

## #Assignment 4

通訊碩一 112523059 馬寧

題目：

### 作業四

- 問題描述：於 Linux 環境，以 C 語言實作 AES 演算法之加密與解密函式，並利用此函式實作可以對一檔案加密與解密應用程式。

系統參數：

- Key = 區塊長度 (為 128 或 256)，
- 應用程式對檔案加密，支援使用 CBC 與 CTR。

程式支援 command line CLI 操作，例如使用 CBC 的演算法：

> `aes_enc "filename" "key" "CBC"`

> `aes_dec "filename" "key" "CBC"`

註：注意資料長度不足 `block length` 的特殊處理做法，網路上面有答案。

- Deadline: 12/06

實現方法：

#### A. Code

- 設定基本參數 `Nr`、`Nb`、`Nk`、`BLOCK_SIZE`、`ROUNDKEY_SIZE`

依 AES 標準規範，若依金鑰輸入長度分類，可分：AES-128, AES-192 及 AES-256。

表 2.1 運算回合數  $N_r$  與  $N_b$  和  $N_k$  之關係

	金鑰區塊數目 ( $N_k$ )	加解密區塊數目 ( $N_b$ )	運算回合次數 ( $N_r$ )
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

AES-128

( $N_b = 4$ ,  $N_k = \text{keysize} / 32 = 4$ ,  $N_r = N_k + 6 = 10$ ,  $\text{BLOCK\_SIZE} = 16$ ,  $\text{ROUNDKEY\_SIZE} = \text{BLOCK\_SIZE} * (N_r + 1) = 176$ )

```
5 #define Nb 4
6 #define Nk 4 // keysize / 32, keysize = 128 in this case
7 #define Nr 10 // Nr = Nk + 6
8 #define BLOCK_SIZE 16 // AES block size in bytes
9 #define ROUNDKEY_SIZE 176 // BLOCK_SIZE * (Nr+1)
```

- 加密主要包含 SubBytes、ShiftRows、MixColumns 及 AddRoundKey，四個步驟

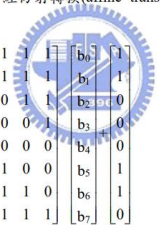
AES 加密演算步驟：在執行第一個回合之前，先把明文與初始金鑰經過 AddRoundKey 運算。再經過( $N_r - 1$ )回合運算，每回合會運用到的四個函數：SubBytes、ShiftRows、MixColumns 及 AddRoundKey。最後一個回合運算省略 MixColumns，只經過 SubBytes、ShiftRows 及 AddRoundKey 運算。

## ☆ SubBytes

### 2.2.1 SubBytes 函數

SubByte 函數轉換是將狀態矩陣(state)，經過下列兩個步驟：

1. 首先對狀態矩陣每一 byte 求出其在有限場乘法反元素。
2. 將第一步運算結果，經仿射轉換(affine transformation)，如下列數學公式 2-2。


$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \quad (2-2)$$

上述兩步驟，可以簡化成S-box表格(表 2.2)。S-box 是一個包含了 256 個byte數值的表格；查詢時我們將每個byte的最高 4 個位元拿來當作列的索引，每個byte的最低 4 個位元當作行的索引，查出所對應的數值。例如：若 $s_{3,0}=(57)_{16}$ ，最高 4 個位元為 5，所以查表第 5 列；最低 4 個位元為 7，所以查表第 7 行，利用S-box(表 2.2)查第 5 列第 7 行對應到 $(5b)_{16}$ ，因此我們知道經S-box轉換可以得到 $s'_{3,0}=(5b)_{16}$ 。

表 2.2 S-box 位元轉換對照表

	Y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
X	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3
	8	cd	0c	13	ec	5f	97	44	17	e4	a7	7e	3d	64	5d	19
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae
	c	ba	78	25	2e	1c	a6	b4	e6	e8	dd	74	1f	4b	bd	8b
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	e1	1d
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb

## ☆ ShiftRows

### 2.2.2 ShiftRows 左旋轉位移函數

ShiftRows 函數為一向左旋轉位移函數。Shiftrows 就是將狀態矩陣的每一列分別做不同程度的旋轉位移。第一列不做任何動作外，第二列向左旋轉位移一個位元組(byte)，第三列向左旋轉位移兩個位元組(byte)，第四列向左旋轉位移三個位元組(byte)。

## ☆ MixColumns

### 2.2.3 MixColumns 函數

MixColumns 是將狀態矩陣的每一行是被視為在 $GF(2^8)$ 中的多項式，乘上一固定多項式 $\alpha(x)=\{03\}x^3+\{01\}x^2+\{01\}x+\{02\}$ 之後，如果發生溢位則同餘 $(x^4+1)$ 。

我們可將其簡化為矩陣乘法，令 $s'(x)=\alpha(x)\otimes s(x)$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad (2-3)$$

展開得到如下式子：

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c})$$

## ☆ AddRoundKey

### 2.2.4 AddRoundKey 函數

AddRoundKey 主要運算是將狀態矩陣(state)與每回合運算出來的子金鑰執行互斥或的運算。每回合子金鑰產生是經由初始密鑰經過金鑰排程(key schedule)所產生。

- 解密主要包含 SubBytes、ShiftRows、MixColumns 及 AddRoundKey，四個步驟

AES 解密使用的回合子金鑰，與加密使用的回合子金鑰相同，只是順序相反。  
 AES 解密演算法步驟：執行第一個回合前，先將密文與回合子金鑰執行 AddRoundKey。  
 再經過(Nr-1)回合運算(round)，每回合運用到四個函數：InvShiftRows、InvSubBytes、  
 AddRoundKey 及 InvMixColumns。最後一個回合省略 InvMixColumns，只經過  
 InvSubBytes、InvShiftRows 及 AddRoundKey 運算。

## ✧ InvSubBytes

### 2.3.1 InvSubBytes 函數

InvSubBytes 函數轉換可經由查 Inverse S-box 表格(如表 2.4)得到。

表 2.3 Inverse S-box 位元轉換對照表

	Y																
	0	1	2	3	4	5	6	7	8	9	a	b	C	d	e	f	
X	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	e1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	af	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Inverse S-box 表格產生如下過程：

InvSubByte 函數轉換是將狀態矩陣(state)每一位元組(byte)，經過下列兩個

步驟完成：

1. 首先將狀態矩陣每一位元組(byte)乘以一個反轉換陣列，如下列數學公式

2-4。

2. 求出其乘法反元素。

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{pmatrix} \quad (2-4)$$

## ✧ InvShiftRows

InvShiftRows 右旋轉函數是將狀態矩陣往右旋轉位移。第一列不做任何動作外，第二列向右旋轉位移一個位元組(byte)，第三列向右旋轉位移兩個位元組(byte)，第四列向右旋轉位移三個位元組(byte)。

## ✧ InvMixColumn

### 2.3.3 InvMixColumn 反混淆運算

InvMixcolumn 反混淆運算是將狀態矩陣的每一行是被視為在  $GF(2^8)$  中的多項式，乘上一固定多項式  $\alpha^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$  之後如果發生溢位則同餘  $(x^4+1)$ 。其中  $\alpha^{-1}(x)$  必需符合下列關係  $\alpha^{-1}(x) \bullet \alpha(x) = 1$ ，其中  $\alpha^{-1}(x)$  與 2.2.3 節  $\alpha(x)$  多項式互為乘法反元素。

可將其簡化為矩陣乘法，令  $s'(x) = \alpha^{-1}(x) \otimes s(x)$

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 0e & 09 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \quad (2-5)$$

$$S'_{0,c} = (\{0e\} \bullet S_{0,c}) \oplus (\{0b\} \bullet S_{1,c}) \oplus (\{0d\} \bullet S_{2,c}) \oplus (\{09\} \bullet S_{3,c})$$

$$S'_{1,c} = (\{09\} \bullet S_{0,c}) \oplus (\{0e\} \bullet S_{1,c}) \oplus (\{0b\} \bullet S_{2,c}) \oplus (\{0d\} \bullet S_{3,c})$$

$$S'_{2,c} = (\{0d\} \bullet S_{0,c}) \oplus (\{09\} \bullet S_{1,c}) \oplus (\{0e\} \bullet S_{2,c}) \oplus (\{0b\} \bullet S_{3,c})$$

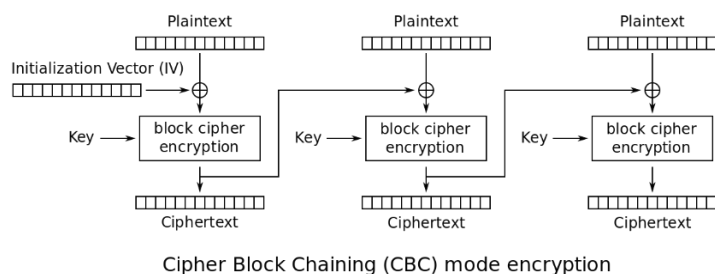
$$S'_{3,c} = (\{0b\} \bullet S_{0,c}) \oplus (\{0d\} \bullet S_{1,c}) \oplus (\{09\} \bullet S_{2,c}) \oplus (\{0e\} \bullet S_{3,c})$$

## ● CBC algorithm

上述四個步驟之加解密運算位於下方的 block cipher

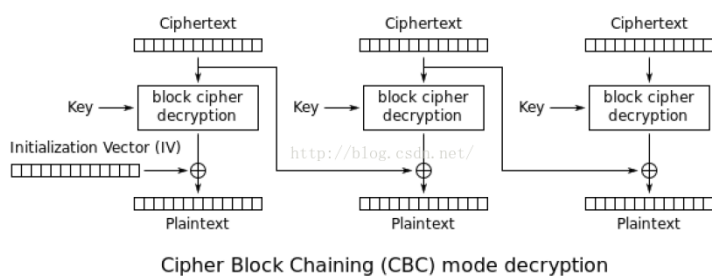
### 加密

CBC algorithm 先將 plaintext block 與 iv 或上一次的 ciphertext block 作 XOR 後，再進行上述 AES 的加密步驟，產生 ciphertext block，並保存該次 ciphertext block 以作為下一次加密前作 XOR 的元素。



### 解密

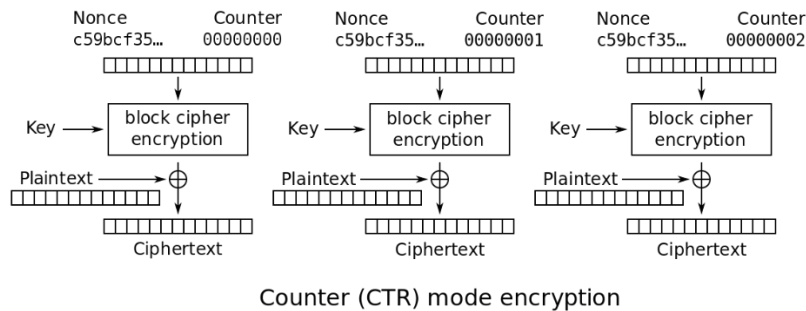
先將 ciphertext block 進行上述 AES 的解密步驟，並保存該次 ciphertext block 以作為下次解密後作 XOR 的元素，解密後與 iv 或上一次的 ciphertext block 作 XOR，產生 plaintext block。



- CTR algorithm

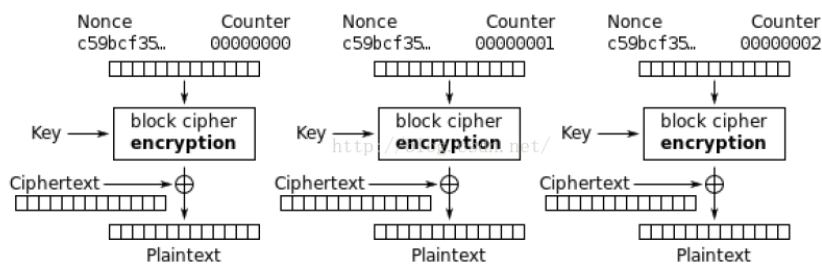
加密

CTR algorithm 先將 iv (nonce 和 counter 的組合，nonce 為隨機產生，counter 從 0 開始每次加 1)作為 input 進行上述加密步驟後，再與 plaintext block 作 XOR 得出 ciphertext block。



解密

先將 iv (nonce 和 counter 的組合，nonce 為隨機產生，counter 從 0 開始 每次加 1)作為 input 進行上述加密步驟後，再與 ciphertext block 作 XOR 得出 plaintext block。



## B. Execute the program

### (a) Enter the required data

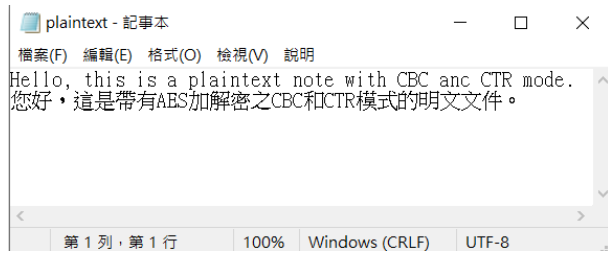
需先得知欲做加密或解密、針對哪個 file 進行加/解密、欲使用的 key 及演算法(CBC or CTR mode)。

```
PS D:\文件\碩一上\課程\無線網路協定\HW\HW4\112523059_馬寧_HW4\src> ./main
Enter aes_enc or aes_dec to encrypt or decrypt the file: aes_enc
Enter plaintext file name to encrypt => plaintext.txt
Enter the Ciphertext file name to write out the cipher => ciphertext.txt
Enter the Decrypted file name to write out the decrypt => decrypted.txt
Enter the key (length 16): abcdefghijklmnop
Enter AES mode (Only CBC or CTR): CBC
```

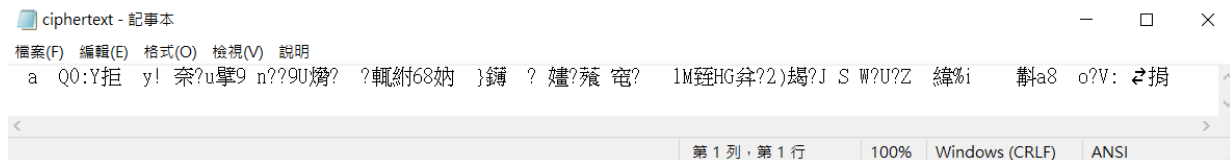
### (b) Result

#### (i) CBC mode

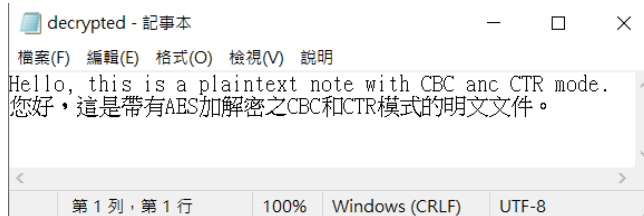
plaintext.txt



## ciphertext.txt

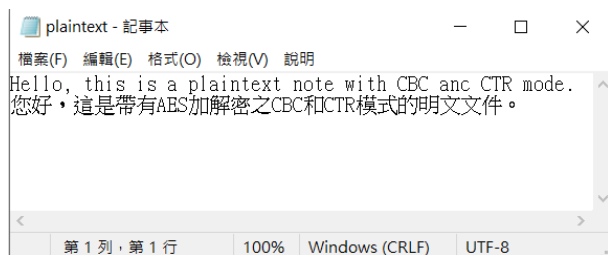


## decrypted.txt

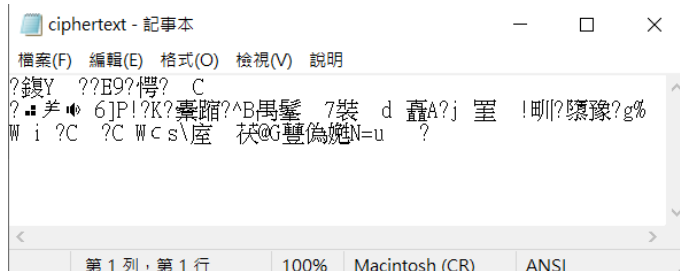


## (ii) CTR mode

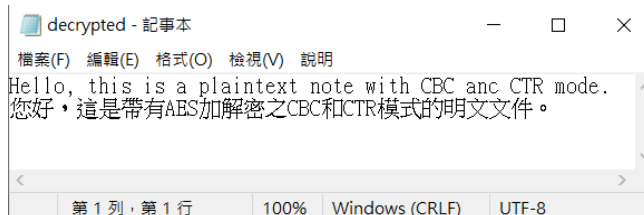
### plaintext.txt



### ciphertext.txt



### decrypted.txt



資料來源：

AES Background Knowledge：<https://ir.nctu.edu.tw/bitstream/11536/41079/3/751403.pdf>

AES Code Reference：<https://github.com/Yunying/Cryptography-AES-implement-in-C/blob/master>

CBC and CTR Encryption Information：<https://ithelp.ithome.com.tw/m/articles/10249953>

CBC and CTR Decryption Information：[https://blog.csdn.net/Lv\\_Victor/article/details/50973330](https://blog.csdn.net/Lv_Victor/article/details/50973330)