

## Experiment Number: 3

UID : 2019140050

Batch: C

Name: Tanvi Sunil Pen

Branch: TE IT

**AIM:** To get familiar with the concepts in secret-key encryption also gain first-hand experience on encryption algorithms, encryption modes, paddings, and initial vector (IV). After this lab should be able to use tools and write programs to encrypt/decrypt messages.

### **PROBLEM STATEMENT:**

**Task 1:** Encryption using different ciphers and modes.

**Task 2:** Encryption Mode – ECB vs. CBC (Image)

**Task 3:** Encryption Mode – Corrupted Cipher Text

**Task 4:** Padding

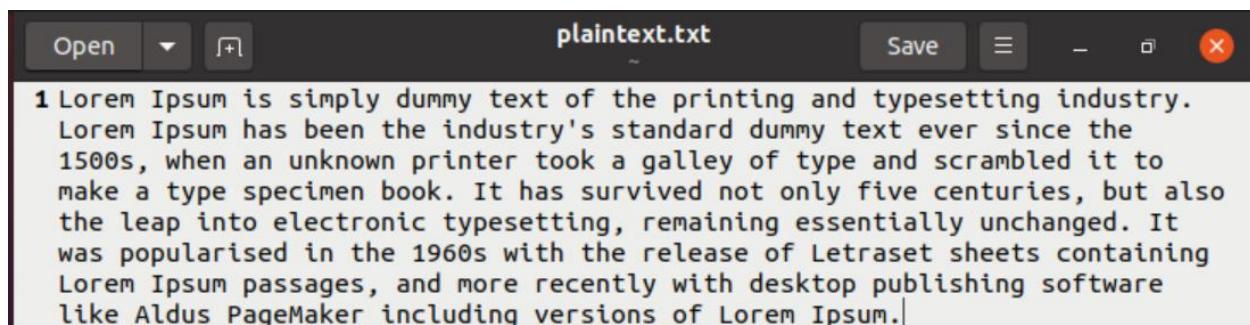
**Task 5:** Programming using the Crypto Library

## 1) Task 1 :

### a) Using the cipher type - aes

#### a.1) -aes-256-cbc

plaintext.txt



```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It
was popularised in the 1960s with the release of Letraset sheets containing
Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.|
```

After encrypting this file and storing the output in encryp.txt

```
tanvi@tanvi:~$ openssl aes-256-cbc -salt -a -e -p -in plaintext.txt -out encryp
.txt
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=BBB9C6EB28D61AAD
key=610FAC2AAC0AFE48B43722F1AA68D6B23A6821082E7DBBAE2BF813A38AF01E5
iv =C719772CB3FBC6C67394076737A064F2
tanvi@tanvi:~$
```

encryp.txt (after encryption)

The screenshot shows a terminal window with the title bar 'encryp.txt'. The window contains 13 lines of encrypted text, starting with '1 U2FsdGVkX1/h...' and ending with '13 mEKkztet4kB82wqFTghN/w=='. The text is in a monospaced font.

```
1 U2FsdGVkX1/h0DStajP0hLUYTzWE9kXJaLWkudo4HvABcBdc1LYTucGUN3RbEI20
2 /TaRi3MzbMppwSua80PXzBRUbdfMhNhQ3uvhRwReRW6eJ4u2Rj6mTWqho8zE7r
3 wthyibnVAInQd4FYkVQDahnRF7Zewek/Qtfh8l2ySapxSjkztjDoFL6pmNjYl/SU
4 cP1H9zAlMRzcMxxBLp+1A0dgttFQlEpMUMeWq/6CSvUQk4rSYkdf4HQQEPONS/iL
5 eA/9C7V5FUoRZPhwKiLp/IMig9Mb+QJ6FR+cN/W4Bl9hbr0sbaMprb+Ll4HJzQTF
6 CYu0grpblhD89y+oSSNaTUpxHUWCk44p5dMwcCJRjSWRv0nbEUdqDGPDpJXbzQZ
7 00T5a7F52AUmpYKivyscPJbqdTNyaXv37MqZRCa3VV0ZBfGeGQPTTf7jpi9IHRF
8 8Vb8KZjnS03BUdXx1eiKx/PsUvbFH1FMe2e3Zq7KoCnGtw/1DjlfpA3sWwli9bSr
9 gc0lZnT9D/DKkqBf4F1t50pXXdU6ghsfdaVvd8NhbAZN8XRG4aPPvYN6ylwgCS1
10 ZUyS8/2zSP86qNX9TcqGlgbUeoDBmh3u9ptC+H9LBkLzKNut6oXtuqyU+PCftxDc
11 uJfx5Chg0dtJ+TgWQTkzjcOKU+Rj5MSyzE4xi6XRthLz80Rf36HAnsK+7kq1YTFM
12 e3M35kDCdpUeE9G3oDDu0H8TuXHZ+2Cn3Ebq+XIKNiWA20bWvDeiwSXYzQ0Q/NT
13 mEKkztet4kB82wqFTghN/w==
```

After decrypting the file 'encryp.txt' and storing the output in decryp.txt

```
tanvi@tanvi:~$ openssl aes-256-cbc -salt -a -d -p -in encryp.txt -out decryp.txt
enter aes-256-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=BBB9C6EB28D61AAD
key=610FAC2AAC0AFE48B43722F1AA68D6B23A6821082E7DBBAE2BF813A38AF01E5
iv =C719772CB3FBC6C67394076737A064F2
tanvi@tanvi:~$
```

The screenshot shows a terminal window with the title bar 'decryp.txt'. The window contains a single line of decrypted text: '1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.>Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.' The text is in a monospaced font.

```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It
was popularised in the 1960s with the release of Letraset sheets containing
Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.
```

## a.2) -aes-256-cbf

plaintext.txt

```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  Lorem Ipsum has been the industry's standard dummy text ever since the
  1500s, when an unknown printer took a galley of type and scrambled it to
  make a type specimen book. It has survived not only five centuries, but also
  the leap into electronic typesetting, remaining essentially unchanged. It
  was popularised in the 1960s with the release of Letraset sheets containing
  Lorem Ipsum passages, and more recently with desktop publishing software
  like Aldus PageMaker including versions of Lorem Ipsum.
```

After encrypting this file and storing the output in encryp.txt

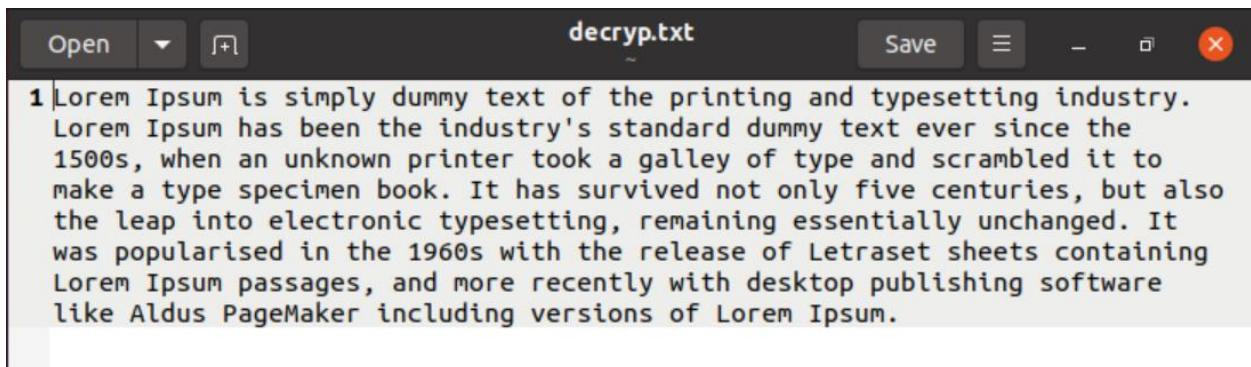
```
tanvi@tanvi:~$ openssl aes-256-cfb -salt -a -e -p -in plaintext.txt -out encryp
.txt
enter aes-256-cfb encryption password:
Verifying - enter aes-256-cfb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=EB9CC47ACEF506E6
key=5D57B73248448759756BBF6FBC7A02FDB15EDEDEBB0F7CABB8AA053B9136B6A8
iv =08CD26D38318132363EA78F15283EC1E
tanvi@tanvi:~$
```

encryp.txt (after encryption)

```
1 J2FsdGVkX1+351SA97F/62vE3/0xWGPrJvdjtWjfxY8lB4rHHJ3V7ivWLX3XrNhe
2 p04zVjKX7w/fa7Lw8Fx8jRUCv6h0P9Irmho1DTMHxMtvdBftF8qENISKKaYdQmCS
3 otUX7tkj7DMWwQWB8aIMelJE/1hiVc6nzXuCgdqIl1bI+xBfxJnStb30edm8zLb7
4 Gm80RnQnnNzHkQCThTEtw/XsQcWveko1SVASA+edpsuqWLjY+jY+VrtvsLG922NWmM
5 udFLXMUnPc1nNT0gEcjM0yMh8Vt5fBp9WT7EHqtWY+oWBSNEYGgk0h0H0ObIFNPO
6 xX8M7NFUUsYLMANCS05BjGBGLDuWn0mLegTFrMXYdpMHDEI3tAod8BMP3ukHySEj
7 FyNZBAwk9WGc+G2MJ9mZD2vIcHsoUsvsqWHENDZqN9VP+omx6bRpU4GFurXNmh+8
8 Kxk21EktlCx9fRmGkT1bGqK50hUMXz1m9keMppaXppHUarmqSozo01iPhAQcUy3Q
9 NS7WGNa0LcRBBydplzGrqvyRsGHwdaj0NkYKmVwzxV7c3m2cePfCa5VrDcCgNpEll
10 628me1n3RbK1y0lf5Ui+sw0jqgYRZcnbroP/o80UepIdpk7KbnM7Lf5f8pRqKe2k
11 YYqpVRKyEHBDc3qV/dk0zo7k7SeXrJHaT5tmFtbuPja0ei8ioaUuyr/w2ljUxG2P
12 jpFqQADiQvxGzs4KVE2WyEiZpQsqswSbkRUF5HoH8MM01GUmpRt8P3lWXtzwQin4
13 yp6qr1dIFD0MbRTeq+3Z
```

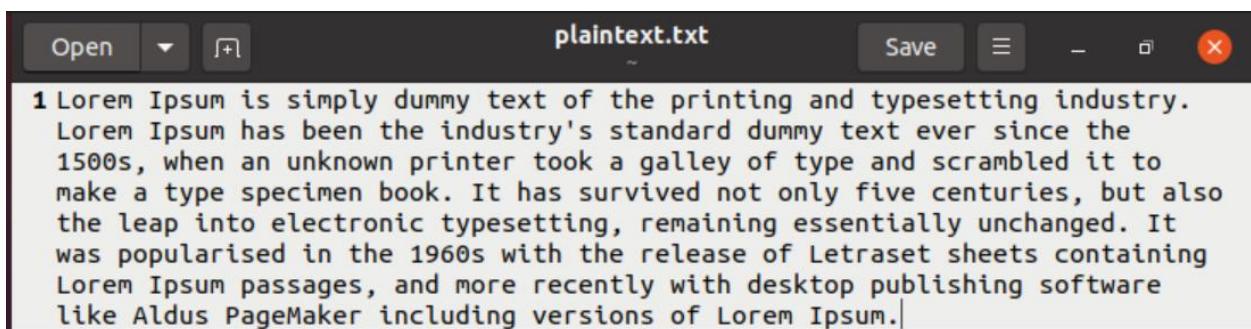
After decrypting the file ‘encryp.txt’ and storing the output in decryp.txt

```
tanvi@tanvi:~$ openssl aes-256-cfb -salt -a -d -p -in encryp.txt -out decryp.txt
enter aes-256-cfb decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=EB9CC47ACEF506E6
key=5D57B73248448759756BBF6FBC7A02FDB15EDEDEBB0F7CABB8AA053B9136B6A8
iv =08CD26D38318132363EA78F15283EC1E
tanvi@tanvi:~$
```



### a.3) -aes-128-cbc

plaintext.txt



After encrypting this file and storing the output in encryp.txt

```
tanvi@tanvi:~$ openssl aes-128-cbc -salt -a -e -p -in plaintext.txt -out encryp.txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=B589940C2A66C1CA
key=341FBFB2DDECCE9481AEDD27A8F5FC8F
iv =AE5A6BB43892C601CB876D557276B789
```

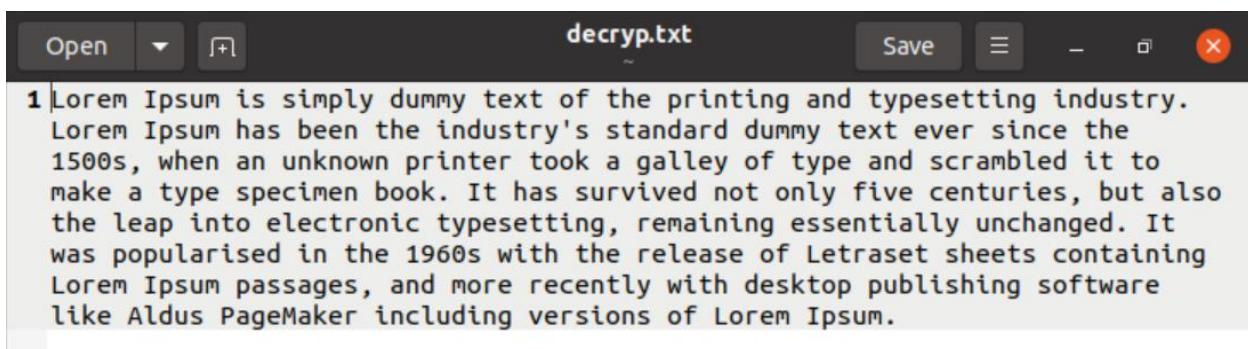
encryp.txt (after encryption)



A screenshot of a terminal window titled "encryp.txt". The window contains 13 lines of highly encrypted binary data represented as ASCII art. The data consists of various characters including numbers, letters, and symbols, such as '1', 'J', 'Y', 'A', 'C', 'G', 'B', 'M', 'P', 'E', 'S', 'D', 'F', 'L', 'R', 'T', 'W', 'N', 'Q', 'Z', 'H', 'V', 'X', 'D', 'O', 'S', 'K', 'I', 'P', 'T', 'B', 'H', 'C', 'B', 'D', 'C', 'E', 'J', 'K', 'Y', 'F', 'G', 'M', 'V', 'Q', 'I', 'e', 's', 'V', 't', 'S', 'V', '3', 'L', 'j', 'C', '6', 't', '+', 'T', 'n', 'X', 'Z', 't', 'K', 'd', 'M', 'G', 'o', 'M', 'i', '5', 'h', 'Z', 'P', 'R', 'e', 'v', 'C', 'l', 't', 'L', 'Y', 'Z', 'z', 'A', 'N', 'c', 'Z', '2', 'S', 'h', 'R', '+', '9', 'I', 'N', 'T', 'z', '0', 'z', '4', 'Y', '9', 't', '1', 'A', '4', 'P', 'a', 'U', 'T', 'z', 'e', 'y', 't', 'y', 'F', '1', 'W', 'J', 'W', '4', 'g', 'p', 'q', 't', 'o', 'f', 'U', 'V', 'm', 'h', 'M', 'a', 'g', 'y', 'g', 'M', 'o', 'w', '0', 'X', '1', 'S', 't', 'w', 'p', '0', '7', 'C', 'R', 'l', 'u', '0', 'l', 'd', 'j', 'D', 's', '9', '9', 'v', 'K', 'I', 'k', 'A', 'u', 's', 'q', 'L', 'c', 'J', 'v', 'A', 'S', 'A', 'X', 'D', 'F', 'Q', 'R', '3', '1', 'S', 'a', 'V', 'v', 'P', '9', 's', 'I', 'p', 'f', 'D', 'O', 'U', 'R', 'F', 'g', '8', 'A', 'z', 'c', '8', 'g', 'h', 'I', 'Y', 'H', '2', 'n', 'F', 'f', '1', 'M', 'Q', 'r', 'P', '+', 'B', 'u', 'J', 'L', 'G', 'x', 'u', 'k', 'V', '0', 'N', 'q', 'm', 'C', 'Q', 'q', 'L', 'M', 'p', 'k', 'M', 'h', 'Y', 'n', 'f', '6', '1', '3', '/', 'd', 'y', 'L', 'Y', 'g', 'e', 'Z', 'F', 'W', 'd', 'C', 'C', 'A', 'w', '2', 'I', '+', 'r', 'm', 'd', 'g', '=='. The terminal prompt is "tanvi@tanvi:~\$".

After decrypting the file 'encryp.txt' and storing the output in decryp.txt

```
tanvi@tanvi:~$ openssl aes-128-cbc -salt -a -d -p -in encryp.txt -out decryp.txt
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=B589940C2A66C1CA
key=341FBFB2DDECCE9481AEDD27A8F5FC8F
iv =AE5A6BB43892C601CB876D557276B789
```



A screenshot of a terminal window titled "decryp.txt". The window contains a single paragraph of text in a standard font. The text is a well-known Lorem Ipsum placeholder, describing its history and use in printing and typesetting. The terminal prompt is "tanvi@tanvi:~\$".

1 Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

## b) Using the cipher type - des

### b.1) -des

plaintext.txt

```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  Lorem Ipsum has been the industry's standard dummy text ever since the
  1500s, when an unknown printer took a galley of type and scrambled it to
  make a type specimen book. It has survived not only five centuries, but also
  the leap into electronic typesetting, remaining essentially unchanged. It
  was popularised in the 1960s with the release of Letraset sheets containing
  Lorem Ipsum passages, and more recently with desktop publishing software
  like Aldus PageMaker including versions of Lorem Ipsum.|
```

After encrypting this file and storing the output in encryp.txt

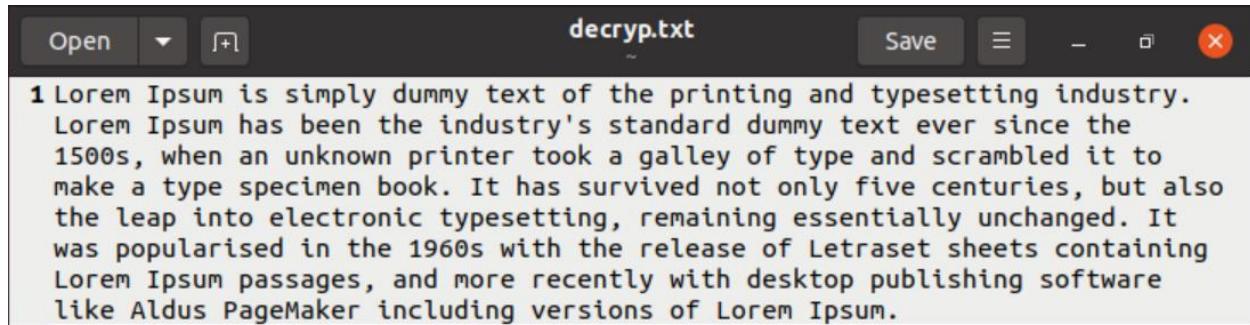
```
tanvi@tanvi:~$ openssl des -salt -a -e -p -in plaintext.txt -out encryp.txt
enter des-cbc encryption password:
Verifying - enter des-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=36FBAD8CAC5BD50B
key=ABEE42D8273ED83A
iv =6210292D95AF83C2
```

encryp.txt (after encryption)

```
1 U2FsdGVkX1+IeDoQQi9dDtVxg4CkjrwJTTtoFlo89QQBnlf0hf+sVp0iTrm0Z0I2u
  2 MXJ483W/xkZWlgNtA2HfobjWr6q1KgY6FzU7i9X1mhXZhvKABW2ofG0lcJ0os9NN
  3 UJJJE/Ua/9KNpK6+Q5zv/wEAglqKdz3Mpj3lJyf7kUEEdVG3eq17RTPjp+B0WND8+
  4 nBX8CSlzq0/mCRaA30mgqFtWxyak8sn28ComVax0zw/8bqemz7Wjl5xK5g9YvEU/
  5 /KRWIr5s34KsT222LcU8yHp9h9Ly09h4zkwwG6n5pUsaOPMTdt/uq/QaZR4CBE3k
  6 8EEEn08onmPyRhhfZLi22ntVnm42cPaQv8Cvzltf7nIJEf/4lueXsGCEIkeu0dq+c
  7 2+k07tR2LL1RJEExDtipHewR7Ecjeqr2sFo440d/XMoDVVMISzub+470D7ZLCiYei
  8 A4ZJC4ULP13tb4/Dfb8YQIPwgBwqsJjZhh+yg3QR9lZI7zVFbxq19GrjjkFpspMj
  9 vpZ7oSzXPKh5HkshDAIfh49kK3MDtZtJIVs6rt/57MxVS26ET9/o0Qx2itnR8R63
  10 ut4nqsdxmZQy0mE5URqiJ/RrrH2Wf70aToVEXwEM1T79V9h5B5Ar9n7HqG0+FfDL
  11 ux0tmY1cMPMdxeFxMfYQyzzqwjQjKcW99vUS7ZhekXPqWCAYmnnyAng5X04r0li
  12 2uv4zV11jTtJdfEZ3lntuCsHFuTQ29XLrXR/gUIUjW9K6/VUPsVBzx1qX11f2rNO
  13 B0P8DQa3QLy7+asGh1pyUA==
```

After decrypting the file ‘encryp.txt’ and storing the output in decryp.txt

```
tanvi@tanvi:~$ openssl des -salt -a -d -p -in encryp.txt -out decryp.txt
enter des-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=36FBAD8CAC5BD50B
key=ABEE42D8273ED83A
iv =6210292D95AF83C2
```

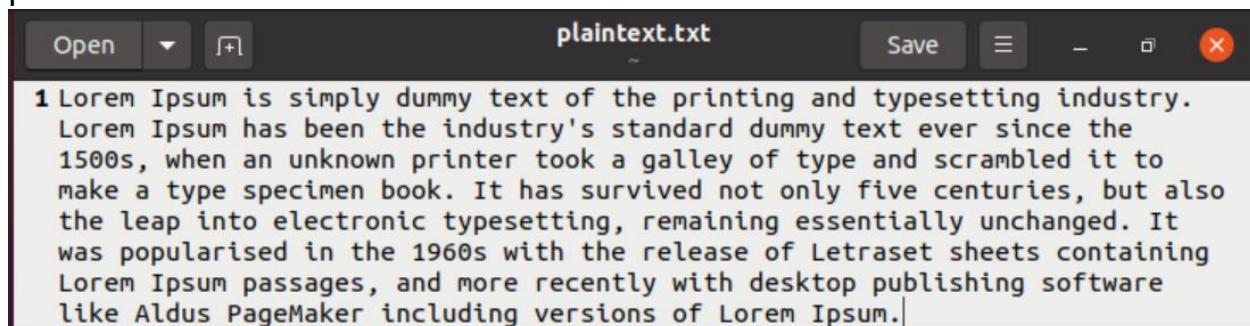


A screenshot of a terminal window titled "decryp.txt". The window contains the following text:

```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It
was popularised in the 1960s with the release of Letraset sheets containing
Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.
```

## b.2) -des-cbc

plaintext.txt



A screenshot of a terminal window titled "plaintext.txt". The window contains the following text:

```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It
was popularised in the 1960s with the release of Letraset sheets containing
Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.
```

After encrypting this file and storing the output in encryp.txt

```
tanvi@tanvi:~$ openssl des-cbc -salt -a -e -p -in plaintext.txt -out encryp.txt
enter des-cbc encryption password:
Verifying - enter des-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=F4D2AFE3F4093037
key=2B7F218EF1CFA4B6
iv =DAF3980EE8966D40
```

encryp.txt (after encryption)

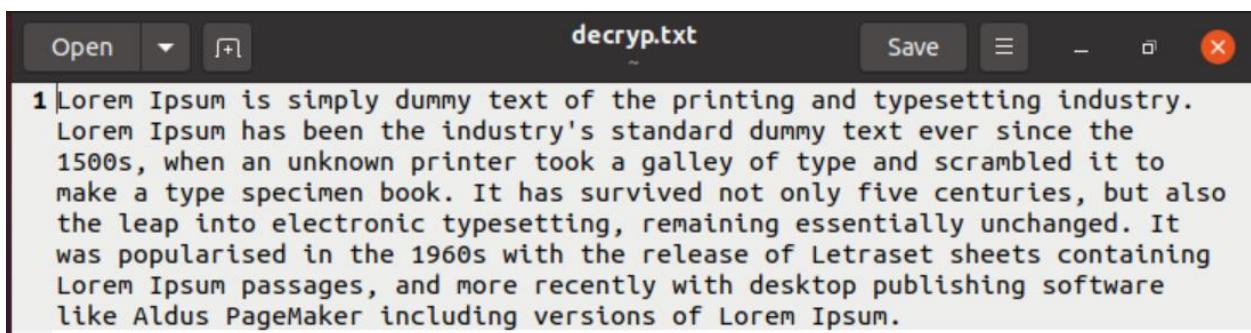


A screenshot of a terminal window titled "encryp.txt". The window contains a large amount of highly encrypted binary data represented as a series of characters. The data starts with "1 J2FsdGVkX1+AqJa/9y9Fjk5xH2K9dc+2AMfQjjc3y9Y4fu2W1MecESGYdl/crb1S" and continues through several lines of similarly encoded text.

```
1 J2FsdGVkX1+AqJa/9y9Fjk5xH2K9dc+2AMfQjjc3y9Y4fu2W1MecESGYdl/crb1S
2 C02hW9YMexi3fPkG5Ncc13HAEzezrTgFGzxeoPebnHpXhnqSVIuVsLHhn05h3v9a
3 yYKbiYp7fIz7aIc/nIgIkrrQ9SP6QTcZMD+VkjNx4xwhhMZv4qX8jpCsZxKUaX
4 AniIvRJHWbC8wUiNJgZYPB1Lxm3Xojw82Cc98HZCgsIZCgjGHILcY8EU7UNWj7kaK
5 b9xr64mImB6tt4GYT0MIZLr0Y0Dfa4ZLLV8JTzqQBRYR+fXa3/ugGyvVyhVvPNpC3
6 3uBVep1nsP0SeaeHTY84SDzjreNKMA7nuG0wOHo/CIPirhmP/3wIUjCfnuq5ZipF
7 moN6lS8K6V8llsFB/Qk0w5u90w4bSOPeZDwauU+hNf74Ej/lz1XgZJxtooVW+Pyo
8 DIgphaMf3Y+Pg8V0j4+7LoEeDTngDLi/9KXkqlU4VTy6hx5sbXff6pmikxjvBIQy
9 dVXqlGb1029EFHVJYW28pKjHrbkMzTo/ChRkMTXKQI55Ta8Wn9ZNDaEBisBoTpeS
10 L2F+lNV/FaYrFioJckoKSmoJ3ZI7AcCr1balVXxF/A5gqW9BdGIFxbXgnid0f6d
11 ppa0L0/cNb85y6SLJ6ghxgiCW9UPpQ01G+fc0bOKPWsxqp31u+3iqg31XLL5/woF
12 g50wi62bu8zg+2Aw1NSKwGWw1vMPybOUTBHeXFYGz29cK/R1+gmIsNH3ut02tWwY
13 Qq86pq+WFSYfD0heVtaW7Q==
```

After decrypting the file 'encryp.txt' and storing the output in decryp.txt

```
tanvi@tanvi:~$ openssl des-cbc -salt -a -d -p -in encryp.txt -out decryp.txt
enter des-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=F4D2AFE3F4093037
key=2B7F218EF1CFA4B6
iv =DAF3980EE8966D40
```



A screenshot of a terminal window titled "decryp.txt". The window displays a single paragraph of text in a standard font. The text is a well-known Lorem Ipsum passage, describing the history and evolution of dummy text in printing and typesetting.

```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  Lorem Ipsum has been the industry's standard dummy text ever since the
  1500s, when an unknown printer took a galley of type and scrambled it to
  make a type specimen book. It has survived not only five centuries, but also
  the leap into electronic typesetting, remaining essentially unchanged. It
  was popularised in the 1960s with the release of Letraset sheets containing
  Lorem Ipsum passages, and more recently with desktop publishing software
  like Aldus PageMaker including versions of Lorem Ipsum.
```

### b.3) -des-cfb

plaintext.txt

```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  Lorem Ipsum has been the industry's standard dummy text ever since the
  1500s, when an unknown printer took a galley of type and scrambled it to
  make a type specimen book. It has survived not only five centuries, but also
  the leap into electronic typesetting, remaining essentially unchanged. It
  was popularised in the 1960s with the release of Letraset sheets containing
  Lorem Ipsum passages, and more recently with desktop publishing software
  like Aldus PageMaker including versions of Lorem Ipsum.|
```

After encrypting this file and storing the output in encryp.txt

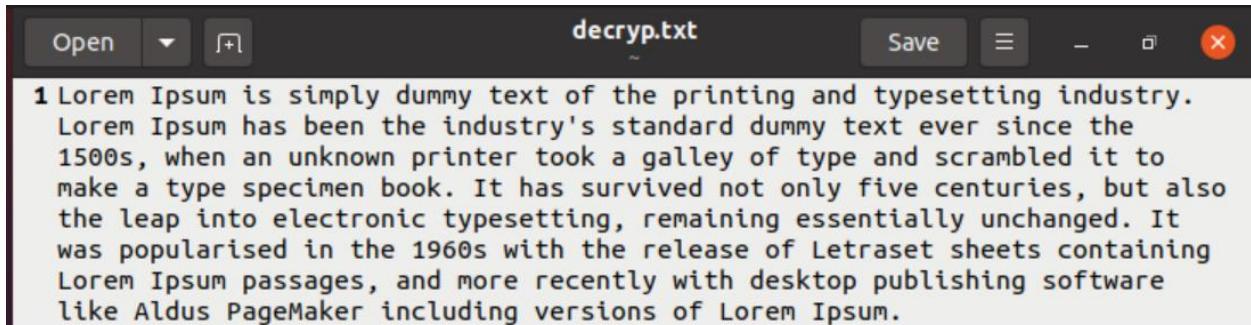
```
tanvi@tanvi:~$ openssl des-cfb -salt -a -e -p -in plaintext.txt -out encryp.txt
enter des-cfb encryption password:
Verifying - enter des-cfb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=81E6BAC51982B8C9
key=0DCB69907109139D
iv =E335F5D2F9CC018B
```

encryp.txt (after encryption)

```
1 J2FsdGVkX1/pgwsbN3Pu/CmH8+qzT8LF0+3Wrh+k5kWCjZXfwe1JVz+urbpVHfA8
2 QVBRArT140uvCtiMKl7SJwIKTffLmQe9AhSy6vp7d4DdP18MiP6IByWzVVRKLNTw
3 cXbL8Uu35qhQk4570JFH0tSnBYt8qlhLA+VBVYviIit0MKaBqH1LsfXCZL6ANFJp
4 v/WBslF4w7GIeqhrM6mxmr82QlGdByxYsr6qKU7GbTvWSZc+lQ405L92piKMvtB
5 VAmtnEd5Ata0ZC40lWYzm+QvOCwijNDrolhbDt/vX9DSe0YgjkRitW22deSccnFz
6 LElkCHM7VKaEqg8+S4lLvmxNXP+EDIUCEDlu70MjHxwdnv9g2wWUhcdQLNQ9vKd
7 UZM0hW4bdQVoThWyGm9YYixjyogELVyPCs+TbmjPcHFIImVHogUDAPdbuTnnP2W9
8 GbPjqwg6be5qonodPKvGm6s0/+6DSkk5I7k/aUFla5iS+KtV9SaqI7JC9/8+w3Su
9 oMK3nSnK3L860hyYkjncN52LD8VmDrVbkZ0/N9rrUvvtaOKp0td24jB6A+Wc+xhG
10 ZBAB8mCI39H+wJBpxD8unHSLLzp2dENcbvHzvjT2SvWdCdFecQe0+G1HTrjV+Ga0
11 RP/UUIDGrpls+yNDhOH56Zfl5LcHIFG/yrl56pEjv0bo15D1FYDcP4FsFtV2f8BnC
12 wSdtUENFff+ngEWMiHMxa7Zc3uZVe0nkLF39yy040ma0/NjE+C0WBvOJYHnsjLh0
13 MShIMvI+Gqji9oaAXC3x
```

After decrypting the file ‘encryp.txt’ and storing the output in decryp.txt

```
tanvi@tanvi:~$ openssl des-cfb -salt -a -d -p -in encryp.txt -out decryp.txt
enter des-cfb decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=81E6BAC51982B8C9
key=0DCB69907109139D
iv =E335F5D2F9CC018B
```



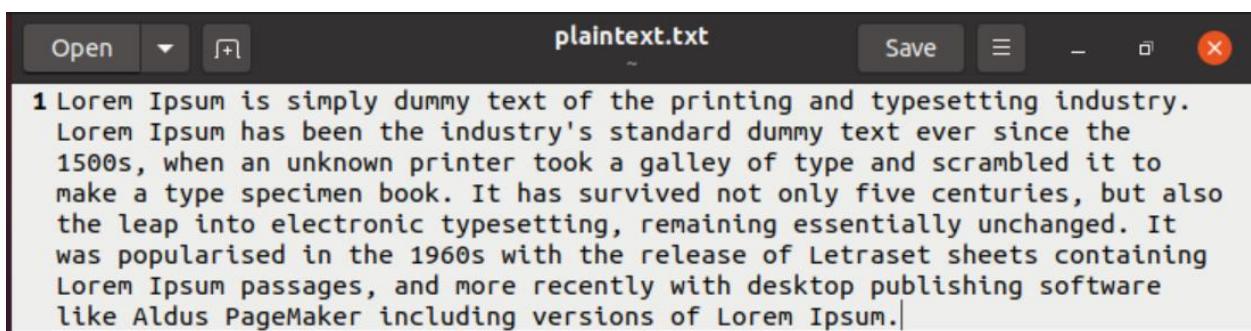
A screenshot of a terminal window titled 'decryp.txt'. The window contains the following text:

```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It
was popularised in the 1960s with the release of Letraset sheets containing
Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.
```

### c) Using the cipher type - rc2

#### c.1) rc2

plaintext.txt



A screenshot of a terminal window titled 'plaintext.txt'. The window contains the following text:

```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It
was popularised in the 1960s with the release of Letraset sheets containing
Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.|
```

After encrypting this file and storing the output in encryp.txt

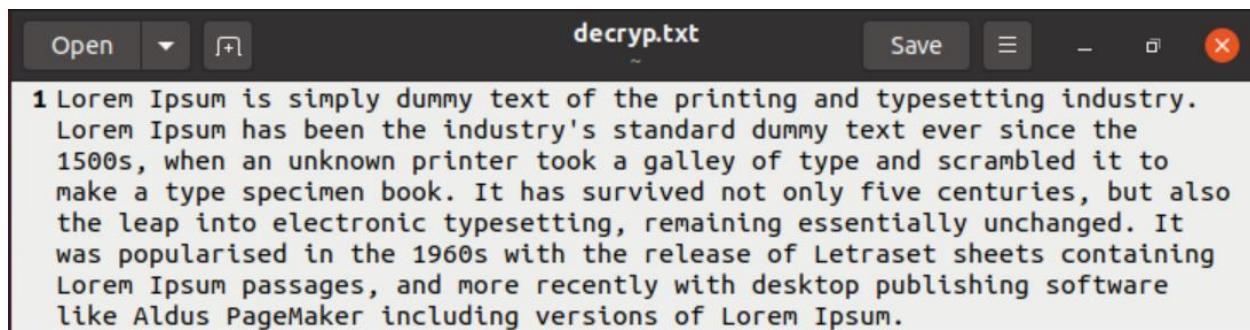
```
tanvi@tanvi:~$ openssl rc2 -salt -a -e -p -in plaintext.txt -out encryp.txt
enter rc2-cbc encryption password:
Verifying - enter rc2-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=DD982CFCE0CD95A9
key=38E9526B5C4779264D441EA8464786B8
iv =5A021FC1EEE4E006
```

encryp.txt (after encryption)



After decrypting the file ‘encryp.txt’ and storing the output in decryp.txt

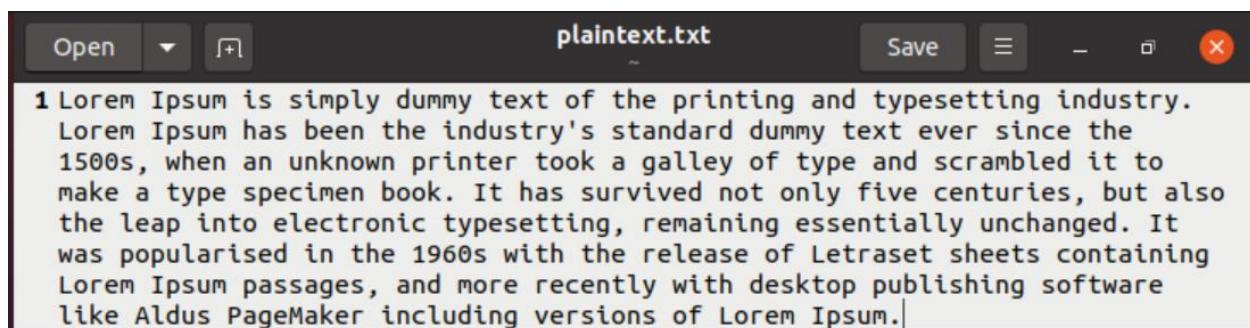
```
tanvi@tanvi:~$ openssl rc2 -salt -a -d -p -in encryp.txt -out decryp.txt
enter rc2-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=DD982CFCE0CD95A9
key=38E9526B5C4779264D441EA8464786B8
iv =5A021FC1EEE4E006
```



```
decryp.txt
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It
was popularised in the 1960s with the release of Letraset sheets containing
Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.
```

## c.2) rc2-64-cbc

plaintext.txt



```
plaintext.txt
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It
was popularised in the 1960s with the release of Letraset sheets containing
Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.|
```

After encrypting this file and storing the output in encryp.txt

```
tanvi@tanvi:~$ openssl rc2-64-cbc -salt -a -e -p -in plaintext.txt -out encryp.
txt
enter rc2-64-cbc encryption password:
Verifying - enter rc2-64-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=4085E2A190B90A67
key=3B5BFDEF3FBA130E
iv =B6844A130D759DEC
```

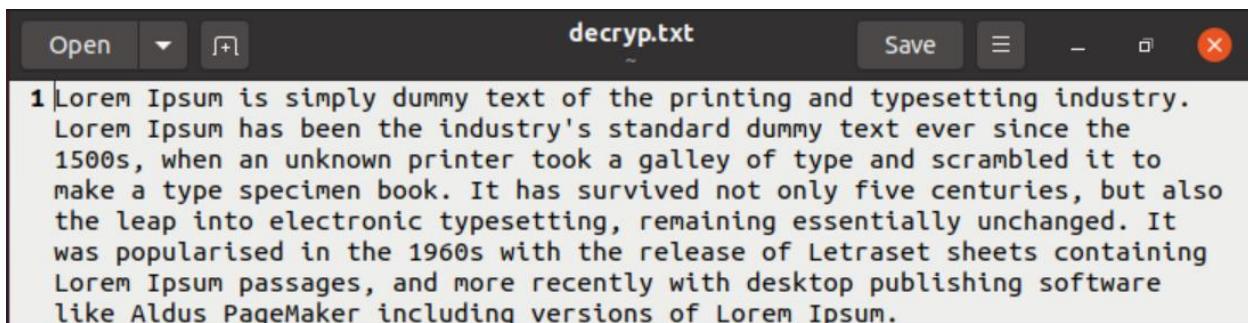
## encryp.txt (after encryption)



```
Open Save encryp.txt - + X
1 U2FsdGVkX19FoI+7je2+Ey5xrWaRrRxwriVq+ds5n5vy8S2TM9sV5c9+5evo0fXS
2 ao0l5PCiYNmuNF5lelKlisja4jRpMPBKEGrtJdhvVMizKAEGKiBzx3aWQ8mmHrvv
3 T9yzvdbDDV6MLTqDsja4jRpMPBKEGrtJdhvVMizKAEGKiBzx3aWQ8mmHrvv
4 PtkG6j/siPtHFug3AoZvjv/DogJH5WLyu9WFewwF/UDF3oTzFGAfRAVkyOVAhrb
5 doU06J3ZBsJg0pzl6/BR0rs5taSjZ4fJD2MmSLRIP0uS8Iu2VjBOEF0WgBHhudIe
6 Ro/PwrFygdiMkrnyKIuVOyAjjYmMyj+ffyS+j1zxjxD6zreZrdU1Bq2lzaE1Ml7A
7 vJzxfdPkJ867bn68pjJ67k+cnDNj+yg5lEJM/kX9nLTgS9Q900+YHHPEZaW01GgF
8 bl1lQQxJEh7gbmrX+4tiITqHm/WJmrQIDkLECtxMqDh9Xvo5XR2LZN890vV9cjSm
9 bvbuU+v0GC3E7tp0GDiWaE1PfTF4NQt2Kjz/r6peMMIJvajxCWP9pcVkJqLkOYJ/
10 PCsg05mBoIqZSw8NRS8Wp8HcCWP0PezFwPwDlKr0iKccfTwnrT0FYQruXBzQxWHS
11 lSEzx4vj3wAtg55ZDdhvDiPaTJCEWDW6WKg9Ps9hc4le9dza0gc8sB0Ts/duiGoM
12 P+qgbUX24ZN7P1qnHOZp1aUCTAWLaPY6h6fmi8mUQRQWucNKoDN4Nwu5oDs8j5ws
13 twc8cvnsne/qaG+lg+ntMNw==
```

After decrypting the file ‘encryp.txt’ and storing the output in decryp.txt

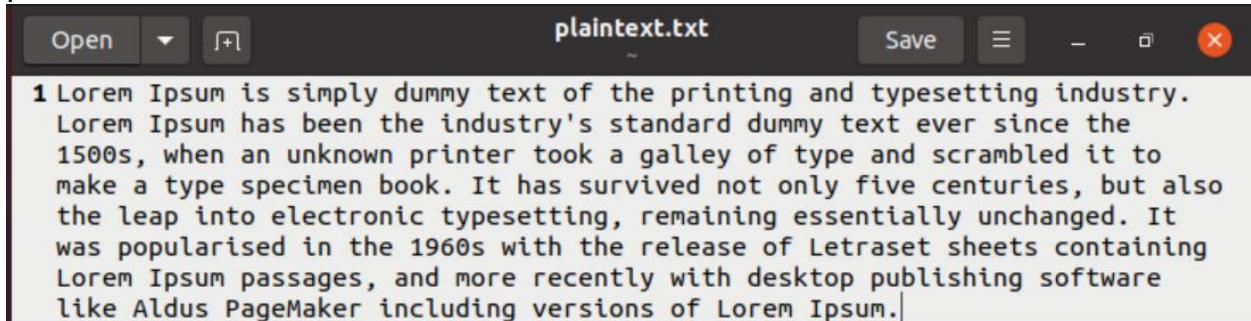
```
tanvi@tanvi:~$ openssl rc2-64-cbc -salt -a -d -p -in encryp.txt -out decryp.txt
enter rc2-64-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=4085E2A190B90A67
key=3B5BFDEF3FBA130E
iv =B6844A130D759DEC
```



```
Open Save decryp.txt - + X
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  Lorem Ipsum has been the industry's standard dummy text ever since the
  1500s, when an unknown printer took a galley of type and scrambled it to
  make a type specimen book. It has survived not only five centuries, but also
  the leap into electronic typesetting, remaining essentially unchanged. It
  was popularised in the 1960s with the release of Letraset sheets containing
  Lorem Ipsum passages, and more recently with desktop publishing software
  like Aldus PageMaker including versions of Lorem Ipsum.
```

### c.3) rc2-40-cbc

plaintext.txt



```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.  
2 Lorem Ipsum has been the industry's standard dummy text ever since the  
3 1500s, when an unknown printer took a galley of type and scrambled it to  
4 make a type specimen book. It has survived not only five centuries, but also  
5 the leap into electronic typesetting, remaining essentially unchanged. It  
6 was popularised in the 1960s with the release of Letraset sheets containing  
7 Lorem Ipsum passages, and more recently with desktop publishing software  
8 like Aldus PageMaker including versions of Lorem Ipsum.|
```

After encrypting this file and storing the output in encryp.txt

```
tanvi@tanvi:~$ openssl rc2-40-cbc -salt -a -e -p -in plaintext.txt -out encryp.  
txt  
enter rc2-40-cbc encryption password:  
Verifying - enter rc2-40-cbc encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
salt=33A40FB70DE7894A  
key=73D57F07D6  
iv =C0FB24A9695299A6
```

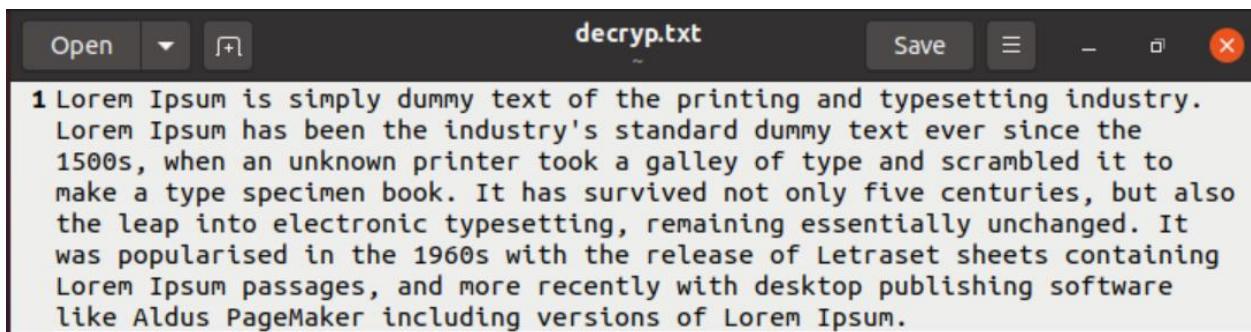
encryp.txt (after encryption)



```
1 J2FsdGVkX18PVS2UGF04E62DkY6qgw0lCR40svtJH93HZ1Pjp9R2omNtLmdRoUWU  
2 eaHCWamknAKMIIiXP3FfyASSwgzSmLZ4NX+aB8IqjtY3KeE89zbbJyeSELWytlh  
3 X7/Wpafhb0wcd/5JtQ9Ki02v/Nwkg+t/L+lMjKvVHpUVct4oX0iPtzBnjVBRQtyq  
4 3bi56WxugYt1kSkogT4LzQoeQfmuJSou4vWdrQzS5r7p/EzCce5WFdqHFHax6ADt  
5 egFieeIZIM8lBgKpLwarFefj4rUhGxbLnF8S2bB6KurcS7VFvjS91u/qckIJVtSy  
6 s3wltI26EqvcKSWGax+uvGGbU3L4tHXJv/ekApVeDLq4xy9z3d6SZ3idC0QjpIfu  
7 z33uD50WfZyzneV6jzeifWcujqr6MiRrSft3mCtXJDMDlKggaD/CYQe3rmolF+  
8 thgkRcsihkkVCy1Bi1ip6FBgGITDzfvAg9dTgg4/fqlt5/hy8QofaPh5H0ZvjC2D  
9 P6MNnv/BJuajXP79FeG7yyAwfmHH9Qflqd2rr85kqI0t+ArEEK92snezfKGvm60  
10 GznYyGzHDhfPgnzPyMuS7AT4uKOuMztffT8tJE6jb3lUanFdH0oF/ujSq8hUydbE  
11 grIthG8u6uZuNmjkMBYve+CQPUDIKW8+9UnEwDNFQUGL+edvROX1/TbI1YZwRv/n  
12 R8DjQpbfmagouILkFKFxkMgkbp/pQwKgPx3z0QtEtApn90duydMV6UDsiQWzg1AJ  
13 JKuDFuA+SmCM/MTVpZQM2Q==
```

After decrypting the file ‘encryp.txt’ and storing the output in decryp.txt

```
tanvi@tanvi:~$ openssl rc2-40-cbc -salt -a -d -p -in encryp.txt -out decryp.txt
enter rc2-40-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=33A40FB70DE7894A
key=73D57F07D6
iv =C0FB24A9695299A6
```



## Conclusion:

- It was observed that the key length of aes and rc2 is variable while des has a fixed key length of 56 bits.
- aes can have key length of 128, 192 and 256 bits, while rc2 can have key length of 40 and 64 bits.
- The –p parameter helped to display the salt key and initial vector (iv).
- The above algorithms are symmetric key algorithms and hence they use the same key to encrypt and decrypt data.
- The size of encrypted text is found to be larger than the plain text, as encryption provides redundancy.

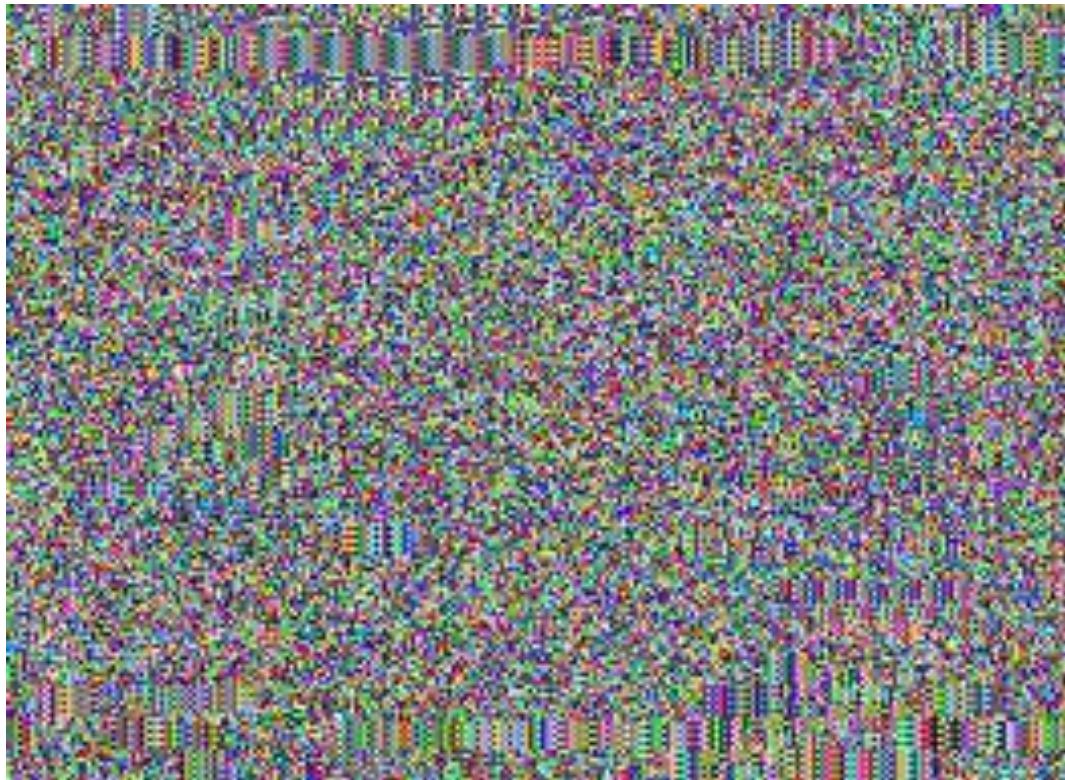
## **2) Task 2:**

### **Original Image:**



#### **a) Using the cipher type - aes-256-ecb**

```
C:\Users\manal\Desktop\Tanvi\Sem5\CCS>openssl enc -aes-256-ecb -e -p -in img.bmp -out encecb.bmp
enter AES-256-ECB encryption password:
Verifying - enter AES-256-ECB encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=624D80481B8B3754
key=9C779A27F3313505D3F82378F94AAFD157768502F17F0BE15947E9D3B2A4916A
```



### b) Using the cipher type - aes-256-cbc

```
C:\Users\manal\Desktop\Tanvi\Sem5\CCS>openssl enc -aes-256-cbc -e -p -in img.bmp -out enccbc.bmp
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=D4947C2CAF9477BB
key=830D70F0A6DF1D6FD35DD16272A0A82FC28D472B82A0933B8C8BCD743728FFFC
iv =CE981FD69103CBA217254F0266B520D6
```

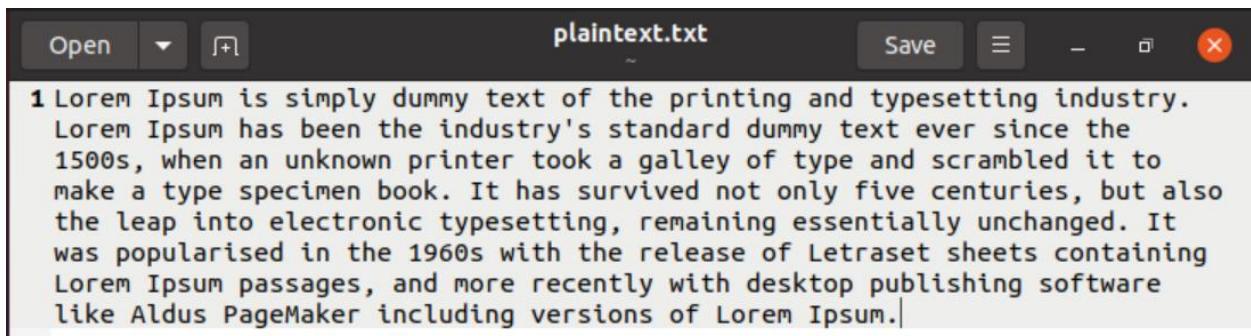


### **Conclusion:**

- The image encrypted using ECB mode still has little shapes in the encrypted image which might help to figure out the original picture.
- The image encrypted using CBC is completely distorted and hence making it impossible to understand the original image.

### 3) Task 3:

The text file:



A screenshot of a text editor window titled "plaintext.txt". The content of the file is a single paragraph of Lorem Ipsum text. The text is as follows:

```
1 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  Lorem Ipsum has been the industry's standard dummy text ever since the
  1500s, when an unknown printer took a galley of type and scrambled it to
  make a type specimen book. It has survived not only five centuries, but also
  the leap into electronic typesetting, remaining essentially unchanged. It
  was popularised in the 1960s with the release of Letraset sheets containing
  Lorem Ipsum passages, and more recently with desktop publishing software
  like Aldus PageMaker including versions of Lorem Ipsum.
```

- Using the cipher type - aes-128-ecb to encrypt the plaintext.txt

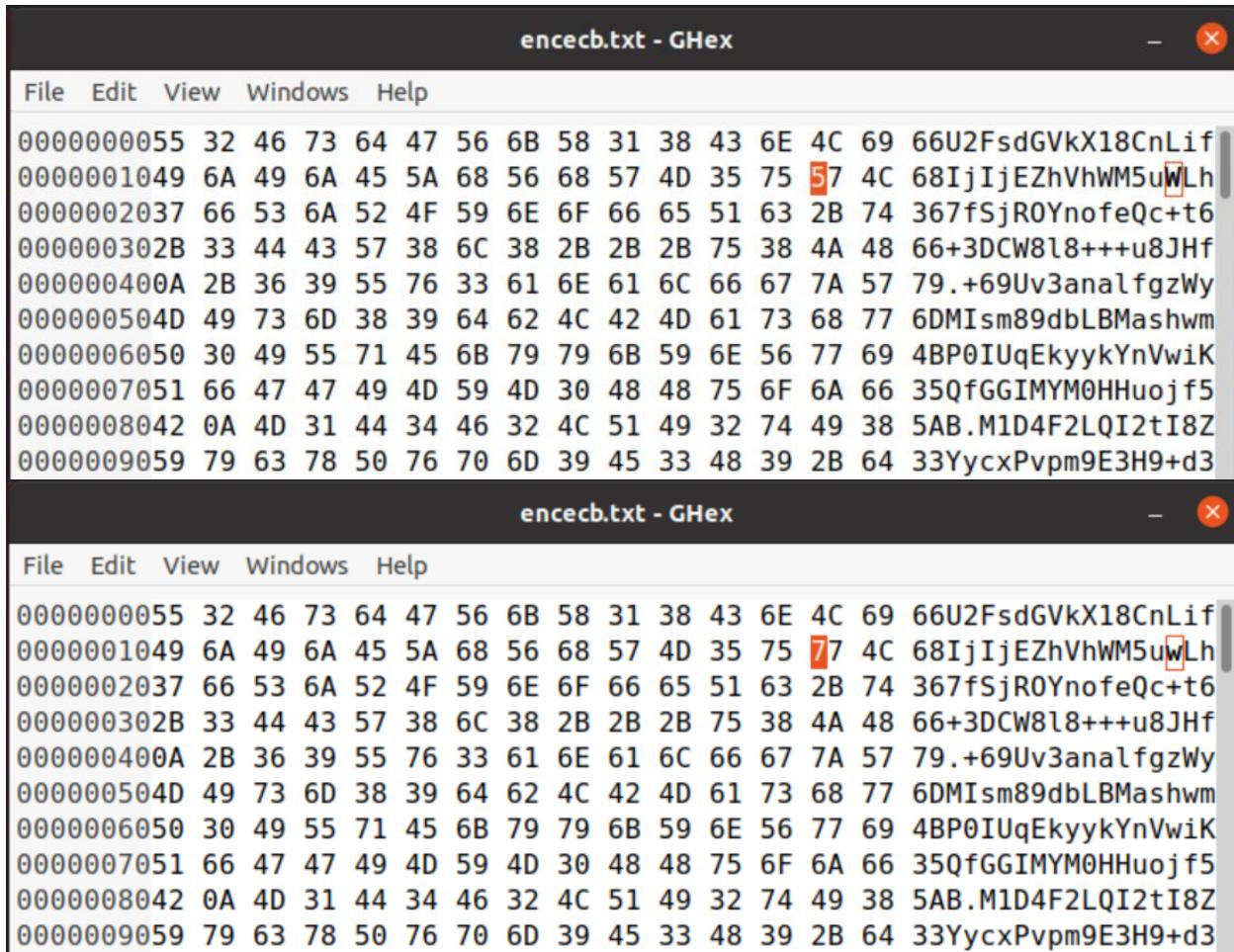
```
tanvi@tanvi:~$ openssl aes-128-ecb -salt -a -e -p -in plaintext.txt -out encecb
.txt
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=029CB89F22322311
key=FC805DFD859AE6ED02AD4256E610A2A3
```



A screenshot of a text editor window titled "encecb.txt". The content of the file is a long string of encrypted data, consisting of 13 lines of hex-encoded ciphertext. The first few lines of the ciphertext are:

```
1 J2FsdGVkX18CnLifIjIjEZhvHWM5uWLh7fSjR0YnofeQc+t6+3DCW8l8+++u8JHf
2 +69Uv3analfgzWyMIsm89dbLBMAshwmP0IUqEkyykYnVwiKQfGGIMYM0HHuojf5B
3 M1D4F2LQI2tI8ZYycxPvpm9E3H9+d3Ihadcv1Utswd5VMcDjNCI3CUu4g3UbIqiB
4 4e4XB2ybBqlQ5lS4YNEFKJ7ytrE4mDT2JIJpwp2Z3a175lAWo1+DG0RsgpOZFFGE
5 nrqxMaTXwmq7rw2yjB1u4j3U1ejKy6y6f56FPstrbrnT+RrJBHt/ZxsCV91bwoN
6 GOX0sn4YupUlbd+CdcAiAMsgb049wPsiaFwci6z48sIhJR2IwBoyej8/TEYDsLmm
7 lh2w8lHzfvuIsSC5E2Q0c2iSrt7jTATluWn0XeJgSuVFzhnCJvBS+AKgAVQHk5v
8 ds8mIVe5R66nVJ1fxb+h182hySjBMsYQbMcOUQ5eS74K5+0h2ISF/CAp1Rf6mlMg
9 Zkjli3y1MFzQLW/wTFt7kJNnC0SvgZNnRgz0Y8CWKgp/bGpFZ/ojNJArIwBhngu
10 j5Uqcw3qnG/oJJ99jeS0xpjSRdSWyl6Qj8v9DuKa8IhGmzLKYhI1nvPGDk4XdhKK
11 fcCPQgX8vn29hEONS5N7tm9+Kk6AEuKiKeWKT6ERWPeeA4vFzMVRxkZ75C2McAq
12 wkUgn+R9hRJFzWRU2uOAZ6RkxaPRbXSqKHCVboZWyBMR9/YU/nNTmj0hdsPC26K
13 78h+3mv/I/v4u8yuHXzaVQ==
```

Corrupting a single bit of the 30th byte:



```
encecb.txt - GHHex
```

```
File Edit View Windows Help
```

```
0000000055 32 46 73 64 47 56 6B 58 31 38 43 6E 4C 69 66U2FsdGVkX18CnLif
0000001049 6A 49 6A 45 5A 68 56 68 57 4D 35 75 57 4C 68IjIjEZhvWM5uWl
0000002037 66 53 6A 52 4F 59 6E 6F 66 65 51 63 2B 74 367fSjR0YnofeQc+t6
000000302B 33 44 43 57 38 6C 38 2B 2B 2B 75 38 4A 48 66+3DCW8l8+++u8JHf
000000400A 2B 36 39 55 76 33 61 6E 61 6C 66 67 7A 57 79.+69Uv3analfgzWy
000000504D 49 73 6D 38 39 64 62 4C 42 4D 61 73 68 77 6DMIsm89dbLB Mashwm
0000006050 30 49 55 71 45 6B 79 79 6B 59 6E 56 77 69 4BP0IUqEkyykYnVwiK
0000007051 66 47 47 49 4D 59 4D 30 48 48 75 6F 6A 66 35QfGGIMYM0HHuojf5
0000008042 0A 4D 31 44 34 46 32 4C 51 49 32 74 49 38 5AB.M1D4F2LQI2tI8Z
0000009059 79 63 78 50 76 70 6D 39 45 33 48 39 2B 64 33YycxPvpm9E3H9+d3
```

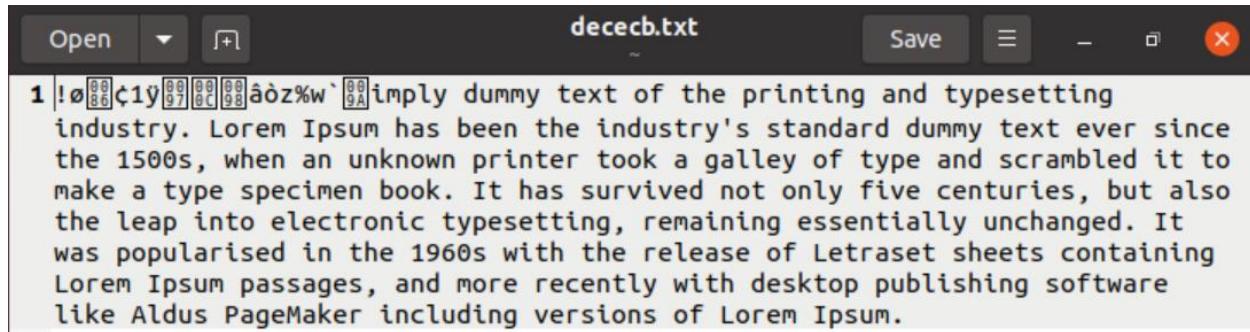
```
encecb.txt - GHHex
```

```
File Edit View Windows Help
```

```
0000000055 32 46 73 64 47 56 6B 58 31 38 43 6E 4C 69 66U2FsdGVkX18CnLif
0000001049 6A 49 6A 45 5A 68 56 68 57 4D 35 75 77 4C 68IjIjEZhvWM5uWl
0000002037 66 53 6A 52 4F 59 6E 6F 66 65 51 63 2B 74 367fSjR0YnofeQc+t6
000000302B 33 44 43 57 38 6C 38 2B 2B 2B 75 38 4A 48 66+3DCW8l8+++u8JHf
000000400A 2B 36 39 55 76 33 61 6E 61 6C 66 67 7A 57 79.+69Uv3analfgzWy
000000504D 49 73 6D 38 39 64 62 4C 42 4D 61 73 68 77 6DMIsm89dbLB Mashwm
0000006050 30 49 55 71 45 6B 79 79 6B 59 6E 56 77 69 4BP0IUqEkyykYnVwiK
0000007051 66 47 47 49 4D 59 4D 30 48 48 75 6F 6A 66 35QfGGIMYM0HHuojf5
0000008042 0A 4D 31 44 34 46 32 4C 51 49 32 74 49 38 5AB.M1D4F2LQI2tI8Z
0000009059 79 63 78 50 76 70 6D 39 45 33 48 39 2B 64 33YycxPvpm9E3H9+d3
```

After decrypting this corrupted file:

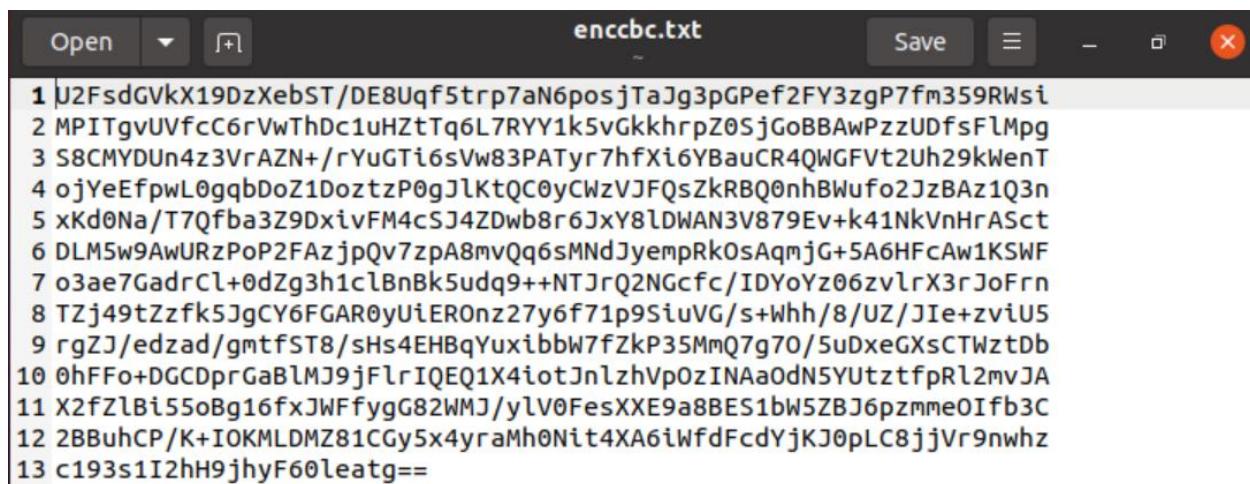
```
tanvi@tanvi:~$ openssl aes-128-ecb -salt -a -d -p -in encecb.txt -out dececb.txt
enter aes-128-ecb decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=029CB89F22322311
key=FC805DFD859AE6ED02AD4256E610A2A3
```



The screenshot shows a text editor window titled "dececb.txt". The content of the file is a portion of the well-known Lorem Ipsum dummy text, starting with "1 !øç1ÿâòz%w`imply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum."

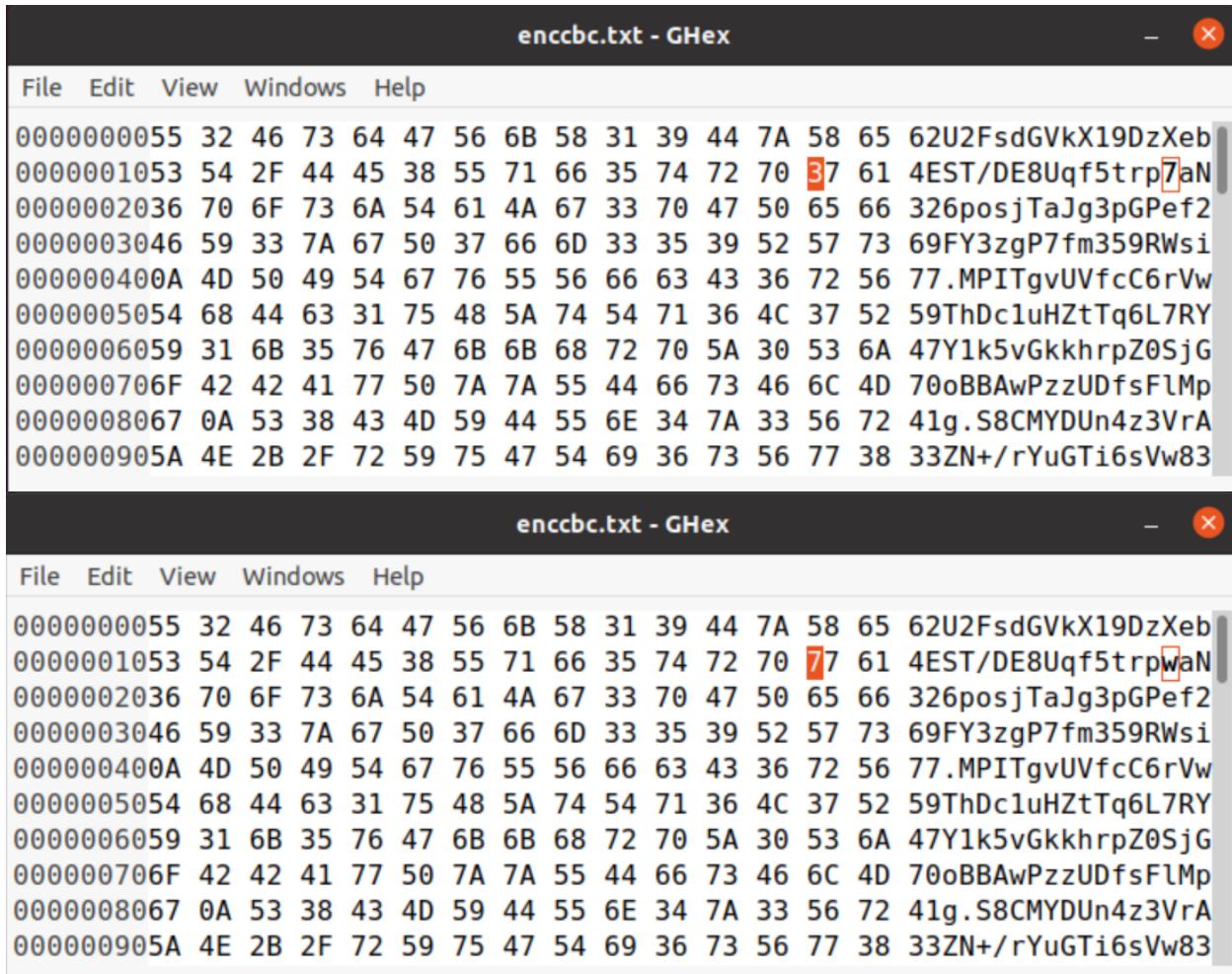
- Using the cipher type - aes-128-cbc to encrypt the plaintext.txt

```
tanvi@tanvi:~$ openssl aes-128-cbc -salt -a -p -e -in plaintext.txt -out enccbc.txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=43CD779B493FC313
key=0151464AF7534DCFB2AA367D291B5C1E
iv =79DDB138DB35C12B7AB67C7788C1F6CC
tanvi@tanvi:~$
```



The screenshot shows a text editor window titled "enccbc.txt". The content of the file is a long string of encrypted binary data, starting with "1 J2FsdGVkX19DzXebST/DE8Uqf5trp7aN6posjTaJg3pGPef2FY3zgP7fm359RWsi". This is the result of encrypting the original Lorem Ipsum text using the OpenSSL command provided.

Corrupting a single bit of the 30th byte:



The screenshot shows two instances of the GHex hex editor. Both windows are titled "enccbc.txt - GHHex". The menu bar includes File, Edit, View, Windows, and Help. The main pane displays a sequence of hex values. In both windows, the byte at index 30 is highlighted in red. In the top window, the value is 7A (hex), and in the bottom window, it is 77 (hex). The rest of the file content remains identical.

Index	Value (Top Window)	Value (Bottom Window)
0	32	32
1	46	46
2	73	73
3	64	64
4	47	47
5	56	56
6	6B	6B
7	58	58
8	31	31
9	39	39
10	44	44
11	7A	7A
12	58	58
13	65	65
14	62U2FsdGVkX19DzXeb	62U2FsdGVkX19DzXeb
15	0000001053	0000001053
16	54	54
17	2F	2F
18	44	44
19	45	45
20	38	38
21	55	55
22	71	71
23	66	66
24	35	35
25	74	74
26	72	72
27	70	70
28	37	77
29	61	61
30	4EST/DE8Uqf5trp7aN	4EST/DE8Uqf5trpwaN
31	0000002036	0000002036
32	70	70
33	6F	6F
34	73	73
35	6A	6A
36	54	54
37	61	61
38	4A	4A
39	67	67
40	33	33
41	70	70
42	47	47
43	50	50
44	66	66
45	55	55
46	63	63
47	76	76
48	6B	6B
49	68	68
50	72	72
51	70	70
52	5A	5A
53	30	30
54	53	53
55	6A	6A
56	47Y1k5vGkkhrpZ0SjG	47Y1k5vGkkhrpZ0SjG
57	0000003046	0000003046
58	59	59
59	33	33
60	7A	7A
61	67	67
62	50	50
63	37	37
64	66	66
65	6D	6D
66	33	33
67	35	35
68	39	39
69	52	52
70	57	57
71	73	73
72	69	69
73	36	36
74	73	73
75	56	56
76	77	77
77	38	38
78	33ZN+/rYuGTi6sVw83	33ZN+/rYuGTi6sVw83

After decrypting this corrupted file:

```
tanvi@tanvi:~$ openssl aes-128-cbc -salt -a -p -d -in enccbc.txt -out deccbc.txt
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=43CD779B493FC313
key=0151464AF7534DCFB2AA367D291B5C1E
iv =79DDB138DB35C12B7AB67C7788C1F6CC
tanvi@tanvi:~$
```

Open ▾ ⌂ deccbc.txt Save ⌂ x  
1 êJ°1 [84][85][86][87] mÈ¶14} 'Ôøµimply Ôummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

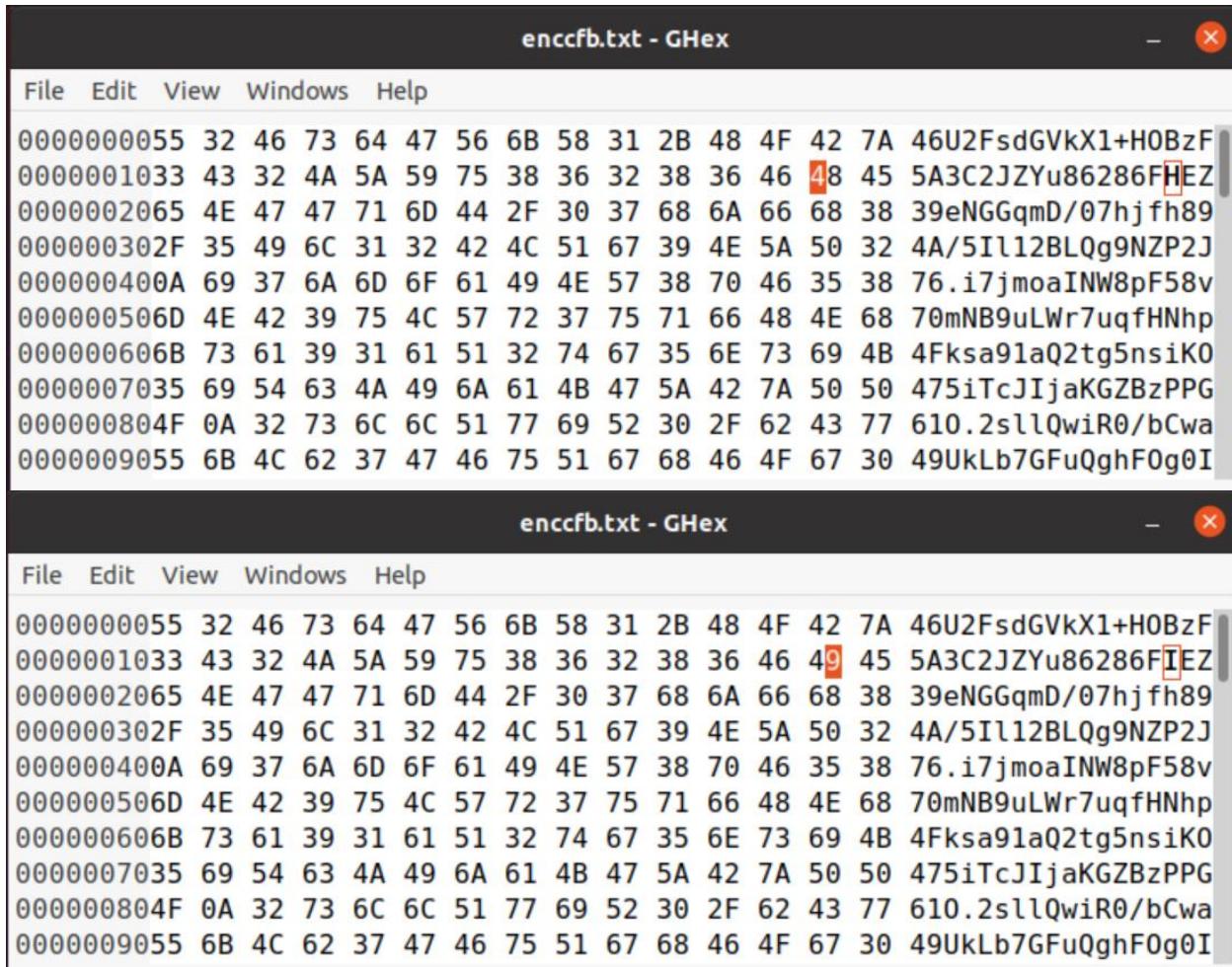
- Using the cipher type - aes-128-cfb to encrypt the plaintext.txt

```
tanvi@tanvi:~$ openssl aes-128-cfb -salt -a -p -e -in plaintext.txt -out enccfb
.txt
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.
salt=87381CC5DC2D8965
key=54634424C90DEAD2B3F830D18DEC339A
iv =CAE7D257CFAFE800EF415A19414E0204
tanvi@tanvi:~$
```

Open Save

```
1 U2FsdGVkX1+H0BzF3C2JZYu86286FHEZeNGGqmD/07hjfh89/5Il12BLQg9NPZ2J
2 i7jmoaINW8pF58vmNB9uLWr7uqfHNhpksa91aQ2tg5nsiK05iTcJIjaKGZBzPPGO
3 2sllQwiR0/bCwaUkLb7GFuQghF0g0IYKDgqJc6Y9kcaCAMszayIplN0U3A0g1DKT
4 i8ST+2/MNYNcvu6uEYJ1Zk5R0Q9rYzTX3+fHVFJ6aC5LK2Hk2IFkSW8uBo0LBQFK
5 2/9SSmRU0U23pmlu2YCNYJfVSY9BbUhOCsuYSumoN14JULB4FH2g/VwNMr+zAjM6
6 cHCC67no/qUDyL1eChtN4sYyTmuC6nBmj3uQs310HoAXzwYwudbp+EXwMcfa80Vq
7 RTp32egVxw/zAEfMoa7dfJrJN4FI/zKNFIL/6RQx671e2ez59EQ4efbBCSr0MXm+
8 +EzpvcNPXETp1rbEkjRkPvHeuVmR75LtJ2hjk6REtKkMgq+N60xlj0slt8XRyB1
9 osDJTFFY6u9G3y8n/mtNAtxiRZQeDJ3cra03igh4fKgN+h52JananP75n36w5zU
10 Mf2UAMps990CAmF9PJ6bRiIF6HKEfPuyG7BqWCojc3FcQuwMJtIzv01CEkxdlnj5
11 AEi6gdyGvkjkFsy+y85kSXLHTf6yLfUbX9UgQCuvR3HfcYdGu7IeCzzAbVmqqkjJ
12 EZn/5tiAaHViR6+Cmvhazes8eBg66+RM2cHURXifwaoCDmVrZyGEgLcufAgKI70A
13 9l0vorjZGZjmoI92QCu+
```

Corrupting a single bit of the 30th byte:

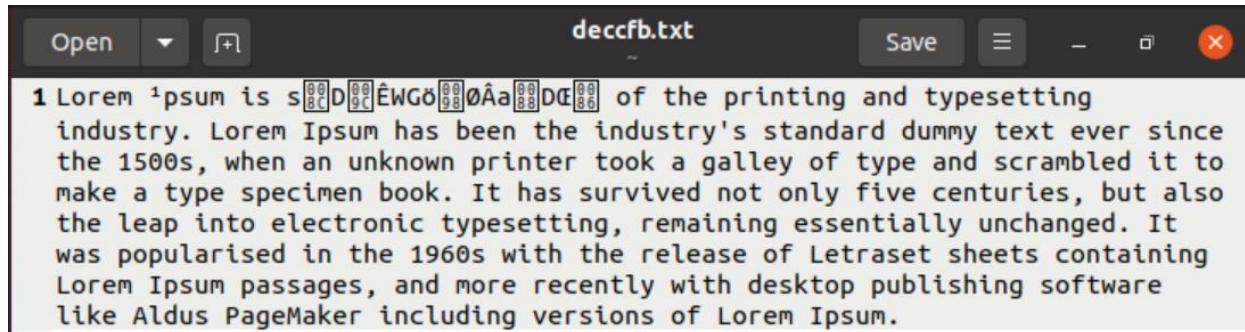


```
enccfb.txt - GHHex
File Edit View Windows Help
0000000055 32 46 73 64 47 56 6B 58 31 2B 48 4F 42 7A 46U2FsdGVkX1+H0BzF
0000001033 43 32 4A 5A 59 75 38 36 32 38 36 46 48 45 5A3C2JZYu86286FHEZ
0000002065 4E 47 47 71 6D 44 2F 30 37 68 6A 66 68 38 39eNGGqmD/07hjf fh89
000000302F 35 49 6C 31 32 42 4C 51 67 39 4E 5A 50 32 4A/5Il12BLQg9NZP2J
000000400A 69 37 6A 6D 6F 61 49 4E 57 38 70 46 35 38 76.i7jmoaINW8pF58v
000000506D 4E 42 39 75 4C 57 72 37 75 71 66 48 4E 68 70mNB9uLWr7uqfHNhp
000000606B 73 61 39 31 61 51 32 74 67 35 6E 73 69 4B 4Fksa91aQ2tg5nsiK0
0000007035 69 54 63 4A 49 6A 61 4B 47 5A 42 7A 50 50 475iTcJIjaKGZBzPPG
000000804F 0A 32 73 6C 6C 51 77 69 52 30 2F 62 43 77 610.2sllQwiR0/bCwa
0000009055 6B 4C 62 37 47 46 75 51 67 68 46 4F 67 30 49UkLb7GFuQghF0g0I

enccfb.txt - GHHex
File Edit View Windows Help
0000000055 32 46 73 64 47 56 6B 58 31 2B 48 4F 42 7A 46U2FsdGVkX1+H0BzF
0000001033 43 32 4A 5A 59 75 38 36 32 38 36 46 49 45 5A3C2JZYu86286FIEZ
0000002065 4E 47 47 71 6D 44 2F 30 37 68 6A 66 68 38 39eNGGqmD/07hjf fh89
000000302F 35 49 6C 31 32 42 4C 51 67 39 4E 5A 50 32 4A/5Il12BLQg9NZP2J
000000400A 69 37 6A 6D 6F 61 49 4E 57 38 70 46 35 38 76.i7jmoaINW8pF58v
000000506D 4E 42 39 75 4C 57 72 37 75 71 66 48 4E 68 70mNB9uLWr7uqfHNhp
000000606B 73 61 39 31 61 51 32 74 67 35 6E 73 69 4B 4Fksa91aQ2tg5nsiK0
0000007035 69 54 63 4A 49 6A 61 4B 47 5A 42 7A 50 50 475iTcJIjaKGZBzPPG
000000804F 0A 32 73 6C 6C 51 77 69 52 30 2F 62 43 77 610.2sllQwiR0/bCwa
0000009055 6B 4C 62 37 47 46 75 51 67 68 46 4F 67 30 49UkLb7GFuQghF0g0I
```

After decrypting this corrupted file:

```
tanvi@tanvi:~$ openssl aes-128-cfb -salt -a -p -d -in enccfb.txt -out deccfb.txt
enter aes-128-cfb decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=87381CC5DC2D8965
key=54634424C90DEAD2B3F830D18DEC339A
iv =CAE7D257CFAFE800EF415A19414E0204
tanvi@tanvi:~$
```



The screenshot shows a terminal window with the file "deccfb.txt" open. The content of the file is a block of Lorem Ipsum text, which includes several special characters like Ä, Ö, and Å. The terminal interface includes standard window controls (Open, Save, Close) and a menu bar.

```
1 Lorem ipsum is sDÉWGöØÅâDŒ of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
```

- Using the cipher type - aes-128-ofb to encrypt the plaintext.txt

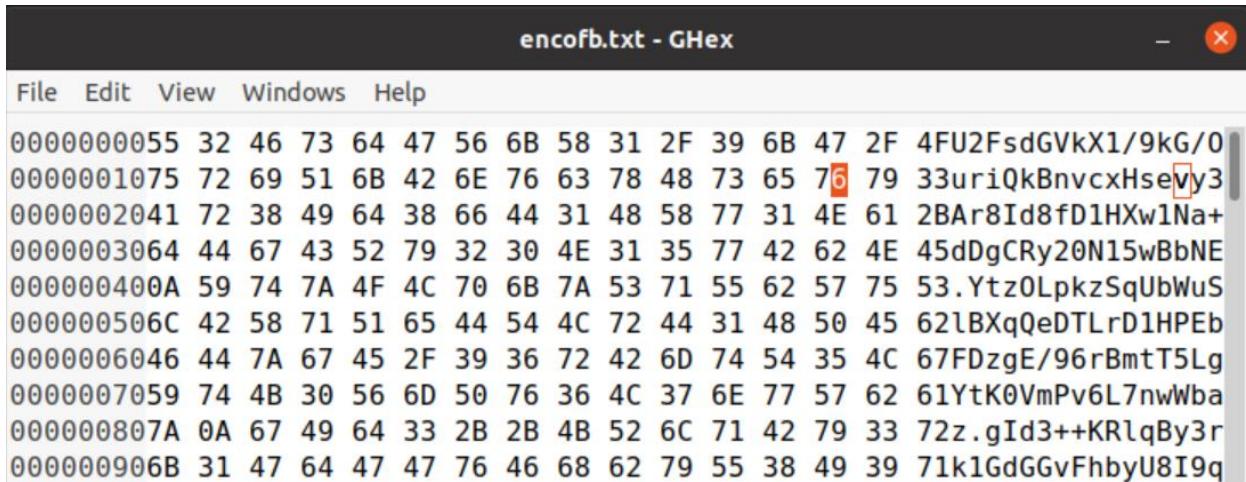
```
tanvi@tanvi:~$ openssl aes-128-ofb -salt -a -p -e -in plaintext.txt -out encofb.txt
enter aes-128-ofb encryption password:
Verifying - enter aes-128-ofb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=FD906FCEBAB89090
key=AE03AE3EF1070FF5E9B42A7730FC40D8
iv =2D3024B7D0E5C9DBBD378FF09053A6A9
tanvi@tanvi:~$
```



The screenshot shows a terminal window with the file "encofb.txt" open. The content of the file is a long string of encrypted data consisting of 13 lines of hex-encoded characters. The terminal interface includes standard window controls (Open, Save, Close) and a menu bar.

```
1 U2FsdGVkX1/9kG/OuriQkBnvCxHsevy3Ar8Id8fD1HXw1Na+dDgCRy20N15wBbNE
2 YtzOLpkzSqUbWuSlBXqQeDTLrD1HPEbFDzgE/96rBmtT5LgYtK0VmPv6L7nwWbaz
3 gId3++KRlqBy3rk1GdGGvFhbyU8I9qBxvFmi0bLiDIJvEUZzRDAPEZU04p5PyanF
4 XWg1xn/X1HU8C1yjQ+fjzyjggHDDHVq2ptBOfgyrP4YDEAw42+nYwf9YleeINz35
5 zubGRbK5oNP1sNcZmpnRN/+1TSt3s7dTdhSPrn5MqjZbWUm2NdWT0UdRe5qth1KC
6 dyOVSzHrVoI4qpkYajxuJcPBiw6An4wSSTLEU1Ld7gpGGPdVKBZ2RMBFUrrR2eUiE
7 R6HUxrY/AarQgYmpHmlG4SR5G/ohyojMpI6C12TqSnZ16DTTdJnXeri50BAq28Mf
8 Ms3Ie2ZadhSLa1MhcoRWmsPV1ISeS0apAzUpQ6M+3FhyZYEpGqvkd1wyzo5mie29
9 xtWhxogKQ3caIgiR/l38RIobnHjBviVon00K2RJ0MQhu85aUxkiJRwUeqLuuPF8G
10 DEZyykhsFn+oGNRnnlo7b7zMMU2kTFvv77zHMHevaKsQF33WPFR9+S206mxLtUgU
11 rtDFPPNY+bPX24W8W11CncDd+sxDKokke+yygvWlMF067WMxsbc19Ar/a5Zr9u2
12 UvL5XmB1WKlpfG8+sRhMLCNKgSemHJPu03VN6swn35Sb4MNzACEj0GyZCCeDwWI6
13 XC3k3d2LZvIJxiUMnlt4
```

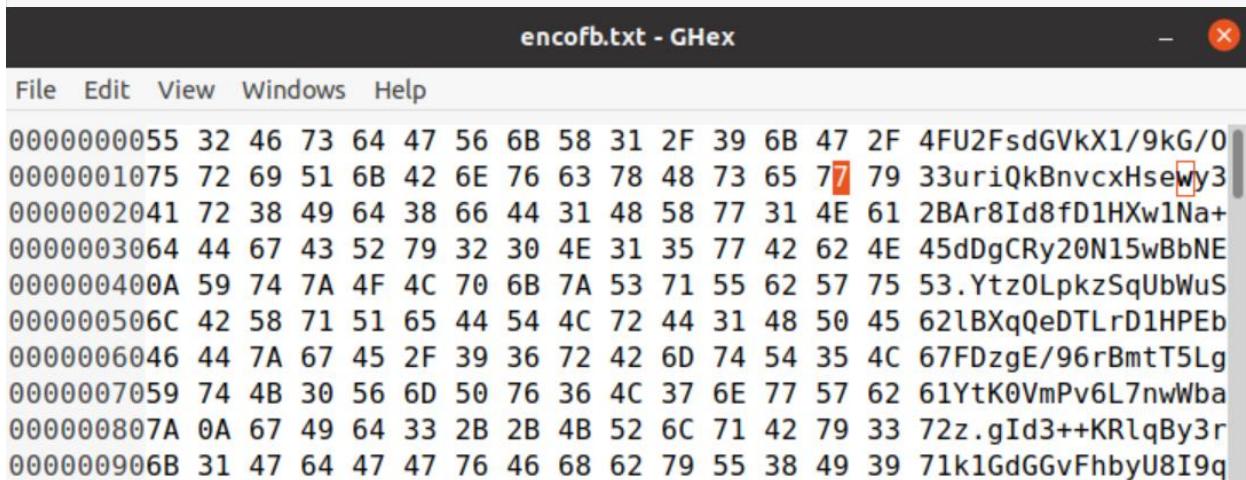
Corrupting a single bit of the 30th byte:



encofb.txt - GHHex

File Edit View Windows Help

0000000055 32 46 73 64 47 56 6B 58 31 2F 39 6B 47 2F 4FU2FsdGVkX1/9kG/0  
0000001075 72 69 51 6B 42 6E 76 63 78 48 73 65 76 79 33uriQkBnvcxHsevy3  
0000002041 72 38 49 64 38 66 44 31 48 58 77 31 4E 61 2BAr8Id8fD1HXw1Na+  
0000003064 44 67 43 52 79 32 30 4E 31 35 77 42 62 4E 45dDgCRy20N15wBbNE  
000000400A 59 74 7A 4F 4C 70 6B 7A 53 71 55 62 57 75 53.Ytz0LpkzSqUbWuS  
000000506C 42 58 71 51 65 44 54 4C 72 44 31 48 50 45 62lBXqQeDTLrD1HPEb  
0000006046 44 7A 67 45 2F 39 36 72 42 6D 74 54 35 4C 67FDzgE/96rBmtT5Lg  
0000007059 74 4B 30 56 6D 50 76 36 4C 37 6E 77 57 62 61YtK0VmPv6L7nwWba  
000000807A 0A 67 49 64 33 2B 2B 4B 52 6C 71 42 79 33 72z.gId3++KRLqBy3r  
000000906B 31 47 64 47 47 76 46 68 62 79 55 38 49 39 71k1GdGGvFhbyU8I9q



encofb.txt - GHHex

File Edit View Windows Help

0000000055 32 46 73 64 47 56 6B 58 31 2F 39 6B 47 2F 4FU2FsdGVkX1/9kG/0  
0000001075 72 69 51 6B 42 6E 76 63 78 48 73 65 77 79 33uriQkBnvcxHsevy3  
0000002041 72 38 49 64 38 66 44 31 48 58 77 31 4E 61 2BAr8Id8fD1HXw1Na+  
0000003064 44 67 43 52 79 32 30 4E 31 35 77 42 62 4E 45dDgCRy20N15wBbNE  
000000400A 59 74 7A 4F 4C 70 6B 7A 53 71 55 62 57 75 53.Ytz0LpkzSqUbWuS  
000000506C 42 58 71 51 65 44 54 4C 72 44 31 48 50 45 62lBXqQeDTLrD1HPEb  
0000006046 44 7A 67 45 2F 39 36 72 42 6D 74 54 35 4C 67FDzgE/96rBmtT5Lg  
0000007059 74 4B 30 56 6D 50 76 36 4C 37 6E 77 57 62 61YtK0VmPv6L7nwWba  
000000807A 0A 67 49 64 33 2B 2B 4B 52 6C 71 42 79 33 72z.gId3++KRLqBy3r  
000000906B 31 47 64 47 47 76 46 68 62 79 55 38 49 39 71k1GdGGvFhbyU8I9q

After decrypting this corrupted file:

```
tanvi@tanvi:~$ openssl aes-128-ofb -salt -a -p -d -in encofb.txt -out decofb.txt
enter aes-128-ofb decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=FD906FCEBAB89090
key=AE03AE3EF1070FF5E9B42A7730FC40D8
iv =2D3024B7D0E5C9DBBD378FF09053A6A9
tanvi@tanvi:~$
```

```
1 Lorem! ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
```

## Conclusion:

- In ecb mode only one block containing the corrupted byte is corrupted as in ecb mode each block gets encrypted and decrypted separately.
- In cbc mode two block are corrupted due to the chaining between input and output, as the block of plain text is XORed with the encrypted block of the previous pass.
- In cfb mode two block are corrupted as similar to cbc this mode uses feedback.
- In ofb mode only one bit is corrupted as this mode uses the feedback before it is XORed.

#### 4) Task 4 :

Here I am using two text files namely 32oct.txt and 20oct.txt containing 32 and 20 bytes each.

```
tanvi@tanvi:~/padding$ ls -l
total 8
-rw-rw-r-- 1 tanvi tanvi 20 Nov 18 21:25 20oct.txt
-rw-rw-r-- 1 tanvi tanvi 32 Nov 18 21:25 32oct.txt
tanvi@tanvi:~/padding$
```

Encrypting the above two text files using **aes-256** in the 4 modes: ecb, cbc, ocb and cfb

#### 20oct.txt

```
tanvi@tanvi:~/padding$ openssl aes-128-ecb -salt -a -p -e -in 20oct.txt -out ec
b20.txt
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=5C48E7B8E1921180
key=68A7A7D639FB06F701E354C2C4ECFF14
tanvi@tanvi:~/padding$ openssl aes-128-cbc -salt -a -p -e -in 20oct.txt -out cb
c20.txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=A6585BE98723BF34
key=F04DB99229AA6124EBA0D4AA33D8FDF8
iv =A87F10CE886DA4639E74B55589BC1DD8
tanvi@tanvi:~/padding$ openssl aes-128-cfb -salt -a -p -e -in 20oct.txt -out cf
b20.txt
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.
salt=8E0152ED710FF0BD
key=028D3E7CDFA0951C13E43E95924B155E
iv =1BCA184D22B4B08284A497F44BD46359
```

```
tanvi@tanvi:~/padding$ openssl aes-128-ofb -salt -a -p -e -in 20oct.txt -out of  
b20.txt  
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.  
salt=995A474C8D986766  
key=E4944CA5C52106B166895133745EE7F0  
iv =53CCC89803578CF5E68FB464DE0C7C5D  
tanvi@tanvi:~/padding$
```

## 32oct.txt

```
tanvi@tanvi:~/padding$ openssl aes-128-ecb -salt -a -p -e -in 32oct.txt -out ec  
b32.txt  
enter aes-128-ecb encryption password:  
Verifying - enter aes-128-ecb encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
salt=8157ED2732467A13  
key=F7CEFAC16F58C6E0B9203DBF0071CD31  
tanvi@tanvi:~/padding$ openssl aes-128-cbc -salt -a -p -e -in 32oct.txt -out cb  
c32.txt  
enter aes-128-cbc encryption password:  
Verifying - enter aes-128-cbc encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
salt=CCEF580635C91EED  
key=5CB7B02673BEB11BE8DB9DD519B01C4E  
iv =744DDEFAD27553ABD5E2B02A02B72D88  
tanvi@tanvi:~/padding$ openssl aes-128-cfb -salt -a -p -e -in 32oct.txt -out cf  
b32.txt  
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.  
salt=27B43EE9F12177DD  
key=80542BB7A3BF074D695C94B0A6EE8DD5  
iv =9C804CA5F90725114D7A29E89977CB0F
```

```
tanvi@tanvi:~/padding$ openssl aes-128-ofb -salt -a -p -e -in 32oct.txt -out of  
b32.txt  
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.  
salt=0E2AD8885A4E8EB6  
key=1DE4E621829E05F4F51263A73DDB0A3A  
iv =3C57F0FCDBE97ADCA8D069A5BC6E02BB  
tanvi@tanvi:~/padding$
```

After encrypting both the files through all the modes:



A terminal window titled "tanvi@tanvi: ~/padding". The command "ls -l" is run, showing the following file list:

```
total 40  
-rw-rw-r-- 1 tanvi tanvi 20 Nov 18 21:25 20oct.txt  
-rw-rw-r-- 1 tanvi tanvi 32 Nov 18 21:25 32oct.txt  
-rw-rw-r-- 1 tanvi tanvi 65 Nov 18 21:44 cbc20.txt  
-rw-rw-r-- 1 tanvi tanvi 90 Nov 18 22:22 cbc32.txt  
-rw-rw-r-- 1 tanvi tanvi 49 Nov 18 21:45 cfb20.txt  
-rw-rw-r-- 1 tanvi tanvi 65 Nov 18 22:24 cfb32.txt  
-rw-rw-r-- 1 tanvi tanvi 65 Nov 18 21:44 ecb20.txt  
-rw-rw-r-- 1 tanvi tanvi 90 Nov 18 22:22 ecb32.txt  
-rw-rw-r-- 1 tanvi tanvi 49 Nov 18 21:46 ofb20.txt  
-rw-rw-r-- 1 tanvi tanvi 65 Nov 18 22:25 ofb32.txt
```

To check whether padding occurred we will decrypt the file using the `-nopad` parameter, it turns off the standard block padding.

## 20oct.txt

```
tanvi@tanvi:~/padding$ openssl aes-128-ecb -salt -nopad -a -p -d -in ecb20.txt
-out ecb20_d.txt
enter aes-128-ecb decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=5C48E7B8E1921180
key=68A7A7D639FB06F701E354C2C4ECFF14
tanvi@tanvi:~/padding$ openssl aes-128-cbc -salt -nopad -a -p -d -in cbc20.txt
-out cbc20_d.txt
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=A6585BE98723BF34
key=F04DB99229AA6124EBA0D4AA33D8FDF8
iv =A87F10CE886DA4639E74B55589BC1DD8
tanvi@tanvi:~/padding$ openssl aes-128-cfb -salt -nopad -a -p -d -in cfb20.txt
-out cfb20_d.txt
enter aes-128-cfb decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=8862674BE1AFCF28
key=3BF683497A9AA1D6DF947D623F53E9B5
iv =52298BA95E2DAA0819748DFE985E8244
tanvi@tanvi:~/padding$ openssl aes-128-ofb -salt -nopad -a -p -d -in ofb20.txt
-out ofb20_d.txt
enter aes-128-ofb decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=995A474C8D986766
key=E4944CA5C52106B166895133745EE7F0
iv =53CCC89803578CF5E68FB464DE0C7C5D
```

## 32oct.txt

```
tanvi@tanvi:~/padding$ openssl aes-128-ecb -salt -nopad -a -p -d -in ecb32.txt  
-out ecb32_d.txt  
enter aes-128-ecb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
salt=8157ED2732467A13  
key=F7CEFAC16F58C6E0B9203DBF0071CD31  
tanvi@tanvi:~/padding$ openssl aes-128-cbc -salt -nopad -a -p -d -in cbc32.txt  
-out cbc32_d.txt  
enter aes-128-cbc decryption password:  
* [WARNING] : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
salt=CCEF580635C91EED  
key=5CB7B02673BEB11BE8DB9DD519B01C4E  
iv =744DDEFAD27553ABD5E2B02A02B72D88  
tanvi@tanvi:~/padding$ openssl aes-128-cfb -salt -nopad -a -p -d -in cfb32.txt  
-out cfb32_d.txt  
enter aes-128-cfb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
salt=27B43EE9F12177DD  
key=80542BB7A3BF074D695C94B0A6EE8DD5  
iv =9C804CA5F90725114D7A29E89977CB0F  
tanvi@tanvi:~/padding$ openssl aes-128-ofb -salt -nopad -a -p -d -in ofb32.txt  
-out ofb32_d.txt  
enter aes-128-ofb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
salt=0E2AD8885A4E8EB6  
key=1DE4E621829E05F4F51263A73DDB0A3A  
iv =3C57F0FCDBE97ADCA8D069A5BC6E02BB  
tanvi@tanvi:~/padding$ █
```

After decrypting both the files through all the modes using the nopad parameter:

```
tanvi@tanvi:~/padding$ ls -l
total 72
-rw-rw-r-- 1 tanvi tanvi 20 Nov 18 21:25 20oct.txt
-rw-rw-r-- 1 tanvi tanvi 32 Nov 18 21:25 32oct.txt
-rw-rw-r-- 1 tanvi tanvi 32 Nov 18 22:31 cbc20_d.txt
-rw-rw-r-- 1 tanvi tanvi 65 Nov 18 21:44 cbc20.txt
-rw-rw-r-- 1 tanvi tanvi 48 Nov 18 22:34 cbc32_d.txt
-rw-rw-r-- 1 tanvi tanvi 90 Nov 18 22:22 cbc32.txt
-rw-rw-r-- 1 tanvi tanvi 20 Nov 18 22:31 cfb20_d.txt
-rw-rw-r-- 1 tanvi tanvi 49 Nov 18 21:45 cfb20.txt
-rw-rw-r-- 1 tanvi tanvi 32 Nov 18 22:34 cfb32_d.txt
-rw-rw-r-- 1 tanvi tanvi 65 Nov 18 22:24 cfb32.txt
-rw-rw-r-- 1 tanvi tanvi 32 Nov 18 22:30 ecb20_d.txt
-rw-rw-r-- 1 tanvi tanvi 65 Nov 18 21:44 ecb20.txt
-rw-rw-r-- 1 tanvi tanvi 48 Nov 18 22:33 ecb32_d.txt
-rw-rw-r-- 1 tanvi tanvi 90 Nov 18 22:22 ecb32.txt
-rw-rw-r-- 1 tanvi tanvi 20 Nov 18 22:32 ofb20_d.txt
-rw-rw-r-- 1 tanvi tanvi 49 Nov 18 21:46 ofb20.txt
-rw-rw-r-- 1 tanvi tanvi 32 Nov 18 22:34 ofb32_d.txt
-rw-rw-r-- 1 tanvi tanvi 65 Nov 18 22:25 ofb32.txt
tanvi@tanvi:~/padding$
```

## Conclusion:

- It was observed that in case of the cfb and ofb modes no padding was required as they are stream ciphers.
- While in case of ecb and cbc modes the padding was required of 12 bytes for the 20 byte file and 16 bytes for the 32 byte file, because these are block cipher and the size of the plaintext should be an exact multiple of the block size.

## 5) Task 5 :

Code:

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
plaintxt = b"This is a top secret."
ciphertext = "8d20e5056a8d24d0462ce74e4904c1b513e10d1df4a2ef2ad4540fae1ca0aaaf9"
result = []
file = open('words.txt', 'r')
lines = file.readlines()
words = [str.strip(line) for line in lines]
for word in words:
    if len(word) >= 16:
        continue
    word = word.lower()
    key = word.encode() + b' '*(16-len(word))
    cipher = AES.new(key, AES.MODE_CBC, iv=bytes.fromhex('0'*32))
    ciphertext = cipher.encrypt(pad(plaintxt, AES.block_size))
    match = "NO MATCH"
    if bytes.hex(ciphertext) == ciphertext:
        match = "MATCH"
        result.append(word)
    bytes.hex(ciphertext)
    print(word,match)
print("\n\nResulting Key:",result)
```

## Output:

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
zunis NO MATCH
zupanate NO MATCH
zupus NO MATCH
zurbar NO MATCH
zurbaran NO MATCH
zurek NO MATCH
zurheide NO MATCH
zurich NO MATCH
zurkow NO MATCH
zurlite NO MATCH
zurn NO MATCH
zurvan NO MATCH
zusman NO MATCH
zutugil NO MATCH
zuurveldt NO MATCH
zuza NO MATCH
zuzana NO MATCH
zu-zu NO MATCH
zwanziger NO MATCH
zwart NO MATCH
zwei NO MATCH
zweig NO MATCH
zwick NO MATCH
zwickau NO MATCH
zwicky NO MATCH
zwieback NO MATCH
zwiebacks NO MATCH
zwiebel NO MATCH
zwieselite NO MATCH
zwingle NO MATCH
zwingli NO MATCH
zwinglian NO MATCH
zwinglianism NO MATCH
zwinglianist NO MATCH
zwitter NO MATCH
zwitterion NO MATCH
zwitterionic NO MATCH
zwolle NO MATCH
zworykin NO MATCH
zz NO MATCH
zzt NO MATCH
zzz NO MATCH
```

Resulting Key: ['median']

PS C:\Users\manal\Desktop\Tanvi\Sem5\CCS> █

PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

```
medevac NO MATCH
medevacs NO MATCH
medfield NO MATCH
medfly NO MATCH
medflies NO MATCH
medford NO MATCH
medi- NO MATCH
media NO MATCH
mediacy NO MATCH
mediacid NO MATCH
mediacies NO MATCH
mediad NO MATCH
mediae NO MATCH
mediaeval NO MATCH
mediaevalism NO MATCH
mediaevalist NO MATCH
mediaevalize NO MATCH
mediaevally NO MATCH
medial NO MATCH
medialization NO MATCH
medialize NO MATCH
medalkaline NO MATCH
medially NO MATCH
medials NO MATCH
median MATCH
medianic NO MATCH
medianimic NO MATCH
medianimity NO MATCH
medianism NO MATCH
medianity NO MATCH
medianly NO MATCH
medians NO MATCH
median's NO MATCH
mediant NO MATCH
mediants NO MATCH
mediapolis NO MATCH
mediary NO MATCH
medias NO MATCH
mediastina NO MATCH
mediastinal NO MATCH
mediastine NO MATCH
```

## Conclusion:

- Here it is observed that if plain text, cipher text and iv is known then by using the brute force approach we can find the key.

Github: <https://github.com/tanvipen/CSS-Lab>