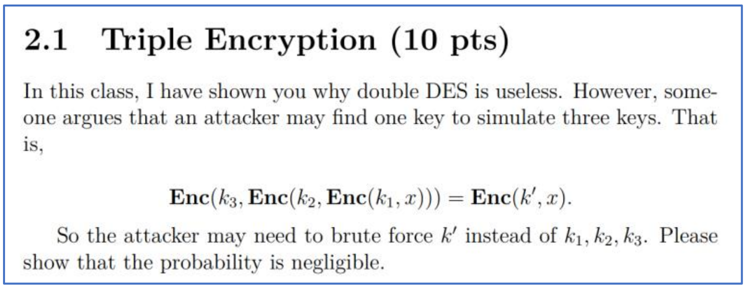


2.1 Triple Encryption (10 pts)

In this class, I have shown you why double DES is useless. However, someone argues that an attacker may find one key to simulate three keys. That is,

$$\mathbf{Enc}(k_3, \mathbf{Enc}(k_2, \mathbf{Enc}(k_1, x))) = \mathbf{Enc}(k', x).$$

So the attacker may need to brute force k' instead of k_1, k_2, k_3. Please show that the probability is negligible.

3DES: k1 加密 -> k2 解密 -> k3 加密

DES 加密算法使用 64 位元的區塊大小和 56 位元的金鑰。在一個金鑰空間中，有 2^{56} 個可能的金鑰，3DES 用三個金鑰，所以是 $2^{56*3} = 2^{168}$ 種組合

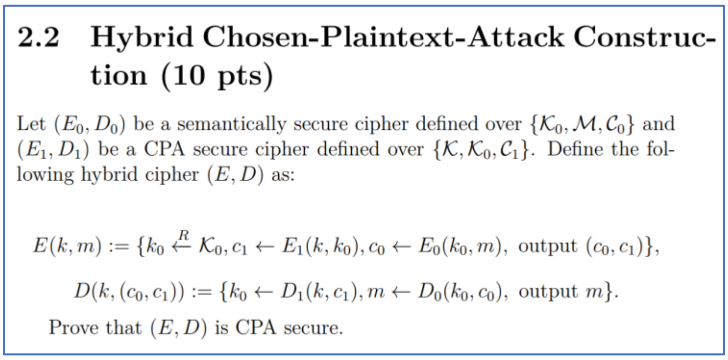
讓 3DES 等同於只用一把金鑰的可能情況:

　　情況 1. 如果 k1=k2，相當於只用 k3 加密

　　情況 2. 如果 k2=k3，相當於只用 k1 加密

如果 k1, k2, k3 是各自獨立選擇，情況 1 和情況 2 的機率就都是 $1/2^{56*2} = 1/2^{112}$

總體發生的機率是 $2/2^{112}$，小於 $1/2^{88}$，所以可以忽略



2.2 Hybrid Chosen-Plaintext-Attack Construction (10 pts)

Let (E_0, D_0) be a semantically secure cipher defined over $\{\mathcal{K}_0, \mathcal{M}, \mathcal{C}_0\}$ and (E_1, D_1) be a CPA secure cipher defined over $\{\mathcal{K}, \mathcal{K}_0, \mathcal{C}_1\}$. Define the following hybrid cipher (E, D) as:

$$E(k, m) := \{k_0 \xleftarrow{R} \mathcal{K}_0, c_1 \leftarrow E_1(k, k_0), c_0 \leftarrow E_0(k_0, m), \text{ output } (c_0, c_1)\},$$

$$D(k, (c_0, c_1)) := \{k_0 \leftarrow D_1(k, c_1), m \leftarrow D_0(k_0, c_0), \text{ output } m\}.$$

Prove that (E, D) is CPA secure.

CPA 安全性要求對於任何給定的兩個 m_0 和 m_1，如果攻擊者只能進行加密和解密操作，那麼他在區分兩個 c_0 和 c_1 的能力應該是可忽略的

用反證法證明(E,D)是 CPA 安全：

設(E,D)不安全(存在一個 CPA 攻擊者可以在可忽略的時間內區分兩個密文)

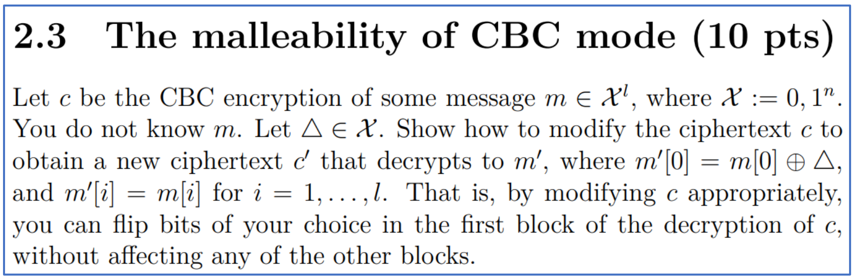
建構攻擊者 A’，它是兩個子加密方案 E_0 和 E_1 的攻擊者

用 A 的結果區分(E,D)的密文 -> 違反(E_0,D_0)和(E_1,D_1)的安全性假設

由於 A’和 A 的時間複雜度相同，且(E_0,D_0)和(E_1,D_1)是安全的

所以 A’也無法區分(E,D)的密文

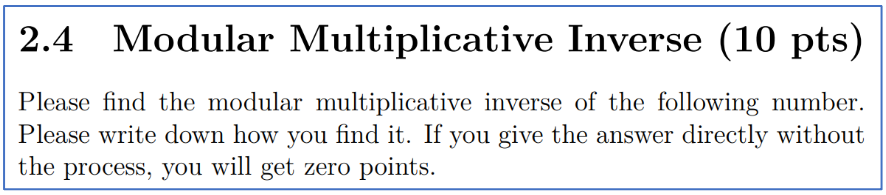
因此，如果(E_0,D_0)和(E_1,D_1)抗 CPA 攻擊，那(E,D)也是 CPA 安全



2.3 The malleability of CBC mode (10 pts)

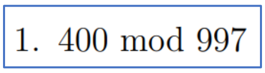
Let c be the CBC encryption of some message $m \in \mathcal{X}^l$, where $\mathcal{X} := 0, 1^n$. You do not know m. Let $\triangle \in \mathcal{X}$. Show how to modify the ciphertext c to obtain a new ciphertext c' that decrypts to m', where $m'[0] = m[0] \oplus \triangle$, and $m'[i] = m[i]$ for $i = 1, \ldots, l$. That is, by modifying c appropriately, you can flip bits of your choice in the first block of the decryption of c, without affecting any of the other blocks.

1. 解密 c，得到 m_0, m_1, m_2,…,m_n
2. 修改 m_0 的位元，使 $m_0[0] = m[0] \oplus \Delta$
3. 將修改後的 m_0 和原本的 m_1, m_2,…,m_n 重新組合成新的 m’
4. 將 m’進行 CBC 加密，得到 c’



2.4 Modular Multiplicative Inverse (10 pts)

Please find the modular multiplicative inverse of the following number. Please write down how you find it. If you give the answer directly without the process, you will get zero points.



1. 400 mod 997

找到 b 使得 400b≡1(mod997)

997 = 2*400 + 197

400 = 2*197 + 6

197 = 32*6 + 5

6 = 1*5 + 1

反回去

1 = 6 – 1*5

= 6 – 1*(197 – 32*6)

= 33*6 – 1*197

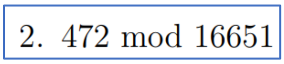
= 33*(400 – 2*197) – 1*197

= 33*400 – 67*197

= 33*400 – 67*(997 – 2*400)

= 167*400 – 67*997

b = 167



2. 472 mod 16651

找到 b 使得 472b≡1(mod16651)

16651 =35*472+431

472 = 1*431 + 41

431 = 10*41 + 21

41 = 1*21 + 20

21 = 1*20 + 1

反回去

1 = 21 – 1*20

= 21 – 1*(41 – 1*21)

= 2*21 – 1*41

= 2*(431 – 10*41) – 1*41

= 2*431 – 21*41

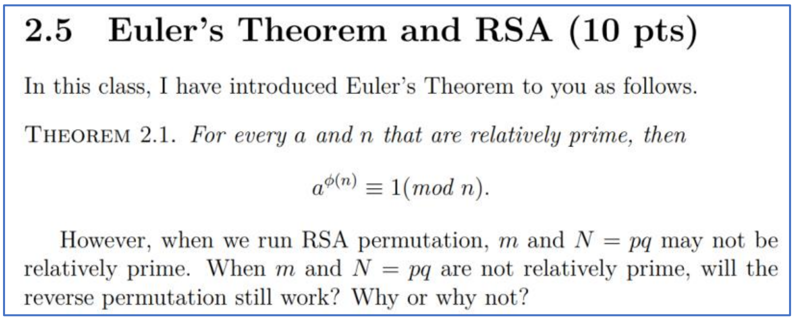
= 2*431 – 21*(472 – 1*431)

= 23*431 – 21*472

= 23*(16651 – 35*472) – 21*472

= −803*472 + 23*16651

b = −803 但要找正的，所以是 b = 16651−803 = 15848



2.5 Euler's Theorem and RSA (10 pts)

In this class, I have introduced Euler's Theorem to you as follows.

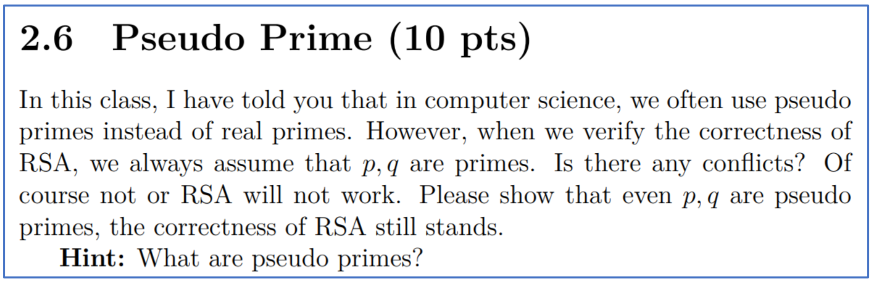
THEOREM 2.1. *For every a and n that are relatively prime, then*

$$a^{\phi(n)} \equiv 1(mod\ n).$$

However, when we run RSA permutation, m and $N = pq$ may not be relatively prime. When m and $N = pq$ are not relatively prime, will the reverse permutation still work? Why or why not?

在 RSA 加密中，我們選擇兩個大質數 p 和 q，然後計算它們的乘積 N = pq 作為模數，當 m 和 N 不互質(m 不是 N 的倍數)，Euler's Theorem 不再適用

因為 Euler's Theorem 僅適用於互質的 a 和 n，這樣才能確保 $a^{\varphi(n)} \equiv 1(\text{mod } n)$，當 m 和 N 不互質時，m 就不會滿足 Euler's Theorem 的條件，因此反向置換不正確



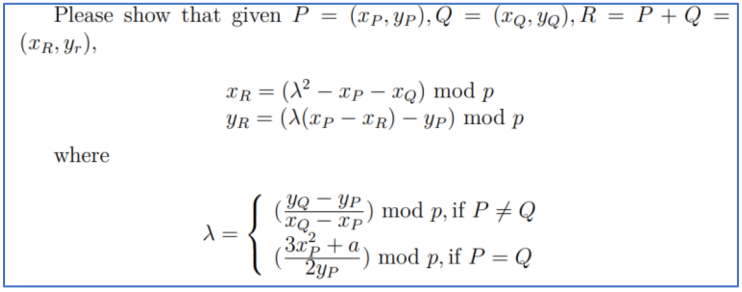
2.6 Pseudo Prime (10 pts)

In this class, I have told you that in computer science, we often use pseudo primes instead of real primes. However, when we verify the correctness of RSA, we always assume that p, q are primes. Is there any conflicts? Of course not or RSA will not work. Please show that even p, q are pseudo primes, the correctness of RSA still stands.

Hint: What are pseudo primes?

因為 RSA 難解的原因是 N = pq 的 p 和 q 不好找，不是因為其他特殊的質數性質，所以只要 p 和 q 夠大，加上 N 的因數分解夠困難，就會讓 RSA 達到安全的效果。

2.7 Elliptic Curve over $\mathbb{Z}_p$ (10 pts)

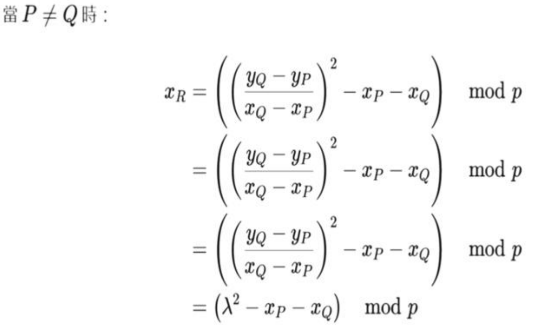


Please show that given $P = (x_P, y_P), Q = (x_Q, y_Q), R = P + Q = (x_R, y_r)$,

$$x_R = (\lambda^2 - x_P - x_Q) \bmod p$$
$$y_R = (\lambda(x_P - x_R) - y_P) \bmod p$$

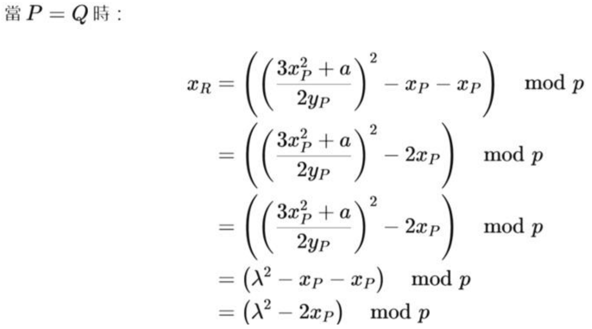
where

$$\lambda = \begin{cases} (\frac{y_Q - y_P}{x_Q - x_P}) \bmod p, \text{if } P \neq Q \\ (\frac{3x_P^2 + a}{2y_P}) \bmod p, \text{if } P = Q \end{cases}$$



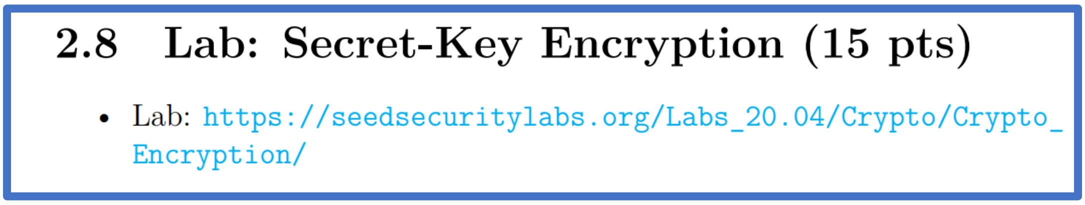
當 $P \neq Q$ 時：

$$\begin{aligned} x_R &= \left(\left(\frac{y_Q - y_P}{x_Q - x_P}\right)^2 - x_P - x_Q\right) \quad \bmod p \\ &= \left(\left(\frac{y_Q - y_P}{x_Q - x_P}\right)^2 - x_P - x_Q\right) \quad \bmod p \\ &= \left(\left(\frac{y_Q - y_P}{x_Q - x_P}\right)^2 - x_P - x_Q\right) \quad \bmod p \\ &= (\lambda^2 - x_P - x_Q) \quad \bmod p \end{aligned}$$



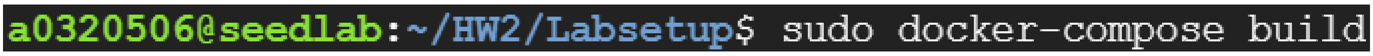
當 $P = Q$ 時：

$$\begin{aligned} x_R &= \left(\left(\frac{3x_P^2 + a}{2y_P}\right)^2 - x_P - x_P\right) \quad \bmod p \\ &= \left(\left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P\right) \quad \bmod p \\ &= \left(\left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P\right) \quad \bmod p \\ &= (\lambda^2 - x_P - x_P) \quad \bmod p \\ &= (\lambda^2 - 2x_P) \quad \bmod p \end{aligned}$$

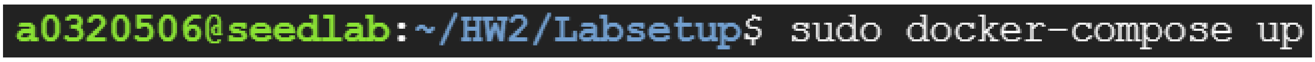


2.8 Lab: Secret-Key Encryption (15 pts)

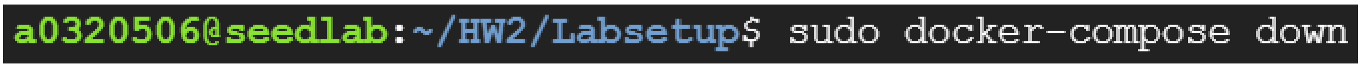
- Lab: https://seedsecuritylabs.org/Labs_20.04/Crypto/Crypto_Encryption/



```
a0320506@seedlab:~/HW2/Labsetup$ sudo docker-compose build
```

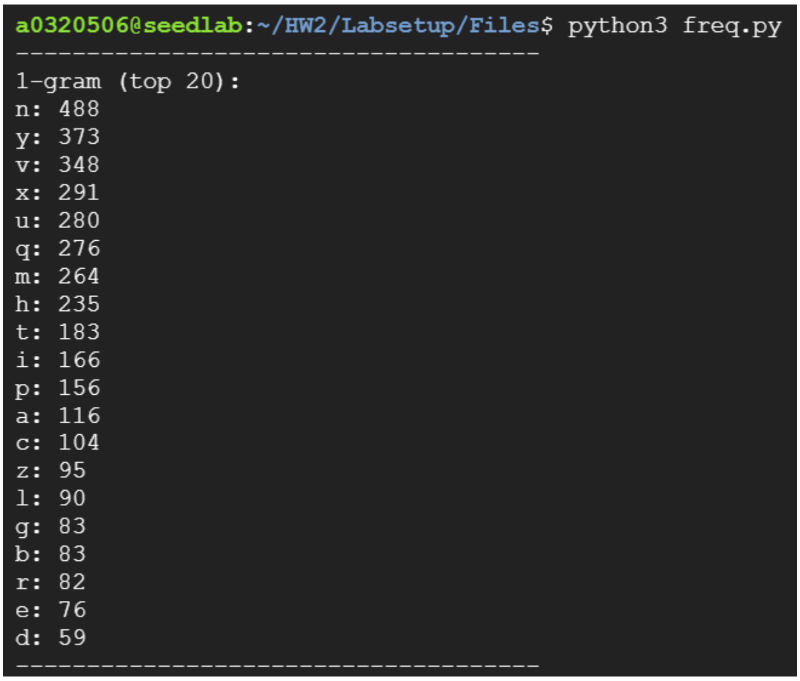


```
a0320506@seedlab:~/HW2/Labsetup$ sudo docker-compose up
```

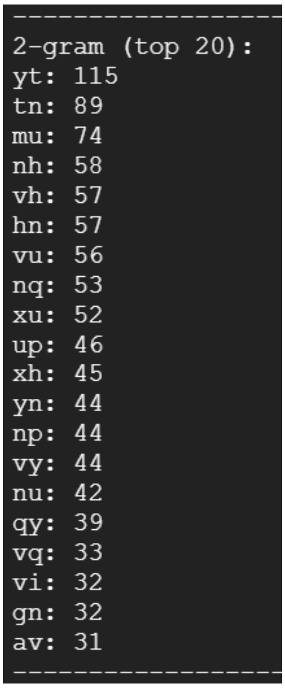


```
a0320506@seedlab:~/HW2/Labsetup$ sudo docker-compose down
```

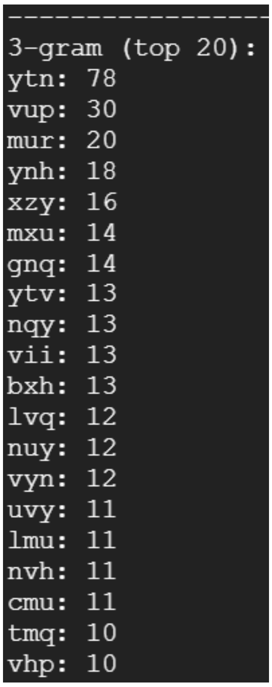
3 Task 1: Frequency Analysis



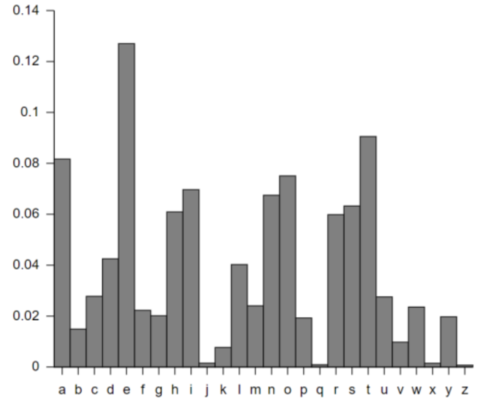
```
a0320506@seedlab:~/HW2/Labsetup/Files$ python3 freq.py
---------------------------------------
1-gram (top 20):
n: 488
y: 373
v: 348
x: 291
u: 280
q: 276
m: 264
h: 235
t: 183
i: 166
p: 156
a: 116
c: 104
z: 95
l: 90
g: 83
b: 83
r: 82
e: 76
d: 59
---------------------------------------
```

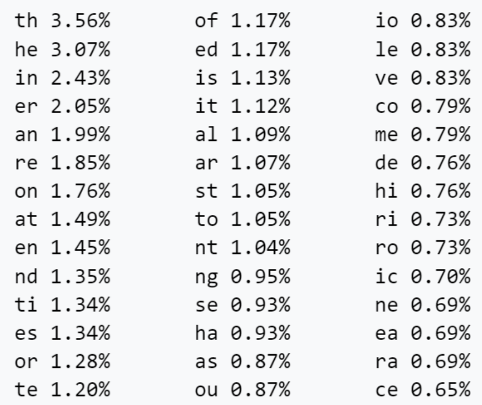


```
--------------------
2-gram (top 20):
yt: 115
tn: 89
mu: 74
nh: 58
vh: 57
hn: 57
vu: 56
nq: 53
xu: 52
up: 46
xh: 45
yn: 44
np: 44
vy: 44
nu: 42
qy: 39
vq: 33
vi: 32
gn: 32
av: 31
--------------------
```

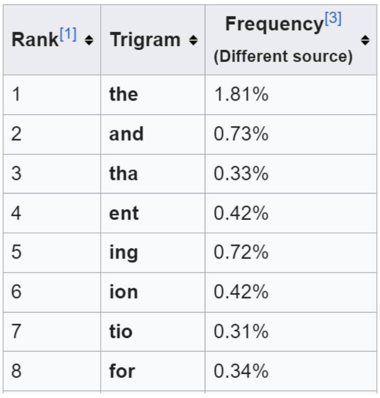


```
--------------------
3-gram (top 20):
ytn: 78
vup: 30
mur: 20
ynh: 18
xzy: 16
mxu: 14
gnq: 14
ytv: 13
nqy: 13
vii: 13
bxh: 13
lvq: 12
nuy: 12
vyn: 12
uvy: 11
lmu: 11
nvh: 11
cmu: 11
tmq: 10
vhp: 10
```





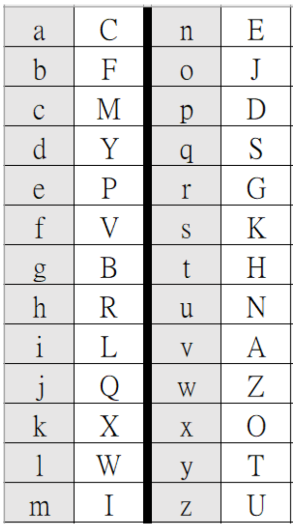
```
th 3.56%     of 1.17%     io 0.83%
he 3.07%     ed 1.17%     le 0.83%
in 2.43%     is 1.13%     ve 0.83%
er 2.05%     it 1.12%     co 0.79%
an 1.99%     al 1.09%     me 0.79%
re 1.85%     ar 1.07%     de 0.76%
on 1.76%     st 1.05%     hi 0.76%
at 1.49%     to 1.05%     ri 0.73%
en 1.45%     nt 1.04%     ro 0.73%
nd 1.35%     ng 0.95%     ic 0.70%
ti 1.34%     se 0.93%     ne 0.69%
es 1.34%     ha 0.93%     ea 0.69%
or 1.28%     as 0.87%     ra 0.69%
te 1.20%     ou 0.87%     ce 0.65%
```



Rank[1] ⬍	Trigram ⬍	Frequency[3] (Different source) ⬍
1	the	1.81%
2	and	0.73%
3	tha	0.33%
4	ent	0.42%
5	ing	0.72%
6	ion	0.42%
7	tio	0.31%
8	for	0.34%

解密過程

```
sudo tr ’mnrtuvy’ ’IEGHNAT’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvybp’ ’IEGHNATSD’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvybpx’ ’IEGHNATSDO’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvybpxh’ ’IEGHNATSDOL’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxh’ ’IEGHNATSDOL’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhb’ ’IEGHNATSDOLF’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhb’ ’IEGHNATSDORF’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbi’ ’IEGHNATSDORFL’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbiz’ ’IEGHNATSDORFLU’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizg’ ’IEGHNATSDORFLUB’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgd’ ’IEGHNATSDORFLUBY’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgds’ ’IEGHNATSDORFLUBYK’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgdsl’ ’IEGHNATSDORFLUBYKW’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgdslc’ ’IEGHNATSDORFLUBYKWC’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgdsla’ ’IEGHNATSDORFLUBYKWC’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgdslac’ ’IEGHNATSDORFLUBYKWCM’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgdslace’ ’IEGHNATSDORFLUBYKWCMP’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgdslacef’ ’IEGHNATSDORFLUBYKWCMPV’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgdslacefj’ ’IEGHNATSDORFLUBYKWCMPVQ’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgdslacefjw’ ’IEGHNATSDORFLUBYKWCMPVQM’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgdslacefjwo’ ’IEGHNATSDORFLUBYKWCMPVQMJ’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgdslacefjwok’ ’IEGHNATSDORFLUBYKWCMPVQMJX’ <ciphertext.txt> message.txt
sudo tr ’mnrtuvyqpxhbizgdslacefjwok’ ’IEGHNATSDORFLUBYKWCMPVQZJX’ <ciphertext.txt> message.txt
```



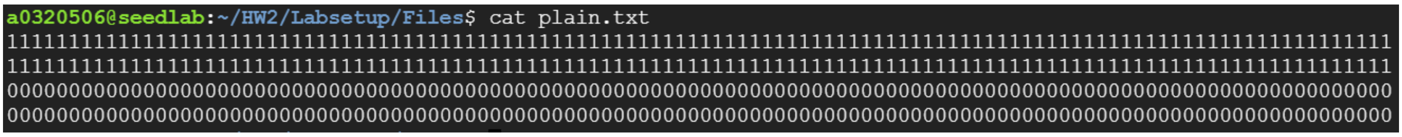
a	C	n	E
b	F	o	J
c	M	p	D
d	Y	q	S
e	P	r	G
f	V	s	K
g	B	t	H
h	R	u	N
i	L	v	A
j	Q	w	Z
k	X	x	O
l	W	y	T
m	I	z	U

整理後的對照表

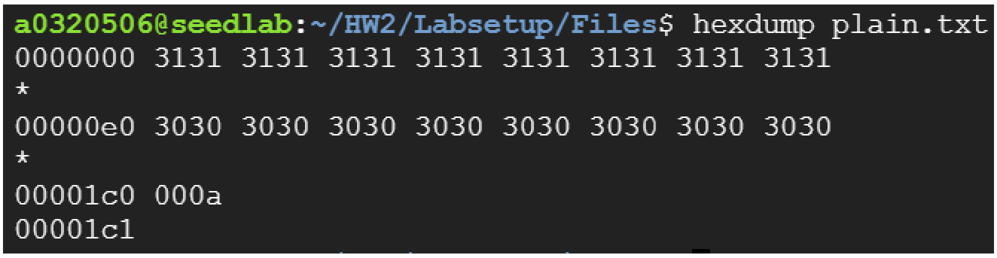
abcdefghijklmnopqrstuvwxyz -> cfmypvbrlqxwiejdsgkhnazotu

解密結果在 message.txt

Task 2: Encryption using Different Ciphers and Modes

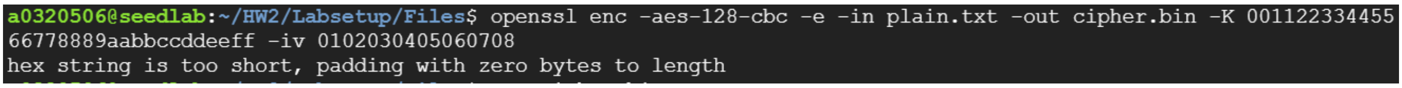


```
a0320506@seedlab:~/HW2/Labsetup/Files$ cat plain.txt
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

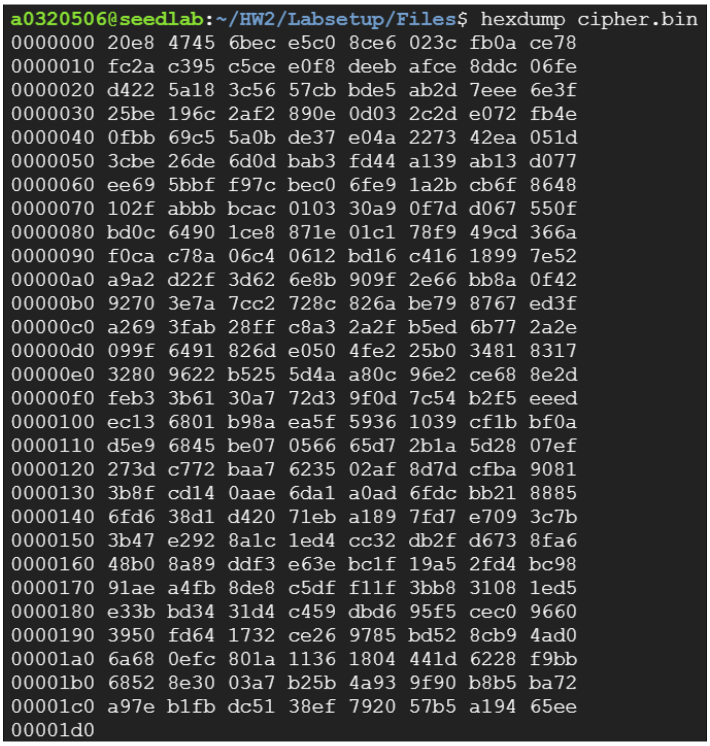


```
a0320506@seedlab:~/HW2/Labsetup/Files$ hexdump plain.txt
0000000 3131 3131 3131 3131 3131 3131 3131 3131
*
00000e0 3030 3030 3030 3030 3030 3030 3030 3030
*
00001c0 000a
00001c1
```

openssl enc -aes-128-cbc -e -in plain.txt -out cipher.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708



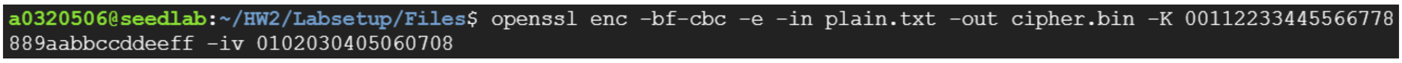
```
a0320506@seedlab:~/HW2/Labsetup/Files$ openssl enc -aes-128-cbc -e -in plain.txt -out cipher.bin -K 001122334455
66778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
```



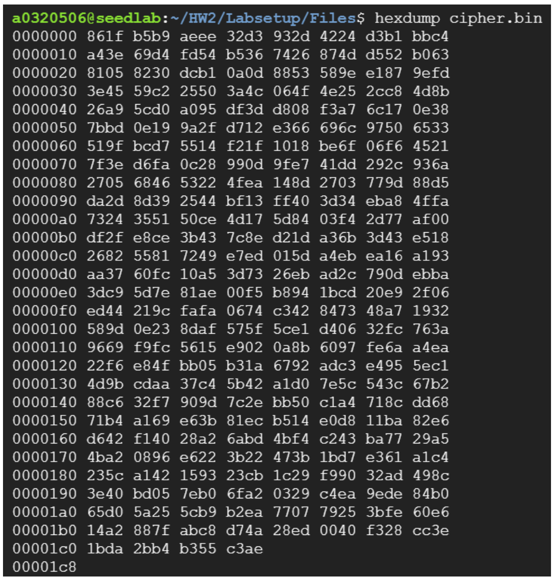
```
a0320506@seedlab:~/HW2/Labsetup/Files$ hexdump cipher.bin
0000000 20e8 4745 6bec e5c0 8ce6 023c fb0a ce78
0000010 fc2a c395 c5ce e0f8 deeb afce 8ddc 06fe
0000020 d422 5a18 3c56 57cb bde5 ab2d 7eee 6e3f
0000030 25be 196c 2af2 890e 0d03 2c2d e072 fb4e
0000040 0fbb 69c5 5a0b de37 e04a 2273 42ea 051d
0000050 3cbe 26de 6d0d bab3 fd44 a139 ab13 d077
0000060 ee69 5bbf f97c bec0 6fe9 1a2b cb6f 8648
0000070 102f abbb bcac 0103 30a9 0f7d d067 550f
0000080 bd0c 6490 1ce8 871e 01c1 78f9 49cd 366a
0000090 f0ca c78a 06c4 0612 bd16 c416 1899 7e52
00000a0 a9a2 d22f 3d62 6e8b 909f 2e66 bb8a 0f42
00000b0 9270 3e7a 7cc2 728c 826a be79 8767 ed3f
00000c0 a269 3fab 28ff c8a3 2a2f b5ed 6b77 2a2e
00000d0 099f 6491 826d e050 4fe2 25b0 3481 8317
00000e0 3280 9622 b525 5d4a a80c 96e2 ce68 8e2d
00000f0 feb3 3b61 30a7 72d3 9f0d 7c54 b2f5 eeed
0000100 ec13 6801 b98a ea5f 5936 1039 cf1b bf0a
0000110 d5e9 6845 be07 0566 65d7 2b1a 5d28 07ef
0000120 273d c772 baa7 6235 02af 8d7d cfba 9081
0000130 3b8f cd14 0aae 6da1 a0ad 6fdc bb21 8885
0000140 6fd6 38d1 d420 71eb a189 7fd7 e709 3c7b
0000150 3b47 e292 8a1c 1ed4 cc32 db2f d673 8fa6
0000160 48b0 8a89 ddf3 e63e bc1f 19a5 2fd4 bc98
0000170 91ae a4fb 8de8 c5df f11f 3bb8 3108 1ed5
0000180 e33b bd34 31d4 c459 dbd6 95f5 cec0 9660
0000190 3950 fd64 1732 ce26 9785 bd52 8cb9 4ad0
00001a0 6a68 0efc 801a 1136 1804 441d 6228 f9bb
00001b0 6852 8e30 03a7 b25b 4a93 9f90 b8b5 ba72
00001c0 a97e b1fb dc51 38ef 7920 57b5 a194 65ee
00001d0
```

aes-128-cbc 加密結果

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



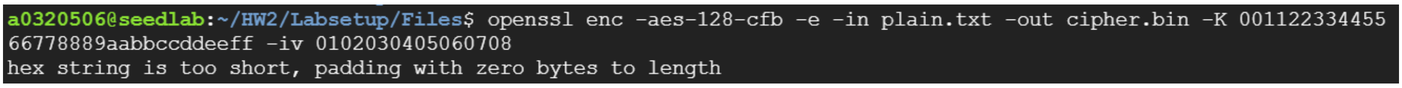
```
a0320506@seedlab:~/HW2/Labsetup/Files$ openssl enc -bf-cbc -e -in plain.txt -out cipher.bin -K 00112233445566778
889aabbccddeeff -iv 0102030405060708
```



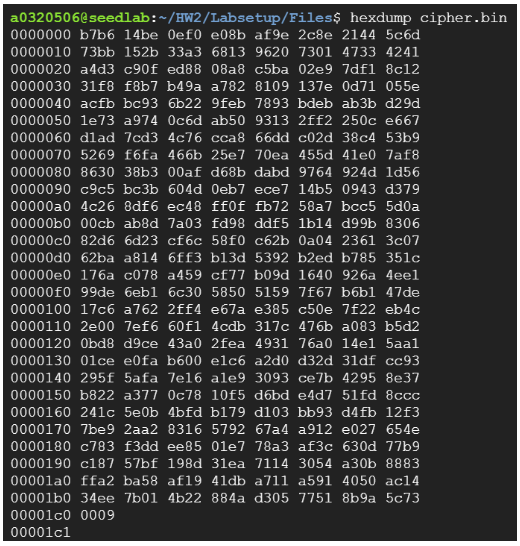
```
a0320506@seedlab:~/HW2/Labsetup/Files$ hexdump cipher.bin
0000000 861f b5b9 aeee 32d3 932d 4224 d3b1 bbc4
0000010 a43e 69d4 fd54 b536 7426 874d d552 b063
0000020 8105 8230 dcb1 0a0d 8853 589e e187 9efd
0000030 3e45 59c2 2550 3a4c 064f 4e25 2cc8 4d8b
0000040 26a9 5cd0 a095 df3d d808 f3a7 6c17 0e38
0000050 7bbd 0e19 9a2f d712 e366 696c 9750 6533
0000060 519f bcd7 5514 f21f 1018 be6f 06f6 4521
0000070 7f3e d6fa 0c28 990d 9fe7 41dd 292c 936a
0000080 2705 6846 5322 4fea 148d 2703 779d 88d5
0000090 da2d 8d39 2544 bf13 ff40 3d34 eba8 4ffa
00000a0 7324 3551 50ce 4d17 5d84 03f4 2d77 af00
00000b0 df2f e8ce 3b43 7c8e d21d a36b 3d43 e518
00000c0 2682 5581 7249 e7ed 015d a4eb ea16 a193
00000d0 aa37 60fc 10a5 3d73 26eb ad2c 790d ebba
00000e0 3dc9 5d7e 81ae 00f5 b894 1bcd 20e9 2f06
00000f0 ed44 219c fafa 0674 c342 8473 48a7 1932
0000100 589d 0e23 8daf 575f 5ce1 d406 32fc 763a
0000110 9669 f9fc 5615 e902 0a8b 6097 fe6a a4ea
0000120 22f6 e84f bb05 b31a 6792 adc3 e495 5ec1
0000130 4d9b cdaa 37c4 5b42 a1d0 7e5c 543c 67b2
0000140 88c6 32f7 909d 7c2e bb50 c1a4 718c dd68
0000150 71b4 a169 e63b 81ec b514 e0d8 11ba 82e6
0000160 d642 f140 28a2 6abd 4bf4 c243 ba77 29a5
0000170 4ba2 0896 e622 3b22 473b 1bd7 e361 a1c4
0000180 235c a142 1593 23cb 1c29 f990 32ad 498c
0000190 3e40 bd05 7eb0 6fa2 0329 c4ea 9ede 84b0
00001a0 65d0 5a25 5cb9 b2ea 7707 7925 3bfe 60e6
00001b0 14a2 887f abc8 d74a 28ed 0040 f328 cc3e
00001c0 1bda 2bb4 b355 c3ae
00001c8
```

bf-cbc 加密結果

openssl enc -aes-128-cfb -e -in plain.txt -out cipher.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708



```
a0320506@seedlab:~/HW2/Labsetup/Files$ openssl enc -aes-128-cfb -e -in plain.txt -out cipher.bin -K 001122334455
66778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
```

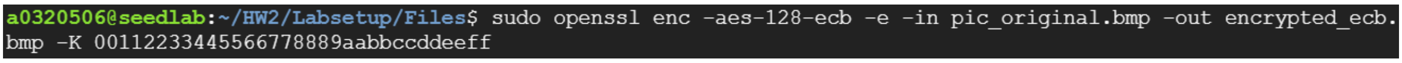


```
a0320506@seedlab:~/HW2/Labsetup/Files$ hexdump cipher.bin
0000000 b7b6 14be 0ef0 e08b af9e 2c8e 2144 5c6d
0000010 73bb 152b 33a3 6813 9620 7301 4733 4241
0000020 a4d3 c90f ed88 08a8 c5ba 02e9 7df1 8c12
0000030 31f8 f8b7 b49a a782 8109 137e 0d71 055e
0000040 acfb bc93 6b22 9feb 7893 bdeb ab3b d29d
0000050 1e73 a974 0c6d ab50 9313 2ff2 250c e667
0000060 d1ad 7cd3 4c76 cca8 66dd c02d 38c4 53b9
0000070 5269 f6fa 466b 25e7 70ea 455d 41e0 7af8
0000080 8630 38b3 00af d68b dabd 9764 924d 1d56
0000090 c9c5 bc3b 604d 0eb7 ece7 14b5 0943 d379
00000a0 4c26 8df6 ec48 ff0f fb72 58a7 bcc5 5d0a
00000b0 00cb ab8d 7a03 fd98 ddf5 1b14 d99b 8306
00000c0 82d6 6d23 cf6c 58f0 c62b 0a04 2361 3c07
00000d0 62ba a814 6ff3 b13d 5392 b2ed b785 351c
00000e0 176a c078 a459 cf77 b09d 1640 926a 4ee1
00000f0 99de 6eb1 6c30 5850 5159 7f67 b6b1 47de
0000100 17c6 a762 2ff4 e67a e385 c50e 7f22 eb4c
0000110 2e00 7ef6 60f1 4cdb 317c 476b a083 b5d2
0000120 0bd8 d9ce 43a0 2fea 4931 76a0 14e1 5aa1
0000130 01ce e0fa b600 e1c6 a2d0 d32d 31df cc93
0000140 295f 5afa 7e16 a1e9 3093 ce7b 4295 8e37
0000150 b822 a377 0c78 10f5 d6bd e4d7 51fd 8ccc
0000160 241c 5e0b 4bfd b179 d103 bb93 d4fb 12f3
0000170 7be9 2aa2 8316 5792 67a4 a912 e027 654e
0000180 c783 f3dd ee85 01e7 78a3 af3c 630d 77b9
0000190 c187 57bf 198d 31ea 7114 3054 a30b 8883
00001a0 ffa2 ba58 af19 41db a711 a591 4050 ac14
00001b0 34ee 7b01 4b22 884a d305 7751 8b9a 5c73
00001c0 0009
00001c1
```

aes-128-cfb 加密結果

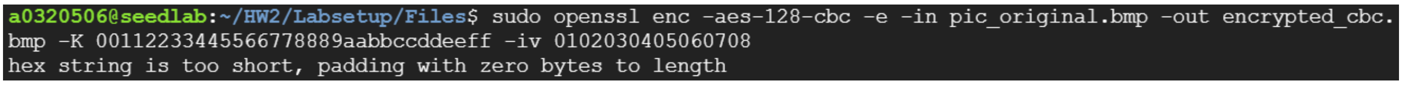
5 Task 3: Encryption Mode – ECB vs. CBC

openssl enc -aes-128-ecb -e -in pic_original.bmp -out encrypted_ecb.bmp -K 00112233445566778889aabbccddeeff



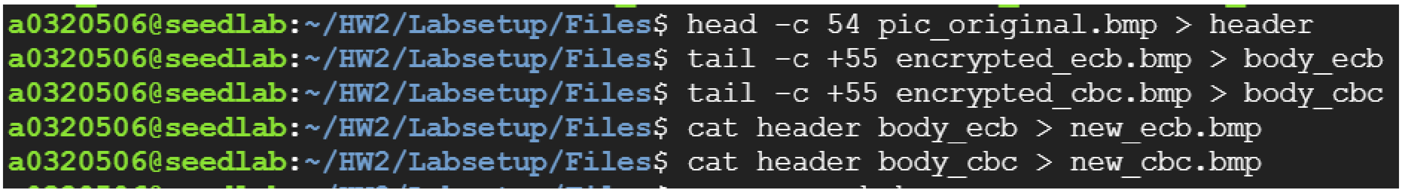
```
a0320506@seedlab:~/HW2/Labsetup/Files$ sudo openssl enc -aes-128-ecb -e -in pic_original.bmp -out encrypted_ecb.
bmp -K 00112233445566778889aabbccddeeff
```

openssl enc -aes-128-cbc -e -in pic_original.bmp -out encrypted_cbc.bmp -K 00112233445566778889aabbccddeeff -iv 0102030405060708

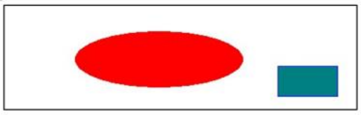


```
a0320506@seedlab:~/HW2/Labsetup/Files$ sudo openssl enc -aes-128-cbc -e -in pic_original.bmp -out encrypted_cbc.
bmp -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
```

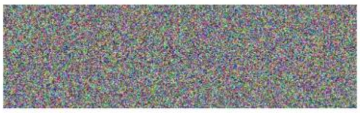
修改標頭



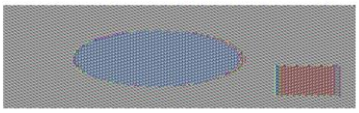
```
a0320506@seedlab:~/HW2/Labsetup/Files$ head -c 54 pic_original.bmp > header
a0320506@seedlab:~/HW2/Labsetup/Files$ tail -c +55 encrypted_ecb.bmp > body_ecb
a0320506@seedlab:~/HW2/Labsetup/Files$ tail -c +55 encrypted_cbc.bmp > body_cbc
a0320506@seedlab:~/HW2/Labsetup/Files$ cat header body_ecb > new_ecb.bmp
a0320506@seedlab:~/HW2/Labsetup/Files$ cat header body_cbc > new_cbc.bmp
```



pic_original.bmp



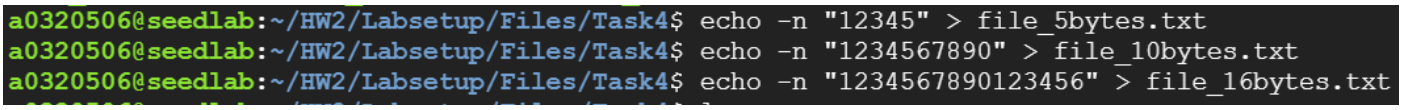
new_cbc.bmp



new_ecb.bmp 結果

6 Task 4: Padding

創建三種長度的原文檔案



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ echo -n "12345" > file_5bytes.txt
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ echo -n "1234567890" > file_10bytes.txt
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ echo -n "1234567890123456" > file_16bytes.txt
```

aes-128-cbc

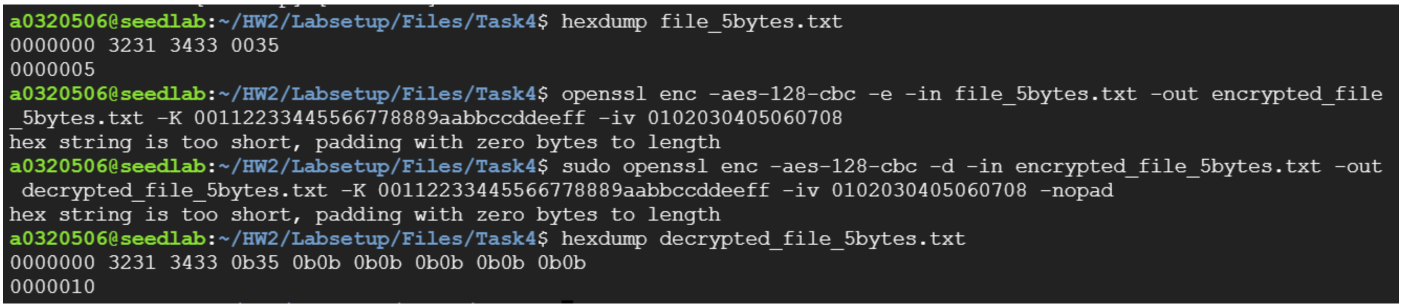
5 bytes

hexdump file_5bytes.txt

openssl enc -aes-128-cbc -e -in file_5bytes.txt -out encrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708

sudo openssl enc -aes-128-cbc -d -in encrypted_file_5bytes.txt -out decrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad

hexdump decrypted_file_5bytes.txt

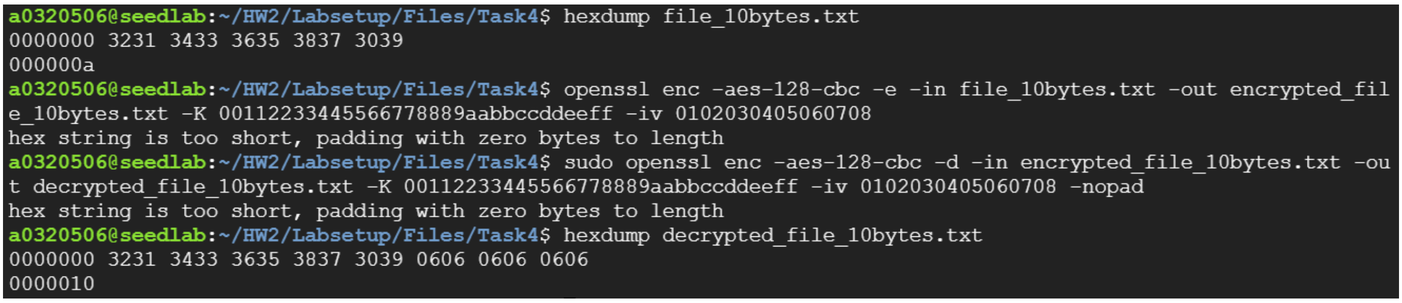


```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_5bytes.txt
0000000 3231 3433 0035
0000005
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-cbc -e -in file_5bytes.txt -out encrypted_file
_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-cbc -d -in encrypted_file_5bytes.txt -out
 decrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_5bytes.txt
0000000 3231 3433 0b35 0b0b 0b0b 0b0b 0b0b 0b0b
0000010
```

aes-128-cbc

10 bytes

```
hexdump file_10bytes.txt
openssl enc -aes-128-cbc -e -in file_10bytes.txt -out encrypted_file_10bytes.txt -K
00112233445566778889aabbccddeeff -iv 0102030405060708
sudo openssl enc -aes-128-cbc -d -in encrypted_file_10bytes.txt -out decrypted_file_10bytes.txt -K
00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hexdump decrypted_file_10bytes.txt
```

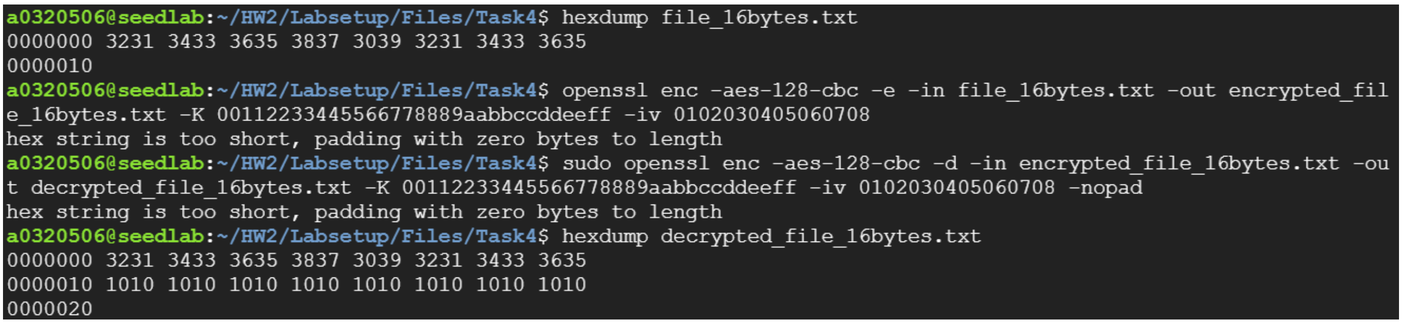


```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_10bytes.txt
0000000 3231 3433 3635 3837 3039
000000a
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-cbc -e -in file_10bytes.txt -out encrypted_fil
e_10bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-cbc -d -in encrypted_file_10bytes.txt -ou
t decrypted_file_10bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_10bytes.txt
0000000 3231 3433 3635 3837 3039 0606 0606 0606
0000010
```

aes-128-cbc

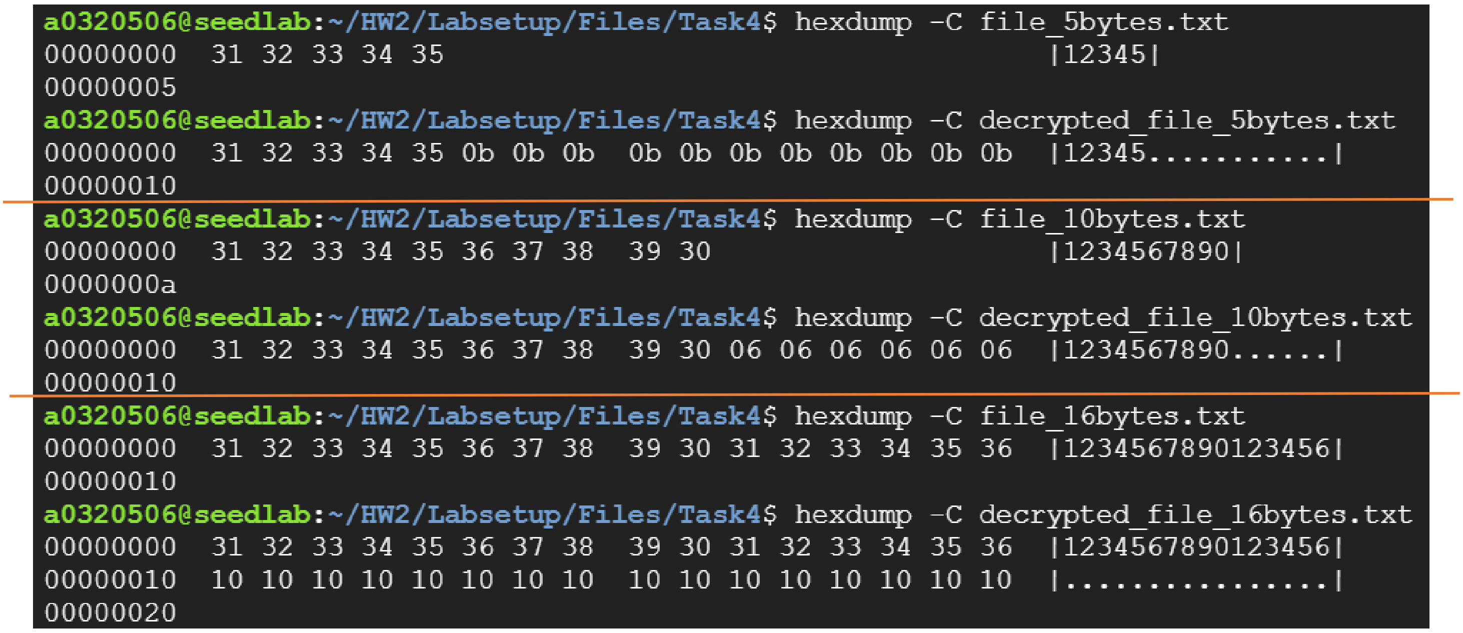
16 bytes

```
hexdump file_16bytes.txt
openssl enc -aes-128-cbc -e -in file_16bytes.txt -out encrypted_file_16bytes.txt -K
00112233445566778889aabbccddeeff -iv 0102030405060708
sudo openssl enc -aes-128-cbc -d -in encrypted_file_16bytes.txt -out decrypted_file_16bytes.txt -K
00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hexdump decrypted_file_16bytes.txt
```



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_16bytes.txt
0000000 3231 3433 3635 3837 3039 3231 3433 3635
0000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-cbc -e -in file_16bytes.txt -out encrypted_fil
e_16bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-cbc -d -in encrypted_file_16bytes.txt -ou
t decrypted_file_16bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_16bytes.txt
0000000 3231 3433 3635 3837 3039 3231 3433 3635
0000010 1010 1010 1010 1010 1010 1010 1010 1010
0000020
```

aes-128-cbc 比較



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_5bytes.txt
00000000  31 32 33 34 35                                    |12345|
00000005
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_5bytes.txt
00000000  31 32 33 34 35 0b 0b 0b  0b 0b 0b 0b 0b 0b 0b 0b  |12345...........|
00000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_10bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30                    |1234567890|
0000000a
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_10bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30 06 06 06 06 06 06  |1234567890......|
00000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_16bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_16bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010  10 10 10 10 10 10 10 10  10 10 10 10 10 10 10 10  |................|
00000020
```

要填充

aes-128-ecb

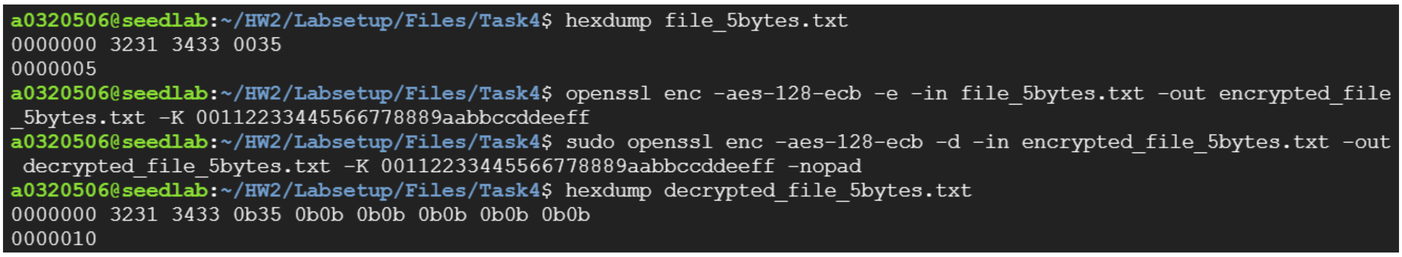
5 bytes

hexdump file_5bytes.txt

openssl enc -aes-128-ecb -e -in file_5bytes.txt -out encrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff

sudo openssl enc -aes-128-ecb -d -in encrypted_file_5bytes.txt -out decrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff -nopad

hexdump decrypted_file_5bytes.txt



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_5bytes.txt
0000000 3231 3433 0035
0000005
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-ecb -e -in file_5bytes.txt -out encrypted_file
_5bytes.txt -K 00112233445566778889aabbccddeeff
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-ecb -d -in encrypted_file_5bytes.txt -out
 decrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff -nopad
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_5bytes.txt
0000000 3231 3433 0b35 0b0b 0b0b 0b0b 0b0b 0b0b
0000010
```

aes-128-ecb

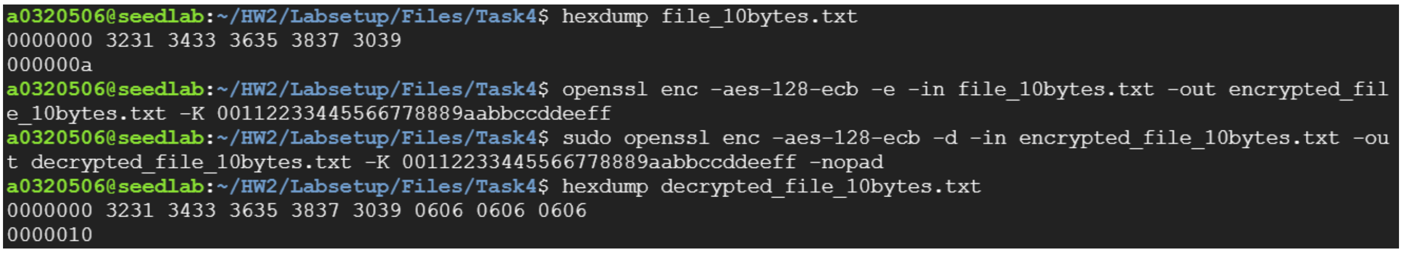
10 bytes

hexdump file_10bytes.txt

openssl enc -aes-128-ecb -e -in file_10bytes.txt -out encrypted_file_10bytes.txt -K 00112233445566778889aabbccddeeff

sudo openssl enc -aes-128-ecb -d -in encrypted_file_10bytes.txt -out decrypted_file_10bytes.txt -K 00112233445566778889aabbccddeeff -nopad

hexdump decrypted_file_10bytes.txt



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_10bytes.txt
0000000 3231 3433 3635 3837 3039
000000a
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-ecb -e -in file_10bytes.txt -out encrypted_fil
e_10bytes.txt -K 00112233445566778889aabbccddeeff
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-ecb -d -in encrypted_file_10bytes.txt -ou
t decrypted_file_10bytes.txt -K 00112233445566778889aabbccddeeff -nopad
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_10bytes.txt
0000000 3231 3433 3635 3837 3039 0606 0606 0606
0000010
```

aes-128-ecb

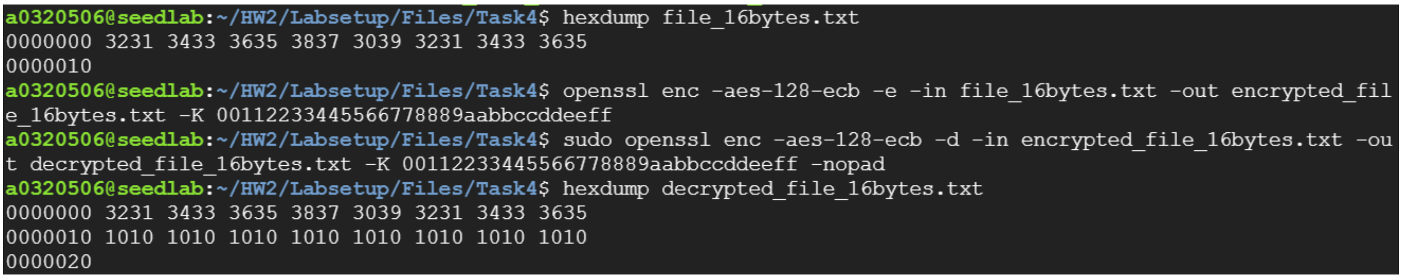
16 bytes

hexdump file_16bytes.txt

openssl enc -aes-128-ecb -e -in file_16bytes.txt -out encrypted_file_16bytes.txt -K 00112233445566778889aabbccddeeff

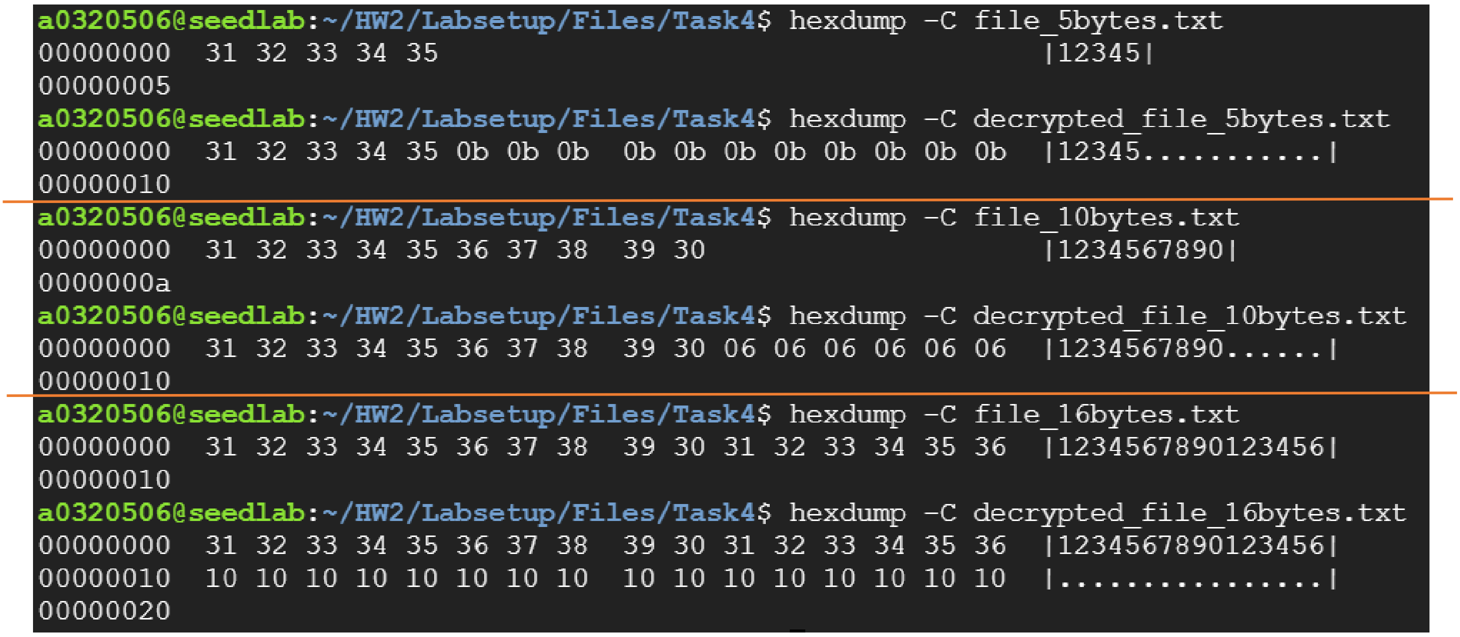
sudo openssl enc -aes-128-ecb -d -in encrypted_file_16bytes.txt -out decrypted_file_16bytes.txt -K 00112233445566778889aabbccddeeff -nopad

hexdump decrypted_file_16bytes.txt



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_16bytes.txt
0000000 3231 3433 3635 3837 3039 3231 3433 3635
0000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-ecb -e -in file_16bytes.txt -out encrypted_fil
e_16bytes.txt -K 00112233445566778889aabbccddeeff
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-ecb -d -in encrypted_file_16bytes.txt -ou
t decrypted_file_16bytes.txt -K 00112233445566778889aabbccddeeff -nopad
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_16bytes.txt
0000000 3231 3433 3635 3837 3039 3231 3433 3635
0000010 1010 1010 1010 1010 1010 1010 1010 1010
0000020
```

aes-128-ecb 比較



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_5bytes.txt
00000000  31 32 33 34 35                                    |12345|
00000005
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_5bytes.txt
00000000  31 32 33 34 35 0b 0b 0b  0b 0b 0b 0b 0b 0b 0b 0b  |12345...........|
00000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_10bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30                    |1234567890|
0000000a
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_10bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30 06 06 06 06 06 06  |1234567890......|
00000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_16bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_16bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010  10 10 10 10 10 10 10 10  10 10 10 10 10 10 10 10  |................|
00000020
```

要填充

aes-128-cfb

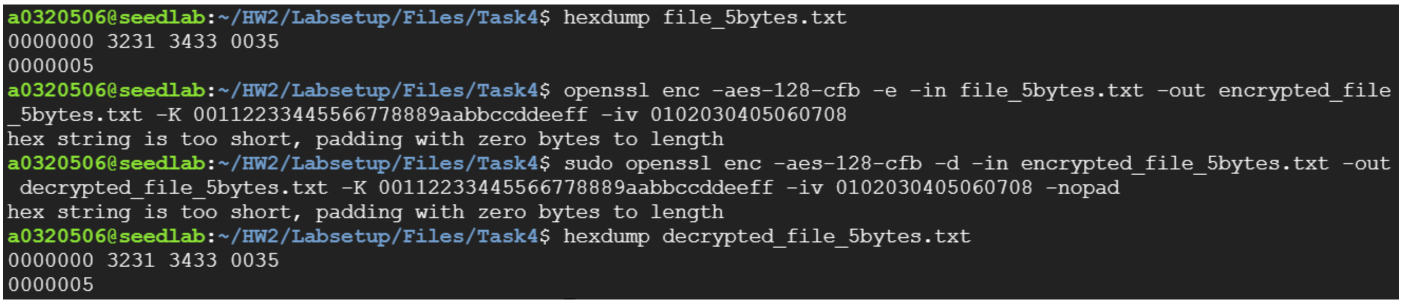
5 bytes

hexdump file_5bytes.txt

openssl enc -aes-128-cfb -e -in file_5bytes.txt -out encrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708

sudo openssl enc -aes-128-cfb -d -in encrypted_file_5bytes.txt -out decrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad

hexdump decrypted_file_5bytes.txt



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_5bytes.txt
0000000 3231 3433 0035
0000005
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-cfb -e -in file_5bytes.txt -out encrypted_file
_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-cfb -d -in encrypted_file_5bytes.txt -out
 decrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_5bytes.txt
0000000 3231 3433 0035
0000005
```

aes-128-cfb

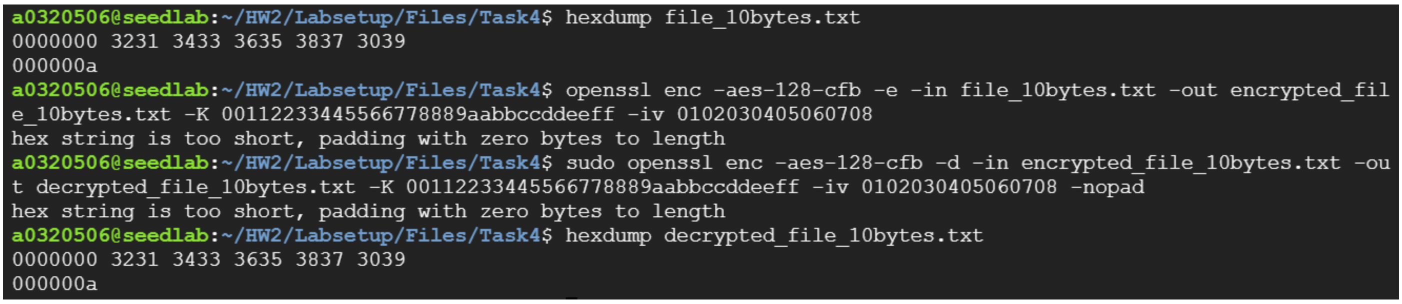
10 bytes

hexdump file_10bytes.txt

openssl enc -aes-128-cfb -e -in file_10bytes.txt -out encrypted_file_10bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708

sudo openssl enc -aes-128-cfb -d -in encrypted_file_10bytes.txt -out decrypted_file_10bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad

hexdump decrypted_file_10bytes.txt



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_10bytes.txt
0000000 3231 3433 3635 3837 3039
000000a
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-cfb -e -in file_10bytes.txt -out encrypted_fil
e_10bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-cfb -d -in encrypted_file_10bytes.txt -ou
t decrypted_file_10bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_10bytes.txt
0000000 3231 3433 3635 3837 3039
000000a
```

aes-128-cfb

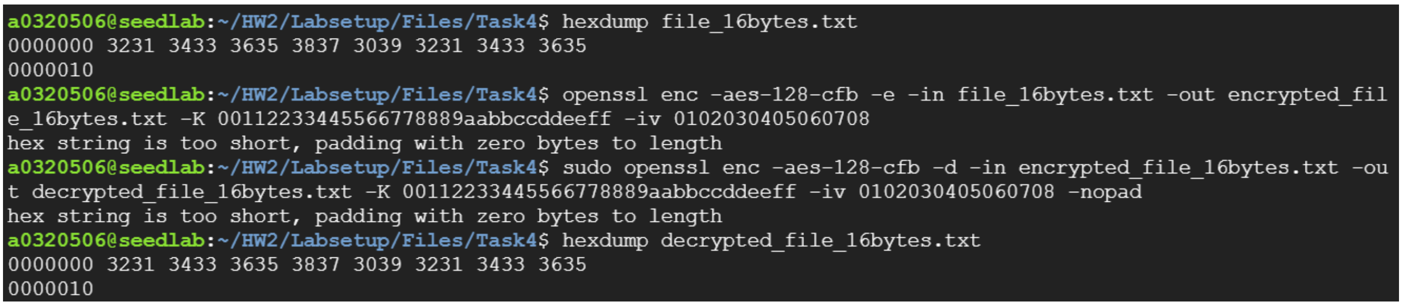
16 bytes

hexdump file_16bytes.txt

openssl enc -aes-128-cfb -e -in file_16bytes.txt -out encrypted_file_16bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708

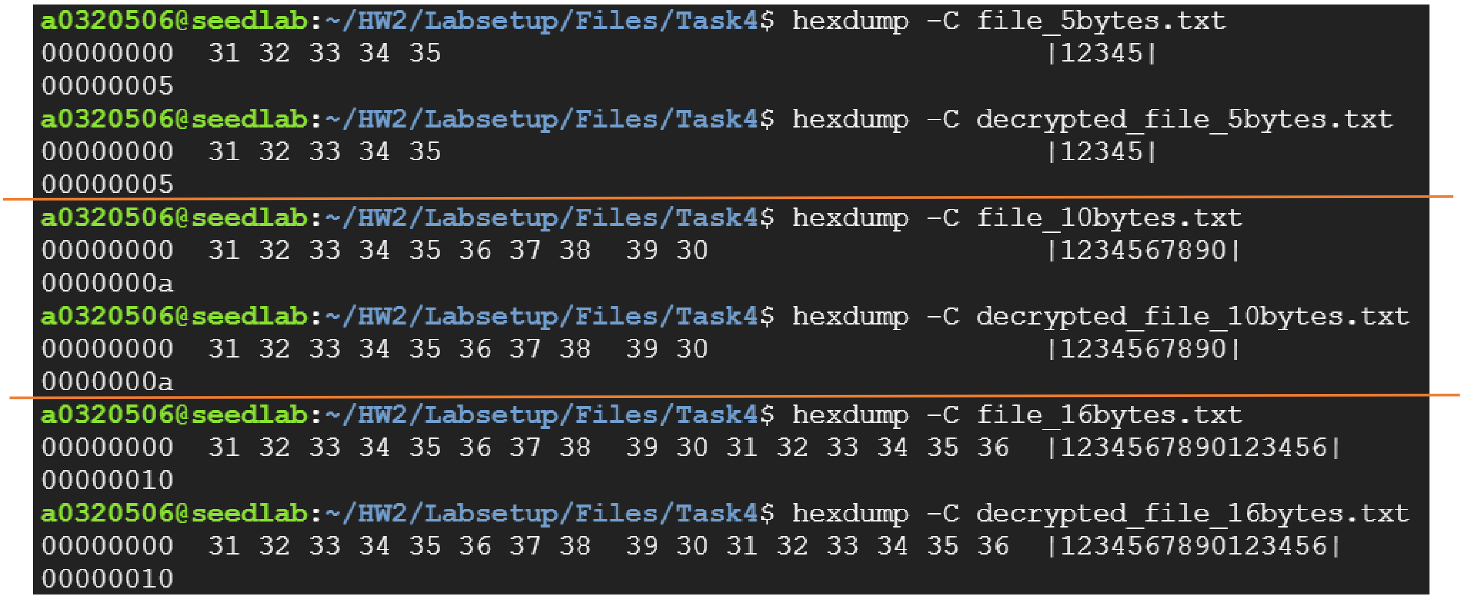
sudo openssl enc -aes-128-cfb -d -in encrypted_file_16bytes.txt -out decrypted_file_16bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad

hexdump decrypted_file_16bytes.txt



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_16bytes.txt
0000000 3231 3433 3635 3837 3039 3231 3433 3635
0000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-cfb -e -in file_16bytes.txt -out encrypted_fil
e_16bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-cfb -d -in encrypted_file_16bytes.txt -ou
t decrypted_file_16bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_16bytes.txt
0000000 3231 3433 3635 3837 3039 3231 3433 3635
0000010
```

aes-128-cfb 比較



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_5bytes.txt
00000000  31 32 33 34 35                                    |12345|
00000005
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_5bytes.txt
00000000  31 32 33 34 35                                    |12345|
00000005
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_10bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30                    |1234567890|
0000000a
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_10bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30                    |1234567890|
0000000a
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_16bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_16bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010
```

不填充

aes-128-ofb

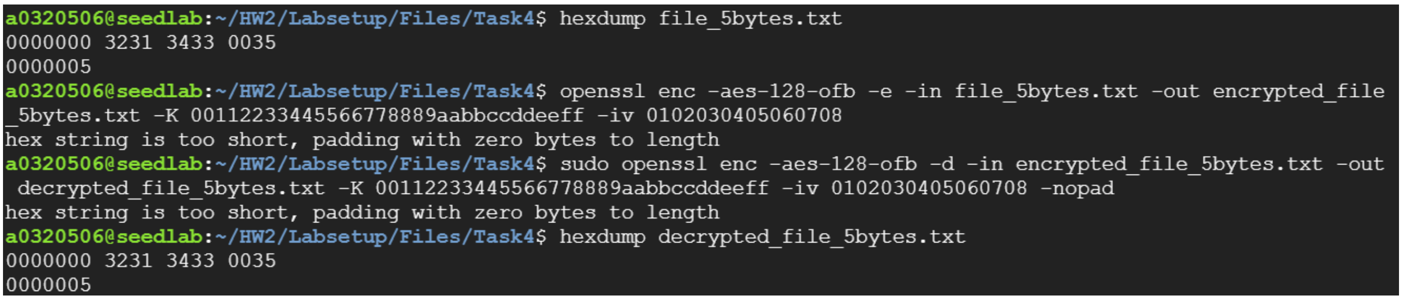
5 bytes

hexdump file_5bytes.txt

openssl enc -aes-128-ofb -e -in file_5bytes.txt -out encrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708

sudo openssl enc -aes-128-ofb -d -in encrypted_file_5bytes.txt -out decrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad

hexdump decrypted_file_5bytes.txt



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_5bytes.txt
0000000 3231 3433 0035
0000005
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-ofb -e -in file_5bytes.txt -out encrypted_file
_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-ofb -d -in encrypted_file_5bytes.txt -out
 decrypted_file_5bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_5bytes.txt
0000000 3231 3433 0035
0000005
```

aes-128-ofb

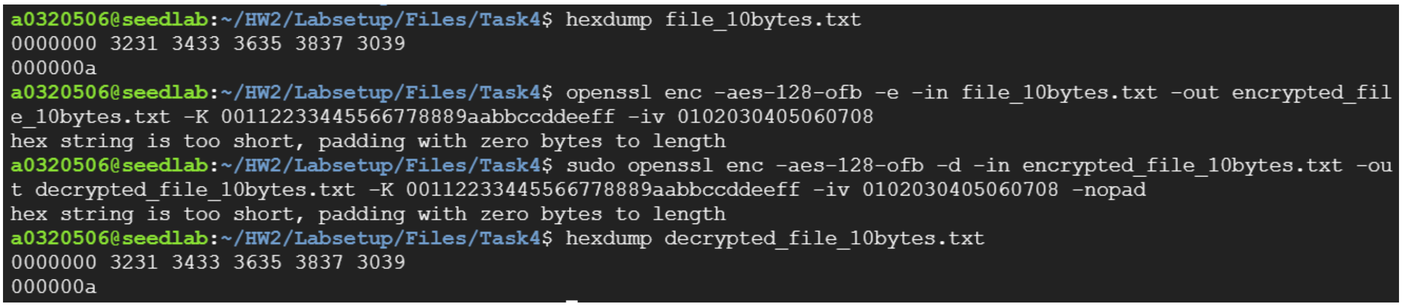
10 bytes

hexdump file_10bytes.txt

openssl enc -aes-128-ofb -e -in file_10bytes.txt -out encrypted_file_10bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708

sudo openssl enc -aes-128-ofb -d -in encrypted_file_10bytes.txt -out decrypted_file_10bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad

hexdump decrypted_file_10bytes.txt



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_10bytes.txt
0000000 3231 3433 3635 3837 3039
000000a
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-ofb -e -in file_10bytes.txt -out encrypted_fil
e_10bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-ofb -d -in encrypted_file_10bytes.txt -ou
t decrypted_file_10bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_10bytes.txt
0000000 3231 3433 3635 3837 3039
000000a
```

aes-128-ofb

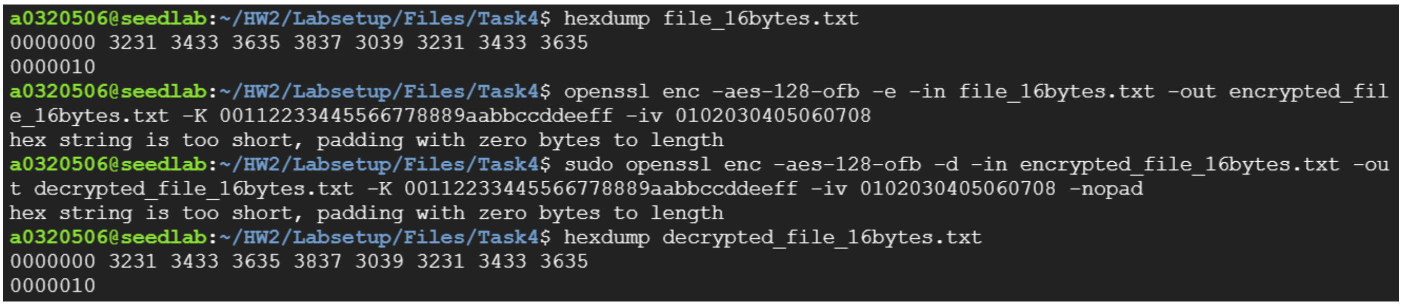
16 bytes

hexdump file_16bytes.txt

openssl enc -aes-128-ofb -e -in file_16bytes.txt -out encrypted_file_16bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708

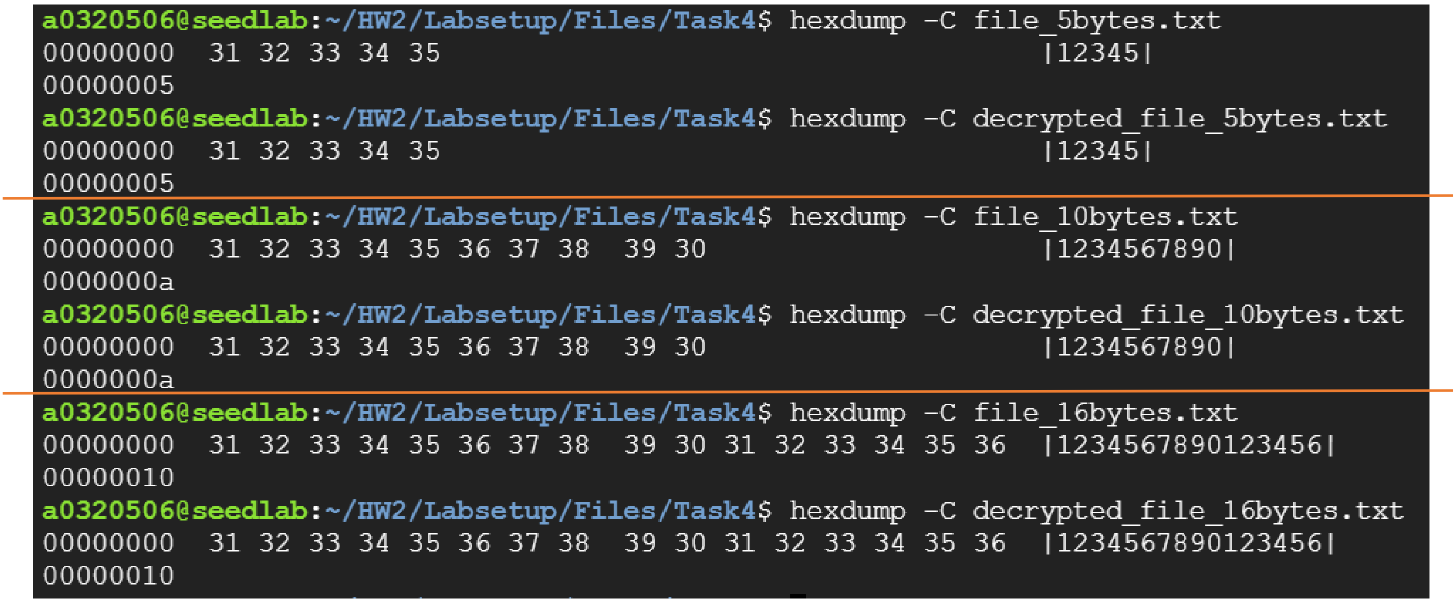
sudo openssl enc -aes-128-ofb -d -in encrypted_file_16bytes.txt -out decrypted_file_16bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad

hexdump decrypted_file_16bytes.txt



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump file_16bytes.txt
0000000 3231 3433 3635 3837 3039 3231 3433 3635
0000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ openssl enc -aes-128-ofb -e -in file_16bytes.txt -out encrypted_fil
e_16bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ sudo openssl enc -aes-128-ofb -d -in encrypted_file_16bytes.txt -ou
t decrypted_file_16bytes.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump decrypted_file_16bytes.txt
0000000 3231 3433 3635 3837 3039 3231 3433 3635
0000010
```

aes-128-ofb 比較



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_5bytes.txt
00000000  31 32 33 34 35                                    |12345|
00000005
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_5bytes.txt
00000000  31 32 33 34 35                                    |12345|
00000005
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_10bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30                    |1234567890|
0000000a
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_10bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30                    |1234567890|
0000000a
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C file_16bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010
a0320506@seedlab:~/HW2/Labsetup/Files/Task4$ hexdump -C decrypted_file_16bytes.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010
```

不填充

ECB、CBC 要填充，因為在他們把明文分成固定大小的塊進行加密。如果明文的大小不是加密演算法要求的固定塊大小的倍數，就需要對明文進行填充，讓每塊的大小相同

CFB、OFB 不填充，因為他們的加密是以 byte 為單位進行，所以即使明文的大小不是固定塊大小的倍數，也不需要填充

---

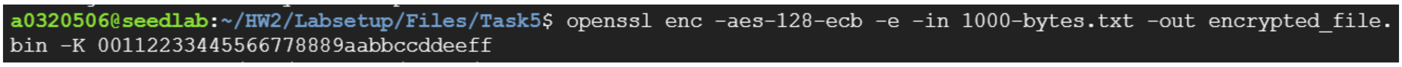
7 Task 5: Error Propagation – Corrupted Cipher Text

文本檔案在 1000-bytes.txt

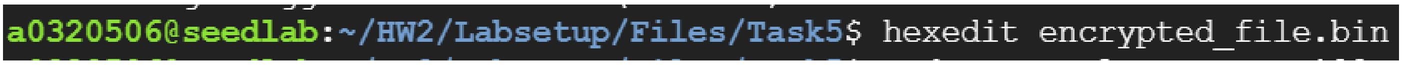
-iv 0102030405060708

aes-128-ecb

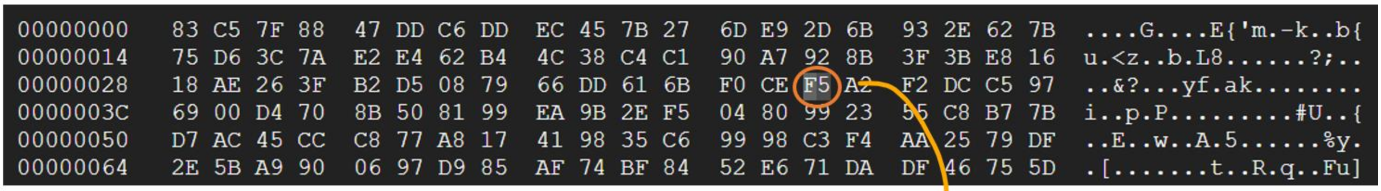
openssl enc -aes-128-ecb -e -in 1000-bytes.txt -out encrypted_file.bin -K 00112233445566778889aabbccddeeff



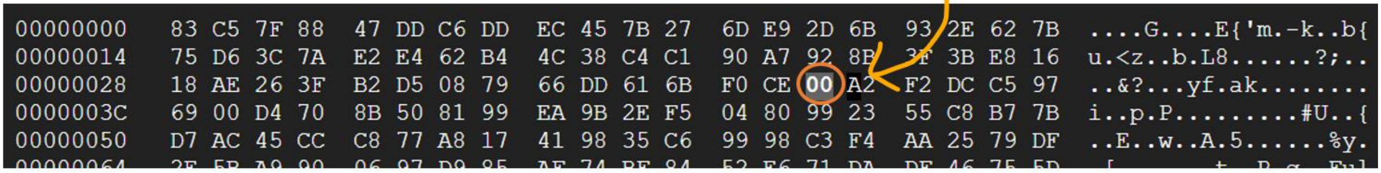
```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ openssl enc -aes-128-ecb -e -in 1000-bytes.txt -out encrypted_file.
bin -K 00112233445566778889aabbccddeeff
```



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ hexedit encrypted_file.bin
```

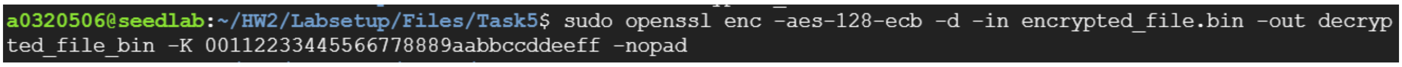


```
00000000   83 C5 7F 88   47 DD C6 DD   EC 45 7B 27   6D E9 2D 6B   93 2E 62 7B   ....G....E{'m.-k..b{
00000014   75 D6 3C 7A   E2 E4 62 B4   4C 38 C4 C1   90 A7 92 8B   3F 3B E8 16   u.<z..b.L8......?;..
00000028   18 AE 26 3F   B2 D5 08 79   66 DD 61 6B   F0 CE F5 A2   F2 DC C5 97   ..&?...yf.ak........
0000003C   69 00 D4 70   8B 50 81 99   EA 9B 2E F5   04 80 99 23   55 C8 B7 7B   i..p.P.........#U..{
00000050   D7 AC 45 CC   C8 77 A8 17   41 98 35 C6   99 98 C3 F4   AA 25 79 DF   ..E..w..A.5......%y.
00000064   2E 5B A9 90   06 97 D9 85   AF 74 BF 84   52 E6 71 DA   DF 46 75 5D   .[.......t..R.q..Fu]
```

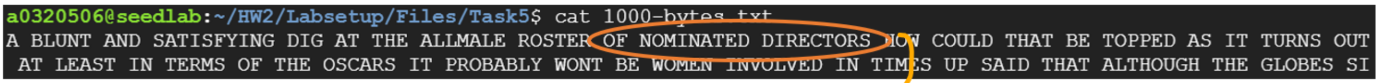


```
00000000   83 C5 7F 88   47 DD C6 DD   EC 45 7B 27   6D E9 2D 6B   93 2E 62 7B   ....G....E{'m.-k..b{
00000014   75 D6 3C 7A   E2 E4 62 B4   4C 38 C4 C1   90 A7 92 8B   3F 3B E8 16   u.<z..b.L8......?;..
00000028   18 AE 26 3F   B2 D5 08 79   66 DD 61 6B   F0 CE 00 A2   F2 DC C5 97   ..&?...yf.ak........
0000003C   69 00 D4 70   8B 50 81 99   EA 9B 2E F5   04 80 99 23   55 C8 B7 7B   i..p.P.........#U..{
00000050   D7 AC 45 CC   C8 77 A8 17   41 98 35 C6   99 98 C3 F4   AA 25 79 DF   ..E..w..A.5......%y.
00000064   2E 5B A9 90   06 97 D9 85   AF 74 BF 84   52 E6 71 DA   DF 46 75 5D   .[.......t..R.q..Fu]
```

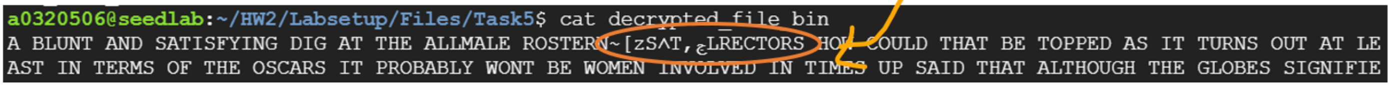
sudo openssl enc -aes-128-ecb -d -in encrypted_file.bin -out decrypted_file_bin -K 00112233445566778889aabbccddeeff -nopad



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ sudo openssl enc -aes-128-ecb -d -in encrypted_file.bin -out decryp
ted_file_bin -K 00112233445566778889aabbccddeeff -nopad
```



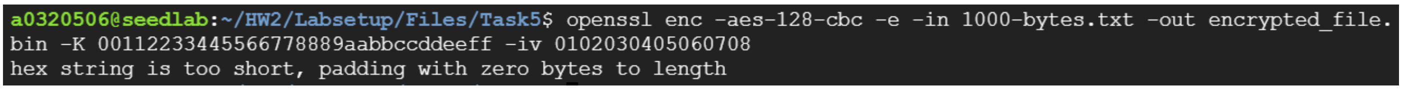
```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ cat 1000-bytes.txt
A BLUNT AND SATISFYING DIG AT THE ALLMALE ROSTER OF NOMINATED DIRECTORS HOW COULD THAT BE TOPPED AS IT TURNS OUT
 AT LEAST IN TERMS OF THE OSCARS IT PROBABLY WONT BE WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SI
```



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ cat decrypted_file_bin
A BLUNT AND SATISFYING DIG AT THE ALLMALE ROSTERN~[zS^T,ج LRECTORS HOW COULD THAT BE TOPPED AS IT TURNS OUT AT LE
AST IN TERMS OF THE OSCARS IT PROBABLY WONT BE WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SIGNIFIE
```

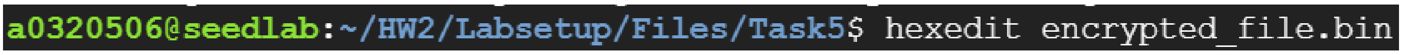
aes-128-cbc

openssl enc -aes-128-cbc -e -in 1000-bytes.txt -out encrypted_file.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708

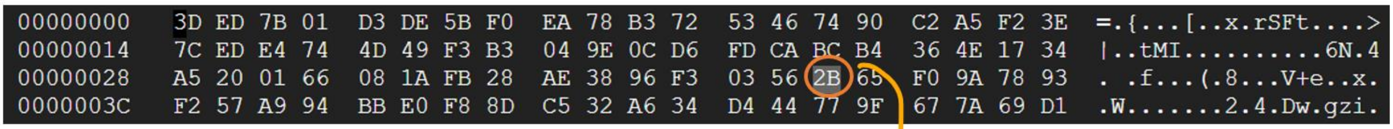


```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ openssl enc -aes-128-cbc -e -in 1000-bytes.txt -out encrypted_file.
bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
```

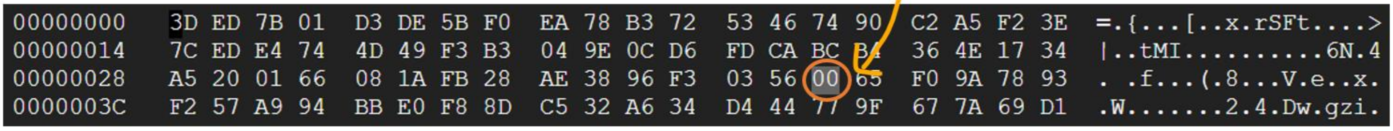
hexedit encrypted_file.bin



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ hexedit encrypted_file.bin
```

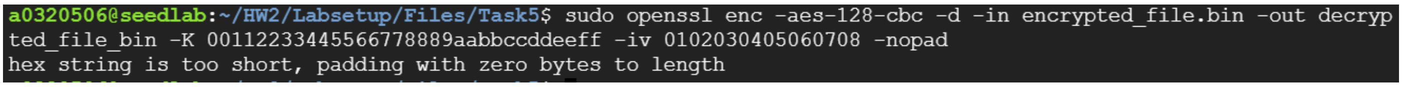


```
00000000   3D ED 7B 01   D3 DE 5B F0   EA 78 B3 72   53 46 74 90   C2 A5 F2 3E   =.{...[..x.rSFt....>
00000014   7C ED E4 74   4D 49 F3 B3   04 9E 0C D6   FD CA BC B4   36 4E 17 34   |..tMI..........6N.4
00000028   A5 20 01 66   08 1A FB 28   AE 38 96 F3   03 56 2B 65   F0 9A 78 93   . .f...(.8...V+e..x.
0000003C   F2 57 A9 94   BB E0 F8 8D   C5 32 A6 34   D4 44 77 9F   67 7A 69 D1   .W.......2.4.Dw.gzi.
```



```
00000000   3D ED 7B 01   D3 DE 5B F0   EA 78 B3 72   53 46 74 90   C2 A5 F2 3E   =.{...[..x.rSFt....>
00000014   7C ED E4 74   4D 49 F3 B3   04 9E 0C D6   FD CA BC B4   36 4E 17 34   |..tMI..........6N.4
00000028   A5 20 01 66   08 1A FB 28   AE 38 96 F3   03 56 00 65   F0 9A 78 93   . .f...(.8...V.e..x.
0000003C   F2 57 A9 94   BB E0 F8 8D   C5 32 A6 34   D4 44 77 9F   67 7A 69 D1   .W.......2.4.Dw.gzi.
```

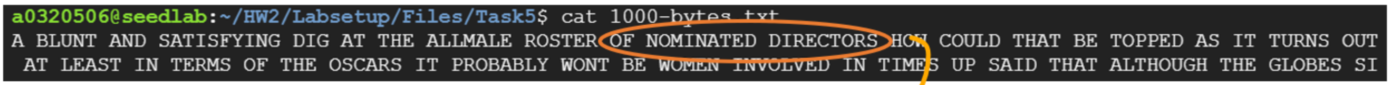
sudo openssl enc -aes-128-cbc -d -in encrypted_file.bin -out decrypted_file_bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad



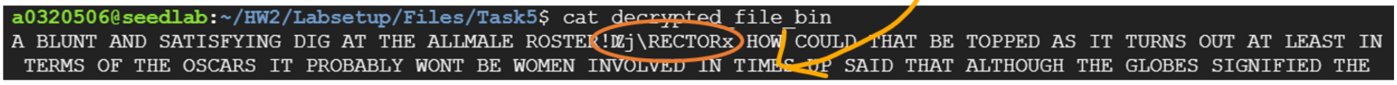
```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ sudo openssl enc -aes-128-cbc -d -in encrypted_file.bin -out decryp
ted_file_bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
```

cat decrypted_file_bin

cat 1000-bytes.txt



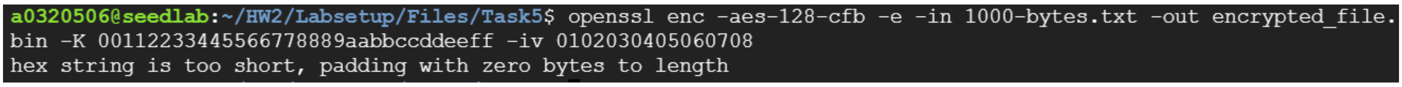
```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ cat 1000-bytes.txt
A BLUNT AND SATISFYING DIG AT THE ALLMALE ROSTER OF NOMINATED DIRECTORS HOW COULD THAT BE TOPPED AS IT TURNS OUT
 AT LEAST IN TERMS OF THE OSCARS IT PROBABLY WONT BE WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SI
```



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ cat decrypted_file_bin
A BLUNT AND SATISFYING DIG AT THE ALLMALE ROSTER!Zj\RECTORx HOW COULD THAT BE TOPPED AS IT TURNS OUT AT LEAST IN
 TERMS OF THE OSCARS IT PROBABLY WONT BE WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SIGNIFIED THE
```

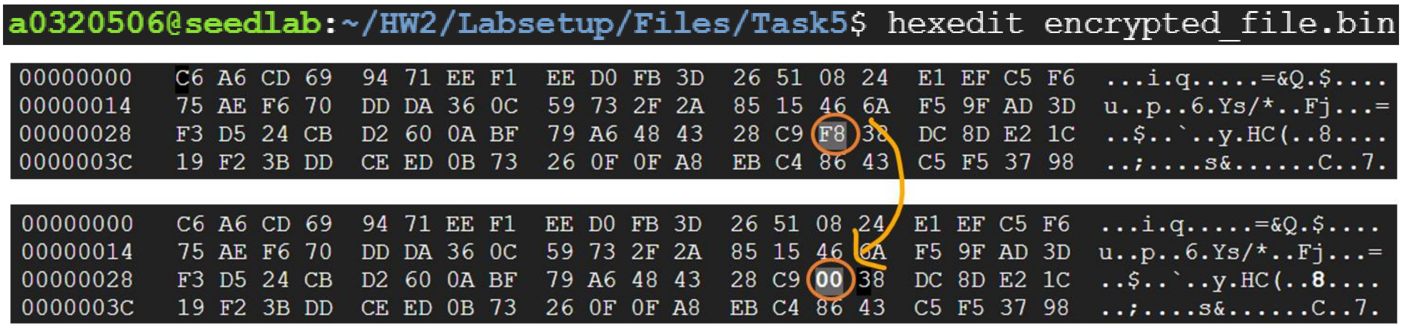
aes-128-cfb

openssl enc -aes-128-cfb -e -in 1000-bytes.txt -out encrypted_file.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ openssl enc -aes-128-cfb -e -in 1000-bytes.txt -out encrypted_file.
bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
```

hexedit encrypted_file.bin

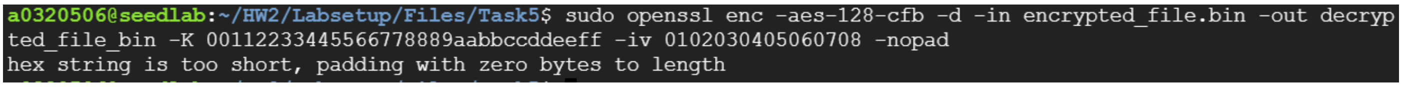


```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ hexedit encrypted_file.bin
```

```
00000000   C6 A6 CD 69  94 71 EE F1  EE D0 FB 3D  26 51 08 24  E1 EF C5 F6  ...i.q.....=&Q.$....
00000014   75 AE F6 70  DD DA 36 0C  59 73 2F 2A  85 15 46 6A  F5 9F AD 3D  u..p..6.Ys/*..Fj...=
00000028   F3 D5 24 CB  D2 60 0A BF  79 A6 48 43  28 C9 F8 38  DC 8D E2 1C  ..$..`..y.HC(..8....
0000003C   19 F2 3B DD  CE ED 0B 73  26 0F 0F A8  EB C4 86 43  C5 F5 37 98  ..;....s&......C..7.
```

```
00000000   C6 A6 CD 69  94 71 EE F1  EE D0 FB 3D  26 51 08 24  E1 EF C5 F6  ...i.q.....=&Q.$....
00000014   75 AE F6 70  DD DA 36 0C  59 73 2F 2A  85 15 46 6A  F5 9F AD 3D  u..p..6.Ys/*..Fj...=
00000028   F3 D5 24 CB  D2 60 0A BF  79 A6 48 43  28 C9 00 38  DC 8D E2 1C  ..$..`..y.HC(..8....
0000003C   19 F2 3B DD  CE ED 0B 73  26 0F 0F A8  EB C4 86 43  C5 F5 37 98  ..;....s&......C..7.
```

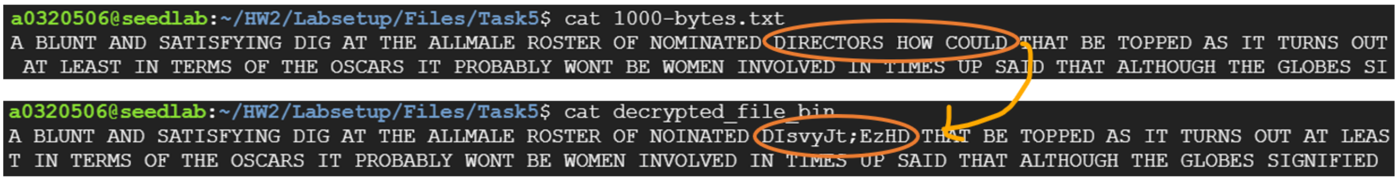
sudo openssl enc -aes-128-cfb -d -in encrypted_file.bin -out decrypted_file_bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ sudo openssl enc -aes-128-cfb -d -in encrypted_file.bin -out decryp
ted_file_bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
```

cat decrypted_file_bin

cat 1000-bytes.txt

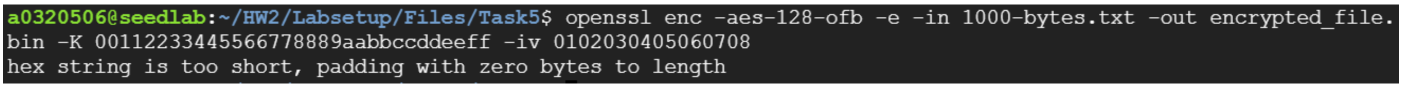


```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ cat 1000-bytes.txt
A BLUNT AND SATISFYING DIG AT THE ALLMALE ROSTER OF NOMINATED DIRECTORS HOW COULD THAT BE TOPPED AS IT TURNS OUT
 AT LEAST IN TERMS OF THE OSCARS IT PROBABLY WONT BE WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SI
```

```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ cat decrypted_file_bin
A BLUNT AND SATISFYING DIG AT THE ALLMALE ROSTER OF NOINATED DIsvyJt;EzHD THAT BE TOPPED AS IT TURNS OUT AT LEAS
T IN TERMS OF THE OSCARS IT PROBABLY WONT BE WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SIGNIFIED
```

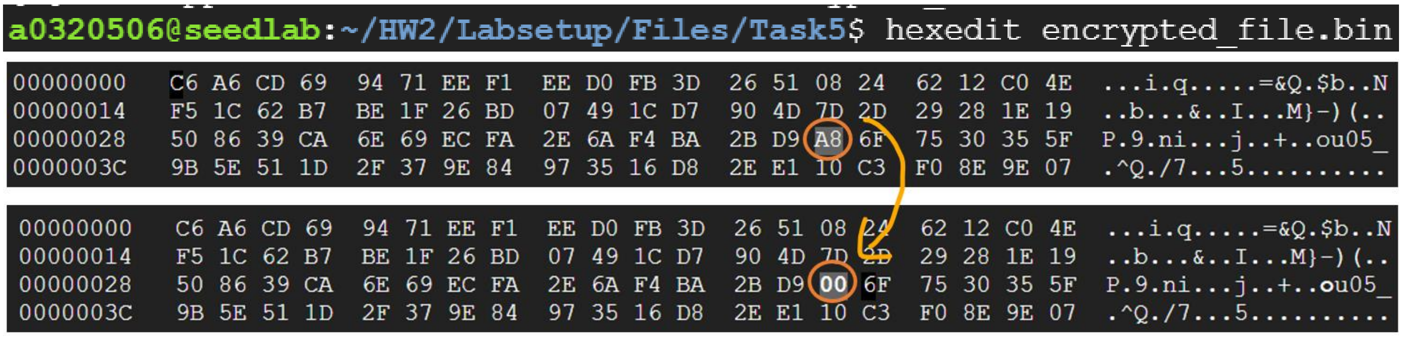
aes-128-ofb

openssl enc -aes-128-ofb -e -in 1000-bytes.txt -out encrypted_file.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ openssl enc -aes-128-ofb -e -in 1000-bytes.txt -out encrypted_file.
bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
```

hexedit encrypted_file.bin

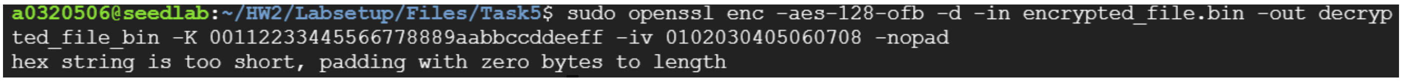


```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ hexedit encrypted_file.bin
```

```
00000000   C6 A6 CD 69  94 71 EE F1  EE D0 FB 3D  26 51 08 24  62 12 C0 4E  ...i.q.....=&Q.$b..N
00000014   F5 1C 62 B7  BE 1F 26 BD  07 49 1C D7  90 4D 7D 2D  29 28 1E 19  ..b...&..I...M}-)(..
00000028   50 86 39 CA  6E 69 EC FA  2E 6A F4 BA  2B D9 A8 6F  75 30 35 5F  P.9.ni...j..+..ou05_
0000003C   9B 5E 51 1D  2F 37 9E 84  97 35 16 D8  2E E1 10 C3  F0 8E 9E 07  .^Q./7...5..........
```

```
00000000   C6 A6 CD 69  94 71 EE F1  EE D0 FB 3D  26 51 08 24  62 12 C0 4E  ...i.q.....=&Q.$b..N
00000014   F5 1C 62 B7  BE 1F 26 BD  07 49 1C D7  90 4D 7D 2D  29 28 1E 19  ..b...&..I...M}-)(..
00000028   50 86 39 CA  6E 69 EC FA  2E 6A F4 BA  2B D9 00 6F  75 30 35 5F  P.9.ni...j..+..ou05_
0000003C   9B 5E 51 1D  2F 37 9E 84  97 35 16 D8  2E E1 10 C3  F0 8E 9E 07  .^Q./7...5..........
```

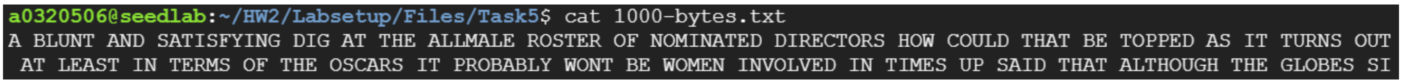
sudo openssl enc -aes-128-ofb -d -in encrypted_file.bin -out decrypted_file_bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad



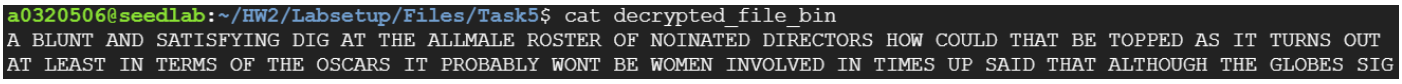
```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ sudo openssl enc -aes-128-ofb -d -in encrypted_file.bin -out decryp
ted_file_bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708 -nopad
hex string is too short, padding with zero bytes to length
```

cat decrypted_file_bin

cat 1000-bytes.txt



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ cat 1000-bytes.txt
A BLUNT AND SATISFYING DIG AT THE ALLMALE ROSTER OF NOMINATED DIRECTORS HOW COULD THAT BE TOPPED AS IT TURNS OUT
 AT LEAST IN TERMS OF THE OSCARS IT PROBABLY WONT BE WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SI
```



```
a0320506@seedlab:~/HW2/Labsetup/Files/Task5$ cat decrypted_file_bin
A BLUNT AND SATISFYING DIG AT THE ALLMALE ROSTER OF NOINATED DIRECTORS HOW COULD THAT BE TOPPED AS IT TURNS OUT
AT LEAST IN TERMS OF THE OSCARS IT PROBABLY WONT BE WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SIG
```

不受影響

實驗前猜測: ECB 和 CBC 會受較大影響，因為他們是用區塊為單位做加密，CFB 和 OFB 影響則較小，因為他們是用 byte 為單位做加密

實驗後觀察:

ECB 和 CBC 被影響到的部分: OF NOMINATED DIRECTORS，理論上 ECB 只影響到單個塊的解密結果，CBC 會影響到錯誤塊他的下一塊的解密結果，但在這裡剛好相同

CFB 被影響到的部分: DIRECTORS HOW COULD，影響到錯誤位元及其後續位元的解密結果

因為 ECB、CBC 以塊為單位，CFB 以 byte 為單位，所以他們受影響的起始位置不同

OFB 不受影響，因為他的加密過程中不需要使用到明文，因此即使在加密過程中某一位元發生了錯誤，也不會對解密結果產生影響

---

8 Task 6: Initial Vector (IV) and Common Mistakes

hexdump P.txt

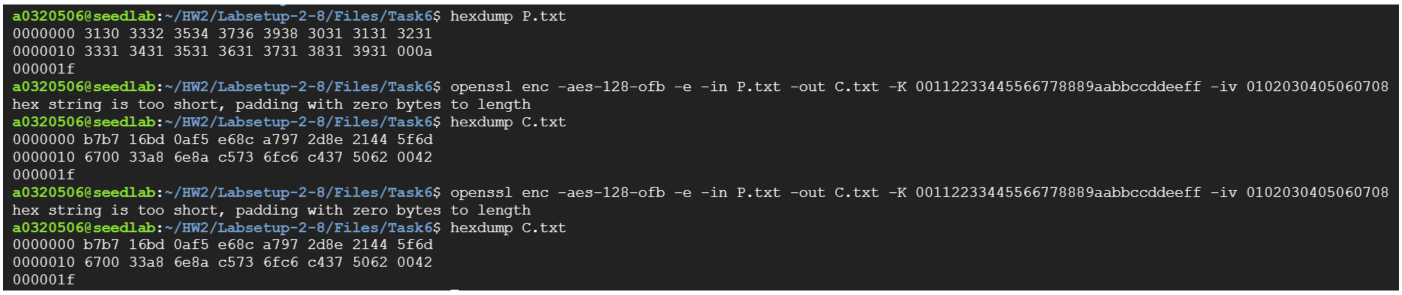
openssl enc -aes-128-ofb -e -in P.txt -out C.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708

hexdump C.txt

openssl enc -aes-128-ofb -e -in P.txt -out C.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708

hexdump C.txt

IV 相同，都是 0102030405060708



```
a0320506@seedlab:~/HW2/Labsetup-2-8/Files/Task6$ hexdump P.txt
0000000 3130 3332 3534 3736 3938 3031 3131 3231
0000010 3331 3431 3531 3631 3731 3831 3931 000a
000001f
a0320506@seedlab:~/HW2/Labsetup-2-8/Files/Task6$ openssl enc -aes-128-ofb -e -in P.txt -out C.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup-2-8/Files/Task6$ hexdump C.txt
0000000 b7b7 16bd 0af5 e68c a797 2d8e 2144 5f6d
0000010 6700 33a8 6e8a c573 6fc6 c437 5062 0042
000001f
a0320506@seedlab:~/HW2/Labsetup-2-8/Files/Task6$ openssl enc -aes-128-ofb -e -in P.txt -out C.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup-2-8/Files/Task6$ hexdump C.txt
0000000 b7b7 16bd 0af5 e68c a797 2d8e 2144 5f6d
0000010 6700 33a8 6e8a c573 6fc6 c437 5062 0042
000001f
```

加密結果完全相同

hexdump P.txt

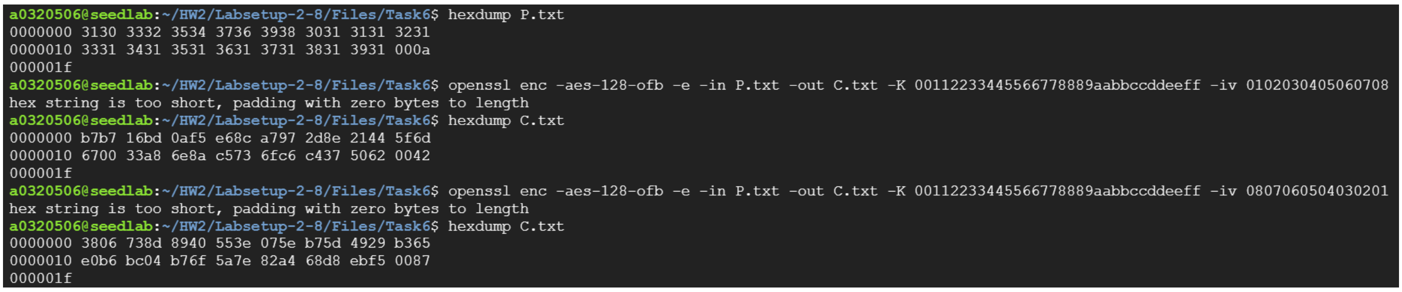
openssl enc -aes-128-ofb -e -in P.txt -out C.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708

hexdump C.txt

openssl enc -aes-128-ofb -e -in P.txt -out C.txt -K 00112233445566778889aabbccddeeff -iv 0807060504030201

hexdump C.txt

IV 不同，0102030405060708 和 0807060504030201



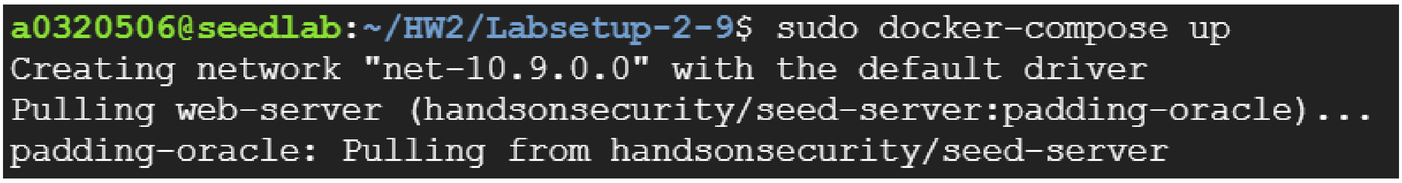
```
a0320506@seedlab:~/HW2/Labsetup-2-8/Files/Task6$ hexdump P.txt
0000000 3130 3332 3534 3736 3938 3031 3131 3231
0000010 3331 3431 3531 3631 3731 3831 3931 000a
000001f
a0320506@seedlab:~/HW2/Labsetup-2-8/Files/Task6$ openssl enc -aes-128-ofb -e -in P.txt -out C.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup-2-8/Files/Task6$ hexdump C.txt
0000000 b7b7 16bd 0af5 e68c a797 2d8e 2144 5f6d
0000010 6700 33a8 6e8a c573 6fc6 c437 5062 0042
000001f
a0320506@seedlab:~/HW2/Labsetup-2-8/Files/Task6$ openssl enc -aes-128-ofb -e -in P.txt -out C.txt -K 00112233445566778889aabbccddeeff -iv 0807060504030201
hex string is too short, padding with zero bytes to length
a0320506@seedlab:~/HW2/Labsetup-2-8/Files/Task6$ hexdump C.txt
0000000 3806 738d 8940 553e 075e b75d 4929 b365
0000010 e0b6 bc04 b76f 5a7e 82a4 68d8 ebf5 0087
000001f
```

加密結果不同

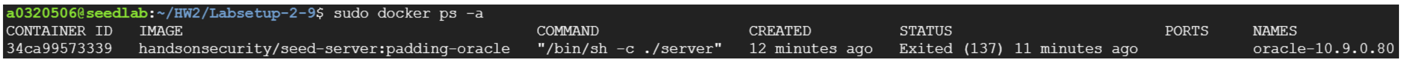
why IV needs to be unique　不然可能會發生加密結果一樣的情況

2.9 Lab: Padding Oracle Attack (15 pts)

要在 docker 下執行

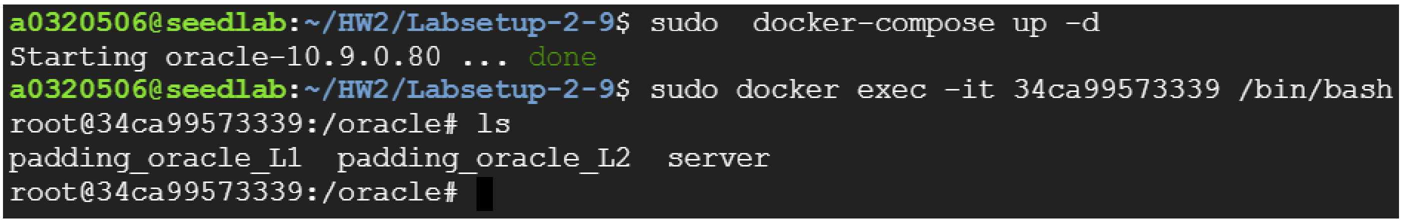


```
a0320506@seedlab:~/HW2/Labsetup-2-9$ sudo docker-compose up
Creating network "net-10.9.0.0" with the default driver
Pulling web-server (handsonsecurity/seed-server:padding-oracle)...
padding-oracle: Pulling from handsonsecurity/seed-server
```



```
a0320506@seedlab:~/HW2/Labsetup-2-9$ sudo docker ps -a
CONTAINER ID   IMAGE                                      COMMAND                  CREATED          STATUS                         PORTS     NAMES
34ca99573339   handsonsecurity/seed-server:padding-oracle "/bin/sh -c ./server"    12 minutes ago   Exited (137) 11 minutes ago              oracle-10.9.0.80
```

ID: 34ca99573339



```
a0320506@seedlab:~/HW2/Labsetup-2-9$ sudo  docker-compose up -d
Starting oracle-10.9.0.80 ... done
a0320506@seedlab:~/HW2/Labsetup-2-9$ sudo docker exec -it 34ca99573339 /bin/bash
root@34ca99573339:/oracle# ls
padding_oracle_L1  padding_oracle_L2  server
root@34ca99573339:/oracle# 
```

sudo docker-compose up -d

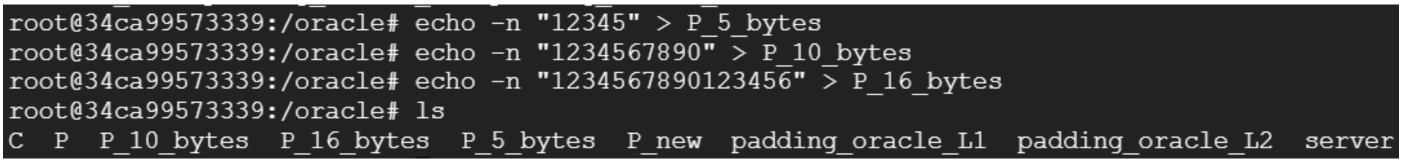
docker exec -it 34ca99573339 /bin/bash

3 Task 1: Getting Familiar with Padding

echo -n "12345" > P_5_bytes

echo -n "1234567890" > P_10_bytes

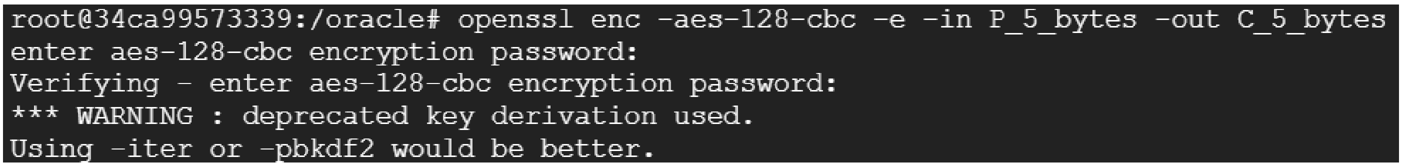
echo -n "1234567890123456" > P_16_bytes



```
root@34ca99573339:/oracle# echo -n "12345" > P_5_bytes
root@34ca99573339:/oracle# echo -n "1234567890" > P_10_bytes
root@34ca99573339:/oracle# echo -n "1234567890123456" > P_16_bytes
root@34ca99573339:/oracle# ls
C  P  P_10_bytes  P_16_bytes  P_5_bytes  P_new  padding_oracle_L1  padding_oracle_L2  server
```

openssl enc -aes-128-cbc -e -in P_5_bytes -out C_5_bytes

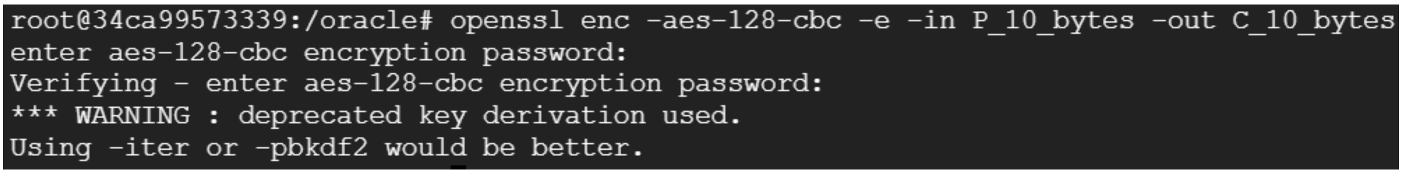
密碼: 00112233445566778889aabbccddeeff



```
root@34ca99573339:/oracle# openssl enc -aes-128-cbc -e -in P_5_bytes -out C_5_bytes
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.
```

openssl enc -aes-128-cbc -e -in P_10_bytes -out C_10_bytes

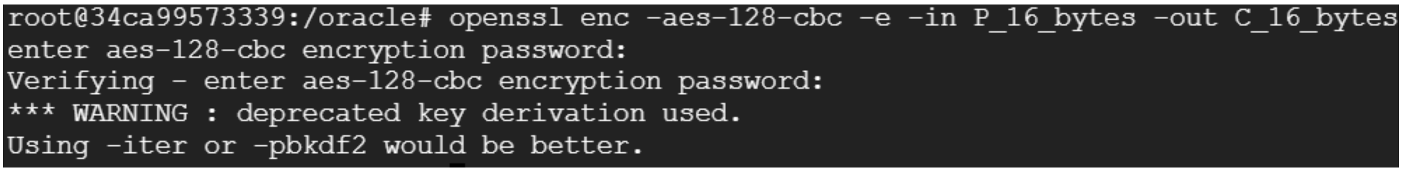
密碼: 00112233445566778889aabbccddeeff



```
root@34ca99573339:/oracle# openssl enc -aes-128-cbc -e -in P_10_bytes -out C_10_bytes
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.
```

openssl enc -aes-128-cbc -e -in P_16_bytes -out C_16_bytes

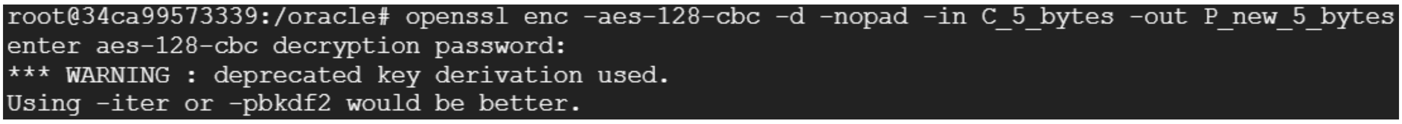
密碼: 00112233445566778889aabbccddeeff



```
root@34ca99573339:/oracle# openssl enc -aes-128-cbc -e -in P_16_bytes -out C_16_bytes
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.
```

openssl enc -aes-128-cbc -d -nopad -in C_5_bytes -out P_new_5_bytes

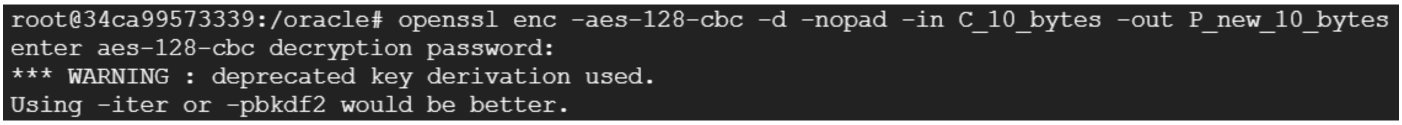
密碼: 00112233445566778889aabbccddeeff



```
root@34ca99573339:/oracle# openssl enc -aes-128-cbc -d -nopad -in C_5_bytes -out P_new_5_bytes
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

openssl enc -aes-128-cbc -d -nopad -in C_10_bytes -out P_new_10_bytes

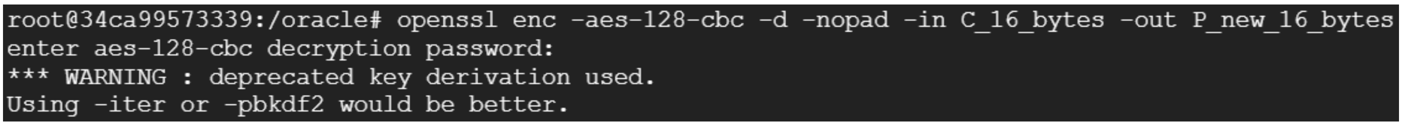
密碼: 00112233445566778889aabbccddeeff



```
root@34ca99573339:/oracle# openssl enc -aes-128-cbc -d -nopad -in C_10_bytes -out P_new_10_bytes
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

openssl enc -aes-128-cbc -d -nopad -in C_16_bytes -out P_new_16_bytes

密碼: 00112233445566778889aabbccddeeff

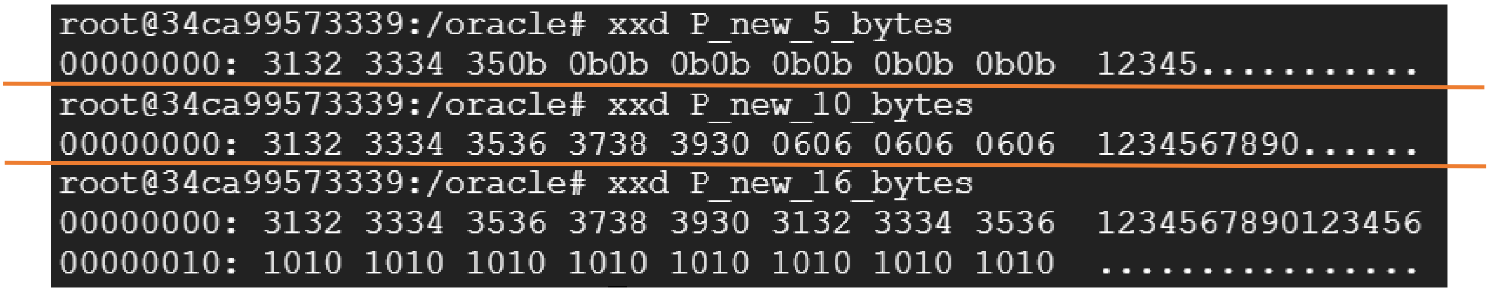


```
root@34ca99573339:/oracle# openssl enc -aes-128-cbc -d -nopad -in C_16_bytes -out P_new_16_bytes
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

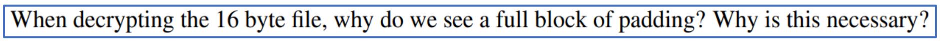
xxd P_new_5_bytes

xxd P_new_10_bytes

xxd P_new_16_bytes



```
root@34ca99573339:/oracle# xxd P_new_5_bytes
00000000: 3132 3334 350b 0b0b 0b0b 0b0b 0b0b 0b0b  12345...........
root@34ca99573339:/oracle# xxd P_new_10_bytes
00000000: 3132 3334 3536 3738 3930 0606 0606 0606  1234567890......
root@34ca99573339:/oracle# xxd P_new_16_bytes
00000000: 3132 3334 3536 3738 3930 3132 3334 3536  1234567890123456
00000010: 1010 1010 1010 1010 1010 1010 1010 1010  ................
```



When decrypting the 16 byte file, why do we see a full block of padding? Why is this necessary?

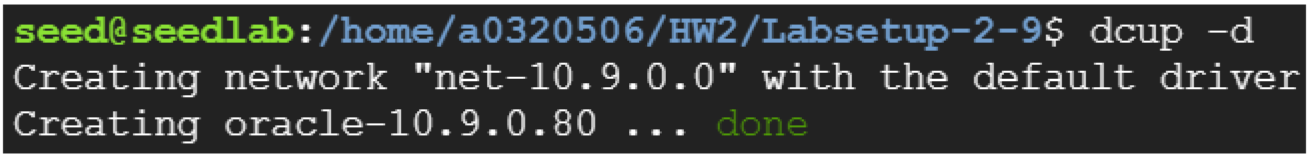
用 AES-128-CBC 模式對一個 16 bytes 的文件加密時，即使原始文件的大小已經是密鑰塊大小的倍數(在這裡是 16 bytes)，仍會看到一個完整的填充塊

這是因為在 CBC 模式下，加密演算法要求明文的大小必須是密鑰塊大小的倍數(即使原始文件已經是密鑰塊大小的倍數也一樣)

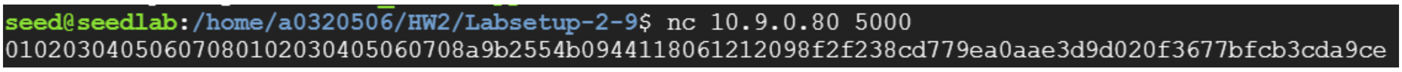
填充是為了確保明文能夠完全填滿一個密鑰塊，避免在加密過程中出現明文被分塊的情況，在 CBC 模式下一定要填充，因為每個明文塊都需要與前一個密文塊進行 XOR，這需要每個明文塊都有一個完整的密鑰塊大小

所以就算原始文件的大小已經是密鑰塊大小的倍數，還是需要填充來確保加密操作的正確

4 Task 2: Padding Oracle Attack (Level 1)



```
seed@seedlab:/home/a0320506/HW2/Labsetup-2-9$ dcup -d
Creating network "net-10.9.0.0" with the default driver
Creating oracle-10.9.0.80 ... done
```

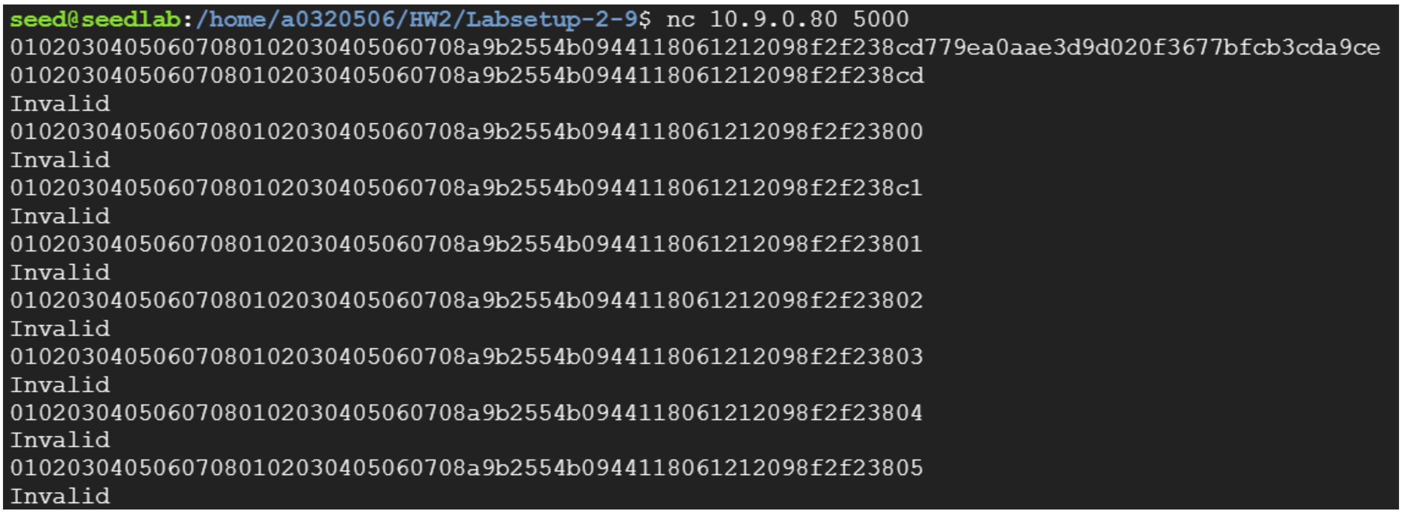


```
seed@seedlab:/home/a0320506/HW2/Labsetup-2-9$ nc 10.9.0.80 5000
01020304050607080102030405060708a9b2554b0944118061212098f2f238cd779ea0aae3d9d020f3677bfcb3cda9ce
```

01020304050607080102030405060708a9b2554b0944118061212098f2f238cd779ea0aae3d9d020f3677bfcb3cda9ce

IV: 0102030405 0607080102 0304050607 08

C: a9b2554b09 4411806121 2098f2f238 cd779ea0aa e3d9d020f3 677bfcb3cd a9 ce

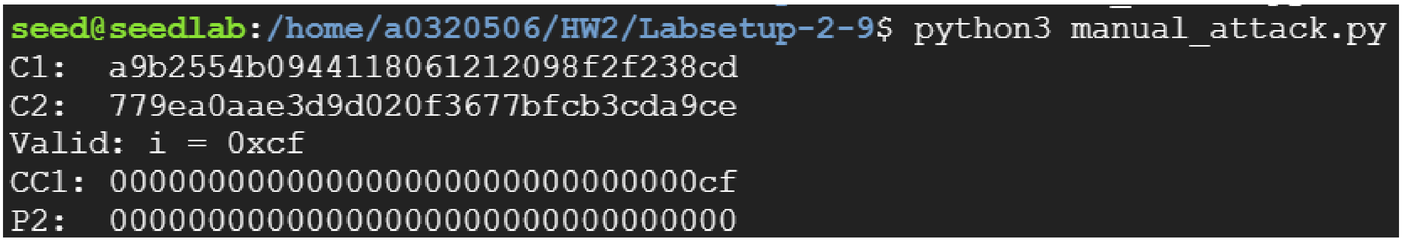


```
seed@seedlab:/home/a0320506/HW2/Labsetup-2-9$ nc 10.9.0.80 5000
01020304050607080102030405060708a9b2554b0944118061212098f2f238cd779ea0aae3d9d020f3677bfcb3cda9ce
01020304050607080102030405060708a9b2554b0944118061212098f2f238cd
Invalid
01020304050607080102030405060708a9b2554b0944118061212098f2f23800
Invalid
01020304050607080102030405060708a9b2554b0944118061212098f2f238c1
Invalid
01020304050607080102030405060708a9b2554b0944118061212098f2f23801
Invalid
01020304050607080102030405060708a9b2554b0944118061212098f2f23802
Invalid
01020304050607080102030405060708a9b2554b0944118061212098f2f23803
Invalid
01020304050607080102030405060708a9b2554b0944118061212098f2f23804
Invalid
01020304050607080102030405060708a9b2554b0944118061212098f2f23805
Invalid
```

理論上往下找就會找到正確的，但我目前試不出來

4.2 Deriving the Plaintext Manually

K = 1



```
seed@seedlab:/home/a0320506/HW2/Labsetup-2-9$ python3 manual_attack.py
C1:  a9b2554b0944118061212098f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0xcf
CC1: 000000000000000000000000000000cf
P2:  00000000000000000000000000000000
```