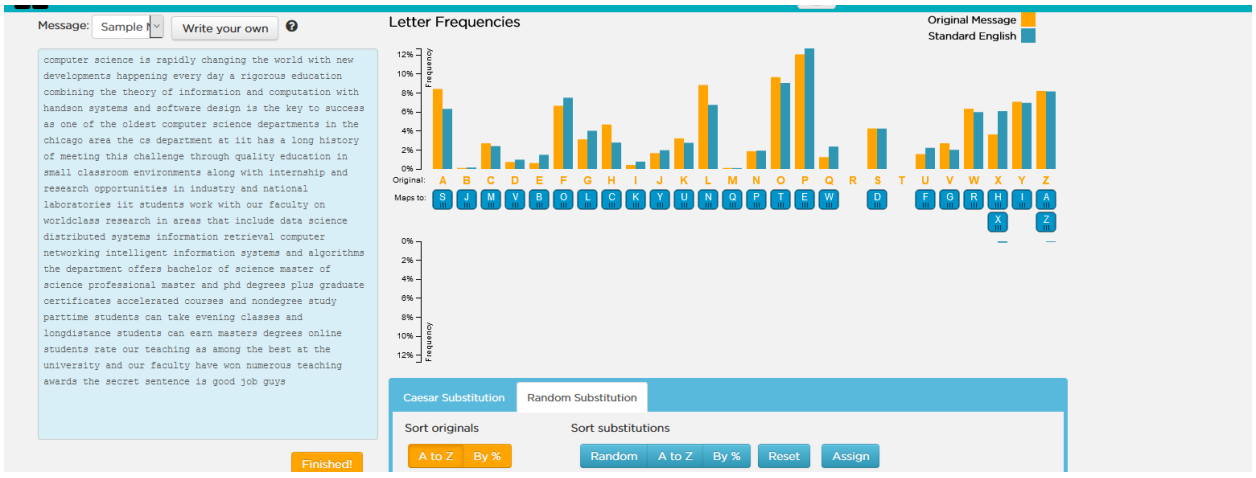
## Lab 10: Secret Key Encryption

### Introduction:

The practice of encrypting information for the sake of confidentiality dates as far back as the Roman Empire in the form of “Caesar’s Cipher”: a simple shift cipher used to keep military orders secret from the enemy. Though today’s ciphers and encryption methods are considerably more advanced than they were in Roman times, the underlying purpose of maintaining information confidentiality remains the same. As such, the focus of this lab was to gain a better understanding of modern cryptography. Namely using OpenSSL and its myriad number of ciphers.

### Task 1: Frequency Analysis

Frequency analysis can be employed to crack any mono or poly alphabetic cipher. Commonly used bigrams and trigrams such as “and”, “the”, and “of” offer the most insight as to the mapping used in encrypting the plaintext. A frequency analysis for the provided ciphertext along with the command used to translate it are shown below.




The trigram “oxp” and the bigram “fu” were observed many times in the ciphertext. Simple trial and error was used to determine the mappings, first assuming that “oxp”

mapped to “and” and fu mapped to “it”. This turned out to be true, and made it exponentially easier to crack the cipher.

```
[03/03/21]seed@VM:~/.../lab02$ tr 'a-z' 'sjmvbolckyunqptewxdzfgrhia' < ciphertex
t.txt > plaintext.txt
[03/03/21]seed@VM:~/.../lab02$
```

```
computer science is rapidly changing the world with new developments happening every day a rigorous education combining the theory of informa
tion and computation with handson systems and software design is the key to success as one of the oldest computer science departments in the
chicago area the cs department at iit has a long history of meeting this challenge through quality education in small classroom environments
along with internship and research opportunities in industry and national laboratories
iit students work with our faculty on worldclass research in areas that include data science distributed systems information retrieval comput
er networking intelligent information systems and algorithms
the department offers bachelor of science master of science professional master and phd degrees plus graduate certificates accelerated course
s and nondegree study parttime students can take evening classes and longdistance students can earn masters degrees online students rate our
teaching as among the best at the university and our faculty have won numerous teaching awards
the secret sentence is good job guys
~
~
~
```

Final plaintext as viewed through the vi editor.

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

Different ciphers, as well as different modes of the same cipher produce wildly different results when applied to the same plaintext. The Preamble of The United States Constitution was encrypted using the aes-128 cipher in both cipher feedback, and block chaining modes. In addition, both blowfish and cast 5 ciphers were used in output feedback mode. The results are summarized below.

The plaintext:

```
We The People, in order to form a more perfect union,
establish justice ensure domestic tranquility.
Provide for the common defense, promote the general welfare,
and secure the blessings of liberty for ourselves and our posterity.
Do ordain and establish this constitution of The United States of America.
~
~
~
```

aes-128 cbc mode

```
[03/03/21]seed@VM:~/.../lab02$ openssl enc -aes-128-cbc -e -in Preamble.txt -out Ciphertext1.bin \-K  000102030405060708090aabbccddeeff \-iv 0
a0b0c0d0e0f010203040506070809
[03/03/21]seed@VM:~/.../lab02$ xxd Ciphertext1.bin
00000000: 34e3 3dc8 3d36 66eb d737 788e d062 5feb  4.=.=6f..7x..b_.
00000010: 298b 3f15 4192 e7d9 6f7f 8058 a6e3 9bc1  ).?.A...o..X....
00000020: e248 f69e bd24 ca43 2d1e 51b0 8584 3c09  .H...$.C-.Q...<.
00000030: d4e1 61c2 48fc 652d 9134 1984 1ac7 0e5b  ..a.H.e-.4.....[
00000040: fbf1 3d9e dbca 96c8 b685 31b4 02b5 49e1  ..=.......1...I.
00000050: 2872 2cfd edc7 b394 38e0 f31b 17da 7e74  (r,.....8.....~t
00000060: 9063 29a0 0cc5 b262 5c5b 315d 97bf fcad  .c)....b\[1]....
00000070: ad0f 66e7 e64e 0fa8 4adc 41a0 99ab 4a0d  ..f..N..J.A...J.
00000080: 3a83 b8eb 0411 d9af 786e 2a76 9594 f1f4  :.......xn*v....
00000090: cef1 d971 e502 6b00 ea21 a743 d996 a8be  ...q..k..!.C....
000000a0: a598 b0ab 966a cc47 1197 d949 1cd8 f171  .....j.G...I...q
000000b0: f211 3e51 063f 9696 e42a cf4f 1202 dbce  ..>Q.?...*.O....
000000c0: ad5a 1cc2 d2f3 ea15 04b8 5ca0 93d1 7be3  .Z........\...{.
000000d0: 72e0 56fb 9557 c27f cfb6 8b38 2292 c65d  r.V..W.....8"..]
000000e0: 7f43 cdf5 2393 f15d 7645 a737 55d2 675e  .C..#..]vE.7U.g^
000000f0: b9ff 68cb 83e1 f2a8 07f8 fcca 3860 9907  ..h.........8`..
00000100: f0c0 1a75 b3a2 e3ce 877b 4830 9b9b 18c3  ...u.....{H0....
00000110: b1b8 3668 9390 d10c 98bf ce03 1d0f a4bc  ..6h............
00000120: 6f3e 52a1 c33e 564b e1cf 56db a392 ddc0  o>R..>VK..V.....
00000130: 46f6 f818 fab4 a4b3 a387 fcd8 a3d3 6cc0  F.............l.
[03/03/21]seed@VM:~/.../lab02$
```

Note: The same key and initialization vector were used for most of the experiments across all tasks. Also note that the ciphertext is displayed in hexadecimal rather than the familiar ASCII format.

aes-128 cfb mode

```
[03/03/21]seed@VM:~/.../lab02$ openssl enc -aes-128-cfb -e -in Preamble.txt -out Ciphertext2.bin \-K  000102030405060708090aabbccddeeff \-iv 0
a0b0c0d0e0f010203040506070809
[03/03/21]seed@VM:~/.../lab02$ xxd Ciphertext2.bin
00000000: 92de a9a1 44e9 674f 3b76 3540 74e3 a683  ....D.gO;v5@t...
00000010: 20d6 25bc 0cd6 0bfd c560 7aa9 d940 0c62   .%.....`z..@.b
00000020: 9539 9cfd e285 e375 b91e 0132 36b8 715a  .9.....u...26.qZ
00000030: 1fb5 7d3f e905 5474 eb3b ed7d 529e 886f  ..}?..Tt.;.}R..o
00000040: 5678 f4bb 6160 b884 543c e1f2 f22d 39e7  Vx..a`..T<....-9.
00000050: 11b2 2af0 5414 7cd5 ecc7 e192 d4d7 6fa5  ..*.T.|.......o.
00000060: fb2e 42da ab41 8e0c d9a5 99da 7b04 515d  ..B..A......{.Q]
00000070: 7974 4874 4475 c7d6 3885 ce49 366f f151  ytHtDu..8..I6o.Q
00000080: 40c2 6e75 a92f 660e e9d3 df72 cbb0 164d  @.nu./f....r...M
00000090: f730 5657 7c2c c35b e626 be2f 4b8e 69d8  .0VW|,.[.&./K.i.
000000a0: 561f 67ac ab97 ceb7 b4a7 a9bb fec1 4a41  V.g...........JA
000000b0: 60dc d7b6 e649 1dd5 f6d2 08d3 26e5 dad6  `....I......&...
000000c0: 8e12 6f76 0275 3f2e 8f24 7918 7301 cacf  ..ov.u?..$y.s...
000000d0: 73ff 2e38 d7d5 4780 b679 4868 ddb9 87f9  s..8..G..yHh....
000000e0: e1a1 b96b 3980 0bcb ad48 93c1 8e00 eda6  ...k9....H......
000000f0: 1f41 186d bfc5 ae22 f918 fb70 b991 8f9c  .A.m..."...p....
00000100: bf9d e8f3 0c4e 1d1f ce58 917b a5a9 10b7  .....N...X.{....
00000110: 1ed1 4a32 7c10 79be 015f f50b 81b2 fcb6  ..J2|.y...._....
00000120: 2a07 596c 7a40 c55e 4340 3930 cfbf 0034  *.Ylz@.^C@90...4
00000130: 9cc5                                     ..
[03/03/21]seed@VM:~/.../lab02$
```

Blowfish ofb mode

```
[03/10/21]seed@VM:~/.../lab02$ openssl enc -bf-ofb -e -in Preamble.txt -out Ciphertext3.bin \-K 00010203040506070809aabbccddeeff \-iv 0a0b0c0
d0e0f010203040506070809
[03/10/21]seed@VM:~/.../lab02$ xxd Ciphertext3.bin
00000000: f41c df39 c858 5ee8 7d8d eee4 724f 7790  ...9.X^.}...rOw.
00000010: c4ac 00a1 d1eb 13a8 8df9 f6f3 cd69 63e2  .............ic.
00000020: 3d60 011c c1a4 e8bb 12af acd9 53e6 a184  =`..........S...
00000030: 3ea4 5d32 f4f7 5c24 6e96 8a0c 7584 6253  >.]2..\$n...u.bS
00000040: 9840 b7d7 b5bd 2fe0 4f45 66d3 4345 a220  .@..../.OEf.CE.
00000050: c15b 19cb 89e5 4241 cff6 5ae1 f74b 59d5  .[....BA..Z..KY.
00000060: 0f58 d5de b5ff 3832 4ee5 87c9 850b 5278  .X....82N.....Rx
00000070: 3609 4a46 505c 5e3f 7cc6 28cc dec5 c5f1  6.JFP\^?|.(.....
00000080: 16f1 1eb2 ef0f baa1 74e6 f201 012f 8075  ........t..../.u
00000090: 2dec e57b cb6d e125 1f1b b4db f850 13e9  -..{.m.%.....P..
000000a0: fbfa 89d6 3b4e 2cf6 a809 2e93 54e8 688c  ....;N,.....T.h.
000000b0: c0a7 3fdb d2aa a772 9072 ca82 d37e 425c  ..?....r.r...~B\
000000c0: 4297 9381 171f 7e77 9926 4c92 c2de 318c  B.....~w.&L...1.
000000d0: 20ef 65e4 72ac 64cc 2cf9 5cae 6b89 f435   .e.r.d.,.\.k..5
000000e0: 74a7 4a53 eeb5 fea6 13ce 19c7 c63d 0032  t.JS.........=.2
000000f0: 76f2 d597 a217 2d39 7a03 af0c 6b1f ec3b  v.....-9z...k..;
00000100: 91e0 9e4f fd05 40bf 4d34 a8b8 b949 3303  ...O..@.M4...I3.
00000110: 1c26 4277 bfdc 4ec7 3997 1a5b 5608 aab2  .&Bw..N.9..[V...
00000120: 9c9e 2aab a1c7 0a3d 606d b603 5117 bced  ..*....=`m..Q...
00000130: ab64                                     .d
[03/10/21]seed@VM:~/.../lab02$
```

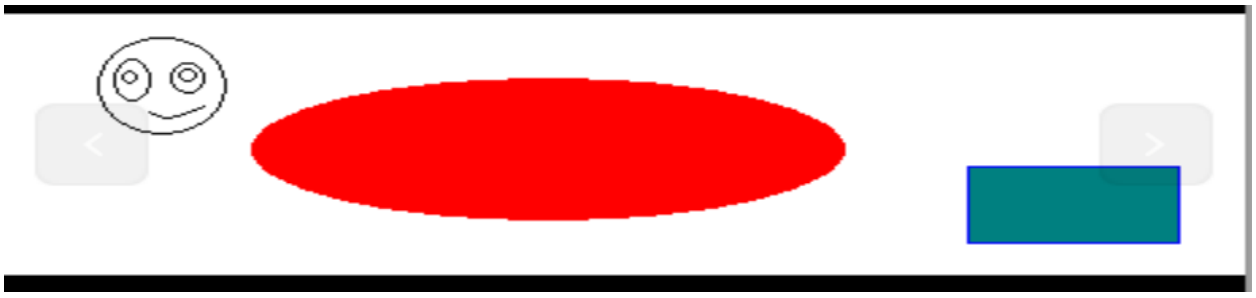
Cast 5 ofb mode

```
[03/10/21]seed@VM:~/.../lab02$ xxd Ciphertext4.bin
00000000: 131f a428 fa3d 201a 9d49 1e35 1070 50fd  ...(.= ..I.5.pP.
00000010: 1458 2318 acb0 0b17 2591 4495 c83f c9b9  .X#.....%.D..?..
00000020: 7cbb 58b5 7c78 d9ca 82d9 0d20 ffd2 720a  |.X.|x.....  ..r.
00000030: d44f 8a82 08f8 6530 f8de c9b6 f834 7390  .O....e0.....4s.
00000040: c6f5 5b4e b7d3 c23d 43bd 70ed cc83 4373  ..[N...=C.p...Cs
00000050: 33a9 4a66 a7ed f433 377c 65e0 df5b af6e  3.Jf...37|e..[.n
00000060: bf91 99bf 0224 1127 e299 827d f0ba 48d0  .....$.'...}..H.
00000070: b3cb 405b 3314 b26d 2653 5acd d329 8ded  ..@[3..m&SZ..)..
00000080: 95ff 1993 88d5 f7e2 6b61 65fb 0280 66a6  ........kae...f.
00000090: d577 b6c6 c1d8 6e7d b840 34a1 b0b5 e6f6  .w....n}.@4.....
000000a0: eee7 9b3b 966c 7a77 8869 0430 b77e 993b  ...;.lzw.i.0.~.;
000000b0: 1495 9a2f 1e4f a007 6387 27d4 09bb aa8d  .../.O..c.'.....
000000c0: e21e 5e33 b141 9ee5 48fb dfe2 496f b4f4  ..^3.A..H...Io..
000000d0: c11c 2bfd 0dca 5268 7c56 d962 6dd1 c48c  ..+...Rh|V.bm...
000000e0: c589 596c d7e4 08e7 c3fa 8471 4e7f d106  ..Yl.......qN...
000000f0: dce7 9483 8ac4 dcc0 41ca d057 932b bca4  ........A..W.+..
00000100: 1545 f1bd 3202 9bee 6016 2c66 7c3f fce0  .E..2...`.,f|?..
00000110: eabe 67f0 f65f 7158 187a b8c1 c79c e8c1  ..g.._qX.z......
00000120: e9b6 ad84 812b 847b 60ee 91b1 a1f8 55f0  .....+.{`.....U.
00000130: 7a02                                     z.
[03/10/21]seed@VM:~/.../lab02$
```

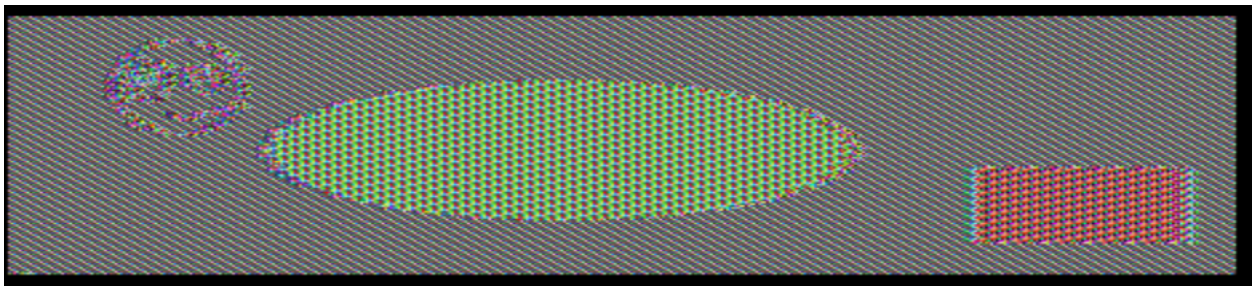
**Task 3: Encryption Mode - ECB VS. CBC**

Electronic code book and cipher block chaining modes produce very different results even when used with the same cipher. ECB is generally regarded as less secure as it encrypts identical blocks of plaintext to identical blocks of ciphertext. Conversely, CBC is usually regarded as more secure as the same plaintext will yield different ciphertext. The aes-128 cipher was applied to bitmap image files to observe this difference.

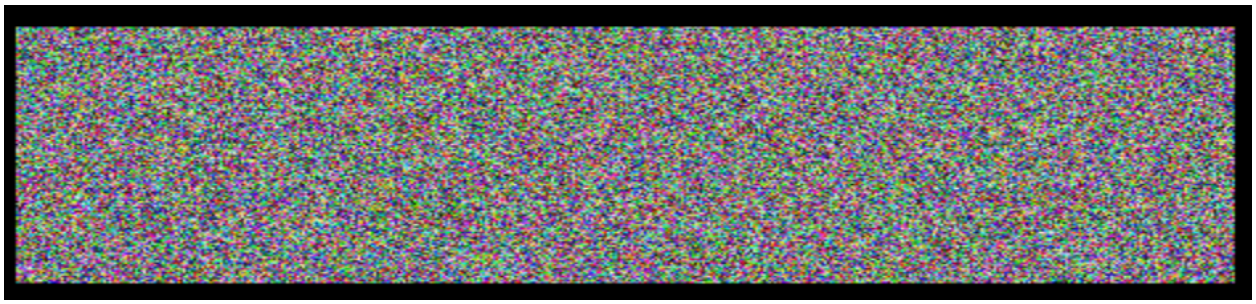
Original bmp file



ECB mode



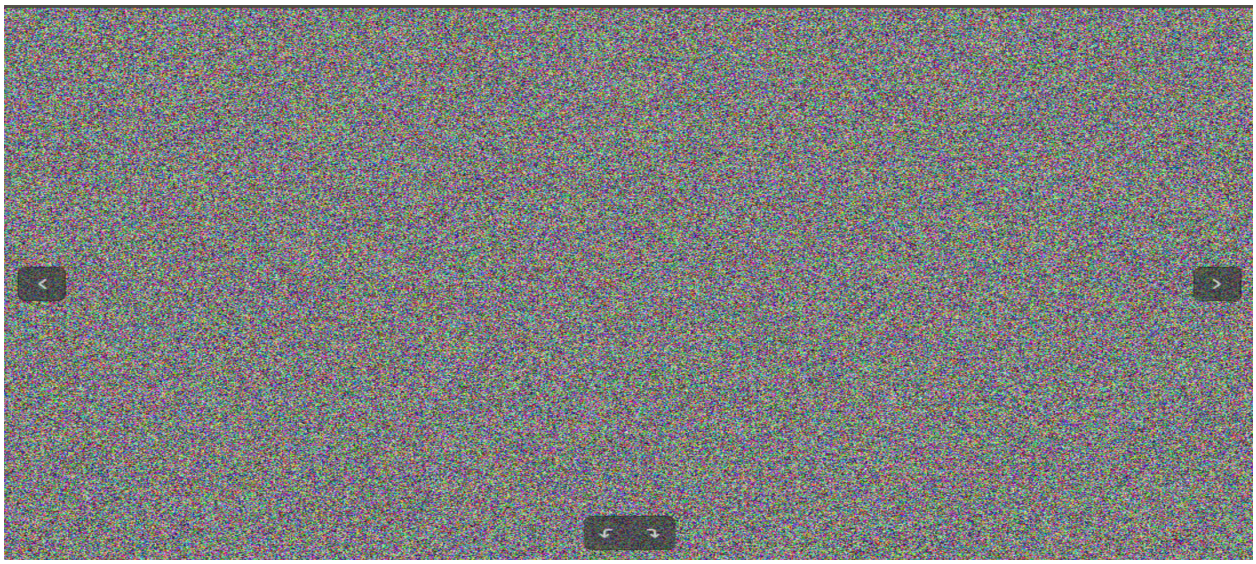
CBC mode



The same technique was applied to an image of the American Flag to verify the replicability of these findings.

ECB mode

CBC mode



As one can plainly see, images encrypted using ECB mode still largely resemble their original forms. Compare this with CBC mode, where the images bear no resemblance whatsoever to the originals. CBC mode is much more effective in hiding data than ECB, even with the strongest ciphers.

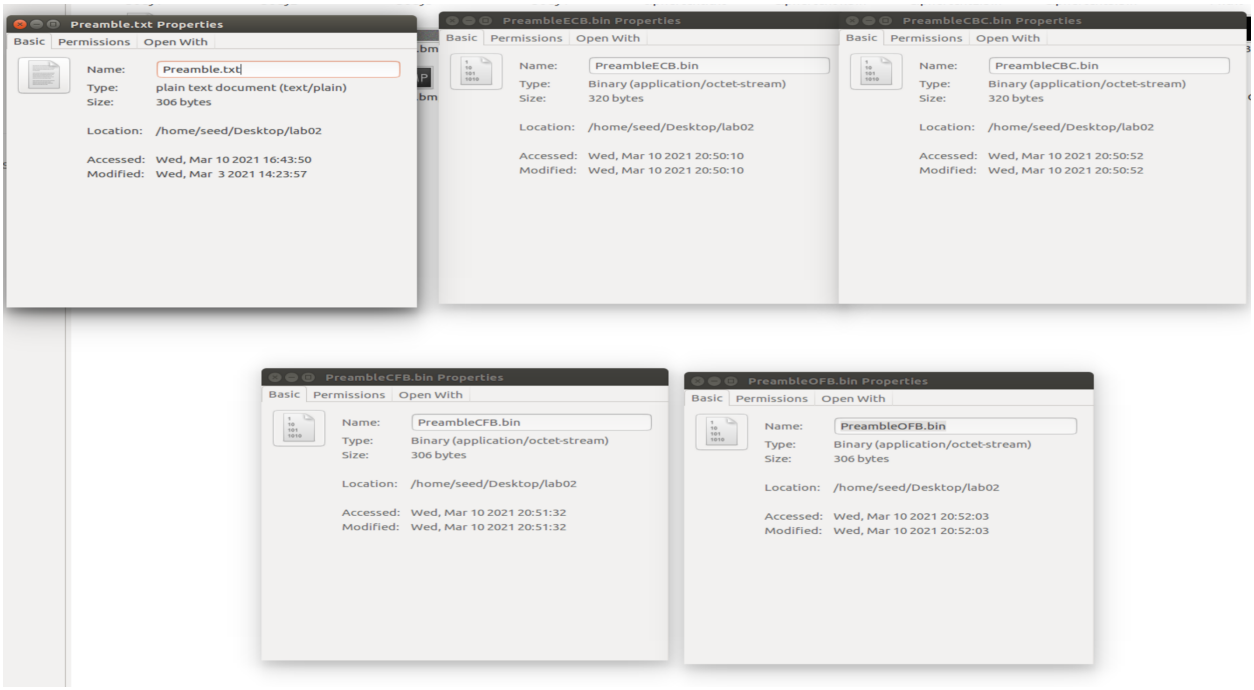
Commands used:

```
[03/03/21]seed@VM:~/.../lab02$ openssl enc -aes-128-ecb -in original.bmp -out new1.bmp
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
[03/03/21]seed@VM:~/.../lab02$ head -c 54 original.bmp > header1
[03/03/21]seed@VM:~/.../lab02$ tail -c +55 new1.bmp > body1
[03/03/21]seed@VM:~/.../lab02$ cat header1 body1 > final1.bmp
[03/03/21]seed@VM:~/.../lab02$
```

```
[03/03/21]seed@VM:~/.../lab02$ openssl enc -aes-128-cbc -e -in original.bmp -out new2.bmp
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[03/03/21]seed@VM:~/.../lab02$ head -c 54 original.bmp > header2
[03/03/21]seed@VM:~/.../lab02$ tail -c +55 new2.bmp > body2
[03/03/21]seed@VM:~/.../lab02$ cat header2 body2 > final2.bmp
[03/03/21]seed@VM:~/.../lab02$
```

**Task 4: Padding**

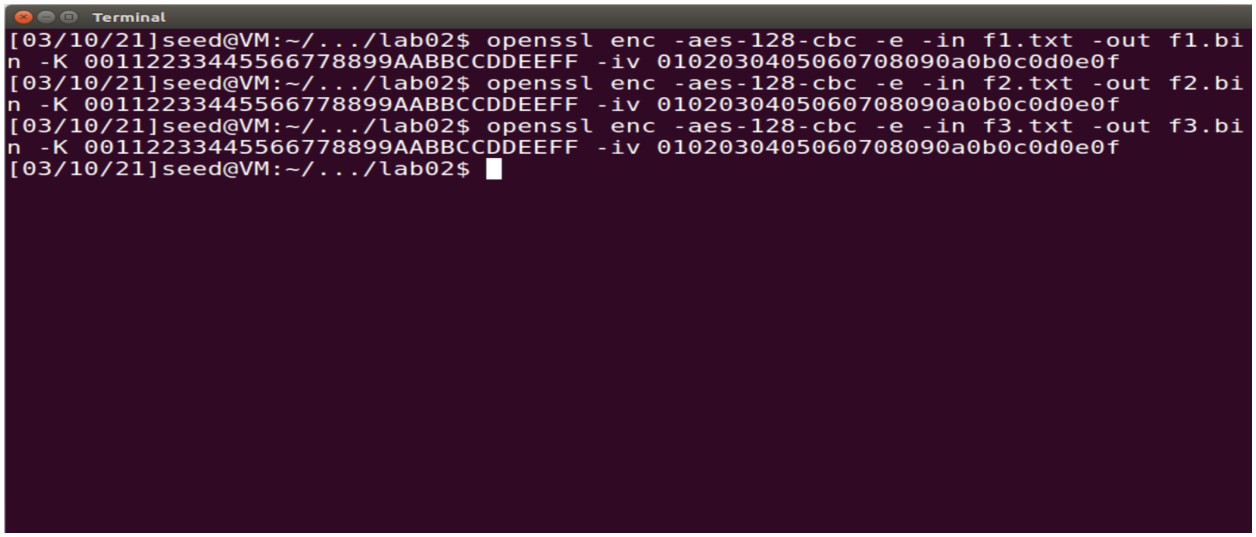
Padding is necessary for block ciphers when the size of the plaintext file or file to be encrypted is not a multiple of the block size. ECB and CBC modes use padding while CFB and OFB modes do not. This is because ECB and CBC modes operate on block ciphers, while CFB and OFB modes essentially convert block ciphers to stream ciphers that require no such padding. The Preamble plaintext file was encrypted with the aes-128 cipher in these four modes, and the sizes of the resultant files observed for evidence of padding.




The ciphertext files encrypted using ECB and CBC modes differ in size from the original, while those encrypted in CFB and OFB modes do not. Therefore ECB and CBC modes incorporate padding if necessary.

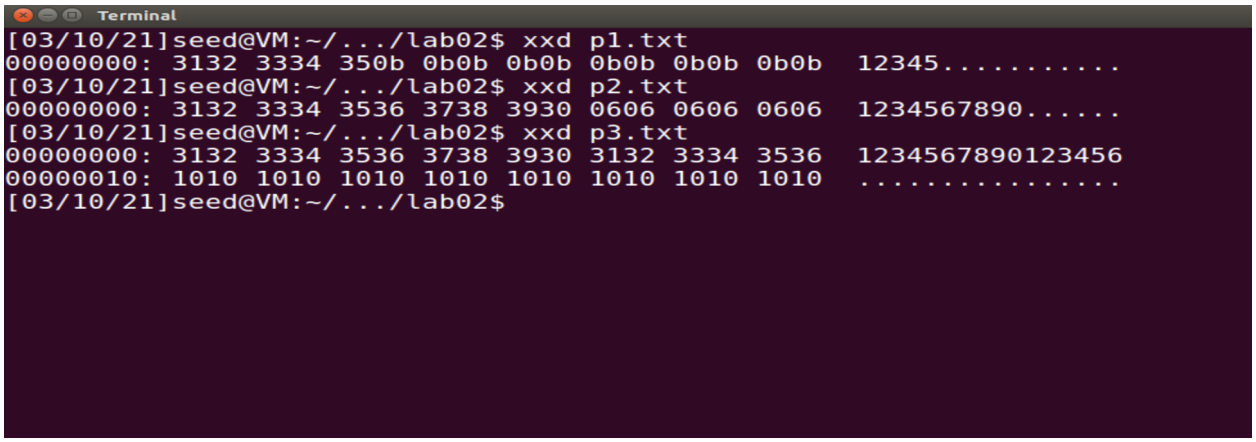
It is possible to directly observe the content that a ciphertext file is padded with by using the -nopad flag to decrypt and a simple hex viewing tool. This process is shown below for files of various sizes.

Encryption:




Decryption:

Resulting hex dumps:




sizeof(f1.txt) = 5 bytes, sizeof(p1.txt) = 16 bytes = 11 bytes of padding (0B * 11)

sizeof(f2.txt) = 10 bytes, sizeof(p2.txt) = 16 bytes = 6 bytes of padding (06 * 6)

sizeof(f3.txt) = 16 bytes, sizeof(p3.txt) = 32 bytes = 16 bytes of padding (10 * 16)

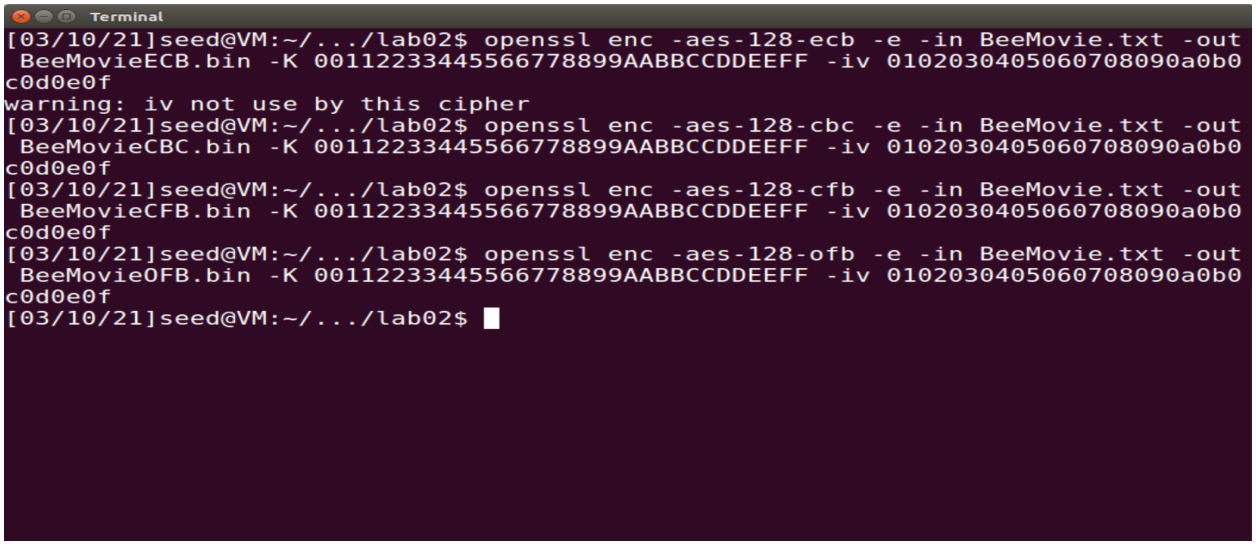
**Task 5: Error Propagation - Corrupted Ciphertext**

Modifying so much as a single bit of an encrypted file, intentionally or otherwise leads to varying degrees of error in decryption depending on the mode used to encrypt. It is hypothesized that decryption using ECB and CBC modes will produce the most error if an encrypted file is corrupted. It is assumed that only the bytes up until the corrupted one will decrypt correctly, with every subsequent byte being incorrect. This hypothesis was tested on an excerpt of *The Bee Movie* script using the aes-128 cipher in ECB, CBC, CFB, and OFB modes. The results are shown below.

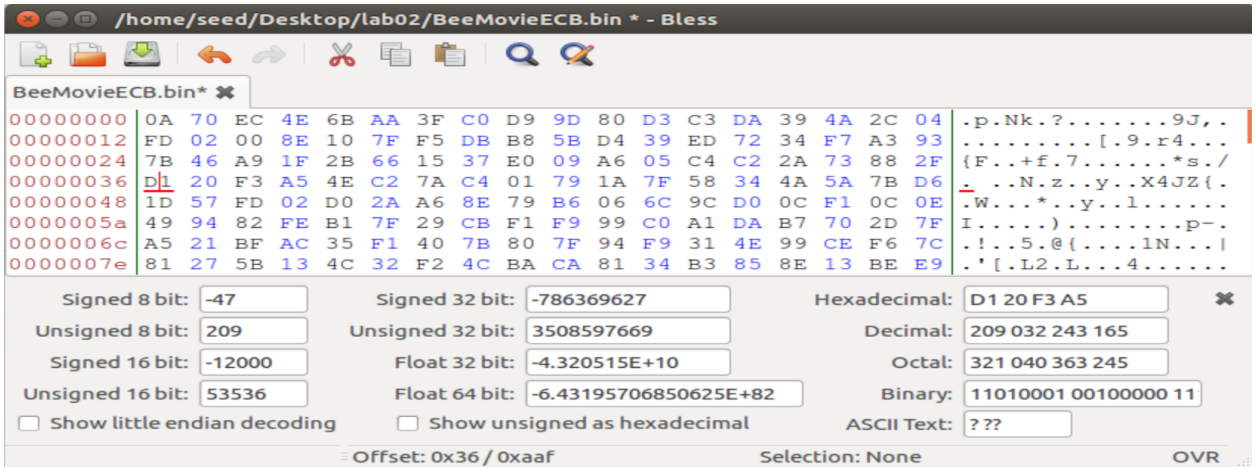
Original script:

```
ccording to all known laws
of aviation,

there is no way a bee
should be able to fly.

Its wings are too small to get
its fat little body off the ground.

The bee, of course, flies anyway

because bees don't care
what humans think is impossible.

Yellow, black. Yellow, black.
Yellow, black. Yellow, black.

Ooh, black and yellow!
Let's shake it up a little.

Barry! Breakfast is ready!

Ooming!

Hang on a second.
```

Encrypting the script in various modes:




Modification of bits using bless hex editor (only ECB mode shown for brevity)




(55th byte of each ciphertext file was modified by one hex value)

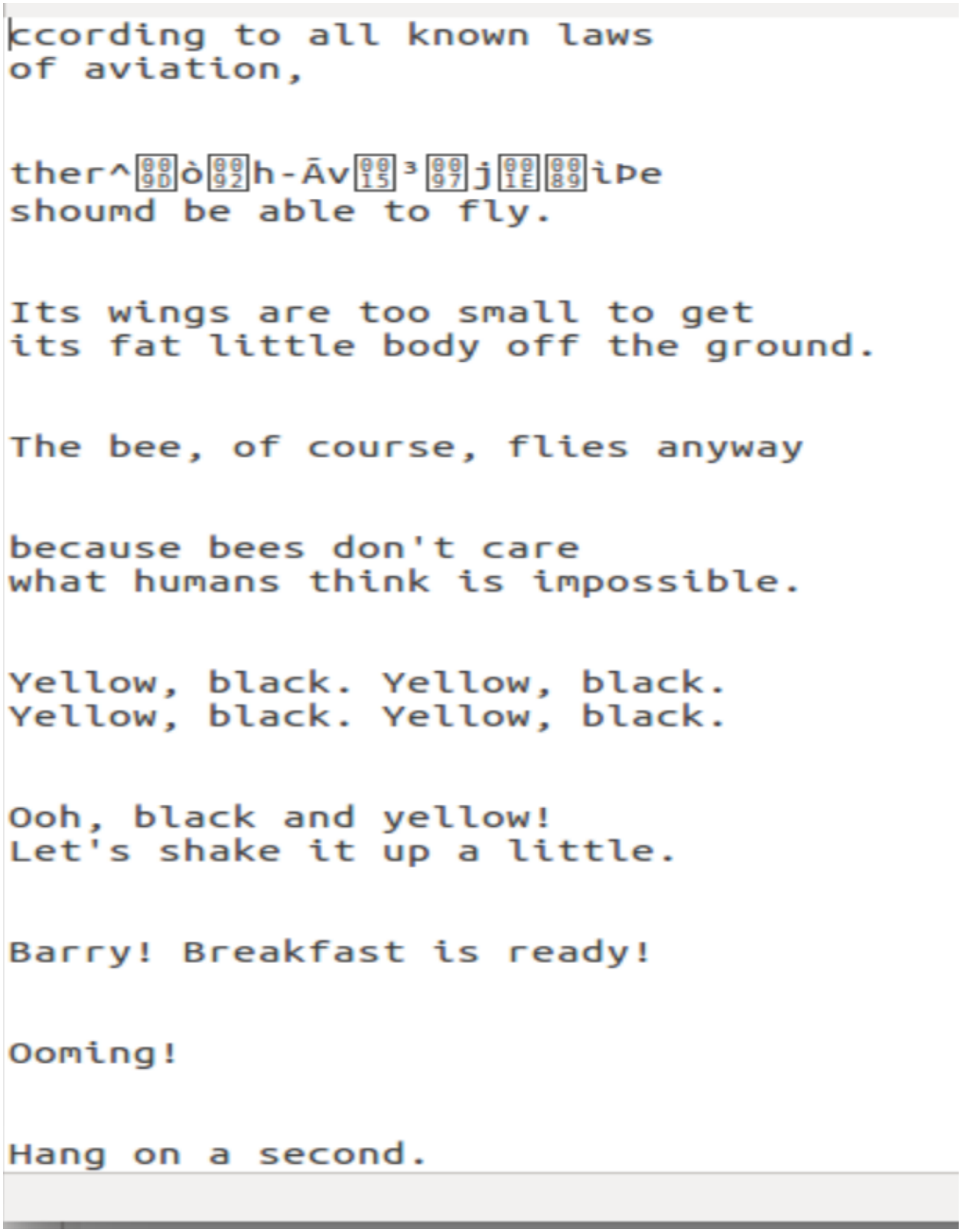
Decrypting the script after bit modification:

```
[03/10/21]seed@VM:~/.../lab02$ openssl enc -aes-128-ecb -d -in BeeMovieECB.bin -
out BeeMovieECB.txt -K 00112233445566778899AABBCCDDEEFF -iv 0102030405060708090a
0b0c0d0e0f
warning: iv not use by this cipher
[03/10/21]seed@VM:~/.../lab02$ openssl enc -aes-128-cbc -d -in BeeMovieCBC.bin -
out BeeMovieCBC.txt -K 00112233445566778899AABBCCDDEEFF -iv 0102030405060708090a
0b0c0d0e0f
[03/10/21]seed@VM:~/.../lab02$ openssl enc -aes-128-cfb -d -in BeeMovieCFB.bin -
out BeeMovieCFB.txt -K 00112233445566778899AABBCCDDEEFF -iv 0102030405060708090a
0b0c0d0e0f
[03/10/21]seed@VM:~/.../lab02$ openssl enc -aes-128-ofb -d -in BeeMovieOFB.bin -
out BeeMovieOFB.txt -K 00112233445566778899AABBCCDDEEFF -iv 0102030405060708090a
0b0c0d0e0f
[03/10/21]seed@VM:~/.../lab02$
```

ECB mode:

```
ccording to all known laws
of aviation,


therÖ6]ÏVv[][1c]šÚRæó[][15]V,we
should be able to fly.


Its wings are too small to get
its fat little body off the ground.


The bee, of course, flies anyway


because bees don't care
what humans think is impossible.


Yellow, black. Yellow, black.
Yellow, black. Yellow, black.


Ooh, black and yellow!
Let's shake it up a little.


Barry! Breakfast is ready!


Ooming!


Hang on a second.
```

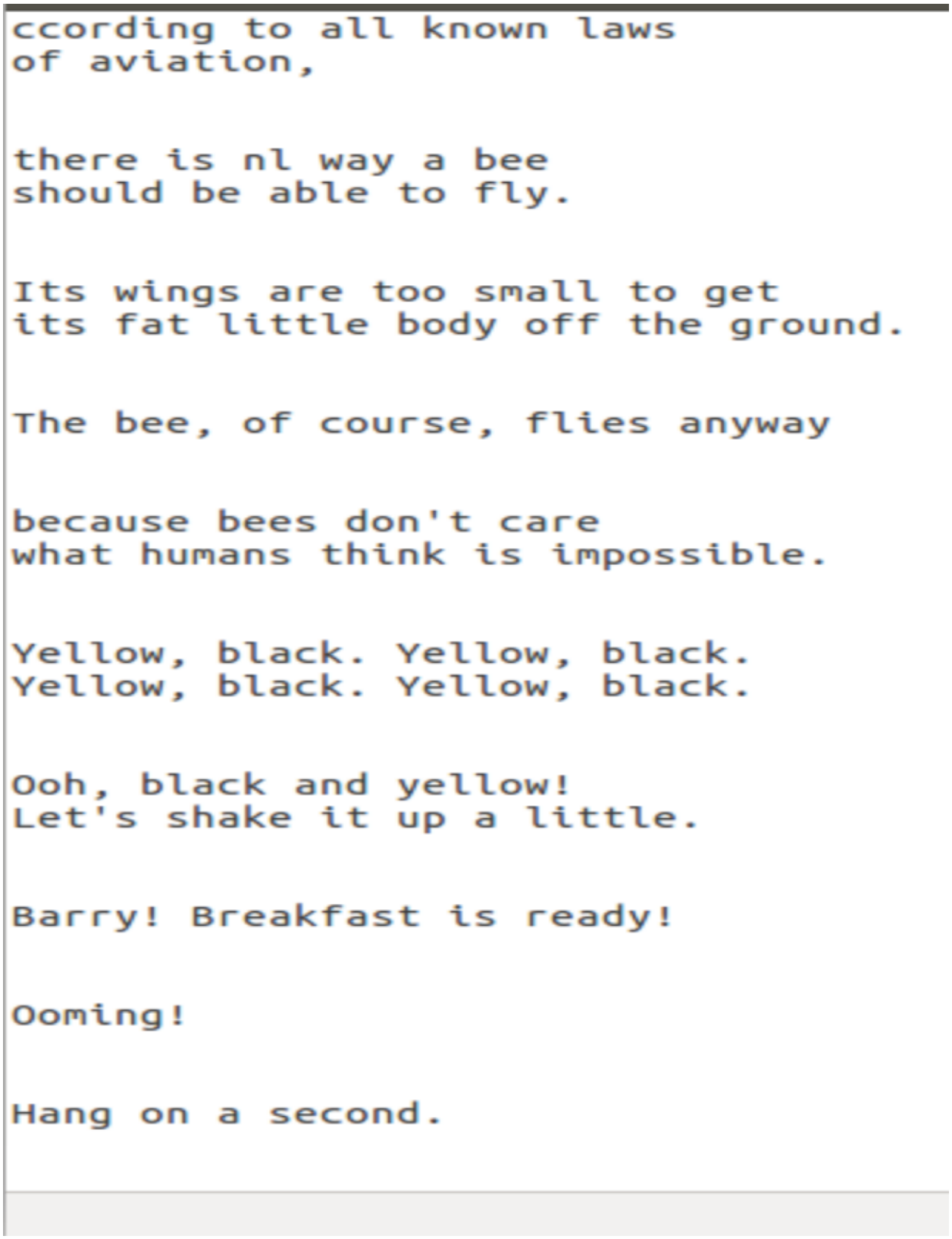
CBC mode:


ccording to all known laws
of aviation,
ther^ò h-Āv ³ j ìÞe
shoumd be able to fly.
Its wings are too small to get
its fat little body off the ground.
The bee, of course, flies anyway
because bees don't care
what humans think is impossible.
Yellow, black. Yellow, black.
Yellow, black. Yellow, black.
Ooh, black and yellow!
Let's shake it up a little.
Barry! Breakfast is ready!
Ooming!
Hang on a second.

CFB mode:

```
ccording to all known laws
of aviation,

there is nl way a beÂ9[00 8D]e±k[00 18]wô[00 18]£Q©[00 08]I[00 84] to fly.

Its wings are too small to get
its fat little body off the ground.

The bee, of course, flies anyway

because bees don't care
what humans think is impossible.

Yellow, black. Yellow, black.
Yellow, black. Yellow, black.

Ooh, black and yellow!
Let's shake it up a little.

Barry! Breakfast is ready!

Ooming!

Hang on a second.
```

OFB mode:


ccording to all known laws
of aviation,
there is nl way a bee
should be able to fly.
Its wings are too small to get
its fat little body off the ground.
The bee, of course, flies anyway
because bees don't care
what humans think is impossible.
Yellow, black. Yellow, black.
Yellow, black. Yellow, black.
Ooh, black and yellow!
Let's shake it up a little.
Barry! Breakfast is ready!
Ooming!
Hang on a second.

The hypothesis is proven to be false. CFB mode produced the most error with 3 lines affected. This is because CFB encrypts the previous ciphertext block and XORs it with the current plaintext block to create the final ciphertext, leading to a ripple effect of error. No such corruption after the 55th byte was observed for ECB and CBC modes. It is unclear whether or not the large amount of whitespace in the original text affected the results in any meaningful way.