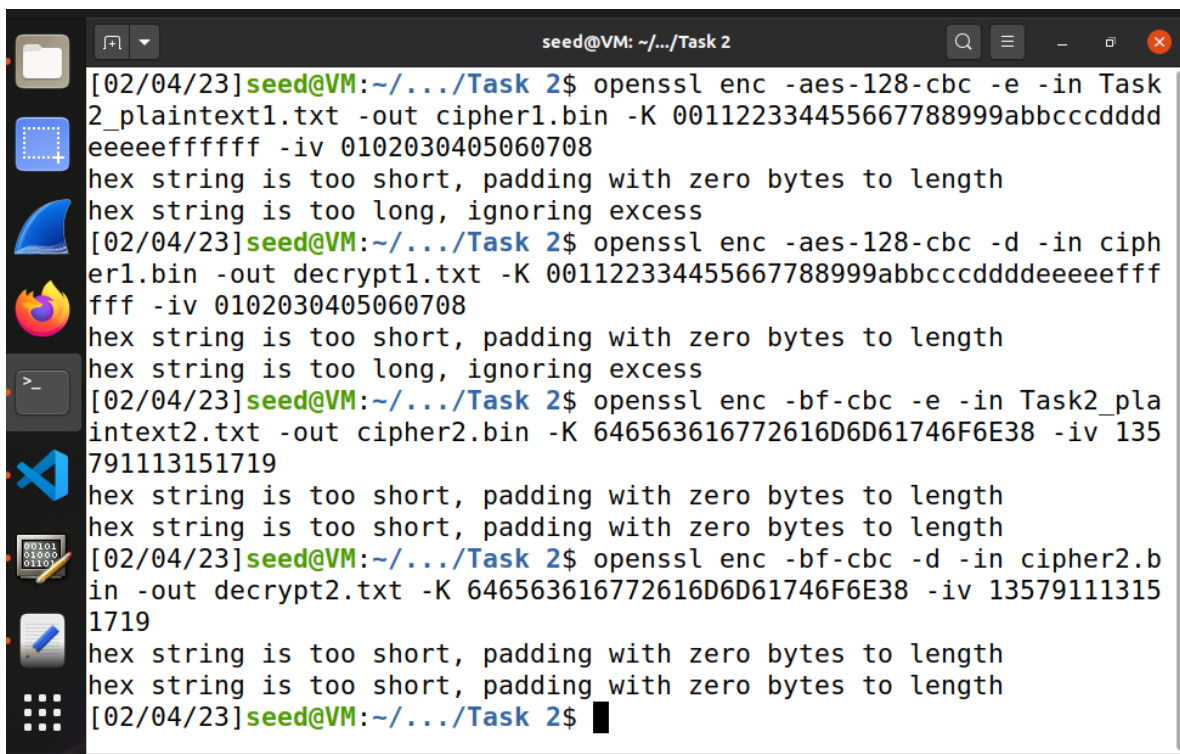


Lab 1 Report

Task 1: For the ciphered text, I used this website <https://crypto.interactive-maths.com/frequency-analysis-breaking-the-code.html> to decrypt the encrypted text. I also decrypt the text according to the frequency of alphabets.

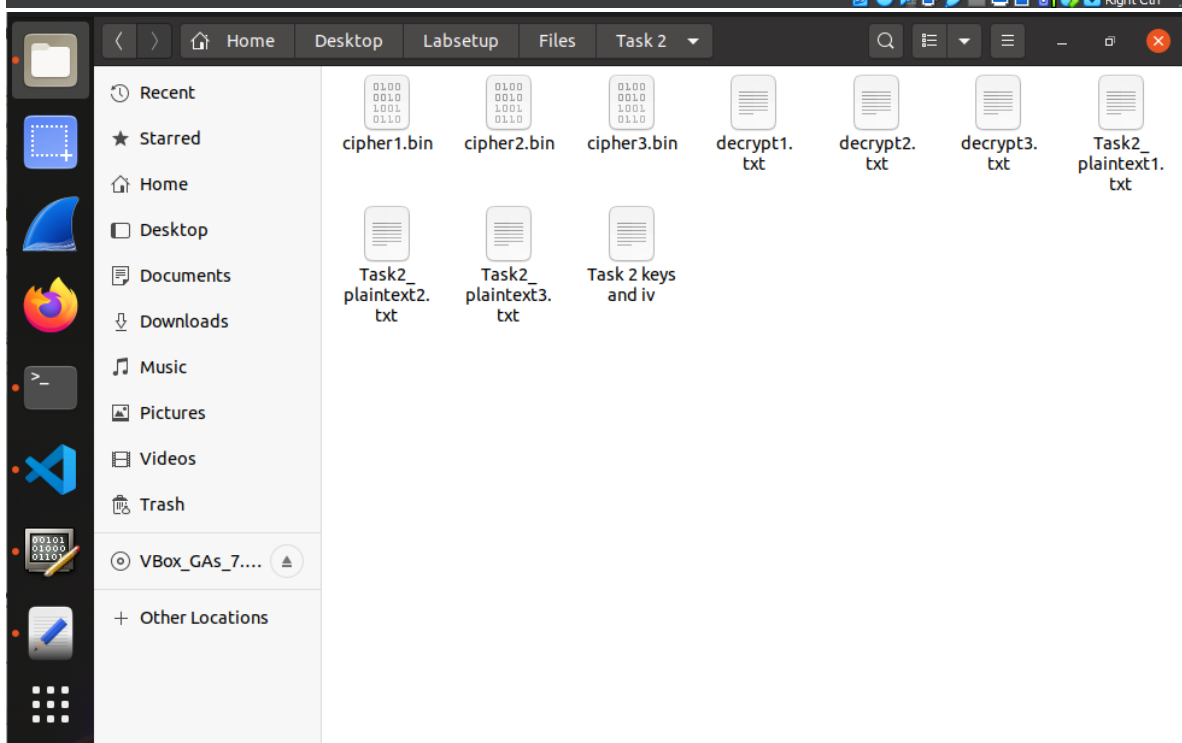
At last, I found out that the key is: **vgapnbrtmosicuxejhqyzflkdw**, and the decrypted text is from the article: *What to Expect (and Not Expect) at the Oscars*

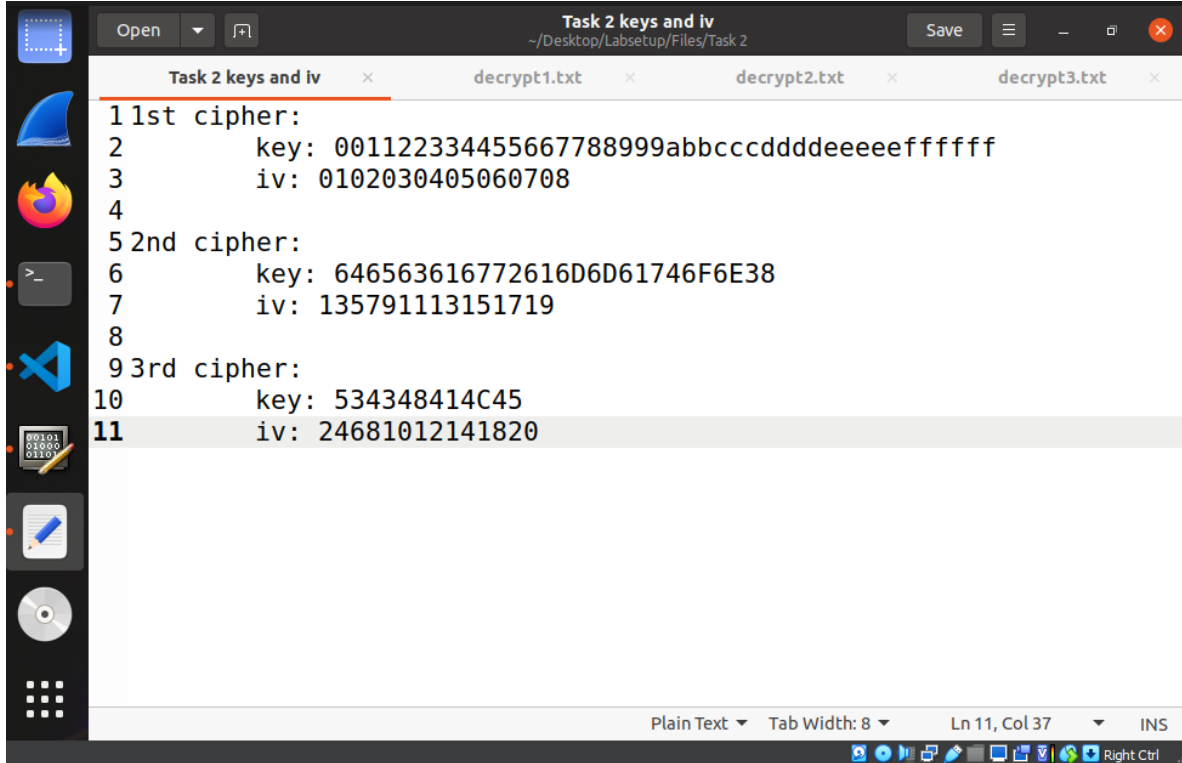
Task2: In this task, I had 3 different cipher type. aes-128-cbc, aes-128-cfb, and bf-cbc. With the key and vi, it will be encrypted and can be decrypted with correct key and vi.



```
seed@VM: ~/.../Task 2
[02/04/23]seed@VM:~/.../Task 2$ openssl enc -aes-128-cbc -e -in Task2_plaintext1.txt -out cipher1.bin -K 0011223344556677889999abbccddddd
eeeeeffffffff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
hex string is too long, ignoring excess
[02/04/23]seed@VM:~/.../Task 2$ openssl enc -aes-128-cbc -d -in cipher1.bin -out decrypt1.txt -K 0011223344556677889999abbccddddd
eeeeeffffffff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
hex string is too long, ignoring excess
[02/04/23]seed@VM:~/.../Task 2$ openssl enc -bf-cbc -e -in Task2_plaintext2.txt -out cipher2.bin -K 646563616772616D6D61746F6E38 -iv 135791113151719
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[02/04/23]seed@VM:~/.../Task 2$ openssl enc -bf-cbc -d -in cipher2.bin -out decrypt2.txt -K 646563616772616D6D61746F6E38 -iv 135791113151719
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[02/04/23]seed@VM:~/.../Task 2$
```

```
seed@VM: ~/.../Task 2
hex string is too short, padding with zero bytes to length
hex string is too long, ignoring excess
[02/04/23]seed@VM:~/.../Task 2$ openssl enc -bf-cbc -e -in Task2_pla
intext2.txt -out cipher2.bin -K 646563616772616D6D61746F6E38 -iv 135
791113151719
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[02/04/23]seed@VM:~/.../Task 2$ openssl enc -bf-cbc -d -in cipher2.b
in -out decrypt2.txt -K 646563616772616D6D61746F6E38 -iv 13579111315
1719
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[02/04/23]seed@VM:~/.../Task 2$ openssl enc -aes-128-cfb -e -in Task
2_plaintext3.txt -out cipher3.bin -K 534348414C45 -iv 24681012141820
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[02/04/23]seed@VM:~/.../Task 2$ openssl enc -aes-128-cfb -d -in ciph
er3.bin -out decrypt3.txt -K 534348414C45 -iv 24681012141820
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[02/04/23]seed@VM:~/.../Task 2$
```





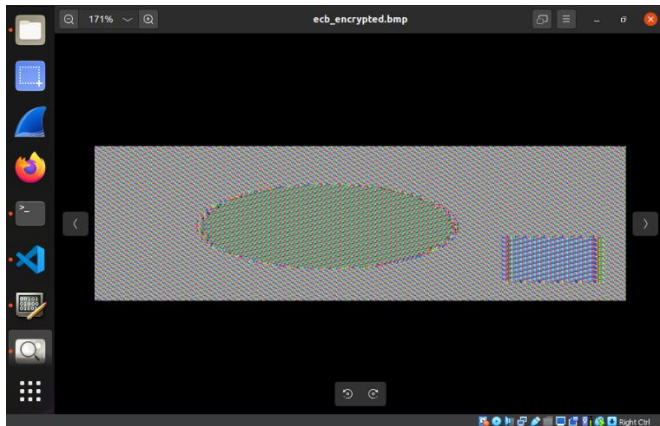
The screenshot shows a code editor window with the title "Task 2 keys and iv" and a file path of "~/Desktop/Labsetup/Files/Task 2". The editor has four tabs: "Task 2 keys and iv" (active), "decrypt1.txt", "decrypt2.txt", and "decrypt3.txt". The content of the active tab is as follows:

```
1 1st cipher:
2     key: 001122334455667788999abbccdddeeeeffffff
3     iv: 0102030405060708
4
5 2nd cipher:
6     key: 646563616772616D6D61746F6E38
7     iv: 135791113151719
8
9 3rd cipher:
10    key: 534348414C45
11    iv: 24681012141820
```

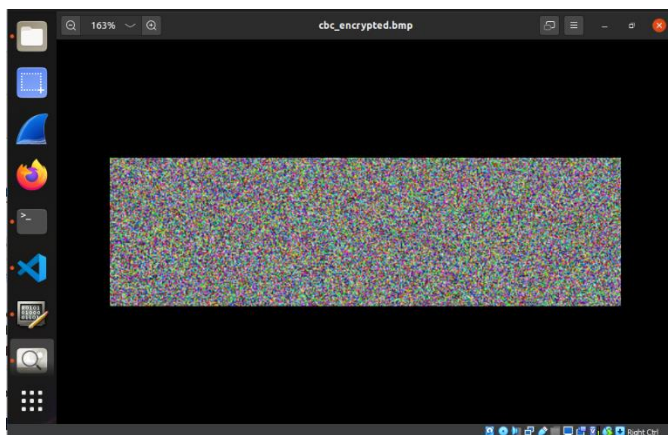
The status bar at the bottom indicates "Plain Text", "Tab Width: 8", "Ln 11, Col 37", and "INS". The system tray at the bottom right shows various icons and the text "Right Ctrl".

Task 3: Here are 2 photos that are encrypted in 2 ways. One is ECB and another one is CBC. It is clear that ECB has better encryption which completely disoriented the original picture.

ECB

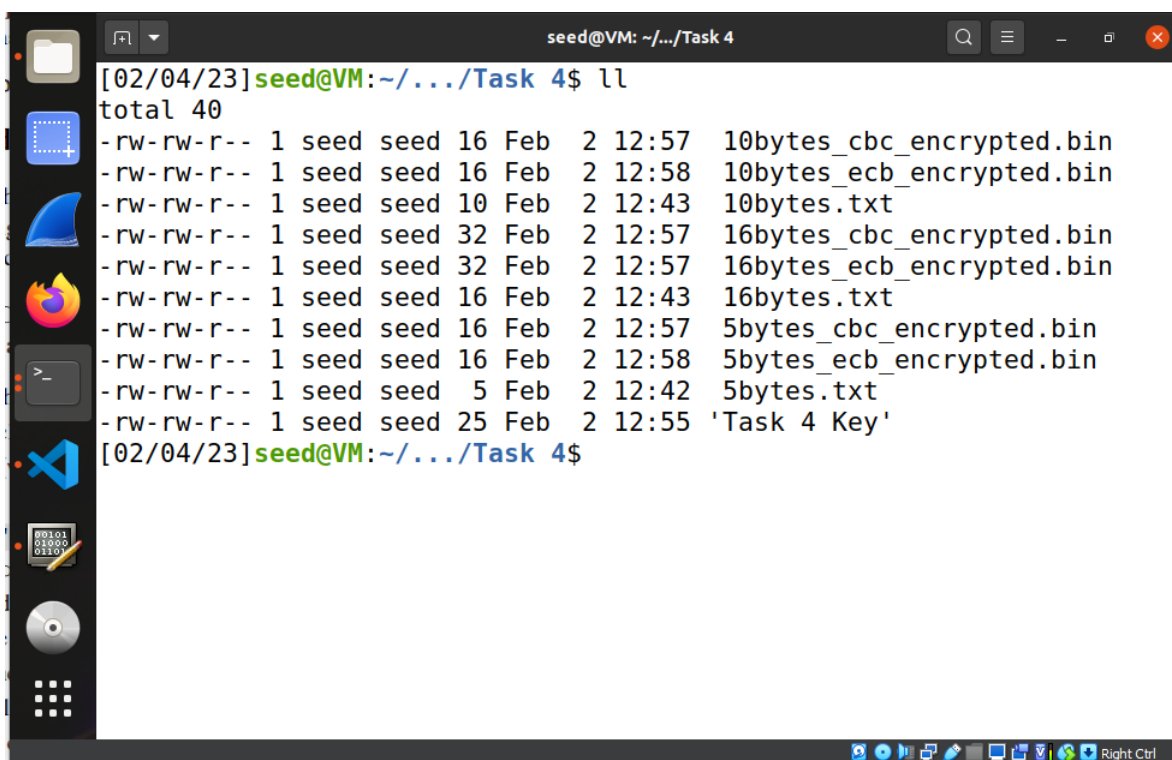


CBC



Task 4:

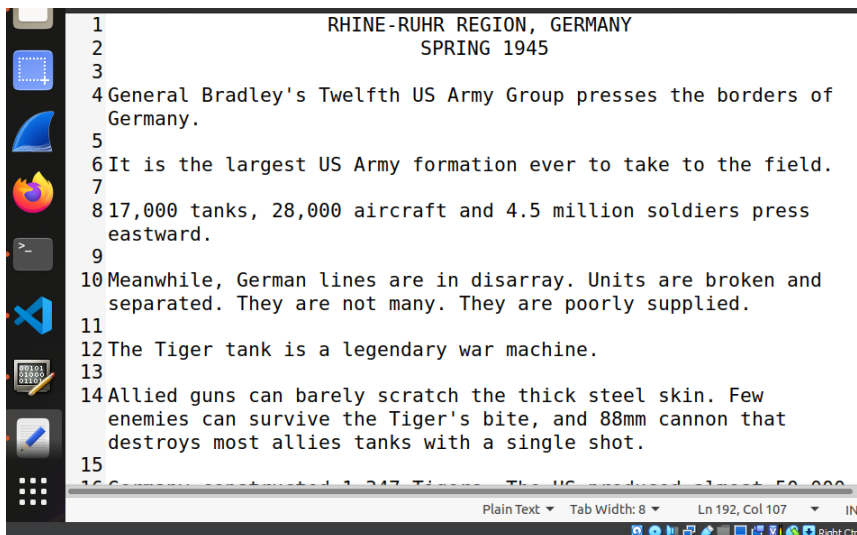
I created 3 different files which are 5, 10, and 16 bytes. In plain text, the file size will be corresponding to bytes it has. However, when I encrypt with either cbc or ecb, the file will automatically round up in 16 bytes. That is, if the plain text is 15 bytes, the ciphered one will be 16 bytes, and if the plain text is 17 bytes, then it will round up to 32 bytes as padding.

A terminal window titled 'seed@VM: ~/.../Task 4' with a search icon, menu icon, and window controls. The terminal shows the command 'll' and its output, which is a file listing. The listing shows 11 files with permissions, owner, group, size, date, time, and filename. The files are: 10bytes_cbc_encrypted.bin, 10bytes_ecb_encrypted.bin, 10bytes.txt, 16bytes_cbc_encrypted.bin, 16bytes_ecb_encrypted.bin, 16bytes.txt, 5bytes_cbc_encrypted.bin, 5bytes_ecb_encrypted.bin, 5bytes.txt, and 'Task 4 Key'. The terminal prompt is '[02/04/23] seed@VM: ~/.../Task 4\$'.

```
[02/04/23] seed@VM: ~/.../Task 4$ ll
total 40
-rw-rw-r-- 1 seed seed 16 Feb  2 12:57 10bytes_cbc_encrypted.bin
-rw-rw-r-- 1 seed seed 16 Feb  2 12:58 10bytes_ecb_encrypted.bin
-rw-rw-r-- 1 seed seed 10 Feb  2 12:43 10bytes.txt
-rw-rw-r-- 1 seed seed 32 Feb  2 12:57 16bytes_cbc_encrypted.bin
-rw-rw-r-- 1 seed seed 32 Feb  2 12:57 16bytes_ecb_encrypted.bin
-rw-rw-r-- 1 seed seed 16 Feb  2 12:43 16bytes.txt
-rw-rw-r-- 1 seed seed 16 Feb  2 12:57 5bytes_cbc_encrypted.bin
-rw-rw-r-- 1 seed seed 16 Feb  2 12:58 5bytes_ecb_encrypted.bin
-rw-rw-r-- 1 seed seed  5 Feb  2 12:42 5bytes.txt
-rw-rw-r-- 1 seed seed 25 Feb  2 12:55 'Task 4 Key'
[02/04/23] seed@VM: ~/.../Task 4$
```

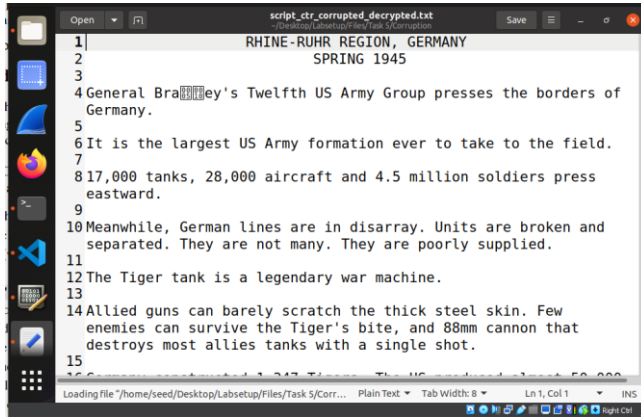
Task 5: In this task, I prepared a length more than 1000 byte plaintext. I encrypted it with cbc, cfb, ctr, and ect. Then, I tampered with those encrypted file's 55th byte. After I decrypted them back, I found out that ctr only has 2 characters being corrupted, and the rest of the text are fine.

Plain text:

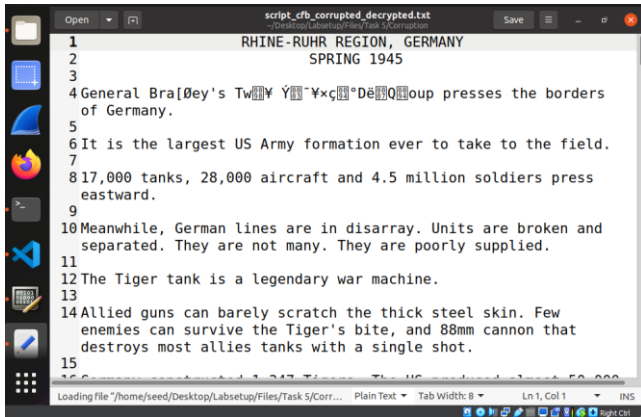


```
1          RHINE-RUHR REGION, GERMANY
2          SPRING 1945
3
4 General Bradley's Twelfth US Army Group presses the borders of
   Germany.
5
6 It is the largest US Army formation ever to take to the field.
7
8 17,000 tanks, 28,000 aircraft and 4.5 million soldiers press
   eastward.
9
10 Meanwhile, German lines are in disarray. Units are broken and
    separated. They are not many. They are poorly supplied.
11
12 The Tiger tank is a legendary war machine.
13
14 Allied guns can barely scratch the thick steel skin. Few
   enemies can survive the Tiger's bite, and 88mm cannon that
   destroys most allies tanks with a single shot.
15
16 Germany constructed 1,347 Tigers. The US produced almost 50,000
```

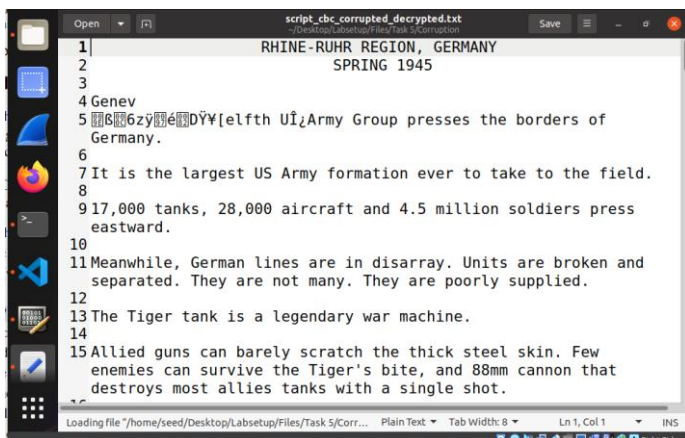
CTR corrupted



CFB corrupted



CBC corrupted

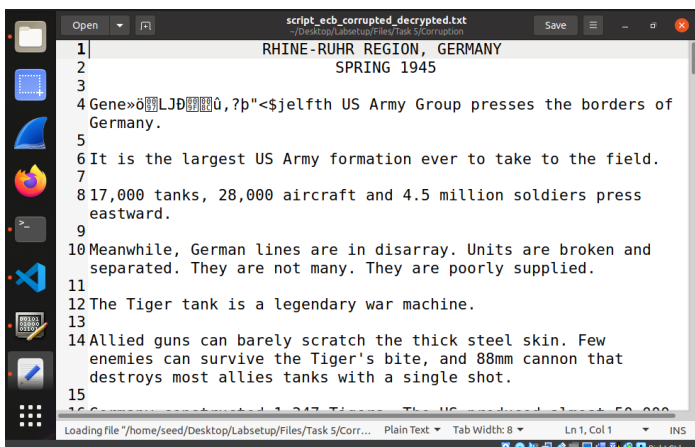


The screenshot shows a text editor window titled "script_cbc_corrupted_decrypted.txt". The text is as follows:

```
1| RHINE-RUHR REGION, GERMANY
2| SPRING 1945
3|
4| Genev
5| 6zzyéDÿY[elfth UîArmy Group presses the borders of
6| Germany.
7| It is the largest US Army formation ever to take to the field.
8|
9| 17,000 tanks, 28,000 aircraft and 4.5 million soldiers press
10| eastward.
11| Meanwhile, German lines are in disarray. Units are broken and
12| separated. They are not many. They are poorly supplied.
13| The Tiger tank is a legendary war machine.
14|
15| Allied guns can barely scratch the thick steel skin. Few
16| enemies can survive the Tiger's bite, and 88mm cannon that
17| destroys most allies tanks with a single shot.
```

The text is corrupted due to a CBC decryption error, with some characters appearing as garbled symbols.

ECB corrupted

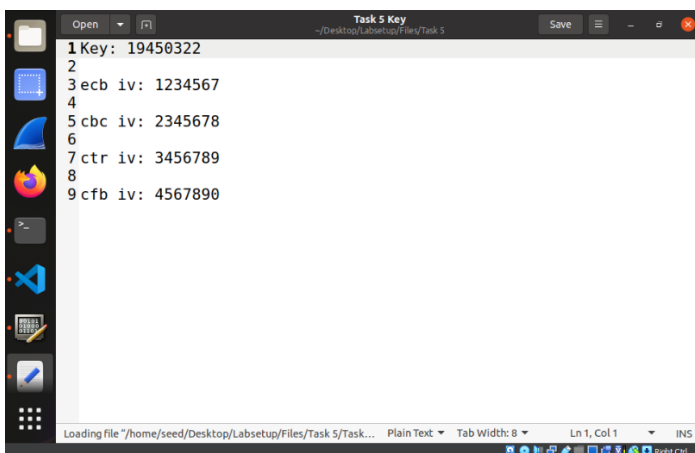


The screenshot shows a text editor window titled "script_ecb_corrupted_decrypted.txt". The text is as follows:

```
1| RHINE-RUHR REGION, GERMANY
2| SPRING 1945
3|
4| Gene>öLJBü,?p"<$jelfth US Army Group presses the borders of
5| Germany.
6| It is the largest US Army formation ever to take to the field.
7|
8| 17,000 tanks, 28,000 aircraft and 4.5 million soldiers press
9| eastward.
10| Meanwhile, German lines are in disarray. Units are broken and
11| separated. They are not many. They are poorly supplied.
12| The Tiger tank is a legendary war machine.
13|
14| Allied guns can barely scratch the thick steel skin. Few
15| enemies can survive the Tiger's bite, and 88mm cannon that
16| destroys most allies tanks with a single shot.
```

The text is corrupted due to an ECB decryption error, with some characters appearing as garbled symbols.

Key



The screenshot shows a text editor window titled "Task 5 Key". The text is as follows:

```
1| Key: 19450322
2|
3| ecb iv: 1234567
4|
5| cbc iv: 2345678
6|
7| ctr iv: 3456789
8|
9| cfb iv: 4567890
```

The text is the decryption key for the task.