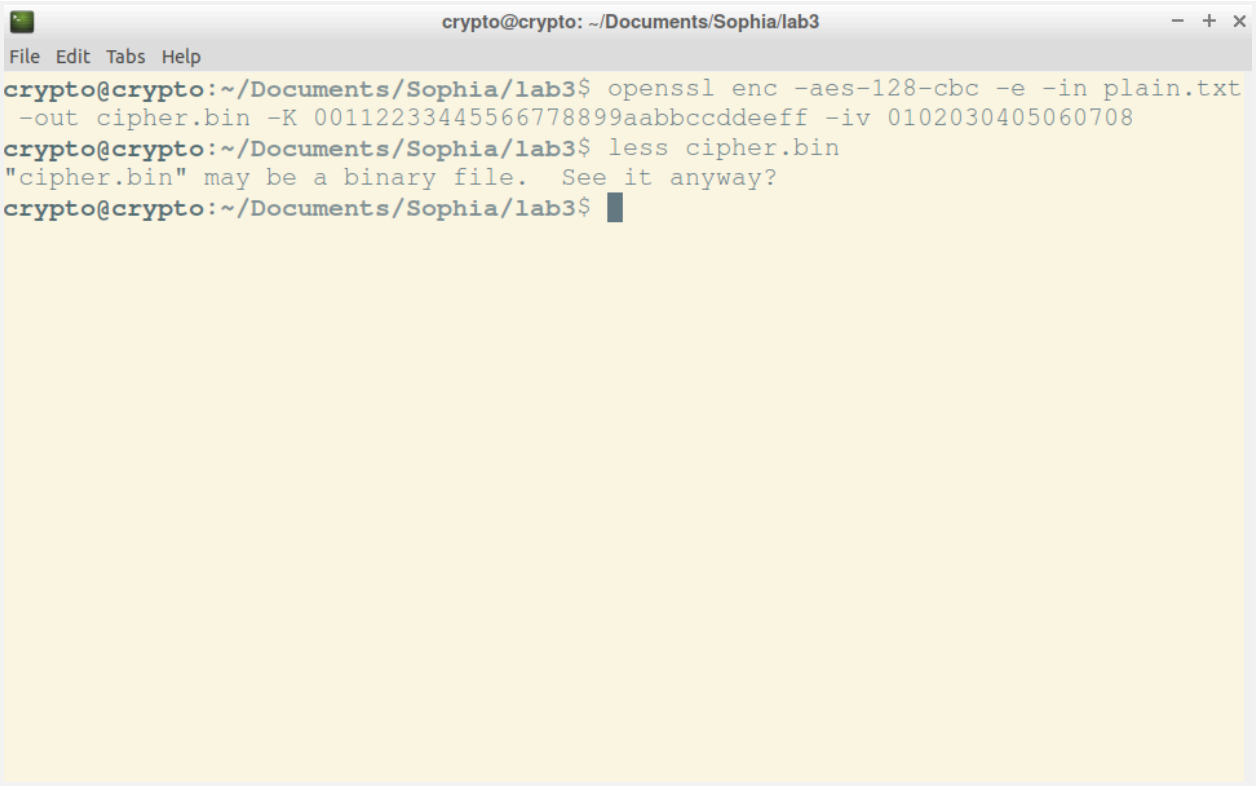
**Block cipher operation**

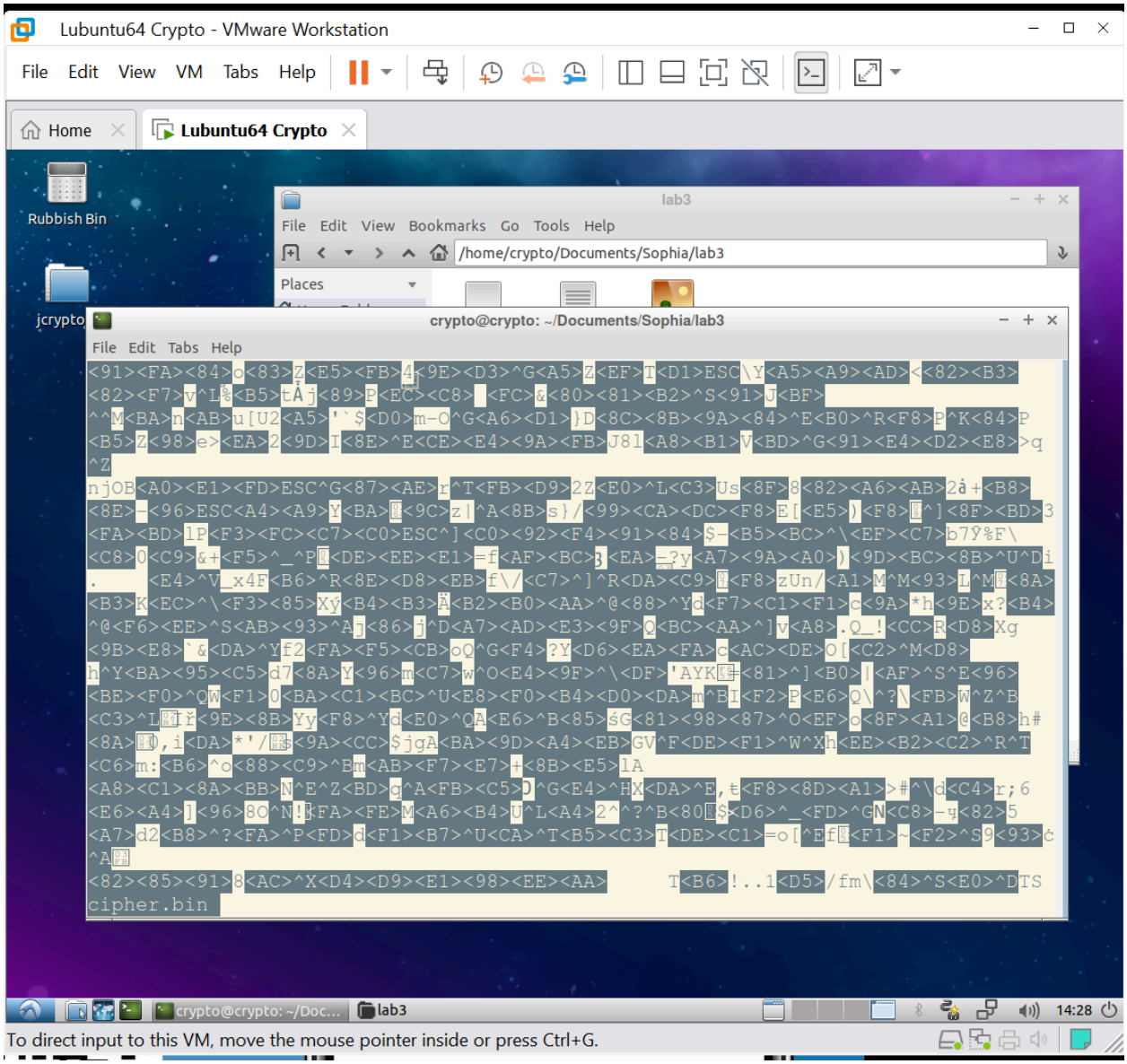
**Task 1- Encryption using different cipher and modes**

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

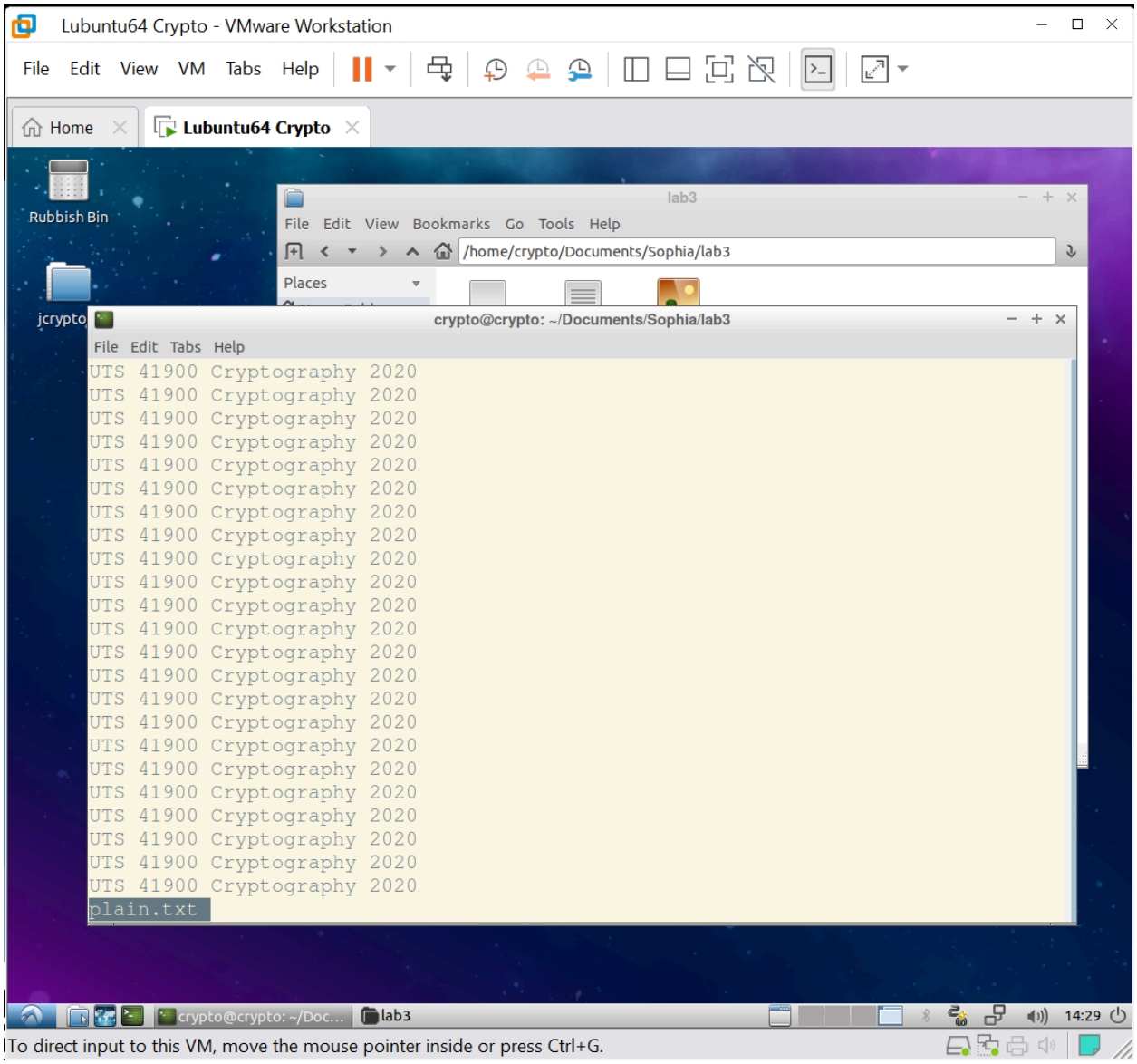
| | |
|---|---|
| -in <file> | input file |
| -out <file> | output file |
| -e | encrypt |
| -d | decrypt |
| -K/-iv | key/iv in hex is the next argument |
| -[pP] | print the iv/key (then exit if -P) |

**aes-128-cbc**

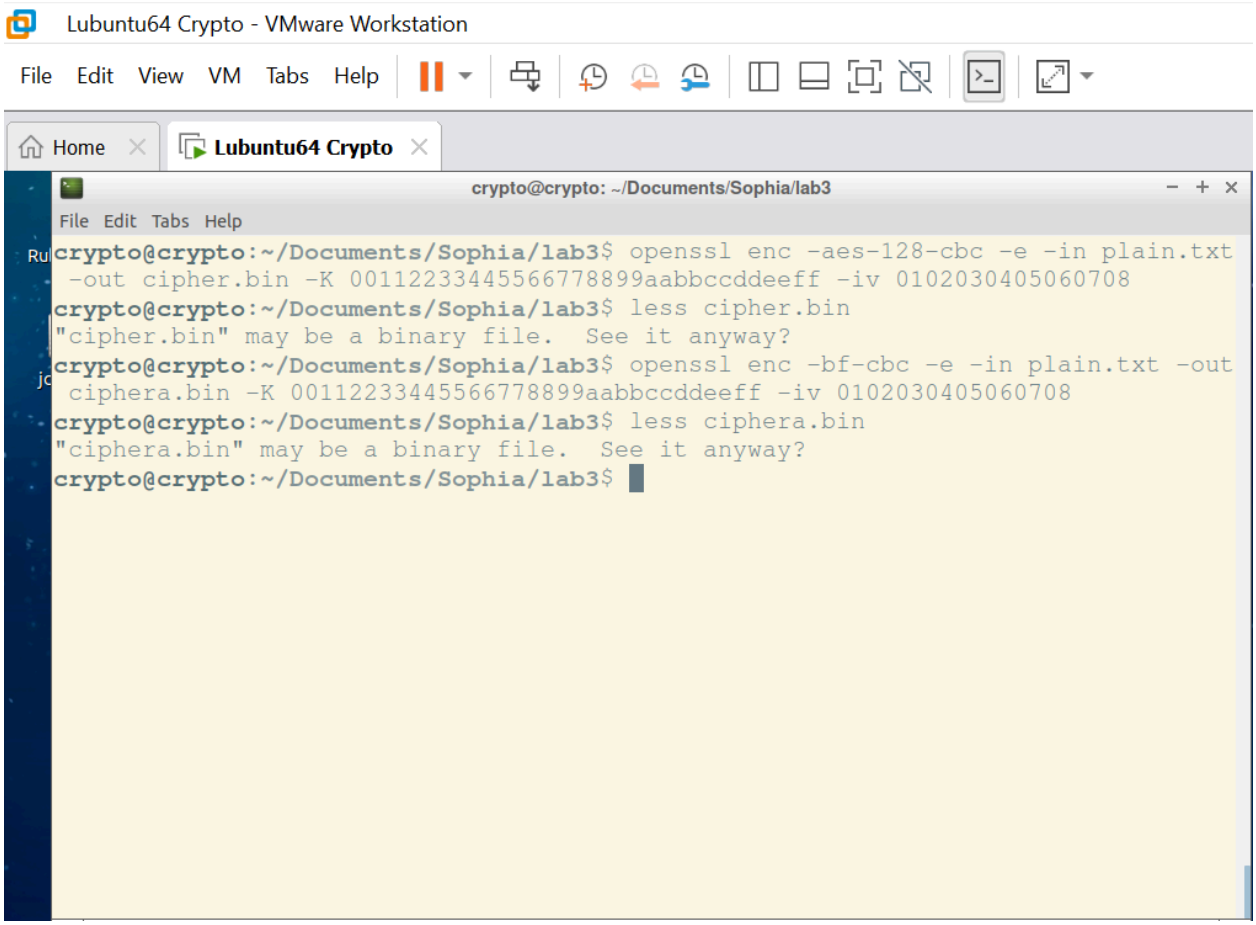


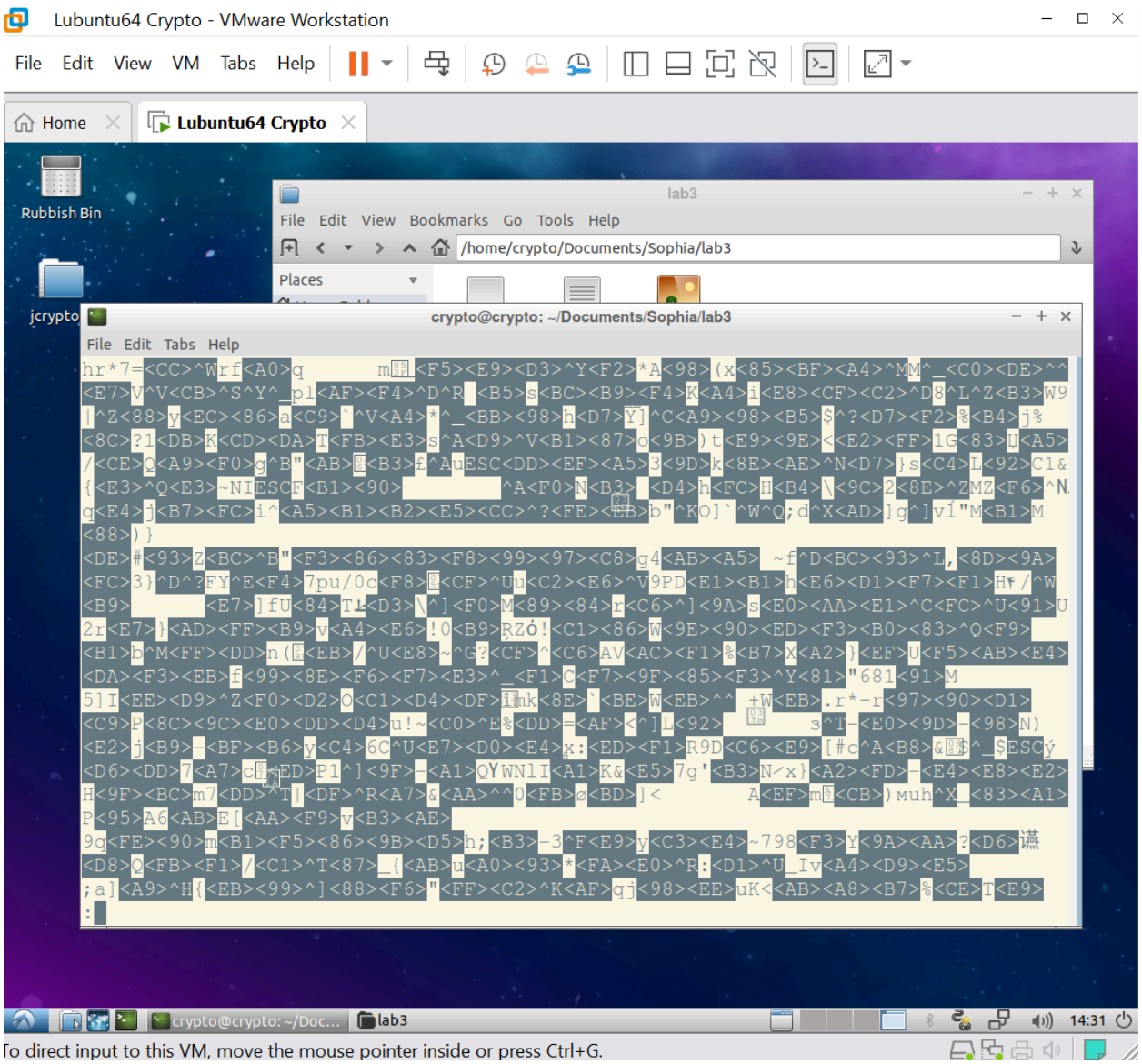



Decrypted file

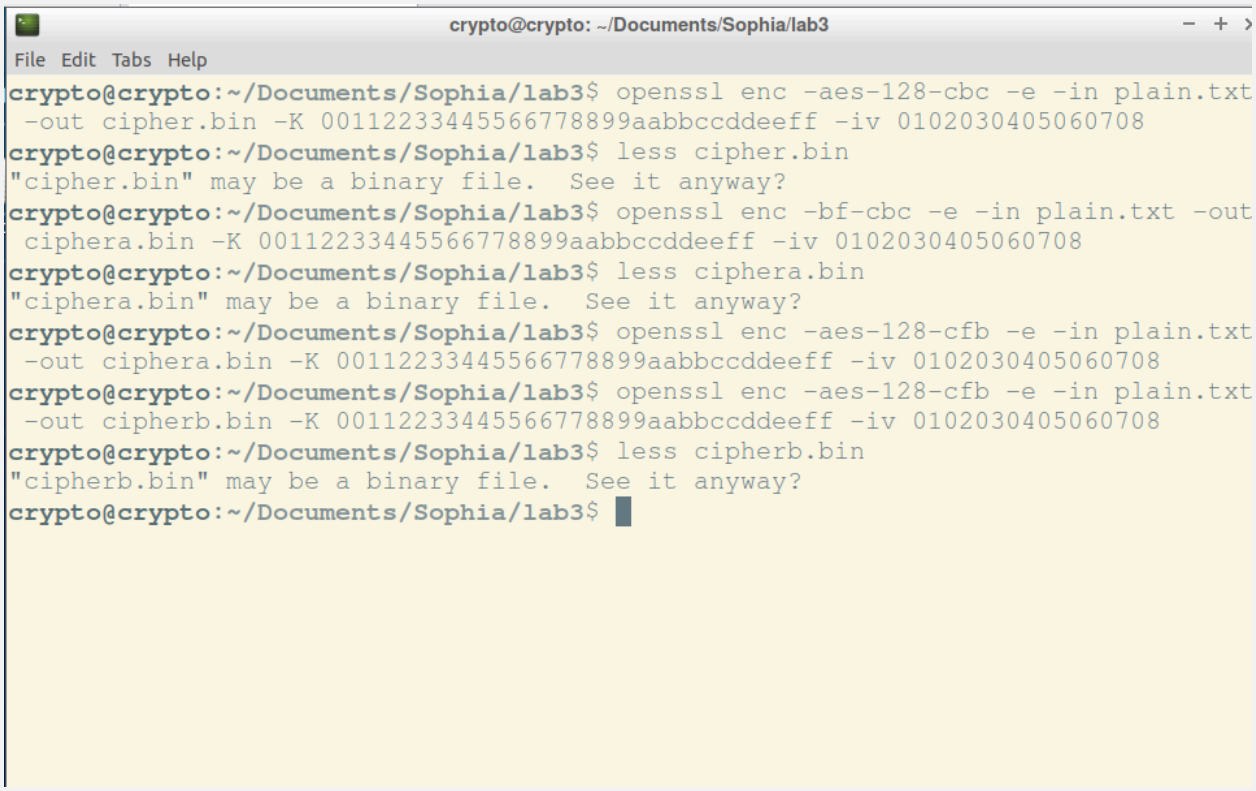

Lubuntu64 Crypto - VMware Workstation
File Edit View VM Tabs Help
Home
Lubuntu64 Crypto
Rubbish Bin
jcrypto
lab3
File Edit View Bookmarks Go Tools Help
/home/crypto/Documents/Sophia/lab3
Places
crypto@crypto: ~/Documents/Sophia/lab3
File Edit Tabs Help
UTS 41900 Cryptography 2020
plain.txt
crypto@crypto: ~/Doc...
lab3
14:29
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

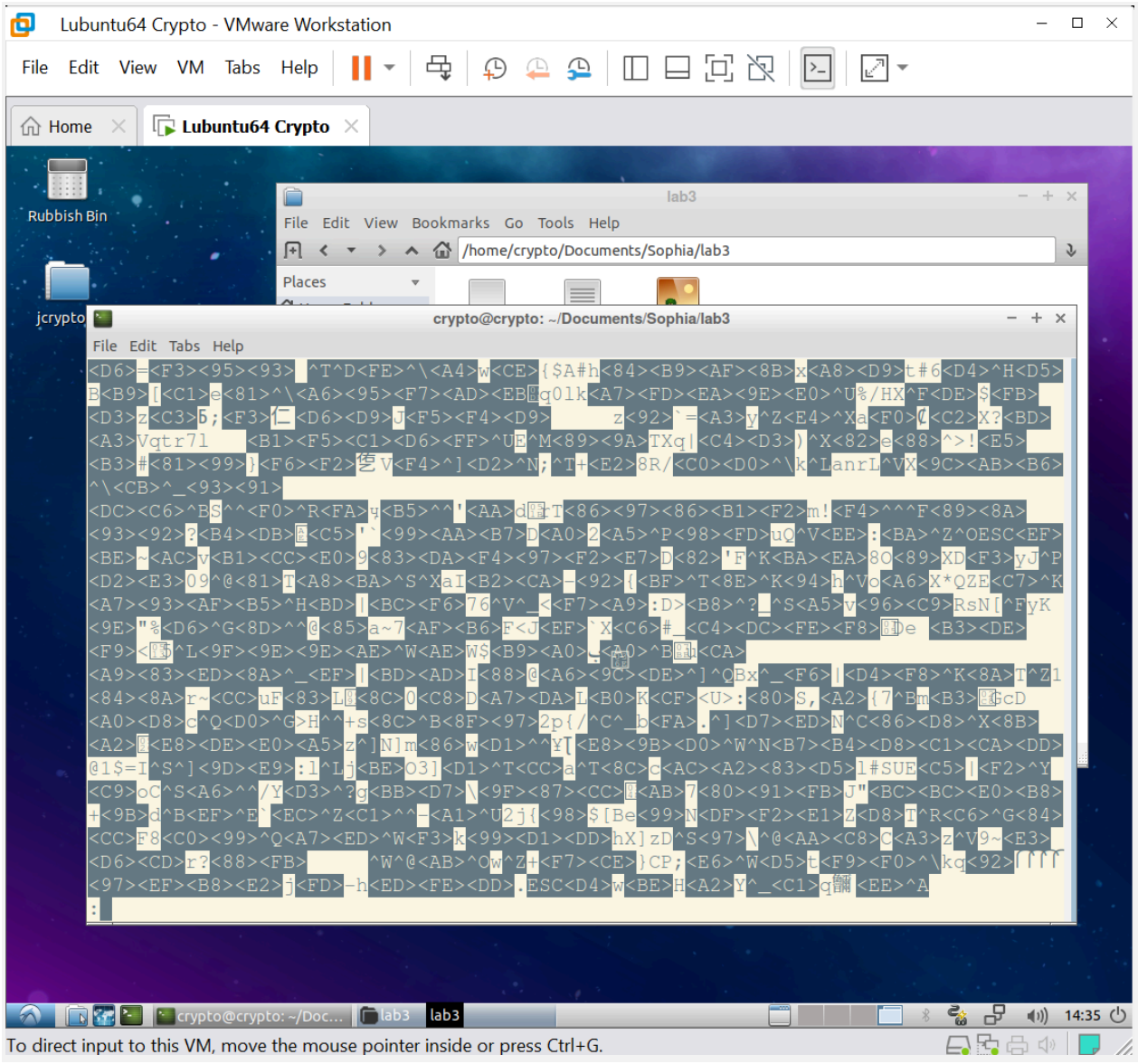
**-bf-cbc**


Lubuntu64 Crypto - VMware Workstation
File Edit View VM Tabs Help
Home
Lubuntu64 Crypto
crypto@crypto: ~/Documents/Sophia/lab3
File Edit Tabs Help
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -aes-128-cbc -e -in plain.txt -out cipher.bin -K 00112233445566778899aabbccddeeff -iv 0102030405060708
crypto@crypto:~/Documents/Sophia/lab3$ less cipher.bin
"cipher.bin" may be a binary file. See it anyway?
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -bf-cbc -e -in plain.txt -out ciphera.bin -K 00112233445566778899aabbccddeeff -iv 0102030405060708
crypto@crypto:~/Documents/Sophia/lab3$ less ciphera.bin
"ciphera.bin" may be a binary file. See it anyway?
crypto@crypto:~/Documents/Sophia/lab3$

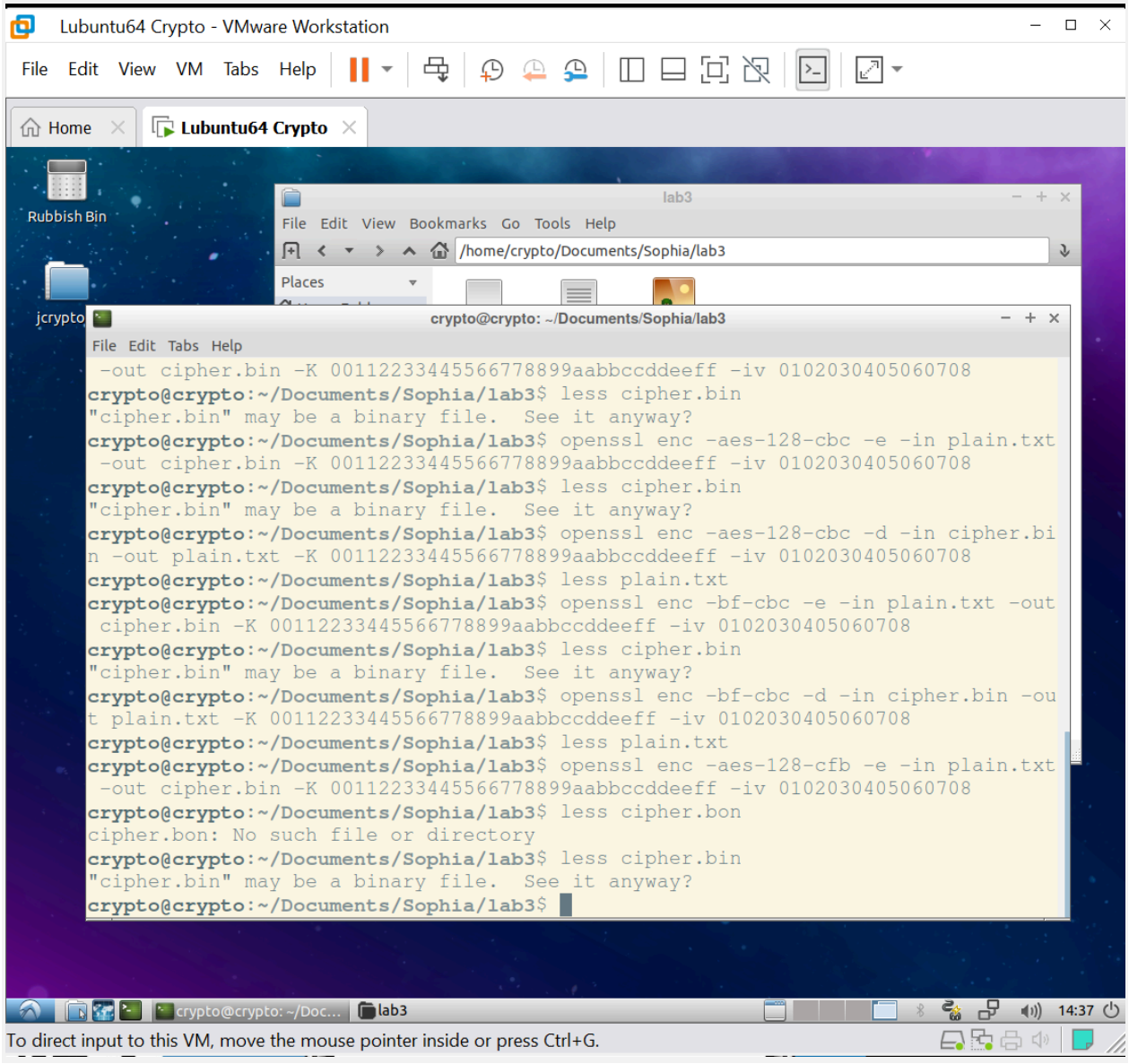

Lubuntu64 Crypto - VMware Workstation
File Edit View VM Tabs Help
Home
Lubuntu64 Crypto
Rubbish Bin
jcrypto
lab3
File Edit View Bookmarks Go Tools Help
/home/crypto/Documents/Sophia/lab3
Places
crypto@crypto: ~/Documents/Sophia/lab3
File Edit Tabs Help
crypto@crypto: ~/Doc...
lab3
14:31
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

**-aes-128-cfb**


crypto@crypto: ~/Documents/Sophia/lab3
File Edit Tabs Help
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -aes-128-cbc -e -in plain.txt -out cipher.bin -K 00112233445566778899aabbccddeeff -iv 0102030405060708
crypto@crypto:~/Documents/Sophia/lab3$ less cipher.bin
"cipher.bin" may be a binary file. See it anyway?
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -bf-cbc -e -in plain.txt -out ciphera.bin -K 00112233445566778899aabbccddeeff -iv 0102030405060708
crypto@crypto:~/Documents/Sophia/lab3$ less ciphera.bin
"ciphera.bin" may be a binary file. See it anyway?
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -aes-128-cfb -e -in plain.txt -out ciphera.bin -K 00112233445566778899aabbccddeeff -iv 0102030405060708
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -aes-128-cfb -e -in plain.txt -out cipherb.bin -K 00112233445566778899aabbccddeeff -iv 0102030405060708
crypto@crypto:~/Documents/Sophia/lab3$ less cipherb.bin
"cipherb.bin" may be a binary file. See it anyway?
crypto@crypto:~/Documents/Sophia/lab3$


Lubuntu64 Crypto - VMware Workstation
File Edit View VM Tabs Help
Home
Lubuntu64 Crypto
Rubbish Bin
jcrypto
lab3
File Edit View Bookmarks Go Tools Help
/home/crypto/Documents/Sophia/lab3
Places
crypto@crypto: ~/Documents/Sophia/lab3
File Edit Tabs Help
crypto@crypto: ~/Doc...
lab3
lab3
14:35
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

## Commands of 3 encryption, 3 decryption

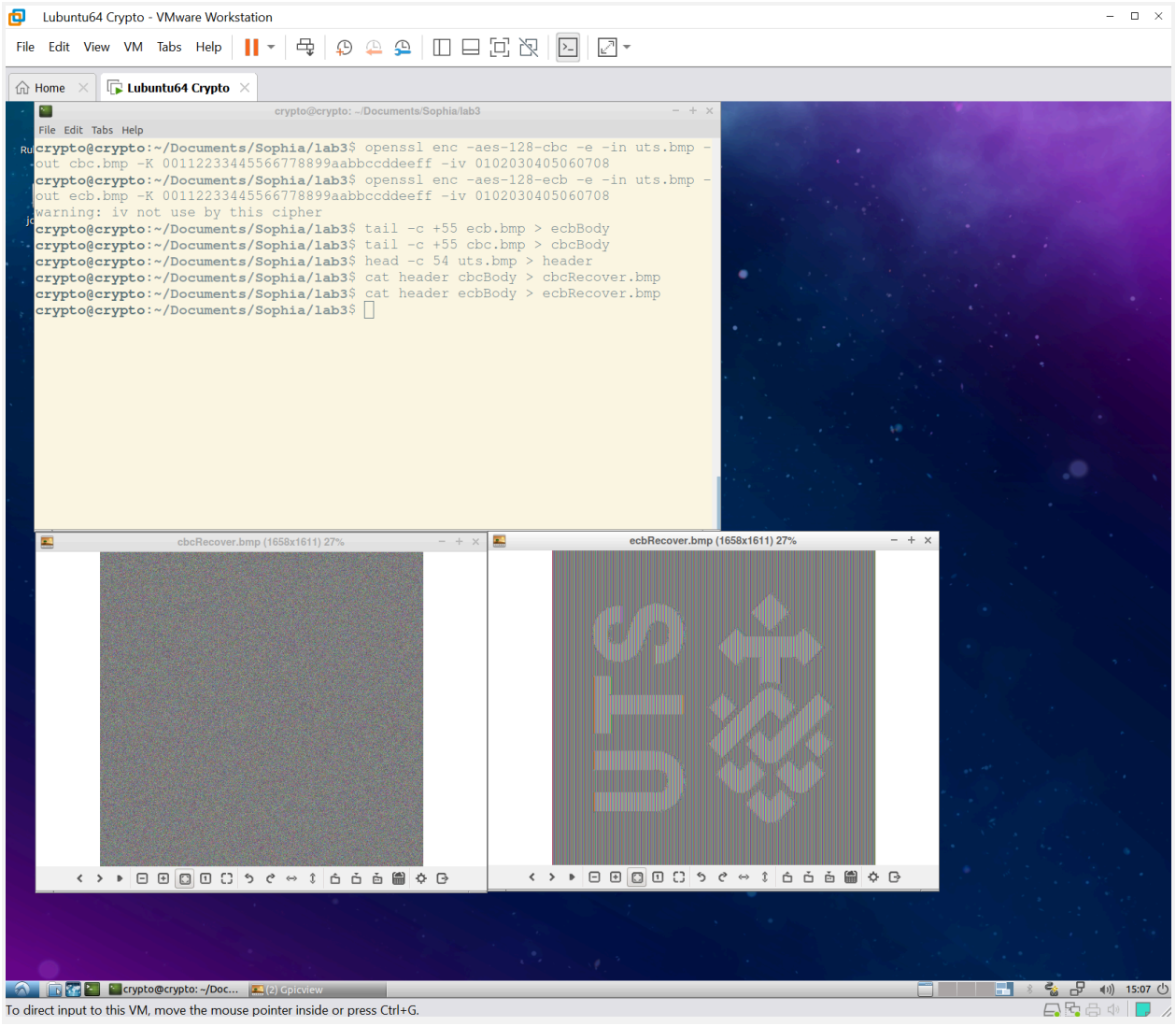


**Task 2 - Encryption mode - ECB vs CBC**

- Encrypting an image
- ECB - electronic code book
- CBC - cipher block chaining

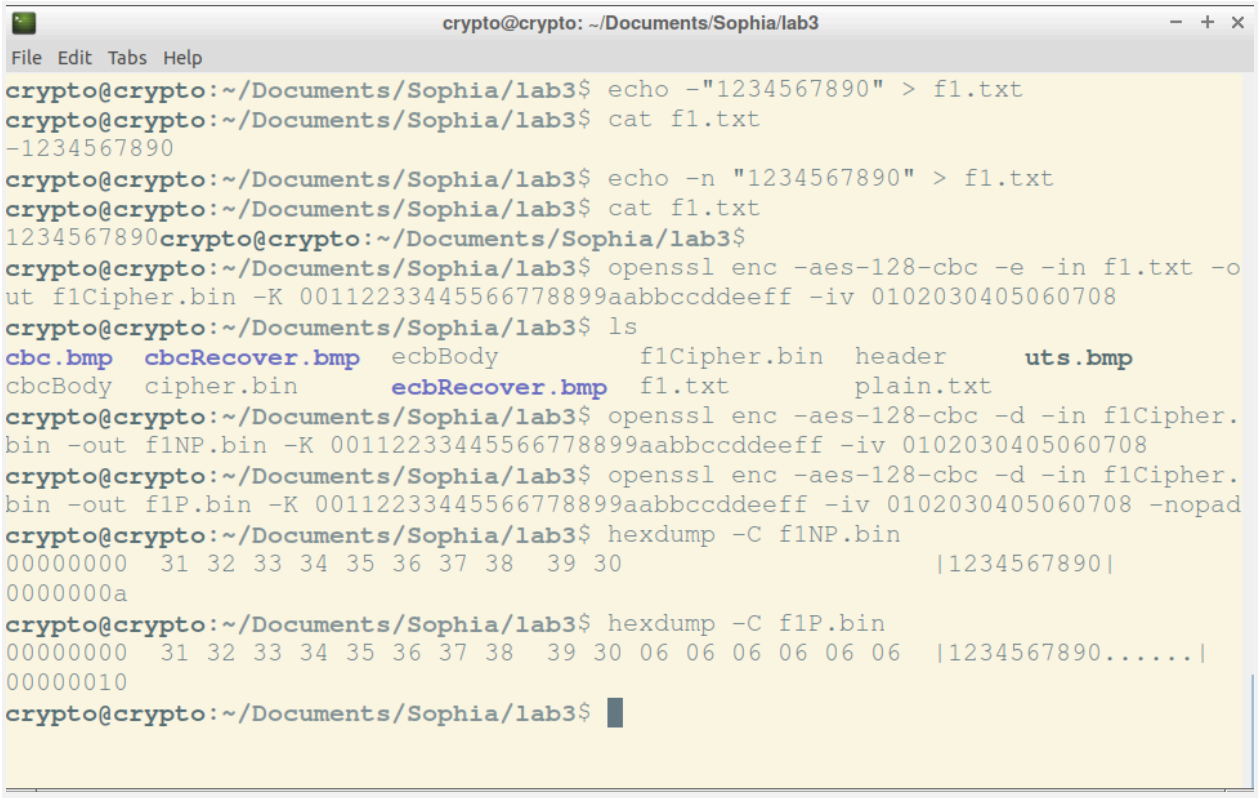
**Q: Can you derive any useful information about the original picture from the encrypted picture? Please explain your observations**

- In ECB mode- the encrypted picture is divided into blocks where each one is encrypted independently using the same key
    - For the UTS logo, the blocks are a more light grey- these ciphertext blocks are identical to one another to create a pattern (weak diffusion, since it's easier to spot patterns)
- In CBC mode- the encrypted picture is more pixelated, there are no repetitive patterns because of the chaining process where the identical blocks of the plaintext are encrypted to different ciphertexts (strong diffusion, small changes in the og image creates a big change to the ciphertext [avalanche effect]- random noise to the encrypted file)
    - CBC is much better than ECB due to its unpredictability with encrypting


Lubuntu64 Crypto - VMware Workstation
File Edit View VM Tabs Help
Home
Lubuntu64 Crypto
crypto@crypto: ~/Documents/Sophia/lab3
File Edit Tabs Help
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -aes-128-cbc -e -in uts.bmp -out cbc.bmp -K 00112233445566778899aabbccddeeff -iv 0102030405060708
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -aes-128-ecb -e -in uts.bmp -out ecb.bmp -K 00112233445566778899aabbccddeeff -iv 0102030405060708
warning: iv not use by this cipher
crypto@crypto:~/Documents/Sophia/lab3$ tail -c +55 ecb.bmp > ecbBody
crypto@crypto:~/Documents/Sophia/lab3$ tail -c +55 cbc.bmp > cbcBody
crypto@crypto:~/Documents/Sophia/lab3$ head -c 54 uts.bmp > header
crypto@crypto:~/Documents/Sophia/lab3$ cat header cbcBody > cbcRecover.bmp
crypto@crypto:~/Documents/Sophia/lab3$ cat header ecbBody > ecbRecover.bmp
crypto@crypto:~/Documents/Sophia/lab3$
cbcRecover.bmp (1658x1611) 27%
ecbRecover.bmp (1658x1611) 27%
15:07
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

**Task 3- Padding** (adding extra data to plaintext before encryption to ensure it's the correct size for the block cipher, encryption algorithm doesn't process incomplete blocks)

- Command: -nopad (disables padding)
- Block ciphers use PKCS#5 padding (standard block padding)
- ECB, CBC- requires padding
- CFB, OFB- doesn't require padding,
    - CFB- turns block cipher into stream cipher instead of encrypting data into blocks (not restricted to blocks)
    - OFB- processes plaintext in smaller chunks rather than encrypting them into blocks
- Steps:
    - 1. Creating two files that contain 10 bytes, 16 bytes
    - 2. Using cbc mode to encrypt
    - 3. decrypting , using -nopad on one file to figure out padding
        - Decrypting automatically removes padding, by putting in -nopad it cancels out the command

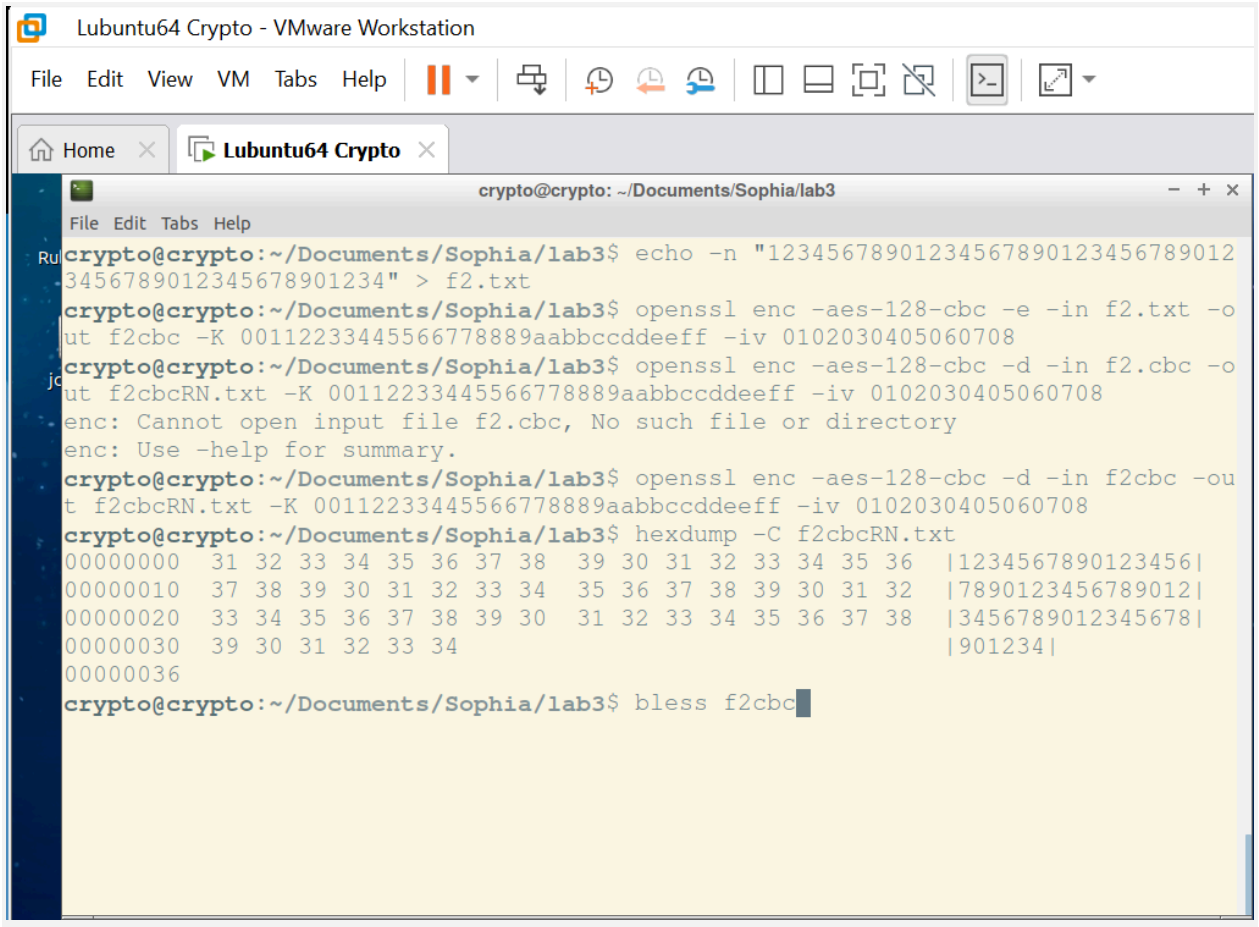


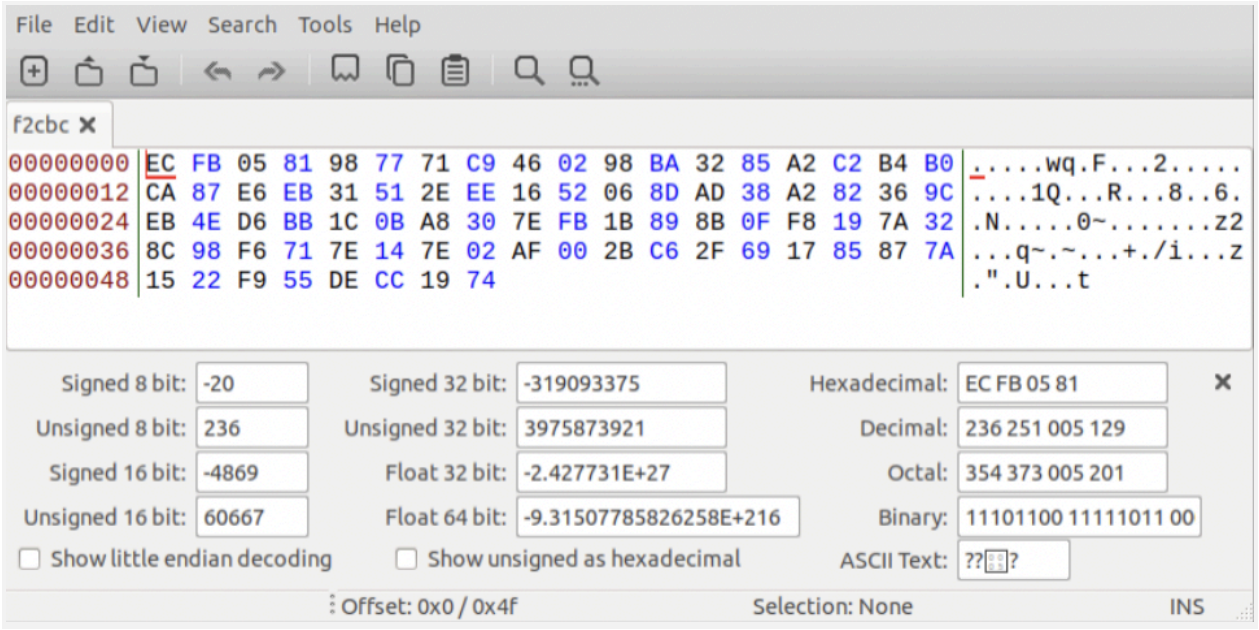
**Task 4- Error propagation - corrupted cipher text**

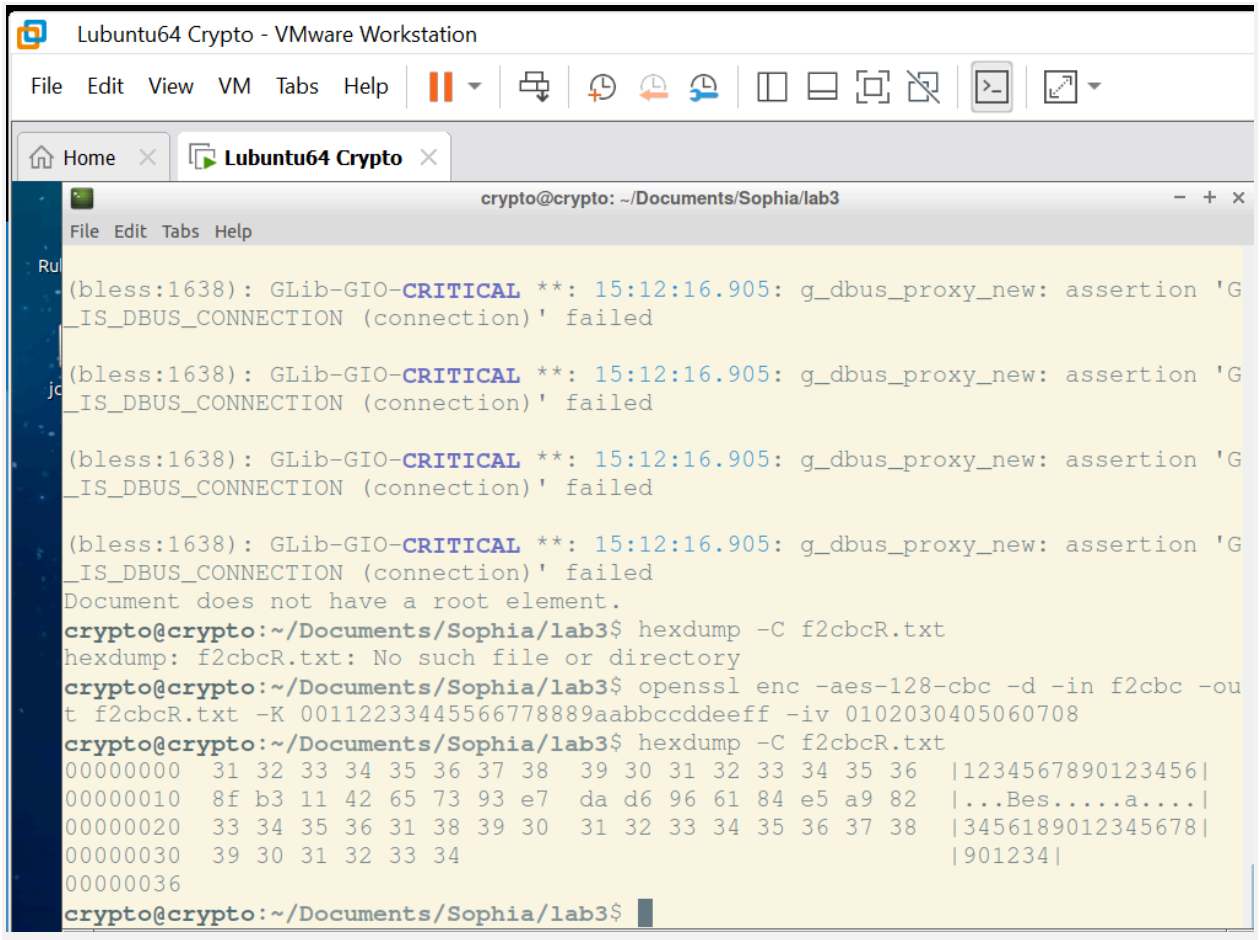
- Steps:
    - 1. Create text file that's 64 bytes long
    - 2. Encrypt using aes-128
    - 3. Corrupt using bless hex editor

**Q: How much information can you recover by decrypting the corrupted file, if the encryption mode is ECB, CBC, CFB, OFB, and CTR respectively? Please find it out after you finish this task and provide justification**

- The information that can be recovered depends on how the mode can handle errors in the ciphertext
- ECB (electronic codebook) - data is divided into blocks, encrypted independently (some blocks can be similar)
    - One single corrupted block wouldn't affect any other blocks
    - Recovered information- everything except for the corrupted block
- CBC (cipher block chaining) - each block to be encrypted depends on the previous block
    - If one block is corrupted, the next block will also be corrupted
    - Recovered information- everything except for the corrupted two consecutive blocks
- CFB (cipher feedback) - estimates the error to a small number to consecutive bytes
    - Corruption for consecutive bytes
    - Recovered information- everything except for the 55th byte and the next byte
- OFB (Output feedback)- block of ciphertext produced by XORing (for error checking by comparing two input bits and generating one output bit) plaintext using the previous block
    - corruption doesn't spread
    - Recovered information- everything except for the corrupted byte
- CTR (counter)- counter increases for each block, resulted value is encrypted to create a keystream that is XORed with the plaintext to produce the ciphertext
    - If the ciphertext byte is corrupted, the corresponding plaintext is corrupted (doesn't spread)
    - Recovered information- everything except for the corrupted data


Lubuntu64 Crypto - VMware Workstation
File Edit View VM Tabs Help
Home
Lubuntu64 Crypto
crypto@crypto: ~/Documents/Sophia/lab3
File Edit Tabs Help
crypto@crypto:~/Documents/Sophia/lab3$ echo -n "12345678901234567890123456789012345678901234567890123456789012345678901234" > f2.txt
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -aes-128-cbc -e -in f2.txt -out f2cbc -K 00112233445566778889aabbccddeeff -iv 0102030405060708
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -aes-128-cbc -d -in f2.cbc -out f2cbcRN.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
enc: Cannot open input file f2.cbc, No such file or directory
enc: Use -help for summary.
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -aes-128-cbc -d -in f2cbc -out f2cbcRN.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
crypto@crypto:~/Documents/Sophia/lab3$ hexdump -C f2cbcRN.txt
00000000 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 |1234567890123456|
00000010 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32 |7890123456789012|
00000020 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 |3456789012345678|
00000030 39 30 31 32 33 34 |901234|
00000036
crypto@crypto:~/Documents/Sophia/lab3$ bless f2cbc


File Edit View Search Tools Help
f2cbc
00000000 EC FB 05 81 98 77 71 C9 46 02 98 BA 32 85 A2 C2 B4 B0 .....wq.F...2.....
00000012 CA 87 E6 EB 31 51 2E EE 16 52 06 8D AD 38 A2 82 36 9C ....1Q...R...8..6.
00000024 EB 4E D6 BB 1C 0B A8 30 7E FB 1B 89 8B 0F F8 19 7A 32 .N.....0~.......z2
00000036 8C 98 F6 71 7E 14 7E 02 AF 00 2B C6 2F 69 17 85 87 7A ...q~.~...+./i...z
00000048 15 22 F9 55 DE CC 19 74 ."U...t
Signed 8 bit: -20
Signed 32 bit: -319093375
Hexadecimal: EC FB 05 81
Unsigned 8 bit: 236
Unsigned 32 bit: 3975873921
Decimal: 236 251 005 129
Signed 16 bit: -4869
Float 32 bit: -2.427731E+27
Octal: 354 373 005 201
Unsigned 16 bit: 60667
Float 64 bit: -9.31507785826258E+216
Binary: 11101100 11111011 00
Show little endian decoding
Show unsigned as hexadecimal
ASCII Text: ??□?
Offset: 0x0 / 0x4f
Selection: None
INS


Lubuntu64 Crypto - VMware Workstation
File Edit View VM Tabs Help
Home
Lubuntu64 Crypto
crypto@crypto: ~/Documents/Sophia/lab3
File Edit Tabs Help
(bless:1638): GLib-GIO-CRITICAL **: 15:12:16.905: g_dbus_proxy_new: assertion 'G
_IS_DBUS_CONNECTION (connection)' failed
(bless:1638): GLib-GIO-CRITICAL **: 15:12:16.905: g_dbus_proxy_new: assertion 'G
_IS_DBUS_CONNECTION (connection)' failed
(bless:1638): GLib-GIO-CRITICAL **: 15:12:16.905: g_dbus_proxy_new: assertion 'G
_IS_DBUS_CONNECTION (connection)' failed
(bless:1638): GLib-GIO-CRITICAL **: 15:12:16.905: g_dbus_proxy_new: assertion 'G
_IS_DBUS_CONNECTION (connection)' failed
Document does not have a root element.
crypto@crypto:~/Documents/Sophia/lab3$ hexdump -C f2cbcR.txt
hexdump: f2cbcR.txt: No such file or directory
crypto@crypto:~/Documents/Sophia/lab3$ openssl enc -aes-128-cbc -d -in f2cbc -ou
t f2cbcR.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
crypto@crypto:~/Documents/Sophia/lab3$ hexdump -C f2cbcR.txt
00000000 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 |1234567890123456|
00000010 8f b3 11 42 65 73 93 e7 da d6 96 61 84 e5 a9 82 |...Bes.....a....|
00000020 33 34 35 36 31 38 39 30 31 32 33 34 35 36 37 38 |3456189012345678|
00000030 39 30 31 32 33 34 |901234|
00000036
crypto@crypto:~/Documents/Sophia/lab3$

**Task 5- Initial vector (IV)**

- Properties of an IV depend on the cryptographic scheme used
- Secures data encryption- it's important to properly select an IV
- Requirement for IV: uniqueness, should never be reused under the same key
    - It needs to be unique because if the same one is used every time, the output ciphertext will be the same if the same plaintext data is used- ciphertext will be predictable to cryptanalysis

Q: What properties the IV should have? Why?

- Randomness- helps with the IV being unique, preventing identical ciphertexts
- Length- 128 bits to match the block size of the cipher for AES (if it's not the right size there will be errors in the encryption)
- One use- an IV shouldn't be repeated to enforce uniqueness
- Should be independent from encryption key- prevents attackers from figuring out a pattern using the IV