out
enter aes-128-cfb encryption password:
Verifying - enter aes-128-cfb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

CBC:

```
[03/12/23] seed@VM: .../files$ openssl enc -aes-128-cbc -e -in f1.txt -out cbc.out
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
```

Result:

```
[03/12/23] seed@VM: .../files$ ls -l
total 800
-rwxrwx--- 1 root vboxsf 32 Mar 12 14:08 cbc.out
-rwxrwx--- 1 root vboxsf 184976 Mar 11 10:45 CBCpic.bmp
-rwxrwx--- 1 root vboxsf 21 Mar 12 13:58 cfb.out
-rwxrwx--- 1 root vboxsf 4759 Dec 5 2020 ciphertext.txt
-rwxrwx--- 1 root vboxsf 32 Mar 12 13:39 ecb.out
-rwxrwx--- 1 root vboxsf 184976 Mar 11 10:45 ECBpic.bmp
-rwxrwx--- 1 root vboxsf 5 Mar 11 20:10 f1.txt
-rwxrwx--- 1 root vboxsf 10 Mar 11 20:10 f2.txt
-rwxrwx--- 1 root vboxsf 16 Mar 11 20:10 f3.txt
-rwxrwx--- 1 root vboxsf 786 Nov 7 2021 freq.py
-rwxrwx--- 1 root vboxsf 21 Mar 12 13:31 ofb.out
-rwxrwx--- 1 root vboxsf 184974 Dec 5 2020 pic_original.bmp
-rwxrwx--- 1 root vboxsf 464 Jan 3 2021 sample_code.py
-rwxrwx--- 1 root vboxsf 206662 Dec 5 2020 words.txt
```

From this screenshot I can see that padding is only needed for ecb and cbc modes. This is because they are block ciphers. Ofb and cfb will not require padding because they are stream ciphers.

File encrypted with cbc

```
seed@VM: .../files
[03/12/23] seed@VM: .../files$ xxd cbc.out
00000000: 5361 6c74 6564 5f5f b4b0 72ea a877 e6d5 Salted_.r..w..
00000010: 74fd 4c85 135a a4a6 ec1f 618f df48 d21e t.L..Z....a..H..
```

Task 5 Error Propagation:

My predictions

ECB: single bit corruption in one block of the ciphertext will only affect the corresponding block of plaintext, but not the rest of the file.

CBC: single bit corruption in one block of the ciphertext will affect the corresponding block of plaintext as well as the next block of plaintext

CFB: single bit corruption in one block of the ciphertext will affect the corresponding block of plaintext as well as the next several blocks of plaintext

OFB: single bit corruption in one block of the ciphertext will not affect any subsequent blocks of the plaintext

Created a text file that is 1000bytes long:

```
seed@VM: .../files
[03/12/23] seed@VM: .../files$ man truncate
[03/12/23] seed@VM: .../files$ truncate -s 1KB task5.txt
[03/12/23] seed@VM: .../files$ ls
cbc.out      ciphertext.txt  f1.txt  freq.py          sample_code.py
CBCpic.bmp   ecb.out       f2.txt  ofb.out         task5.txt
cfb.out      ECBpic.bmp   f3.txt  pic_original.bmp words.txt
[03/12/23] seed@VM: .../files$ -l
-l: command not found
[03/12/23] seed@VM: .../files$ l
cbc.out*     ciphertext.txt* f1.txt*  freq.py*        sample_code.py*
CBCpic.bmp*   ecb.out*      f2.txt*  ofb.out*       task5.txt*
cfb.out*     ECBpic.bmp*   f3.txt*  pic_original.bmp* words.txt*
[03/12/23] seed@VM: .../files$ ls -l
total 800
-rwxrwx--- 1 root vboxsf    32 Mar 12 14:08 cbc.out
-rwxrwx--- 1 root vboxsf 184976 Mar 11 10:45 CBCpic.bmp
-rwxrwx--- 1 root vboxsf    21 Mar 12 13:58 cfb.out
-rwxrwx--- 1 root vboxsf    4759 Dec  5 2020 ciphertext.txt
-rwxrwx--- 1 root vboxsf    32 Mar 12 13:39 ecb.out
-rwxrwx--- 1 root vboxsf 184976 Mar 11 10:45 ECBpic.bmp
-rwxrwx--- 1 root vboxsf     5 Mar 11 20:10 f1.txt
-rwxrwx--- 1 root vboxsf    10 Mar 11 20:10 f2.txt
-rwxrwx--- 1 root vboxsf    16 Mar 11 20:10 f3.txt
-rwxrwx--- 1 root vboxsf    786 Nov  7 2021 freq.py
-rwxrwx--- 1 root vboxsf    21 Mar 12 13:31 ofb.out
-rwxrwx--- 1 root vboxsf 184974 Dec  5 2020 pic_original.bmp
-rwxrwx--- 1 root vboxsf    464 Jan  3 2021 sample_code.py
-rwxrwx--- 1 root vboxsf   1000 Mar 12 14:34 task5.txt
-rwxrwx--- 1 root vboxsf 206662 Dec  5 2020 words.txt
[03/12/23] seed@VM: .../files$
```

Encrypt the file using a AES-128 cipher(cbc):

task5.txt

00000027	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000034	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000041	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000004e	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000005b	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000068	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000075	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000082	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000008f	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000009c	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000a9	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000b6	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000c3	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

After flipping the bit and then decrypting task5.txt I observed no changes in this file. My answer for CBC was wrong.

Task 6 IV and Common Mistakes:

IV Experiment:

```
[03/12/23]seed@VM:.../files$ openssl enc -aes-128-cbc -e -a -nosalt -K 0102030405060708090a0b0c0d0e0f10 -iv 00112233445566778899aabcccddeff -in f2.txt -out f2.txt.enc
[03/12/23]seed@VM:.../files$ openssl enc -aes-128-cbc -e -a -nosalt -K 0102030405060708090a0b0c0d0e0f10 -iv 00112233445566778899aabcccddeaa -in f2.txt -out f2.txt2.enc
[03/12/23]seed@VM:.../files$ cat f2.txt
1234567891[03/12/23]seed@VM:.../files$ cat f2.txt.enc
2u506NParxerBD8iqXjReQ==
[03/12/23]seed@VM:.../files$ cat f2.txt2.enc
q+FxM3jMxpE6enPXUdSCPg==
```

I can observe that the encrypted text is changed by changing iv.

Common Mistake Use the Same IV:
Using CFB mode instead of OFB may potentially allow an attacker to recover more information about P2, as a single bit corruption in the ciphertext in CFB mode is more localized and affects only a few bits of the corresponding plaintext. In contrast, a single bit corruption in OFB mode can propagate and affect the entire block of plaintext.

Common Mistake Use a Predictable IV:

```
[03/12/23]seed@VM:.../files$ nc 10.9.0.80 3000
Bob's secret message is either "Yes" or "No", without quotations.
Bob's ciphertext: 8cf3debff33f09e7faa953dc2ba4e5415
The IV used : ed72bcdca8d5152860030e79e64eb21a
Next IV : f6220f5aa9d5152860030e79e64eb21a
Your plaintext : 54 68 69 73 20 66 73 20 67 79 20 74 65 78 74
Your ciphertext: 755f8945e4108c053b6d3eeae403d73f

Next IV : 1b93fb66a9d5152860030e79e64eb21a
Your plaintext : Your ciphertext: c8e5effee9ec3b13dc77e6bc531f7c699

Next IV : 7cb324d6a9d5152860030e79e64eb21a
Your plaintext : Your ciphertext: ffdbdb9b8e66de0efd4282c6d9fbade22

Next IV : d7549cf0a9d5152860030e79e64eb21a
Your plaintext : Your ciphertext: fa8d8eb3636a45640171c5200d35c419

Next IV : de77594aad5152860030e79e64eb21a
Your plaintext : Your ciphertext: 40e19235c8a5ec475174ff0df48b9c13

Next IV : ea0234acaad5152860030e79e64eb21a
Your plaintext : Your ciphertext: 4c92a9e035e416e1e2c80f71e585d2ce

Next IV : 8fb930fcAAD5152860030e79e64eb21a
Your plaintext : Your ciphertext: 718f1e9484c9ade4530754ada2ccafe3

Next IV : 49753126abd5152860030e79e64eb21a
Your plaintext : Your ciphertext: 012b08b7e5a510a239e9a01efc0054e2
```

Task 7 Programming Using The Crypto Library:

Plaintext is encrypted with a key

```
seed@VM:~/files$ echo -n "This is a top secret." > plaintext1.txt
[03/12/23]seed@VM:~/files$ echo -n "example#####" > key
[03/12/23]seed@VM:~/files$ xxd -p key
6578616d706c6523232323232323232323
[03/12/23]seed@VM:~/files$ openssl enc -aes-128-cbc -e -in plaintext1.txt -out ciphertext.bin -K 6578616d706c652323232323232323 -iv 010203040506070809000a0b0c0d0e0f
[03/12/23]seed@VM:~/files$ xxs -p ciphertext.bin

Command 'xxs' not found, did you mean:

  command 'xjs' from snap xjs (0+git.6419369)
  command 'xbs' from deb xbs (0-10build1)
  command 'xxd' from deb xxd (2:8.1.2269-1ubuntu5.11)
  command 'xx' from deb fex-utils (20160919-1)

See 'snap info <snapname>' for additional versions.

[03/12/23]seed@VM:~/files$ xxd -p ciphertext.bin
e5accdb667e8e569b1b34f423508c15422631198454e104ceb658f591880
)c22
[03/12/23]seed@VM:~/files$
```

Running the program:

```
[03/12/23]seed@VM:~/files$ ls
body          ciphertext.txt  f2.txt2.enc  header      new.bmp        task5.txt
cbc.out       ecb.out        f2.txt.enc   key         ofb.out       words.txt
CBCpic.bmp    ECBpic.bmp    f3.txt      main       pic_original.bmp
cfb.out       f1.txt        file        main.c     plaintext1.txt
ciphertext.bin f2.txt       freq.py    message.enc sample_code.py
[03/12/23]seed@VM:~/files$ gcc -o findkey main.c -lcrypto
[03/12/23]seed@VM:~/files$ ls
body          ciphertext.txt  f2.txt2.enc  freq.py   message.enc    sample_code.py
cbc.out       ecb.out        f2.txt.enc   header   new.bmp      task5.txt
CBCpic.bmp    ECBpic.bmp    f3.txt      key      ofb.out      words.txt
cfb.out       f1.txt        file        main     pic_original.bmp
ciphertext.bin f2.txt       findkey   main.c   plaintext1.txt
[03/12/23]seed@VM:~/files$ ./findkey "This is a top secret." 764aa26b55a4da654df6b19e4
ce00f4ed05e09346fb0e762583cb7da2ac93a2
Key not found
[03/12/23]seed@VM:~/files$
```

```
seed@VM: .../files
// load English word list from file
FILE *wordlist_file = fopen("words.txt", "r");
if (!wordlist_file) {
    printf("Failed to open wordlist file\n");
    return 1;
}

// try each word in the wordlist as the key
char word[16];
while (fgets(word, sizeof(word), wordlist_file)) {
    // remove trailing newline
    int len = strlen(word);
    if (word[len - 1] == '\n') {
        word[len - 1] = '\0';
        len--;
    }

    // append pound signs to form 128-bit key
    memset(key, 0x23, AES_BLOCK_SIZE);
    memcpy(key, word, len);

    // decrypt ciphertext using current key and IV
    AES_KEY aes_key;
    AES_set_decrypt_key(key, 128, &aes_key);
    unsigned char plaintext_decrypted[ciphertext_len + BLOCK_SIZE];
    int plaintext_len_decrypted = 0;
    AES_cbc_encrypt(ciphertext, plaintext_decrypted, ciphertext_len, &aes_key, iv, AES_DECRYPT);

    // check if decrypted plaintext matches original plaintext
    if (strncmp(plaintext, (char*) plaintext_decrypted, strlen(plaintext)) == 0) {
        // found key
        printf("Key found: %s\n", word);
        return 0;
    }
}

// key not found
printf("Key not found\n");
return 1;
}
```