uc3m | Universidad Carlos III de Madrid

CRYPTOGRAPHY AND COMPUTER SECURITY

Ana I. González-Tablas Ferreres
José María de Fuentes García-Romero de Tejada
Lorena González Manzano
Sergio Pastrana Portillo
UC3M | COMPUTER SECURITY LAB (COSEC) GROUP

"Symmetric Encryption: Block ciphers"

Proposed exercises

Exercise 1 :

Assume the following DES key:
10000101 10100100 10001111 10001111 10000101 10100100 10001111 10001111.

a) Compute the first internal subkey generated by the algorithm to encrypt a cleartext.

b) Compute L1 y R1 for the following cleartext: **10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010**

Solution:

a)

1) Initial key:

1-8: 1 0 0 0 0 1 0 1
9-16:1 0 1 0 0 1 0 0
17-24: 1 0 0 0 1 1 1 1
25-32: 1 0 0 0 1 1 1 1
33-40: 1 0 0 0 0 1 0 1
41-48: 1 0 1 0 0 1 0 0
49-56: 1 0 0 0 1 1 1 1
57-64: 1 0 0 0 1 1 1 1.

Key after first permutation PC-1:

1	**1**	**1**	**1**	**1**	**1**	**1**
1	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**1**	**0**	**0**
0	**1**	**0**	**0**	**0**	**0**	**0**
1	**1**	**0**	**0**	**1**	**1**	**0**
0	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**0**	**0**	**1**
1	**0**	**0**	**0**	**0**	**0**	**0**

2) Left shift one position on each half.

C0: 1111111 1000000 0000100 0100000

C0 after shift: 1111111 0000000 0001000 1000001

D0: 1100110 0111111 1111001 1000000

D0 after shift: 1001100 1111111 1110011 0000001

3) Second permutation PC-2, reduces key to 48 bits, being the result:

000011 110100 000100 010001 100100 010111 111100 010111

b)

1. Initial permutation IP, obtaining $\mathbf{L_0}$ y $\mathbf{R_0}$

1-8: **10101010**
9-16: **10101010**
17-24: **10101010**
25-32: **10101010**
33-40: **10101010**
41-48: **10101010**
49-56: **10101010**
57-64: **10101010**

	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$\mathbf{L_0}$	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**
$\mathbf{R_0}$	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**

2. Compute the output of the E box (expansion) taking as input $\mathbf{R_0}$

Output of E-box:

111111

111111

111111

111111

111111

111111

111111

111111

3. Next, we combine the bits from the E-box XORing with bits from the internal key (generated in previous step), obtaining the input bits to S-box.

Subkey	**E box**	Output of E-box =Input to S-box
000011	**111111**	**111100**
110100	**111111**	**001011**
000100	**111111**	**111011**
010001	**111111**	**101110**
100100	**111111**	**011011**
010111	**111111**	**101000**
111100	**111111**	**000011**
010111	**111111**	**101000**

4. We obtain the outputs of the S-boxes

S1: 5 = 0101;	S2: 2 = 0010;	S3: 5 = 0101;	S4: 13 = 1101
S5: 9 = 1001;	S6: 2 = 0010;	S7: 0 = 0000;	S8: 9 = 1001

Then, we obtain the following output: 0101 0010 0101 1101 1001 0010 0000 1001

5. We obtain the P box output:

1110 1101 0010 0001 1001 1000 0100 0010

6. The output is then XORed with L_0 to obtain R_1:

P-box output
1110 1101 0010 0001 1001 1000 0100 0010

L_0
0000 0000 0000 0000 0000 0000 0000 0000

R_1
1110 1101 0010 0001 1001 1000 0100 0010

7. L_1 is R_0. Thus, we finally obtain

L_1 = R_0 (from step 1)	**R_1**
1111 1111 1111 1111 1111 1111 1111 1111	**1110 1101 0010 0001 1001 1000 0100 0010**

Exercise 2:

Consider a DES cipher in CBC mode, and the following data:

The cleartext message M = **10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010 01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101**

The initial value for the registry **Co=11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000**

c) Compute the input value to the S-BOX in the first iteration, assuming that there the IP permutation is not performed, and the first internal subkey is **k1= 000000 111111 000000 111111 000000 111111 000000 111111.**

d) Assuming that, after the first iteration of the encryption process, the output of the cipher is **C1= 01010101 01010101 01010101 01010101 01010101 01010101 01010101**, compute the input to the block cipher in the next iteration.

e) Suppose that C1 is sent over a communication line, and that there is a transmission error which affects 2 bits of this block. Explain and reason how this error would affect the decryption of the message.

Solution:

a)

1. First, we perform the XOR between the first block with C_o, which is $M_1 \oplus C_o$ = 01010101 10101010 01010101 10101010 01010101 10101010 01010101 10101010. This is the input to the DES cipher.

2. Divide the input into Lo y Ro

0 1 0 1 0 1 0 1	
1 0 1 0 1 0 1 0	Lo
0 1 0 1 0 1 0 1	
1 0 1 0 1 0 1 0	
0 1 0 1 0 1 0 1	
1 0 1 0 1 0 1 0	
0 1 0 1 0 1 0 1	Ro
1 0 1 0 1 0 1 0	

3. We obtain the output of E-box from Ro

Output of E-box	
0 0 1 0	1 0
1 0 1 0	1 1
1 1 0 1	0 1
0 1 0 1	0 0
0 0 1 0	1 0
1 0 1 0	1 1
1 1 0 1	0 1
0 1 0 1	0 0

4. Next, we XOR the output bits from E-Blox with the bits from the internal key, obtaining the input bits to S-box.

Key	Output of E-box	Input to S-BOX
000000	0 0 1 01 0	0 0 1 01 0
111111	1 0 1 01 1	0 1 0 10 0
000000	1 1 0 10 1	1 1 0 10 1
111111	0 1 0 10 0	1 0 1 01 1
000000	0 0 1 01 0	0 0 1 01 0
111111	1 0 1 01 1	0 1 0 10 0
000000	1 1 0 10 1	1 1 0 10 1
111111	0 1 0 10 0	1 0 1 01 1

b) We are asked for the calculus of $M_2 \oplus C_1$.
Since the two blocks are the same, the result is 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

c) Given that $M_i = D(C_i, K) \oplus C_{i-1}$, an error in the block C_1 would affect the decryption of blocks M_1 and M_2. $M_1 = D(C_1, K) \oplus C_0$ would be affected at a great extent in their bits, with respect to what would've been received in an error-free transmission. This is due to the *avalanche effect* of DES. $M_2 = D(C_2, K) \oplus C_1$ would be affected in just two bits, concretely those in the positions of the errors from the transmission of C_1

Exercise 3:

We know that a user's DES key is composed by 8 symbols from an alphabet of 26 letters.

Considering that the time needed to test one single key is 1 microsecond, calculate:

a) The time needed to break a cryptogram.

b) The time needed, assuming an alphabet that also includes digits.

Solution:

a) The problem is reduced to calculate the permutation of 26 elements taken eight at a time, i.e, 26^8 = 208827064576 microseconds, or equivalent 2,41 days.

b) Now it's necessary to calculate P(36,8) = 368 = 2821109907456 microseconds = 32,65 days.

Exercise 4:

Given the following intermediate AES state 3 (i.e., the output of the ShiftRows function), calculate the byte from row 1, column 0 (consider that the byte D4 is in position r0,c0):

D4	E0	B8	1E
BF	B4	41	27
5D	52	11	98
30	AE	F1	E5

Solución:

It is necessary to perform the following operation to get the corresponding result for each new byte of the status matrix. Note that it is a combination of several bytes from different rows of the original matrix. We show only the result for r’1,0:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} D4 \\ BF \\ 5D \\ 30 \end{bmatrix}$$

$r'_{1,0} = \{D4\} \oplus (\{02\} \bullet \{BF\}) \oplus (\{03\} \bullet \{5D\}) \oplus \{30\}$

Calculus:

{D4} = $\mathbf{x^7 + x^6 + x^4 + x^2}$

{02} • {BF} = $x\,(x^7 + x^5 + x^4 + x^3 + x^2 + x + 1) =$ $\mathbf{x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + x}$

({03} • {5D}) $= (x+1)\,(x^6 + x^4 + x^3 + x^2 + 1) = x^7 + x^5 + x^4 + x^3 + x + x^6 + x^4 + x^3 + x^2 + 1 =$ $\mathbf{x^7 + x^6 + x^5 + x^2 + x + 1}$

{30} = $\mathbf{x^5 + x^4}$ Thus, the result is:

$r'_{1,0} =$ $\mathbf{(x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + 1) \bmod. (x^8 + x^4 + x^3 + x + 1) = x^6 + x^5 + x^2 + x = 66}$

Exercise 5:

AES SubByte function is a non-linear substitution which is applied independently to every byte within the status matrix (intermediate status 1). For this purpose, the S- BOX substitution table is employed. This table is build using two different transformations

a) First: Calculate the multiplicative inverse of that byte with respect to the polynomial

   $m(x) = x^8 + x^4 + x^3 + x + 1$

b) Second: Apply the following transformation:

$$\begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} = \begin{pmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

where xi bits are parts of the result of the first transformation and yi are the resulting bits of the second transformation (note: subindex 0 indicates the least significant bit)

Suppose the byte A=10001000. Get the resulting byte using the transformations previously described. Check the resulting value using the S-BOX table below.

		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
x	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

Solution:

First Part

- First transformation

Byte A=10001000 corresponds to the polynomial $a(x)= x^7 + x^3$. Now it is necessary to calculate the multiplicative inverse of this polynomial with respect to m(x). For this purpose, the Euclidean extended algorithm can be employed:

$x^8 + x^4 + x^3 + x + 1= x (x^7 + x^3) + x^3 + x + 1$
$x^7 + x^3= (x^4 + x^2 + x)(x^3 + x + 1) + x$
$x^3 + x + 1= (x^2 + 1) x + 1$, and so,
$1= (x^3 + x + 1) - (x^2 + 1) x = (x^3 + x + 1) - (x^2 + 1) [(x^7 + x^3) - (x^4 + x^2 + x)(x^3 + x + 1)]$
$1= (x^3 + x + 1) - (x^2 + 1)(x^7 + x^3) + (x^6 + x^4 + x^3 + x^4 + x^2 + x) (x^3 + x + 1)$
$1= - (x^2 + 1)(x^7 + x^3) + (x^3 + x + 1) (x^6 + x^3 + x^2 + x +1)$
$1= -(x^2 + 1)(x^7 + x^3) + [(x^8 + x^4 + x^3 + x + 1) - x (x^7 + x^3)] (x^6 + x^3 + x^2 + x +1)$
$1= - (x^2 + 1)(x^7 + x^3) + (x^6 + x^3 + x^2 + x +1) (x^8+x^4 +x^3+x+1) - (x^7 + x^4 + x^3 + x^2 + x) (x^7 + x^3)$
$1= (x^6 + x^3 + x^2 + x +1) (x^8 + x^4 + x^3 + x + 1) - (x^7 + x^3) [(x^2 + 1) + (x^7 + x^4 + x^3 + x^2 + x)]$
$1= (x^6 + x^3 + x^2 + x +1) (x^8 + x^4 + x^3 + x + 1) - (x^7 + x^3) (x^7 + x^4 + x^3 + x +1)$
$1= (x^6 + x^3 + x^2 + x +1) (m(x)) - (a(x)) (x^7 + x^4 + x^3 + x + 1)$
inv $(x^7 + x^3)$ mod. $m(x) = (x^7 + x^4 + x^3 + x +1)$
The resulting inverse is $x^7 + x^4 + x^3 + x + 1$. Thus, the output for this first transformation is **X=10011011**

- Second transformation

Using the X value in the matrix:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

The output is Y = 11000100, which corresponds to the hexadecimal value C4.

SECOND PART (i.e. check the result)

The input to the ByteSub function is A=10001000. The first 4 bits indicate the row, and the remaining 4 indicate the column. Both are referred to the S-Box matrix. The result is:
X=1000-> Row 8
Y=1000->Column 8
Using that matrix the result is the same, C4.

Exercise 6:
The following matrix is the input matrix to the ByteSub function:

$$\begin{pmatrix} 09 & 93 & 19 & 27 \\ AE & 52 & 11 & 9D \\ 19 & 21 & A5 & 9C \\ A9 & CC & 33 & 30 \end{pmatrix}$$

Recall that the ByteSub transformation is based on the following table:

	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

a) Calculate the output status matrix of the ByteSub function.

b) After this function, the ShiftRow function is applied in AES. Calculate the output status matrix of the ShiftRow function

c) Afterwards, the MixColumns function is applied. It is based on this transformation:

$$\begin{pmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{pmatrix}$$

Taking the matrix calculated previously as the input state matrix, calculate the transformation of the column number 0 of that matrix

Solution:

a)

$$\begin{pmatrix} 01 & DC & D4 & CC \\ E4 & 00 & 82 & 5E \\ D4 & FD & O6 & DE \\ D3 & 4B & C3 & 04 \end{pmatrix}$$

b) Each byte is shifted to the left as many positions as indicated by its row number.

$$\begin{pmatrix} 01 & DC & D4 & CC \\ 00 & 82 & 5E & E4 \\ 06 & DE & D4 & FD \\ 04 & D3 & 4B & C3 \end{pmatrix}$$

c)

$$\begin{pmatrix} S'_{0,0} \\ S'_{1,0} \\ S'_{2,0} \\ S'_{3,0} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} * \begin{pmatrix} 01 \\ 00 \\ 06 \\ 04 \end{pmatrix} = \begin{pmatrix} 02 + 06 + 04 \\ 01 + 03 \cdot 06 + 04 \\ 01 + 02 \cdot 06 + 03 \cdot 04 \\ 03 + 06 + 02 \cdot 04 \end{pmatrix} = \begin{pmatrix} x + x^2 + x + x^2 \\ 1 + (x+1)(x^2 + x) + x^2 \\ 1 + x(x^2 + x) + (x+1)x^2 \\ x + 1 + x^2 + x + x^3 \end{pmatrix} = \begin{pmatrix} 0 \\ x^3 + x^2 + x + 1 \\ 1 \\ x^3 + x^2 + 1 \end{pmatrix}$$

This value, expressed in hexadecimal notation, is: $\begin{pmatrix} 00 \\ 0F \\ 01 \\ 0D \end{pmatrix}$