Homework-3

Mustafa Tokat

January 10, 2021

1 Introduction

I analyzed several paper together with our lectures during my homework. I studied to understand those, followed step by step entire of the process. Now, we have to 5 problems and will try to solve step by step these.

2 Analysis of problems

2.1 Problem 1

2.1.1

IP(1248842112488421): In this process only place of values change, not themselves of values. We can figure out it with a little codes snippet.

Listing 1: for to permute IP

```
def permutation(ptext, table, nbit):
  permuted = ""
  for i in range(0, nbit):
      permuted = permuted + ptext[table[i]-1]
  return permuted
```

Result: 2211448844882211

2.1.2

E(84211248): By changing the parameters of the code snippet above, we can find the E permutation.

Result: 4081028A4251

2.1.3

P(12488421): Similarly;

Result: 41010C4A

2.1.4

(4) Si(110011) for i = 1; 2; : : : ; 8: We can use below codes:

Listing 2: for to S-Box

```
for i in range(0,8):
          val = S_BOX[i][3][9]
          sboxTable = sboxTable + dec2bin(val)
```

Result: 1011 0110 1111 0100 1111 1110 0101 1100

2.2 Problem 2

2.2.1

Suppose that $K_1 = K_2 = \cdots = K_{16}$. Show that all bits in C_1 are equal and all bits in D_1 are equal

Key scheduling is exactly independent a process from making ciphertext. The first half of the 56-bit key is left, also its second half is right are called as. At the same time, entire of the this process is logic and can be inversion.Now, we can establish the logical connection. K_i values is generated after from come side by side C_i and D_i values,and this process cocnsists two of two steps. (a) Left-shift and (b) PC2. C_i and D_i values are make left-shift for each round. After new values are permuted with PC2. According to our suppose, if entire of the our keys are same, this mean entire of C_i are same and also D_i are same. I encrypted the 'CryptoEn' to explain with an example. The result is as follow.

```
Plaintext giriniz: CryptoEn
C 1 :  FFFFFFF D 1 :  FFFFFFF K 1 :  FFFFFFFFFFFF
C 2 :  FFFFFFF D 2 :  FFFFFFF K 2 :  FFFFFFFFFFFF
C 3 :  FFFFFFF D 3 :  FFFFFFF K 3 :  FFFFFFFFFFFF
C 4 :  FFFFFFF D 4 :  FFFFFFF K 4 :  FFFFFFFFFFFF
C 5 :  FFFFFFF D 5 :  FFFFFFF K 5 :  FFFFFFFFFFFF
C 6 :  FFFFFFF D 6 :  FFFFFFF K 6 :  FFFFFFFFFFFF
C 7 :  FFFFFFF D 7 :  FFFFFFF K 7 :  FFFFFFFFFFFF
C 8 :  FFFFFFF D 8 :  FFFFFFF K 8 :  FFFFFFFFFFFF
C 9 :  FFFFFFF D 9 :  FFFFFFF K 9 :  FFFFFFFFFFFF
C 10 :  FFFFFFF D 10 :  FFFFFFF K 10 :  FFFFFFFFFFFF
C 11 :  FFFFFFF D 11 :  FFFFFFF K 11 :  FFFFFFFFFFFF
C 12 :  FFFFFFF D 12 :  FFFFFFF K 12 :  FFFFFFFFFFFF
C 13 :  FFFFFFF D 13 :  FFFFFFF K 13 :  FFFFFFFFFFFF
C 14 :  FFFFFFF D 14 :  FFFFFFF K 14 :  FFFFFFFFFFFF
C 15 :  FFFFFFF D 15 :  FFFFFFF K 15 :  FFFFFFFFFFFF
C 16 :  FFFFFFF D 16 :  FFFFFFF K 16 :  FFFFFFFFFFFF
Round  1   00BEA4A3   2EA0A9B5   FFFFFFFFFFFF
Round  2   2EA0A9B5   7520B599   FFFFFFFFFFFF
Round  3   7520B599   0A02D290   FFFFFFFFFFFF
Round  4   0A02D290   344FD080   FFFFFFFFFFFF
Round  5   344FD080   50F9FD50   FFFFFFFFFFFF
Round  6   50F9FD50   F254D34A   FFFFFFFFFFFF
Round  7   F254D34A   DB642A76   FFFFFFFFFFFF
Round  8   DB642A76   189C3893   FFFFFFFFFFFF
Round  9   189C3893   4F603EE3   FFFFFFFFFFFF
Round  10   4F603EE3   1A52C542   FFFFFFFFFFFF
Round  11   1A52C542   2F758704   FFFFFFFFFFFF
Round  12   2F758704   2AEAA316   FFFFFFFFFFFF
Round  13   2AEAA316   AE908DF9   FFFFFFFFFFFF
Round  14   AE908DF9   F310184C   FFFFFFFFFFFF
Round  15   F310184C   6A1126A6   FFFFFFFFFFFF
Round  16   9AC8C838   6A1126A6   FFFFFFFFFFFF
Plaintext :  43727970746F456E
Key       :  FEFEFEFEFEFEFEFE
Ciphertext:  20CA0AD5618B9456
```

Figure 1: if K_i values are the same

2.2.2

Show that there are exactly 4 DES keys for which all round keys are the same. They are called weak DES keys.

If we carefully analyze, we can see that dropped 48-bit of PC2 permutation cocnsists its first half from 1-28nth bits and also second half from 29-48nth bits. Therefore, we can easily say that first 24 bits of the K_i values formed from PC2 are left and second 24 bits of the K_i values formed from PC2 are right. The results is follow:

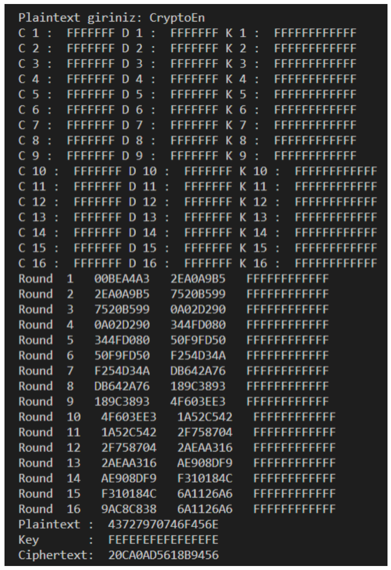
```
Plaintext giriniz: CryptoEn
C 1 :  0000000 D 1 :  FFFFFFF K 1 :  000000FFFFFF
C 2 :  0000000 D 2 :  FFFFFFF K 2 :  000000FFFFFF
C 3 :  0000000 D 3 :  FFFFFFF K 3 :  000000FFFFFF
C 4 :  0000000 D 4 :  FFFFFFF K 4 :  000000FFFFFF
C 5 :  0000000 D 5 :  FFFFFFF K 5 :  000000FFFFFF
C 6 :  0000000 D 6 :  FFFFFFF K 6 :  000000FFFFFF
C 7 :  0000000 D 7 :  FFFFFFF K 7 :  000000FFFFFF
C 8 :  0000000 D 8 :  FFFFFFF K 8 :  000000FFFFFF
C 9 :  0000000 D 9 :  FFFFFFF K 9 :  000000FFFFFF
C 10 :  0000000 D 10 :  FFFFFFF K 10 :  000000FFFFFF
C 11 :  0000000 D 11 :  FFFFFFF K 11 :  000000FFFFFF
C 12 :  0000000 D 12 :  FFFFFFF K 12 :  000000FFFFFF
C 13 :  0000000 D 13 :  FFFFFFF K 13 :  000000FFFFFF
C 14 :  0000000 D 14 :  FFFFFFF K 14 :  000000FFFFFF
C 15 :  0000000 D 15 :  FFFFFFF K 15 :  000000FFFFFF
C 16 :  0000000 D 16 :  FFFFFFF K 16 :  000000FFFFFF
Round  1   00BEA4A3   AE6939E1   000000FFFFFF
Round  2   AE6939E1   21DBEE9D   000000FFFFFF
Round  3   21DBEE9D   3194F799   000000FFFFFF
Round  4   3194F799   0411E67C   000000FFFFFF
Round  5   0411E67C   C8746140   000000FFFFFF
Round  6   C8746140   398CB6BF   000000FFFFFF
Round  7   398CB6BF   38654B59   000000FFFFFF
Round  8   38654B59   0ECBCD0A   000000FFFFFF
Round  9   0ECBCD0A   524E1A19   000000FFFFFF
Round  10   524E1A19   58C91737   000000FFFFFF
Round  11   58C91737   1BA75F33   000000FFFFFF
Round  12   1BA75F33   682193E3   000000FFFFFF
Round  13   682193E3   2E891679   000000FFFFFF
Round  14   2E891679   29EBFFC6   000000FFFFFF
Round  15   29EBFFC6   34998097   000000FFFFFF
Round  16   414BAAEF   34998097   000000FFFFFF
Plaintext :  43727970746F456E
Key       :  1F1F1F1F0E0E0E0E
Ciphertext:  73178335A285512F
```

(a) 1F1F1F1F 0E0E0E0E

```
Plaintext giriniz: CryptoEn
C 1 :  FFFFFFF D 1 :  0000000 K 1 :  FFFFFF000000
C 2 :  FFFFFFF D 2 :  0000000 K 2 :  FFFFFF000000
C 3 :  FFFFFFF D 3 :  0000000 K 3 :  FFFFFF000000
C 4 :  FFFFFFF D 4 :  0000000 K 4 :  FFFFFF000000
C 5 :  FFFFFFF D 5 :  0000000 K 5 :  FFFFFF000000
C 6 :  FFFFFFF D 6 :  0000000 K 6 :  FFFFFF000000
C 7 :  FFFFFFF D 7 :  0000000 K 7 :  FFFFFF000000
C 8 :  FFFFFFF D 8 :  0000000 K 8 :  FFFFFF000000
C 9 :  FFFFFFF D 9 :  0000000 K 9 :  FFFFFF000000
C 10 :  FFFFFFF D 10 :  0000000 K 10 :  FFFFFF000000
C 11 :  FFFFFFF D 11 :  0000000 K 11 :  FFFFFF000000
C 12 :  FFFFFFF D 12 :  0000000 K 12 :  FFFFFF000000
C 13 :  FFFFFFF D 13 :  0000000 K 13 :  FFFFFF000000
C 14 :  FFFFFFF D 14 :  0000000 K 14 :  FFFFFF000000
C 15 :  FFFFFFF D 15 :  0000000 K 15 :  FFFFFF000000
C 16 :  FFFFFFF D 16 :  0000000 K 16 :  FFFFFF000000
Round  1   00BEA4A3   1DB6819C   FFFFFF000000
Round  2   1DB6819C   1FC40C8D   FFFFFF000000
Round  3   1FC40C8D   D8EAC44D   FFFFFF000000
Round  4   D8EAC44D   A613ACFB   FFFFFF000000
Round  5   A613ACFB   18BAC219   FFFFFF000000
Round  6   18BAC219   9775BDCB   FFFFFF000000
Round  7   9775BDCB   31204E4E   FFFFFF000000
Round  8   31204E4E   DBA9CEA7   FFFFFF000000
Round  9   DBA9CEA7   F6C0FEF9   FFFFFF000000
Round  10   F6C0FEF9   83BD9A17   FFFFFF000000
Round  11   83BD9A17   DD8DA216   FFFFFF000000
Round  12   DD8DA216   1C101C56   FFFFFF000000
Round  13   1C101C56   348F42E0   FFFFFF000000
Round  14   348F42E0   95C852EE   FFFFFF000000
Round  15   95C852EE   7DAE4B27   FFFFFF000000
Round  16   13844CA1   7DAE4B27   FFFFFF000000
Plaintext :  43727970746F456E
Key       :  E0E0E0E0F1F1F1F1
Ciphertext:  CB6AB6ACC0A38C31
```

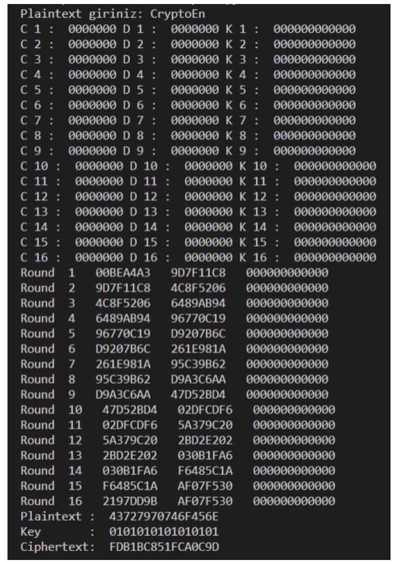
(b) E0E0E0E0 F1F1F1F1

Figure 2: 4 Weak Keys



```
Plaintext giriniz: CryptoEn
C 1 :  FFFFFFF D 1 :  FFFFFFF K 1 :  FFFFFFFFFFFF
C 2 :  FFFFFFF D 2 :  FFFFFFF K 2 :  FFFFFFFFFFFF
C 3 :  FFFFFFF D 3 :  FFFFFFF K 3 :  FFFFFFFFFFFF
C 4 :  FFFFFFF D 4 :  FFFFFFF K 4 :  FFFFFFFFFFFF
C 5 :  FFFFFFF D 5 :  FFFFFFF K 5 :  FFFFFFFFFFFF
C 6 :  FFFFFFF D 6 :  FFFFFFF K 6 :  FFFFFFFFFFFF
C 7 :  FFFFFFF D 7 :  FFFFFFF K 7 :  FFFFFFFFFFFF
C 8 :  FFFFFFF D 8 :  FFFFFFF K 8 :  FFFFFFFFFFFF
C 9 :  FFFFFFF D 9 :  FFFFFFF K 9 :  FFFFFFFFFFFF
C 10 :  FFFFFFF D 10 :  FFFFFFF K 10 :  FFFFFFFFFFFF
C 11 :  FFFFFFF D 11 :  FFFFFFF K 11 :  FFFFFFFFFFFF
C 12 :  FFFFFFF D 12 :  FFFFFFF K 12 :  FFFFFFFFFFFF
C 13 :  FFFFFFF D 13 :  FFFFFFF K 13 :  FFFFFFFFFFFF
C 14 :  FFFFFFF D 14 :  FFFFFFF K 14 :  FFFFFFFFFFFF
C 15 :  FFFFFFF D 15 :  FFFFFFF K 15 :  FFFFFFFFFFFF
C 16 :  FFFFFFF D 16 :  FFFFFFF K 16 :  FFFFFFFFFFFF
Round  1   00BEA4A3   2EA0A9B5   FFFFFFFFFFFF
Round  2   2EA0A9B5   7520B599   FFFFFFFFFFFF
Round  3   7520B599   0A02D290   FFFFFFFFFFFF
Round  4   0A02D290   344FD080   FFFFFFFFFFFF
Round  5   344FD080   50F9FD50   FFFFFFFFFFFF
Round  6   50F9FD50   F254D34A   FFFFFFFFFFFF
Round  7   F254D34A   DB642A76   FFFFFFFFFFFF
Round  8   DB642A76   189C3893   FFFFFFFFFFFF
Round  9   189C3893   4F603EE3   FFFFFFFFFFFF
Round  10   4F603EE3   1A52C542   FFFFFFFFFFFF
Round  11   1A52C542   2F758704   FFFFFFFFFFFF
Round  12   2F758704   2AEAA316   FFFFFFFFFFFF
Round  13   2AEAA316   AE908DF9   FFFFFFFFFFFF
Round  14   AE908DF9   F310184C   FFFFFFFFFFFF
Round  15   F310184C   6A1126A6   FFFFFFFFFFFF
Round  16   9AC8C838   6A1126A6   FFFFFFFFFFFF
Plaintext :  43727970746F456E
Key       :  FEFEFEFEFEFEFEFE
Ciphertext:  20CA0AD5618B9456
```

(a) FEFEFEFE FEFEFEFE



```
Plaintext giriniz: CryptoEn
C 1 :  0000000 D 1 :  0000000 K 1 :  000000000000
C 2 :  0000000 D 2 :  0000000 K 2 :  000000000000
C 3 :  0000000 D 3 :  0000000 K 3 :  000000000000
C 4 :  0000000 D 4 :  0000000 K 4 :  000000000000
C 5 :  0000000 D 5 :  0000000 K 5 :  000000000000
C 6 :  0000000 D 6 :  0000000 K 6 :  000000000000
C 7 :  0000000 D 7 :  0000000 K 7 :  000000000000
C 8 :  0000000 D 8 :  0000000 K 8 :  000000000000
C 9 :  0000000 D 9 :  0000000 K 9 :  000000000000
C 10 :  0000000 D 10 :  0000000 K 10 :  000000000000
C 11 :  0000000 D 11 :  0000000 K 11 :  000000000000
C 12 :  0000000 D 12 :  0000000 K 12 :  000000000000
C 13 :  0000000 D 13 :  0000000 K 13 :  000000000000
C 14 :  0000000 D 14 :  0000000 K 14 :  000000000000
C 15 :  0000000 D 15 :  0000000 K 15 :  000000000000
C 16 :  0000000 D 16 :  0000000 K 16 :  000000000000
Round  1   00BEA4A3   9D7F11C8   000000000000
Round  2   9D7F11C8   4C8F5206   000000000000
Round  3   4C8F5206   6489AB94   000000000000
Round  4   6489AB94   96770C19   000000000000
Round  5   96770C19   D9207B6C   000000000000
Round  6   D9207B6C   261E981A   000000000000
Round  7   261E981A   95C39B62   000000000000
Round  8   95C39B62   D9A3C6AA   000000000000
Round  9   D9A3C6AA   47D52BD4   000000000000
Round  10   47D52BD4   02DFCDF6   000000000000
Round  11   02DFCDF6   5A379C20   000000000000
Round  12   5A379C20   2BD2E202   000000000000
Round  13   2BD2E202   030B1FA6   000000000000
Round  14   030B1FA6   F6485C1A   000000000000
Round  15   F6485C1A   AF07F530   000000000000
Round  16   2197DD9B   AF07F530   000000000000
Plaintext :  43727970746F456E
Key       :  0101010101010101
Ciphertext:  FDB1BC851FCA0C9D
```

(b) 01010101 01010101

Figure 3: 4 Weak Keys

2.2.3

Determine these 4 weak DES keys.

The 4 weak keys were published by NIST in January 2012[1]. These are:

- 01010101 01010101
- FEFEFEFE FEFEFEFE
- E0E0E0E0 F1F1F1F1
- 1F1F1F1F 0E0E0E0E

[1] Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, NIST, January 2012, p.11

2.3 Problem 3

There are other modes of block cipher besides the ones we have learned. One of these modes is named Plaintext Block Chaining (PBC) Mode. On the encryption side, the following is executed to obtain the nth ciphertext: Cn := $E_k(M_n) \oplus M_{n-1}$. Suppose that we need to encrypt M1,...,M5 using the PBC mode. Show the explicit formulas to obtain C1,...,C5. What do you need to use for M0? Also, show the steps on the decryption side to obtain M1,...,M5.

M_0 is Initialization Vector(IV). In this case, we can constitue to encryption and decryption formulas:

- for encryption:
  - $C_1 = E_k(M_1) \oplus \text{IV}$
  - $C_2 = E_k(M_2) \oplus M_1$
  - $C_3 = E_k(M_3) \oplus M_2$
  - $C_4 = E_k(M_4) \oplus M_3$
  - $C_5 = E_k(M_5) \oplus M_4$

Interestingly, we continue toward from start to end in order to decryption, because decryption process continue toward from end to start in some algorithm.

- for decryption:
  - $M_1 = D_k(C_1) \oplus \text{IV}$
  - $M_2 = D_k(C_2) \oplus C_1$
  - $M_3 = D_k(C_3) \oplus C_2$
  - $M_4 = D_k(C_4) \oplus C_3$
  - $M_5 = D_k(C_5) \oplus C_4$

2.4 Problem 4

Consider the AES/Rijndael algorithm and its Galois field GF(2^8)

2.4.1 Compute the sum (a7) + (5c) in GF(2^8)

At first, we should convert these hex values to binary numbers.

$$(\text{a7}) = 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1$$

$$(\text{5c}) = 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0$$

After, we should write to binary numbers as polynomial and should sum to this polynomials(indeed XOR);

$$(x^7 + x^5 + x^2 + x^1 + 1) \oplus (x^6 + x^4 + x^3 + x^2)$$

$$= x^7 + x^6 + x^5 + x^4 + x^3 + x^1 + 1$$

Now, we should convert these polynomials to binary numbers.

$$= \underbrace{1111}_{\text{f}}\ \underbrace{1011}_{\text{b}}$$

Result = (a7) + (5c) = (fb)

2.4.2 Compute the product (a7) × (5c) in GF(2^8)

At first, we should convert these hex values to binary numbers.

(a7) = 1 0 1 0 0 1 1 1

(5c) = 0 1 0 1 1 1 0 0

Besides, I want to solve with two method.

First method

$$(x^7 + x^5 + x^2 + x^1 + 1) \text{ X } (x^6 + x^4 + x^3 + x^2)$$

$$= x^{13} + x^{11} + x^{10} + x^9 + x^{11} + x^9 + x^8 + x^7 + x^8 + x^6 + x^5 + x^4 + x^7 + x^5 + x^4 + x^3 + x^6 + x^4 + x^3 + x^2$$

$$= x^{13} + x^{10} + x^4 + x^2$$

And now, I will divide to p(x) = $x^8 + x^4 + x^3 + x + 1$

$$\begin{array}{rrrrrrrrrrr}
x^{13} & +x^{10} & & & & & +x^4 & & +x^2 & & \\
-x^{13} & & -x^9 & -x^8 & -x^6 & -x^5 & & & & & \\
\cline{1-6}
& x^{10} & -x^9 & -x^8 & -x^6 & -x^5 & +x^4 & & +x^2 & & \\
& -x^{10} & & & -x^6 & -x^5 & & -x^3 & -x^2 & & \\
\cline{2-9}
& & -x^9 & -x^8 & -2x^6 & -2x^5 & +x^4 & -x^3 & & & \\
& & x^9 & & & +x^5 & +x^4 & & +x^2 & +x & \\
\cline{3-10}
& & & -x^8 & -2x^6 & -x^5 & +2x^4 & -x^3 & +x^2 & +x & \\
& & & x^8 & & & +x^4 & +x^3 & & +x & +1 \\
\cline{4-11}
& & & & -2x^6 & -x^5 & +3x^4 & & +x^2 & +2x & +1
\end{array}
\qquad
\begin{array}{|l}
x^8 + x^4 + x^3 + x + 1 \\
\hline
x^5 + x^2 - x - 1
\end{array}$$

$=x^5 + x^4 + x^2 + 1$

Now, we will convert it to binary numbers;

$= \underbrace{0011}_{3}\underbrace{0101}_{5}$

<u>Result:</u> (a7) x (5c) = (35)

<u>Second method</u>

At first, we are convert to $x^{13} + x^{10} + x^4 + x^2$ polynomial to binary numbers and will write its under;

1 0 0 1 0 0 0 0 0 1 0 1 0 0

<u>1 0 0 0 1 1 0 1 1</u>

0 0 0 1 1 1 0 1 1 1 0 1 0 0

<u>0 0 0 1 0 0 0 1 1 0 1 1</u>

0 0 0 0 1 1 0 0 0 1 1 0

0 0 0 0 1 0 0 0 1 1 0 1 1

0 0 0 0 0 1 0 0 1 0 1 1 1 0

0 0 0 0 0 1 0 0 0 1 1 0 1 1

0 0 0 0 0 0 0 0 1 1 0 1 0 1

We take to last 8 numbers and convert it hex number;

$\underbrace{0011}_{3}\ \underbrace{0101}_{5}$

Result: (a7) x (5c) = (35)

2.4.3 Compute S(a7)

I will explain in details solving of this question in next question, for this reason I prefer to solve by looking to the table.

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 4: The SubByte S-Box

Result: S(a7) = (5c)

2.4.4 Compute S^{-1}(5c)

I will do it this way: at first, I will find corresponding of (5c) value from Sub-bytes table . Next, I will compute multiplicative inverse in GF(2^8) of this value.

Also, I will try to find with c = A.b + d formula. In this formula, because of a(x) $\neq$ 0, we will compute its multiplicative inverse. After, of its the result will be equal to b(x). Then, we will multiplication of b(x) with a fixed A matrix. Finally, we will add the result of this process also to a fixed b(x) matrix.

$$(5c) = 01011100 \text{ (as an order; } x^7 + x^6 + \implies + x^1 + x^0 \text{)}$$

Listing 3: for polynomial multiplcation inverse in (GF2^8)

```
from sympy import gcdex
from sympy.abc import x

print(gcdex(x**6 + x**4 + x**3 + x**2, x**8 + x**4 + x**3 + x + 1))
```

$$a^{-1} = \text{b(x)} = x^6 + x^4 + 1$$

$$\text{b(x)} = 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1$$

And we will think as a vector;

$$b(x) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Well, let's multiplicative with fixed A matrix and its the result add with fixed d vector;

$$\begin{bmatrix}1&0&0&0&1&1&1&1\\1&1&0&0&0&1&1&1\\1&1&1&0&0&0&1&1\\1&1&1&1&0&0&0&1\\1&1&1&1&1&0&0&0\\0&1&1&1&1&1&0&0\\0&0&1&1&1&1&1&0\\0&0&0&1&1&1&1&1\end{bmatrix} \text{X} \begin{bmatrix}1\\0\\0\\0\\1\\0\\1\\0\end{bmatrix} \oplus \begin{bmatrix}1\\1\\0\\0\\0\\1\\1\\0\end{bmatrix} =$$

$$S(5c) = \begin{bmatrix}0\\1\\0\\1\\0\\0\\1\\0\end{bmatrix}$$

I will see as a polynomial to this value. Then;

$$= 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0 = x^6 + x^3 + \text{x}.$$

Well, finally, I will compute its multiplicative inverse with the same code snippet.

Listing 4: for polynomial multiplcation inverse in ($\text{GF}2^8$)

```
from sympy import gcdex
from sympy.abc import x

print(gcdex(x**6 + x**3 + x, x**8 + x**4 + x**3 + x + 1))
```

Result: $x^7 + x^5 + x^3 + \text{x} + 1$

2.5 Problem 5

The definition of the SubBytes function is given as

$$\mathbf{f(a) = A}\ a^{-1} \oplus \mathbf{d}$$

for a fixed 8 x 8 matrix A and 8 x 1 vector d. Let a = x + 1. Compute f(a) using the SubBytes lookup table and then using the above definition. Show that these values are equal.

Also, as like above process, I will benefit from the same code snippet. But first up, I need to find looking to the polynomial from Sub-Byte Table

$$\text{a(x+1)} = \underbrace{0000}_{0}\underbrace{0011}_{3}\text{(as binary)} = (03)$$
$$= \textbf{(7b) from Sub-byte table}$$

Listing 5: for polynomial multiplcation inverse in (GF2^8)

```
from sympy import gcdex
from sympy.abc import x

print(gcdex(x + 1, x**8 + x**4 + x**3 + x + 1))
```

$$\text{b(x)} = x^7 + x^6 + x^5 + x^4 + x^2 + \text{x}$$

$$\text{b(x)} = 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\text{(as binary)}$$

Then, we will multiplication of b(x) with a fixed A matrix. Finally, we will add the result of this process also to a fixed b(x) matrix.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \text{X} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} =$$

$$f(f) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Now, we convert it;

$$\mathrm{f(a)} = \underbrace{0111}_{7}\underbrace{1011}_{\mathrm{b}}$$

Result: As it is seen, result of the process and to f(a) Sub-byte table value are equal.