



Universidad
Carlos III de Madrid

COSEC LAB · Dpto. Informática

Universidad Carlos III de Madrid

Ejercicios de cifrado simétrico

Seguridad en las Tecnologías de la Información
Curso 2016/2017

Cifradores de bloque

Parte I - DES

1. Suponiendo que la clave utilizada en el algoritmo DES es: 10000101 10100100 10001111 10001111 10000101 10100100 10001111 10001111.

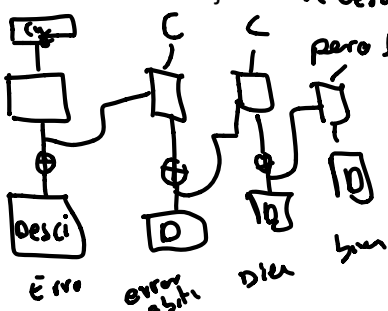
- Calcular la 1ª clave interna que genera el algoritmo, para cifrar un texto en claro.
- Calcular L_1 y R_1 partiendo del mensaje en claro siguiente: 10101010 10101010 10101010 10101010 10101010 10101010 10101010

2. Se dispone de un cifrador DES en modo CBC donde:

El mensaje a cifrar es $M =$ 10101010 10101010 10101010 10101010 10101010 10101010 10101010 01010101 01010101 01010101 01010101 01010101

El valor inicial del registro $C_0 =$ 11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000

- Calcule el valor que habrá a la entrada de la caja S, en la primera iteración, teniendo en cuenta que no se realiza la transformación IP y que el valor de la primera clave interna es $k_1 =$ 000000 111111 000000 111111 000000 111111 000000 111111.
- Suponiendo que después de realizar el primer cifrado tenemos a la salida del cifrador $C_1 =$ 01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101, calcule lo que habrá a la entrada del cifrador, en el cifrado del siguiente bloque.
- Se envía C_1 a través de una línea de comunicación, produciéndose un error que afecta dos bits de este bloque. Explique razonadamente como afectaría esto al descifrado del mensaje.



Al descifrar da error en el propio del error y en el siguiente, pero los demás ya no estarán afectados

Cambiar posiciones, se pasa de 64 a 56 bits segun la matriz

1) a)

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	1
1	1
1	0
0	0

Desplazamiento 1 a la izq cada mitad

...

Juntar ambas mitades siguiendo la tabla de posiciones.

1	2	3	...
...
29	30	31	...
...

b)

L ₀	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
R ₀	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1

⇒ Ampliar
XOR
REDUCIR (Tablas)
Desorden
XOR en L₀

R₁

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

32 → 48 bits

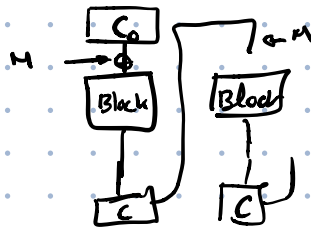
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

XOR con la clave

Tablas S₁ - S₇

Permutación final XOR L₁

2.) DES con CBC



$C_0: L_0:$

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

$R_i:$

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

Esta mal porque no le he hecho xor antes del block, pero proceso correcto.

a) Exp \rightarrow xor $L_i \rightarrow$ Sust \rightarrow ~

1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	1	0	2	5	0	1	1	0
1	0	1	0	1	0	1	1	1	1	0	0	1	0	1	1	0	0	1	0	1	1	1
1	0	1	0	1	0	0	-	-	-	0	1	0	1	0	1	0			1	1	1	1
1	0	1	0	1	0	1	-	-	-	1	0	1	0	1	0	1			0	1	0	1
1	0	1	0	1	0	0	-	-	-	0	1	0	1	0	1	0			1	1	0	1
1	0	1	0	1	0	1	-	-	-	1	0	1	0	1	0	1			0	1	0	0
1	0	1	0	1	0	0	-	-	-	0	1	0	1	0	1	0			0	0	1	1
1	0	1	0	1	0	1	1	1	1	1	0	1	0	1	0	1			1	1	1	0

b) Sale del cifrador y se hace xor con el sig. mensaje.

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

3. Si supiéramos que la clave que un usuario usa en el algoritmo de cifrado DES está compuesta por ocho letras del alfabeto (26 letras), y tomando que el tiempo de cálculo necesario para, haciendo una búsqueda exhaustiva, probar una clave es 1 microsegundo.

Se pide:

- Calcular el tiempo necesario para romper un criptograma.
- Calcularlo también para el caso que el alfabeto sea alfanumérico.

Parte II - AES

1. Dado el Estado Intermedio 3 (salida de la función ShiftRows) en una determinada iteración estándar del algoritmo Rijndael (AES), calcular el byte de la fila 1, columna 0 (el byte D4 del ejemplo ocuparía la posición r0,0):

0			
0	D4	E0	B8
1	<u>BF</u>	B4	41
	5D	52	11
	30	AE	F1
			E5

1011 1111

Add Round
 ByteSub
 ShiftRow
 MixColumn
 AddRoundKey
 ShiftRow
 AddRoundKey

2. La función SubByte de AES es una sustitución no lineal que se aplica a cada byte de la matriz de Estado (Estado Intermedio 1) de forma independiente a través de la tabla de sustitución S-BOX.

Esta tabla se constituye mediante dos transformaciones:

Primero: Se calcula el inverso multiplicativo del byte correspondiente respecto

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

Segundo: Se aplica la siguiente transformación:

$$\begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ h' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ h \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

siendo los b_i bits del byte resultante de la primera transformación e b'_i los bits resultantes de la segunda transformación (el subíndice 0 indica el bit menos significativo)

$$a_{n-1} \quad g_{n-1}$$

$$01011010 \quad 100011011$$

$$2^8-1$$

$$01011010 \bmod 100011011$$

$$2^{254}$$

$$01011010 \bmod 100011011$$

$$(01011010^2)^{127}$$

?

$$(01011010^2)^{127} (01011010)^3$$

$$\begin{array}{r} 01011010 \\ 01011010 \\ \hline \end{array}$$

$$\begin{array}{r} 01011010 \\ 01011010 \\ \hline \end{array}$$

$$\begin{array}{r} 01011010 \\ 01011010 \\ \hline \end{array}$$

$$\begin{array}{r} 01011010 \\ 01011010 \\ \hline \end{array}$$

$$\begin{array}{r} 01011010 \\ 01011010 \\ \hline \end{array}$$

$$\begin{array}{r} 01011010 \\ 01011010 \\ \hline \end{array}$$

$$100011011$$



Dado el byte A=10001000 obtener el byte que obtendríamos con estas dos transformaciones, y comprobar que es el mismo resultado que utilizando la tabla S-Box:

1000 1000 $C4$
 1 8

		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	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	dd	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	b0	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	c1	f8	98	11	69	d9	8c	94	9b	1c	87	c9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

3. Sea la matriz de estado de entrada a la función ByteSub de AES, la siguiente:

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

$\begin{pmatrix} 01 & dc & d4 & cc \\ ae & 00 & 82 & se \\ d4 & fd & 06 & de \\ d3 & 49 & c3 & 04 \end{pmatrix}$

donde se recuerda, que la transformación ByteSub de AES viene dada por la siguiente tabla:

	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

Se pide:

- Halle la matriz de estado a la salida de la función ByteSub.
- A continuación, en AES, se aplica la función ShiftRow. Halle la matriz de estado a la salida de la función ShiftRow.
- Seguidamente, se aplica la función MixColumns dada por la siguiente transformación:

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

Tomando como matriz de estado de entrada, la matriz del resultado anterior, halle la transformación de la columna 0 de dicha matriz.

$$a) \begin{pmatrix} 01 & dc & d4 & cc \\ ae & 00 & 82 & se \\ d4 & fd & 06 & de \\ d3 & 49 & c3 & 04 \end{pmatrix}$$

$$b) \begin{pmatrix} 01 & dc & d4 & cc \\ 00 & 82 & se & ae \\ 06 & de & d4 & fd \\ 04 & d3 & 49 & c3 \end{pmatrix}$$

c)

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} 01 \\ 00 \\ 06 \\ 04 \end{pmatrix} = \begin{pmatrix} 00 \\ 0F \\ 01 \\ 0C \end{pmatrix}$$

$$10 \oplus 110 \oplus 100 = 000 = 00$$

$$1 \oplus 110 \oplus 100 = 1111 = 0F$$

$$1 \oplus 1100 \oplus 1100 = 0001 = 01$$

$$11 \oplus 110 \oplus 1000 = 1101 = 0C$$