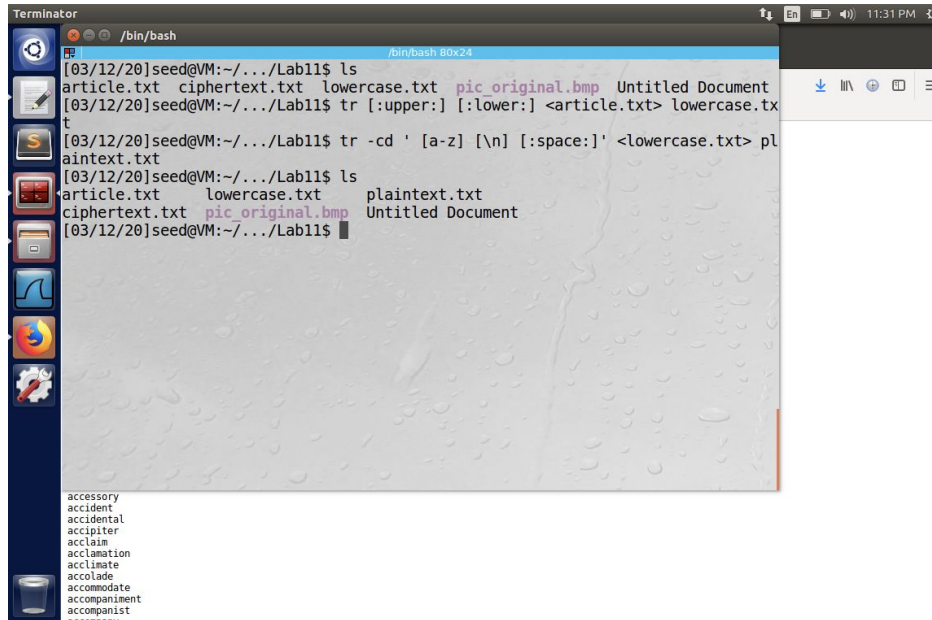


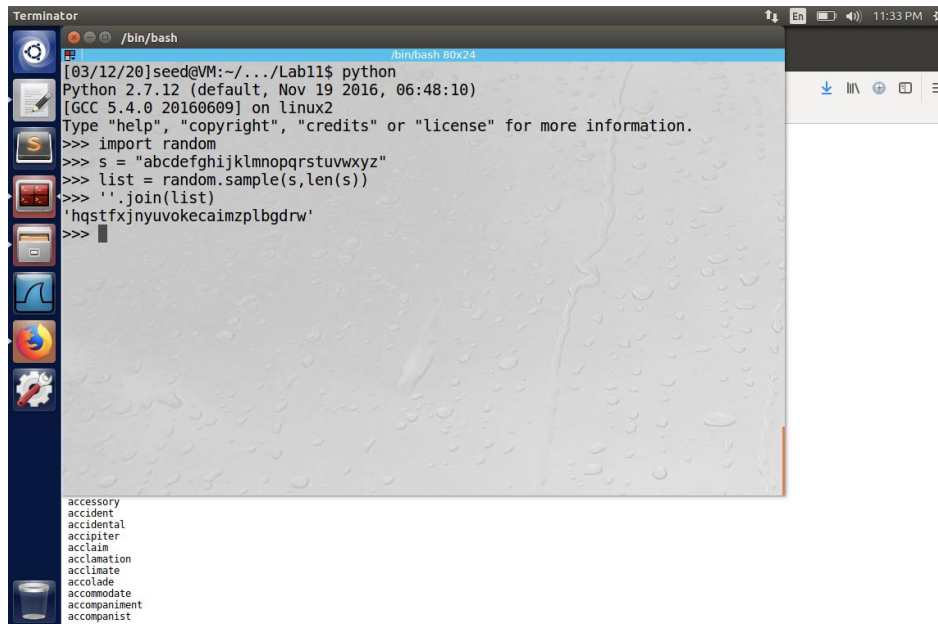
Lab 11  
CSP 544  
Karsen Diepholz

## Task 1



The screenshot shows a Linux terminal window titled "Terminator" with a blue title bar. The terminal prompt is "seed@VM:~/.../Lab11\$". The user has executed several commands: "ls" to list files (article.txt, ciphertext.txt, lowercase.txt, pic\_original.bmp, Untitled Document), "tr [:upper:] [:lower:] <article.txt> lowercase.txt" to convert article.txt to lowercase, and "tr -cd ' [a-z] [\n] [:space:]' <lowercase.txt> plaintext.txt" to remove non-alphabetic characters from lowercase.txt. The terminal output shows the results of these commands. The desktop background is a light gray with a water droplet pattern. The left sidebar contains icons for various applications, and the bottom status bar shows the time as 11:31 PM.

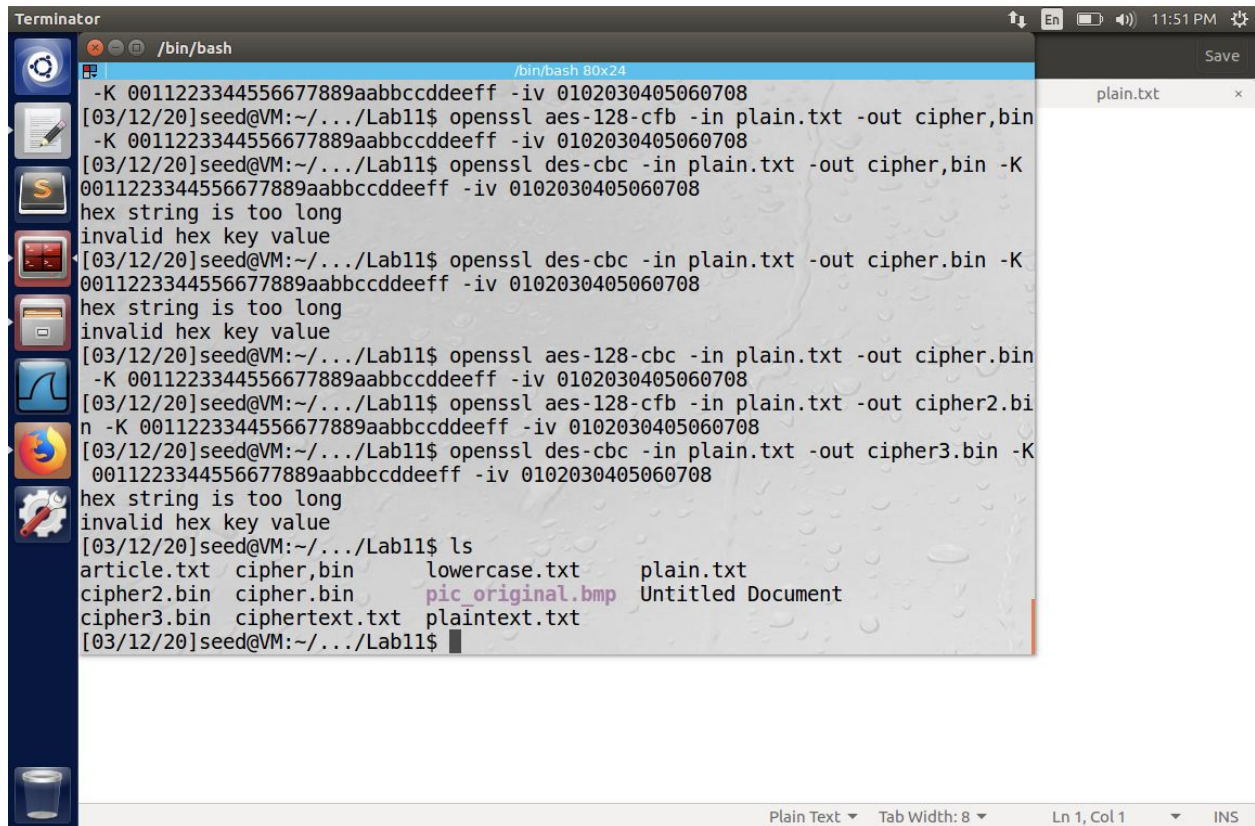
```
Terminator
[03/12/20]seed@VM:~/.../Lab11$ ls
article.txt ciphertext.txt lowercase.txt pic_original.bmp Untitled Document
[03/12/20]seed@VM:~/.../Lab11$ tr [:upper:] [:lower:] <article.txt> lowercase.txt
[03/12/20]seed@VM:~/.../Lab11$ tr -cd ' [a-z] [\n] [:space:]' <lowercase.txt> plaintext.txt
[03/12/20]seed@VM:~/.../Lab11$ ls
article.txt lowercase.txt plaintext.txt
ciphertext.txt pic_original.bmp Untitled Document
[03/12/20]seed@VM:~/.../Lab11$
```



The screenshot shows a Linux terminal window titled "Terminator" with a blue title bar. The terminal prompt is "seed@VM:~/.../Lab11\$". The user has executed the command "python". The terminal output shows the Python version (2.7.12), the default interpreter path, and the GCC version. The user has then entered a series of Python commands to generate a random string: "import random", "s = 'abcdefghijklmnopqrstuvwxyz'", "list = random.sample(s, len(s))", and ".join(list)". The terminal output shows the resulting random string: "hqstfxjnyuvokecaimzplbgdrw". The desktop background is a light gray with a water droplet pattern. The left sidebar contains icons for various applications, and the bottom status bar shows the time as 11:33 PM.

```
Terminator
[03/12/20]seed@VM:~/.../Lab11$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import random
>>> s = "abcdefghijklmnopqrstuvwxyz"
>>> list = random.sample(s, len(s))
>>> ''.join(list)
'hqstfxjnyuvokecaimzplbgdrw'
>>>
```

## Task 2

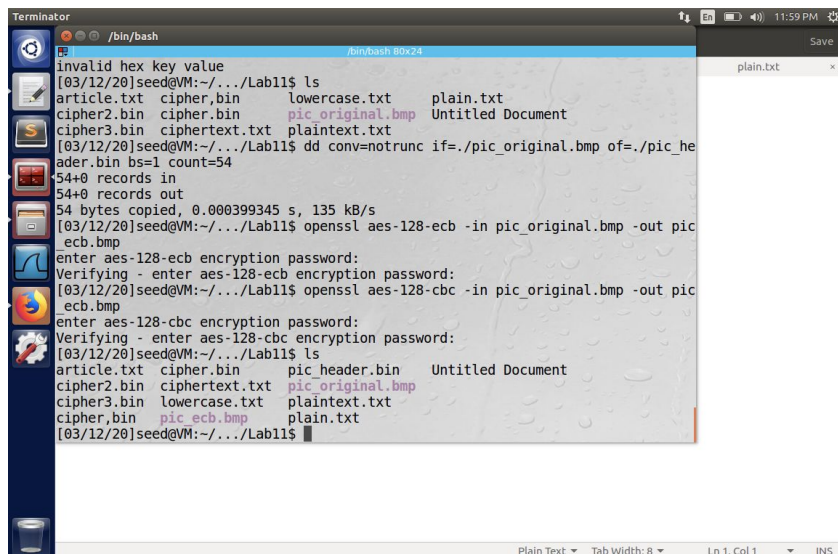


```
Terminator
/bin/bash
/bin/bash 80x24
-K 0011223344556677889aabbccddeeff -iv 0102030405060708
[03/12/20]seed@VM:~/.../Lab11$ openssl aes-128-cfb -in plain.txt -out cipher.bin
-K 0011223344556677889aabbccddeeff -iv 0102030405060708
[03/12/20]seed@VM:~/.../Lab11$ openssl des-cbc -in plain.txt -out cipher.bin -K
0011223344556677889aabbccddeeff -iv 0102030405060708
hex string is too long
invalid hex key value
[03/12/20]seed@VM:~/.../Lab11$ openssl des-cbc -in plain.txt -out cipher.bin -K
0011223344556677889aabbccddeeff -iv 0102030405060708
hex string is too long
invalid hex key value
[03/12/20]seed@VM:~/.../Lab11$ openssl aes-128-cbc -in plain.txt -out cipher.bin
-K 0011223344556677889aabbccddeeff -iv 0102030405060708
[03/12/20]seed@VM:~/.../Lab11$ openssl aes-128-cfb -in plain.txt -out cipher2.bi
n -K 0011223344556677889aabbccddeeff -iv 0102030405060708
[03/12/20]seed@VM:~/.../Lab11$ openssl des-cbc -in plain.txt -out cipher3.bin -K
0011223344556677889aabbccddeeff -iv 0102030405060708
hex string is too long
invalid hex key value
[03/12/20]seed@VM:~/.../Lab11$ ls
article.txt  cipher.bin      lowercase.txt    plain.txt
cipher2.bin  cipher.bin      pic_original.bmp  Untitled Document
cipher3.bin  ciphertext.txt  plaintext.txt
[03/12/20]seed@VM:~/.../Lab11$
```

We encrypt the file using 3 different methods: aes-128-cbc, aes-128-cfb, des-cbc.

### Task 3:

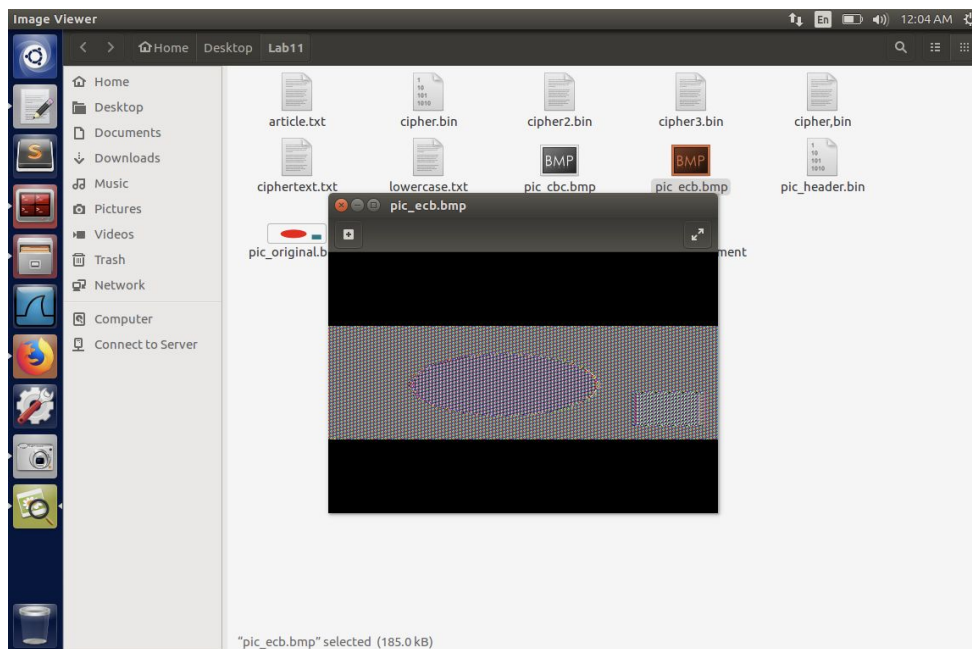
Encrypting the image into two different methods: aes-128-cbc and aes-128-cfb

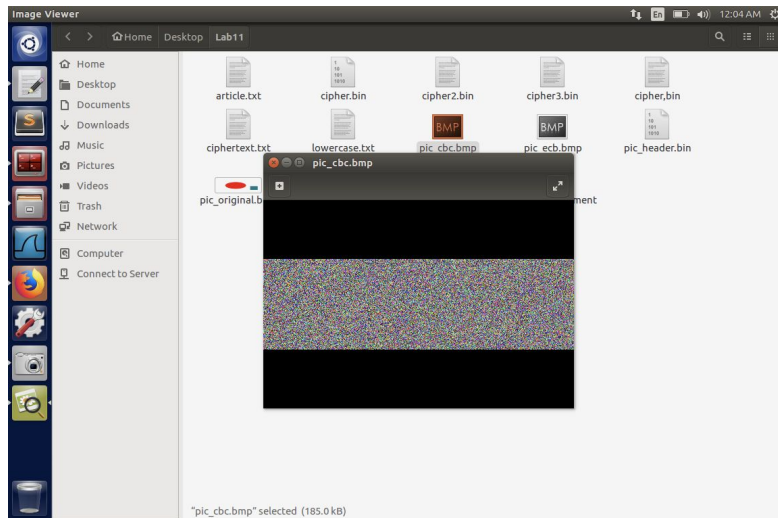


```
Terminator
/bin/bash
/bin/bash 80x24
invalid hex key value
[03/12/20]seed@VM:~/.../Lab11$ ls
article.txt  cipher.bin      lowercase.txt    plain.txt
cipher2.bin  cipher.bin      pic_original.bmp  Untitled Document
cipher3.bin  ciphertext.txt  plaintext.txt
[03/12/20]seed@VM:~/.../Lab11$ dd conv=notrunc if=./pic_original.bmp of=./pic he
ader.bin bs=1 count=54
54+0 records in
54+0 records out
54 bytes copied, 0.000399345 s, 135 kB/s
[03/12/20]seed@VM:~/.../Lab11$ openssl aes-128-ecb -in pic_original.bmp -out pic
ecb.bmp
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
[03/12/20]seed@VM:~/.../Lab11$ openssl aes-128-cbc -in pic_original.bmp -out pic
ecb.bmp
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[03/12/20]seed@VM:~/.../Lab11$ ls
article.txt  cipher.bin      pic_header.bin  Untitled Document
cipher2.bin  ciphertext.txt  pic_original.bmp
cipher3.bin  lowercase.txt  plaintext.txt
cipher.bin   pic_ecb.bmp    plain.txt
[03/12/20]seed@VM:~/.../Lab11$
```

Showing the two different images:

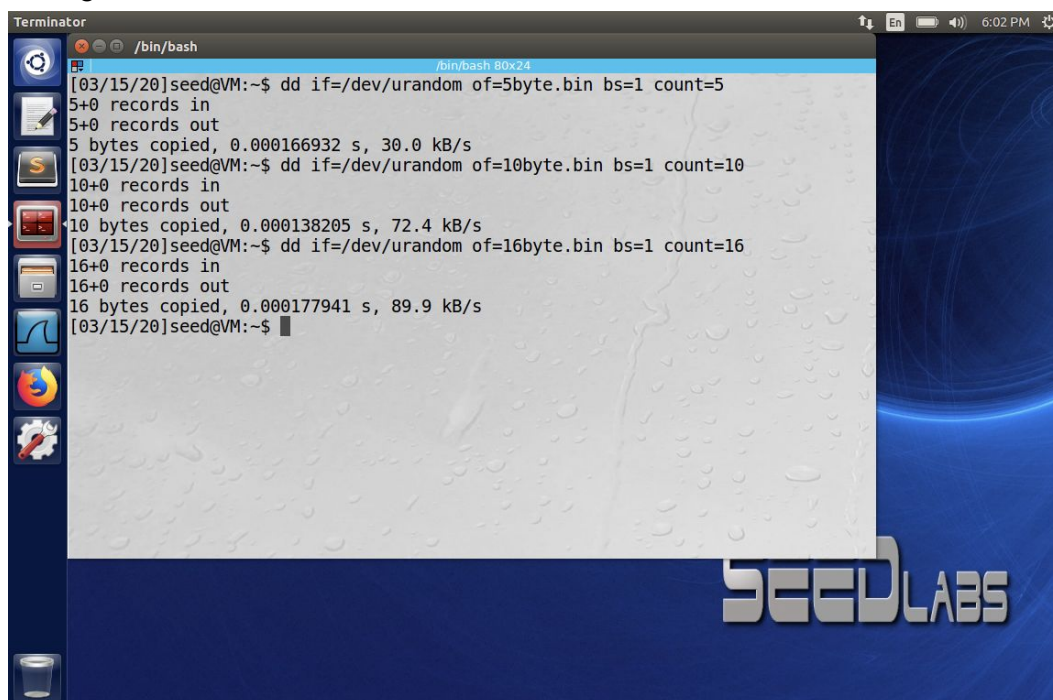
```
Terminator
/bin/bash
. . . . .
dd: failed to open './pic_header.bin': No such file or directory
[03/13/20]seed@VM:~/../Lab11$ openssl aes-128-ecb -in pic_original.bmp -out pic_header.bin
ecb.bmp
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
[03/13/20]seed@VM:~/../Lab11$ openssl aes-128-cbc -in pic_original.bmp -out pic_header.bin
cbc.bmp
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[03/13/20]seed@VM:~/../Lab11$ dd conv=notrunc if=./pic_header.bin of=./pic_ecb
.bmp bs=1 count=54
dd: failed to open './pic_header.bin': No such file or directory
[03/13/20]seed@VM:~/../Lab11$ dd conv=notrunc if=./pic_header.bin of=./pic_ecb
.bmp bs=1 count=54
54+0 records in
54+0 records out
54 bytes copied, 0.000575466 s, 93.8 kB/s
[03/13/20]seed@VM:~/../Lab11$ dd conv=notrunc if=./pic_header.bin of=./pic_cbc
.bmp bs=1 count=54
54+0 records in
54+0 records out
54 bytes copied, 0.000454863 s, 119 kB/s
[03/13/20]seed@VM:~/../Lab11$
```





## Task 4

Making 3 files...



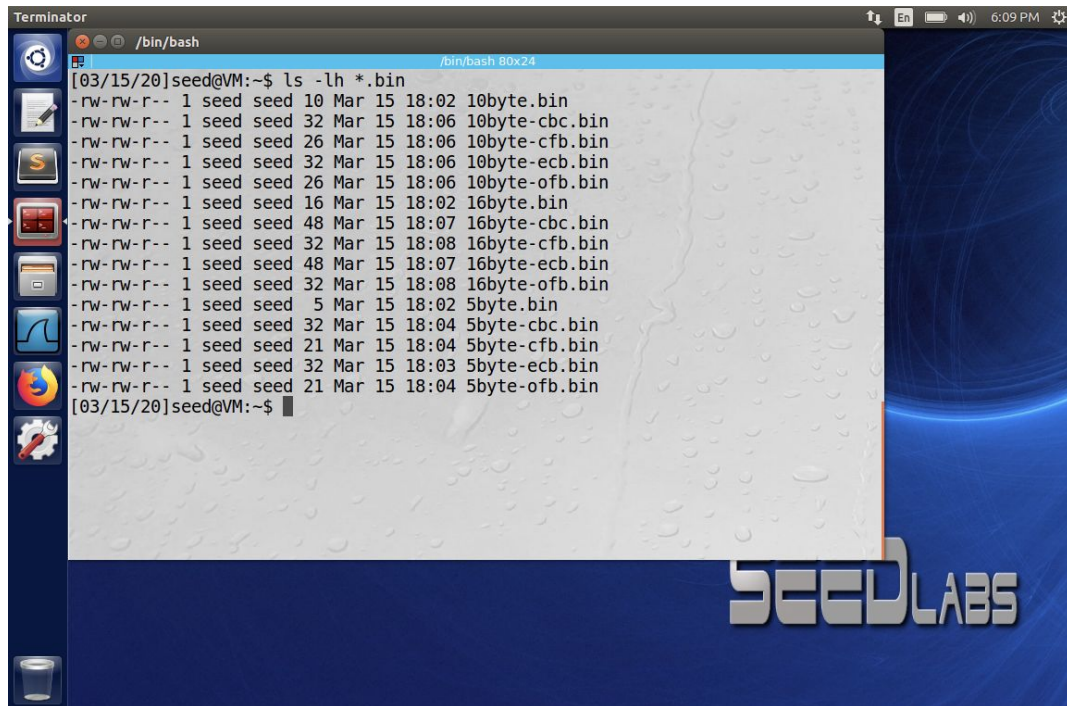
Encrypting...



```
Terminator /bin/bash 6:05 PM
/bin/bash 80x24
5+0 records in
5+0 records out
5 bytes copied, 0.000166932 s, 30.0 kB/s
[03/15/20]seed@VM:~$ dd if=/dev/urandom of=10byte.bin bs=1 count=10
10+0 records in
10+0 records out
10 bytes copied, 0.000138205 s, 72.4 kB/s
[03/15/20]seed@VM:~$ dd if=/dev/urandom of=16byte.bin bs=1 count=16
16+0 records in
16+0 records out
16 bytes copied, 0.000177941 s, 89.9 kB/s
[03/15/20]seed@VM:~$ openssl aes-128-ecb -in 5byte.bin -out 5byte-ecb.bin
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
[03/15/20]seed@VM:~$ openssl aes-128-cbc -in 5byte.bin -out 5byte-cbc.bin
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[03/15/20]seed@VM:~$ openssl aes-128-cfb -in 5byte.bin -out 5byte-cfb.bin
enter aes-128-cfb encryption password:
Verifying - enter aes-128-cfb encryption password:
[03/15/20]seed@VM:~$ openssl aes-128-ofb -in 5byte.bin -out 5byte-ofb.bin
enter aes-128-ofb encryption password:
Verifying - enter aes-128-ofb encryption password:
[03/15/20]seed@VM:~$
```

```
Terminator /bin/bash 6:08 PM
/bin/bash 80x24
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
[03/15/20]seed@VM:~$ openssl aes-128-cbc -in 10byte.bin -out 10byte-cbc.bin
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[03/15/20]seed@VM:~$ openssl aes-128-cfb -in 10byte.bin -out 10byte-cfb.bin
enter aes-128-cfb encryption password:
Verifying - enter aes-128-cfb encryption password:
[03/15/20]seed@VM:~$ openssl aes-128-ofb -in 10byte.bin -out 10byte-ofb.bin
enter aes-128-ofb encryption password:
Verifying - enter aes-128-ofb encryption password:
[03/15/20]seed@VM:~$ openssl aes-128-ecb -in 16byte.bin -out 16byte-ecb.bin
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
[03/15/20]seed@VM:~$ openssl aes-128-cbc -in 16byte.bin -out 16byte-cbc.bin
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[03/15/20]seed@VM:~$ openssl aes-128-cfb -in 16byte.bin -out 16byte-cfb.bin
enter aes-128-cfb encryption password:
Verifying - enter aes-128-cfb encryption password:
[03/15/20]seed@VM:~$ openssl aes-128-ofb -in 16byte.bin -out 16byte-ofb.bin
enter aes-128-ofb encryption password:
Verifying - enter aes-128-ofb encryption password:
[03/15/20]seed@VM:~$
```

Results...



```
Terminator
/bin/bash
[03/15/20]seed@VM:~$ ls -lh *.bin
-rw-rw-r-- 1 seed seed 10 Mar 15 18:02 10byte.bin
-rw-rw-r-- 1 seed seed 32 Mar 15 18:06 10byte-cbc.bin
-rw-rw-r-- 1 seed seed 26 Mar 15 18:06 10byte-cfb.bin
-rw-rw-r-- 1 seed seed 32 Mar 15 18:06 10byte-ecb.bin
-rw-rw-r-- 1 seed seed 26 Mar 15 18:06 10byte-ofb.bin
-rw-rw-r-- 1 seed seed 16 Mar 15 18:02 16byte.bin
-rw-rw-r-- 1 seed seed 48 Mar 15 18:07 16byte-cbc.bin
-rw-rw-r-- 1 seed seed 32 Mar 15 18:08 16byte-cfb.bin
-rw-rw-r-- 1 seed seed 48 Mar 15 18:07 16byte-ecb.bin
-rw-rw-r-- 1 seed seed 32 Mar 15 18:08 16byte-ofb.bin
-rw-rw-r-- 1 seed seed 5 Mar 15 18:02 5byte.bin
-rw-rw-r-- 1 seed seed 32 Mar 15 18:04 5byte-cbc.bin
-rw-rw-r-- 1 seed seed 21 Mar 15 18:04 5byte-cfb.bin
-rw-rw-r-- 1 seed seed 32 Mar 15 18:03 5byte-ecb.bin
-rw-rw-r-- 1 seed seed 21 Mar 15 18:04 5byte-ofb.bin
[03/15/20]seed@VM:~$
```

CFB and OFB pad 16 bytes to the end, CBC and ECB pad until a multiple of 8 is reached.

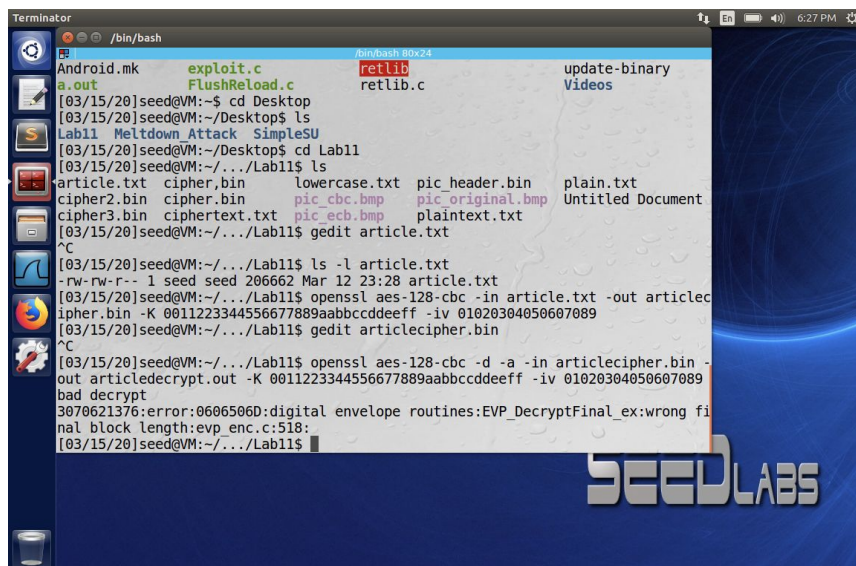
### Task 5:

ECB: All but 1 corrupted block

CBC: All but 2 corrupted block

CFB: All but 2 blocks

OFB: All but 1 block



```
Terminator
/bin/bash
[03/15/20]seed@VM:~$ cd Desktop
[03/15/20]seed@VM:~/Desktop$ ls
Lab11  Meltdown Attack  SimpleSU
[03/15/20]seed@VM:~/Desktop$ cd Lab11
[03/15/20]seed@VM:~/.../Lab11$ ls
article.txt  cipher.bin  lowercase.txt  pic_header.bin  plain.txt
cipher2.bin  cipher3.bin  cipher2.txt    pic_cbc.bmp     pic_original.bmp  plaintext.txt  Untitled Document
[03/15/20]seed@VM:~/.../Lab11$ gedit article.txt
^C
[03/15/20]seed@VM:~/.../Lab11$ ls -l article.txt
-rw-rw-r-- 1 seed seed 206662 Mar 12 23:28 article.txt
[03/15/20]seed@VM:~/.../Lab11$ openssl aes-128-cbc -in article.txt -out articlecipher.bin -K 0011223344556677889aabbccddeeff -iv 01020304050607089
[03/15/20]seed@VM:~/.../Lab11$ gedit articlecipher.bin
^C
[03/15/20]seed@VM:~/.../Lab11$ openssl aes-128-cbc -d -a -in articlecipher.bin -out articledecrypt.out -K 0011223344556677889aabbccddeeff -iv 01020304050607089
bad decrypt
3070621376:error:0606506D:digital envelope routines:EVP_DecryptFinal_ex:wrong final block length:evp enc.c:518:
[03/15/20]seed@VM:~/.../Lab11$
```

Cannot be decrypted because we were the ones who corrupted it!

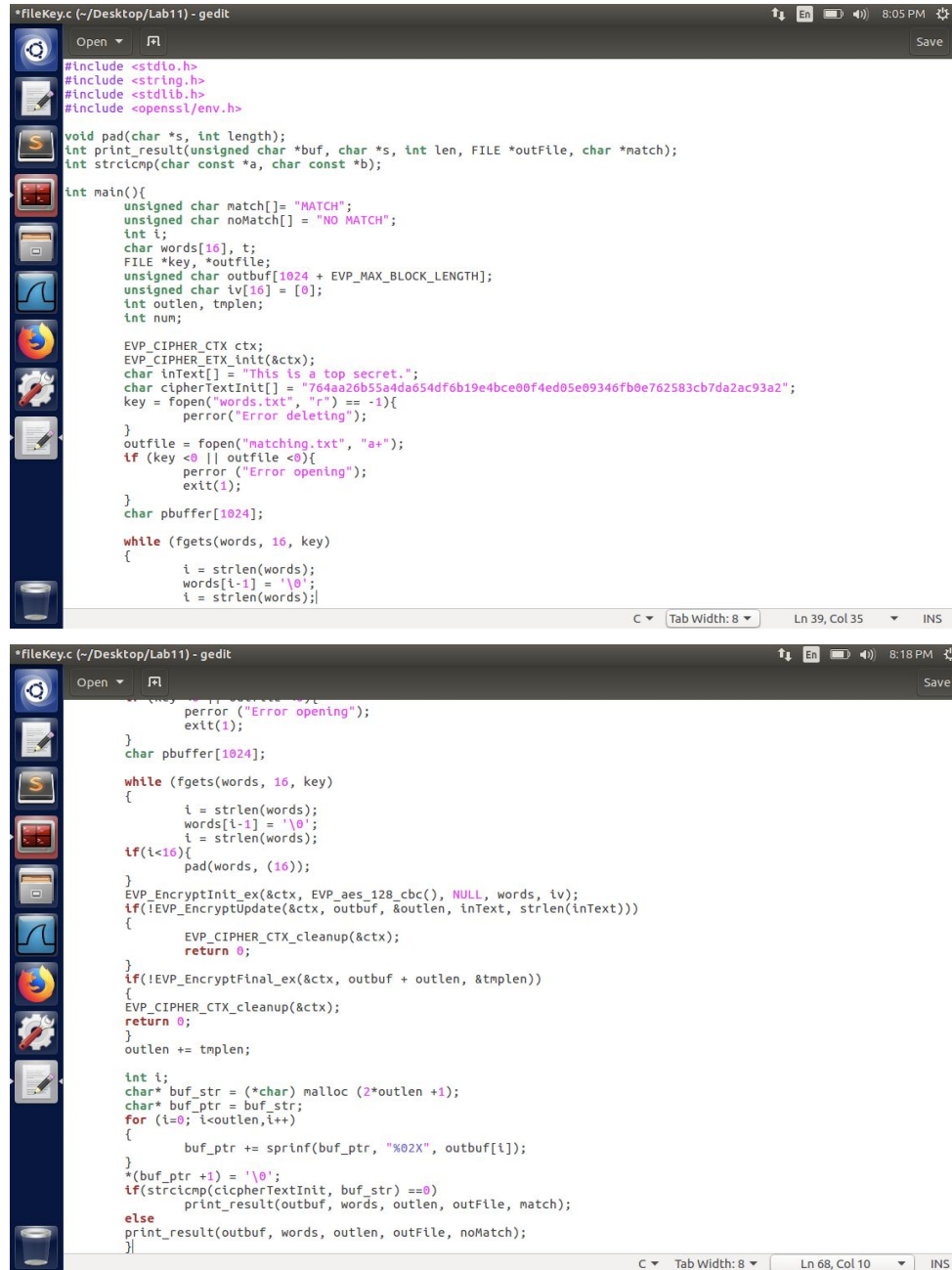
## Task 6

.1

Each IV needs to be different because if they are the same, the attacker can use any message to figure out what the plaintext is for every message that is received, because every word is the same (eg “word” is 6b32 for every instance of it)

.2

## Task 7



```
*fileKey.c (-/Desktop/Lab11) - gedit
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <openssl/env.h>

void pad(char *s, int length);
int print_result(unsigned char *buf, char *s, int len, FILE *outFile, char *match);
int strcmp(char const *a, char const *b);

int main(){
    unsigned char match[] = "MATCH";
    unsigned char noMatch[] = "NO MATCH";
    int i;
    char words[16], t;
    FILE *key, *outFile;
    unsigned char outbuf[1024 + EVP_MAX_BLOCK_LENGTH];
    unsigned char iv[16] = {0};
    int outlen, tmplen;
    int num;

    EVP_CIPHER_CTX ctx;
    EVP_CIPHER_CTX_init(&ctx);
    char inText[] = "This is a top secret.";
    char cipherTextInit[] = "764aa26b55a4da654df6b19e4bce00f4ed05e09346fb0e762583cb7da2ac93a2";
    key = fopen("words.txt", "r");
    if (key == NULL){
        perror("Error deleting");
    }
    outFile = fopen("matching.txt", "a+");
    if (key < 0 || outFile < 0){
        perror("Error opening");
        exit(1);
    }
    char pBuffer[1024];

    while (fgets(words, 16, key))
    {
        i = strlen(words);
        words[i-1] = '\0';
        i = strlen(words);

        if(i<16){
            pad(words, (16));
        }
        EVP_EncryptInit_ex(&ctx, EVP_aes_128_cbc(), NULL, words, iv);
        if(!EVP_EncryptUpdate(&ctx, outbuf, &outlen, inText, strlen(inText)))
        {
            EVP_CIPHER_CTX_cleanup(&ctx);
            return 0;
        }
        if(!EVP_EncryptFinal_ex(&ctx, outbuf + outlen, &tmplen))
        {
            EVP_CIPHER_CTX_cleanup(&ctx);
            return 0;
        }
        outlen += tmplen;

        int i;
        char* buf_str = (*char) malloc (2*outlen +1);
        char* buf_ptr = buf_str;
        for (i=0; i<outlen,i++)
        {
            buf_ptr += sprintf(buf_ptr, "%02X", outbuf[i]);
        }
        *(buf_ptr +1) = '\0';
        if(strcmp(cipherTextInit, buf_str) ==0)
            print_result(outbuf, words, outlen, outFile, match);
        else
            print_result(outbuf, words, outlen, outFile, noMatch);
    }
}
```

```
*fileKey.c (-/Desktop/Lab11) - gedit
Open Save
}
int print_result(unsigned char *buf, char *s, int len, FILE *outFile, char *match)
{
    int i, n, j, k;
    char x = '\n';
    char space = ' ';
    for(j=0; j < strlen(s); j++){
        fprintf(outFile, "%c", s[j]);
    }
    fprintf(outFile, "%c", space);
    for(i=0; i < len; i++){
        fprintf(outFile, "%02x", buf[i]);
    }
    fprintf(outFile, "%c", space);
    for(k=0; k < strlen(match); k++){
        fprintf(outFile, "%c", match[k]);
    }
    fprintf(outFile, "%c", x);
    return(0);
}

void pad(char *s, int length){
    int l;
    l = strlen(s);
    while(l < length){
        s[l] = ' ';
        l++;
    }
    s[l] = '\0';
}

int stricmp(char const *a, char const *b){
    for(;; a++, b++){
        int d = tolower(*a) - tolower(*b);
        if(d != 0 || !*a)
            return d;
    }
}
```

C Tab Width: 8 Ln 110, Col 2 INS