El algoritmo DES

Descripción del algoritmo
Debilidades y ataques
Triple DES

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
KISA

Descripción del algoritmo

Data Encryption Standard (DES)

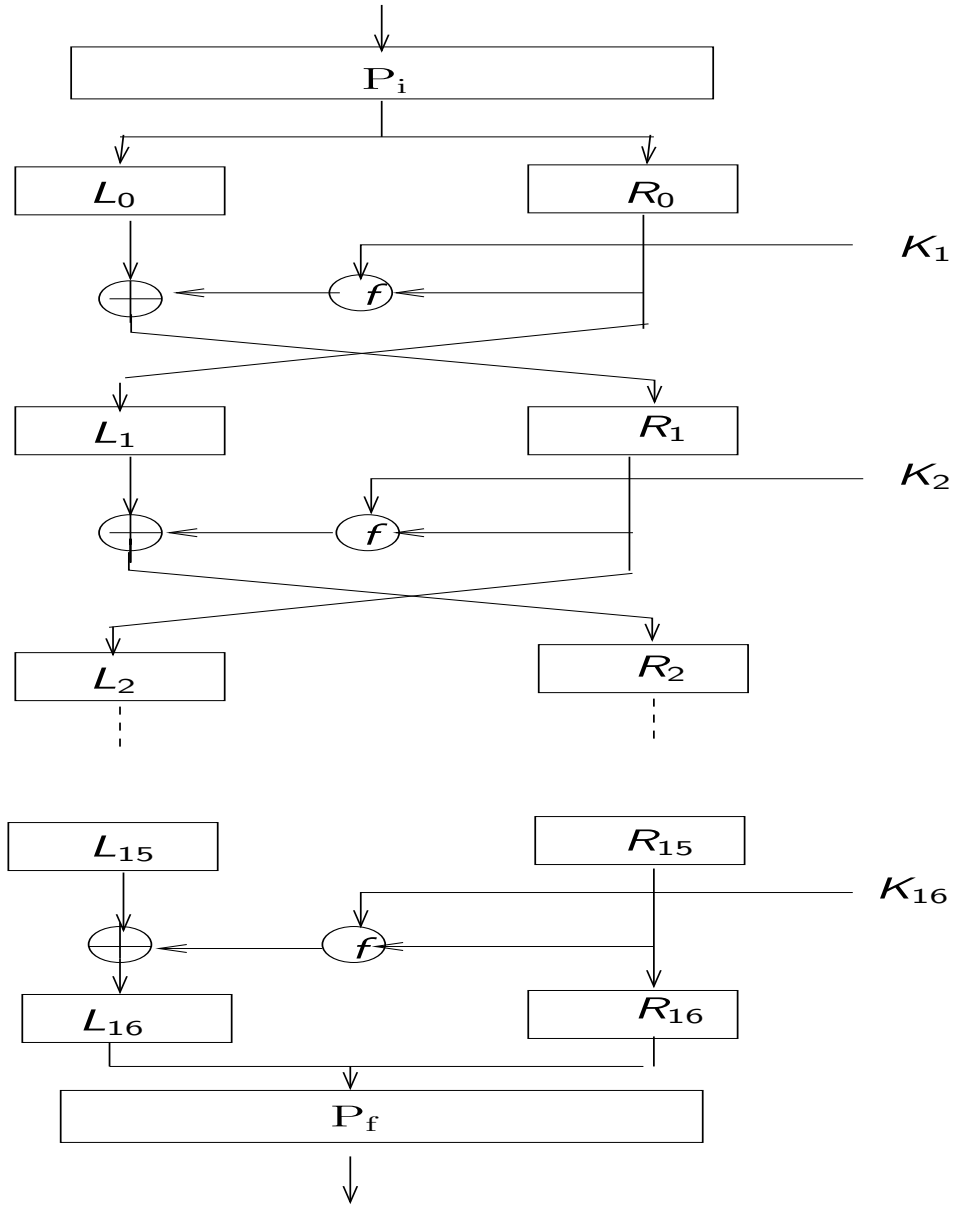
- Es el algoritmo simétrico más extendido mundialmente.
- Se basa en el algoritmo LUCIFER desarrollado por IBM a principios de los setenta.
- Estándar para documentos no clasificados en EE.UU. desde 1976 hasta finales de los noventa.

Algoritmo

- Cifra bloques de 64 bits empleando claves de 56 bits.
- Red de Feistel de 16 rondas.
- Dos permutaciones $\mathrm{P_i}$ (inicial) y $\mathrm{P_f}$ (final) tales que $\mathrm{P_f} = \mathrm{P_i}^{-1}$.

Para descifrar se utiliza el mismo algoritmo en orden inverso.

Esquema del DES



Permutaciones inicial y final

Permutación $\mathrm{P_i}$							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	56	47	39	31	23	15	7

Permutación $\mathrm{P_f} = \mathrm{P_i}^{-1}$							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Las tablas se leen de izquierda a derecha y de arriba abajo.

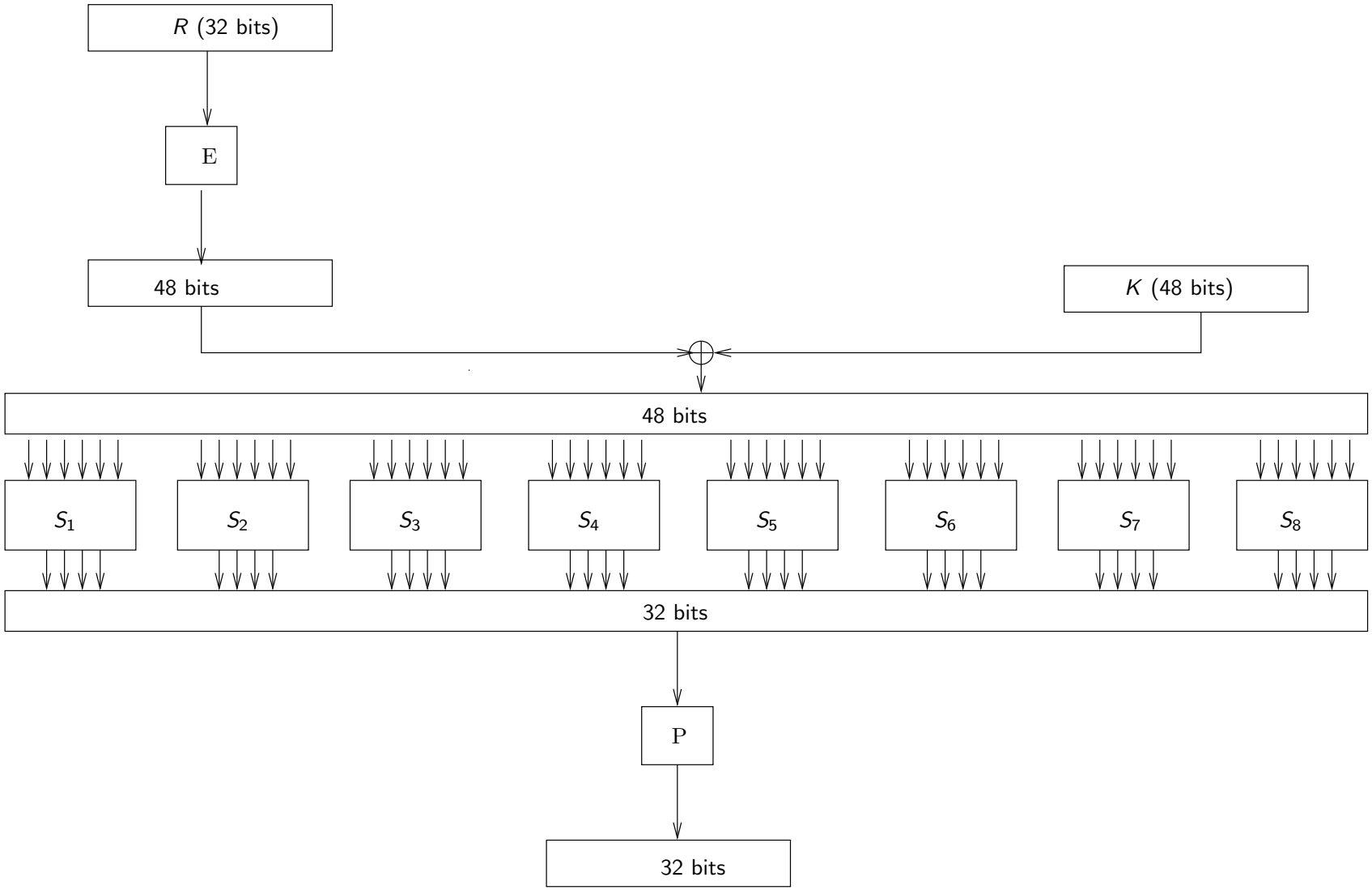
$$\mathrm{P_i}(b_1\, b_2\, b_3 \ldots b_{64}) = b_{58}\, b_{50}\, b_{42} \ldots b_7,$$

$$\mathrm{P_f}(b_1\, b_2\, b_3 \ldots b_{64}) = b_{40}\, b_8\, b_{48} \ldots b_{25}.$$

Función f

- Permutación de expansión (E), que convierte el bloque de 32 bits correspondiente en uno de 48 bits.
- Realiza XOR con el valor K_i, también de 48 bits.
- Aplica ocho S-Cajas de 6×4 bits.
- Efectúa una nueva permutación P.

Esquema de la función f



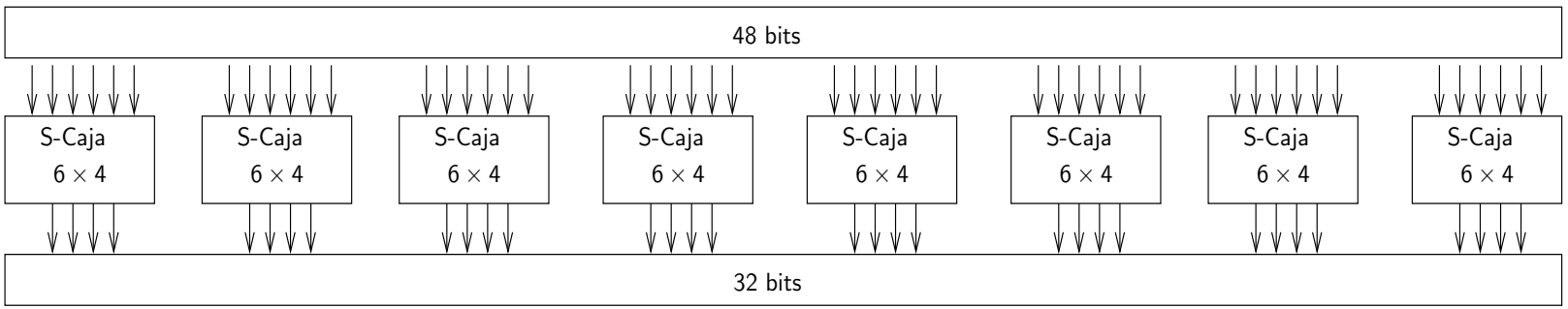
Función de expansión E						Permutación P			
32	1	2	3	4	5	16	7	20	21
4	5	6	7	8	9	29	12	28	17
8	9	10	11	12	13	1	15	23	26
12	13	14	15	16	17	5	18	31	10
16	17	18	19	20	21	2	8	24	14
20	21	22	23	24	25	32	27	3	9
24	25	26	27	28	29	19	13	30	6
28	29	30	31	32	1	22	11	4	25

$$\mathrm{E}(b_1 b_2 \ldots b_{32}) = b_{32} b_1 \ldots b_{32} b_1,$$

$$\mathrm{P}(b_1 b_2 \ldots b_{32}) = b_{16} b_7 \ldots b_4 b_{25}.$$

S-Cajas

El bloque de 48 bits se divide en 8 trozos de 6 bits y cada uno de ellos se sustituye por otro de 4 bits, haciendo uso de la S-caja correspondiente.



Las S-cajas se representan por tablas de 4 filas y 16 columnas:

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	9	12	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

⋮

S_8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Cada caja S_i transforma un bloque de 6 bits $B = b_1 \dots b_6$ en un bloque de 4 bits.

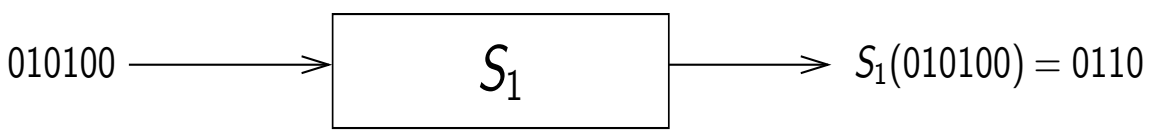
El entero correspondiente a $b_1 b_6$ sirve para seleccionar la fila de la tabla y el correspondiente a $b_2 b_3 b_4 b_5$ sirve para seleccionar la columna.

La salida $S_i(B)$ corresponde a la representación en 4 bits del entero que está en esa fila y esa columna.

Ejemplo

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	9	12	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

$$B = 010100 \rightarrow \left\{ \begin{array}{l} \text{fila } 00 \rightarrow 0 \\ \text{columna } 1010 \rightarrow 10 \end{array} \right\} \rightarrow 6 \rightarrow 0110$$

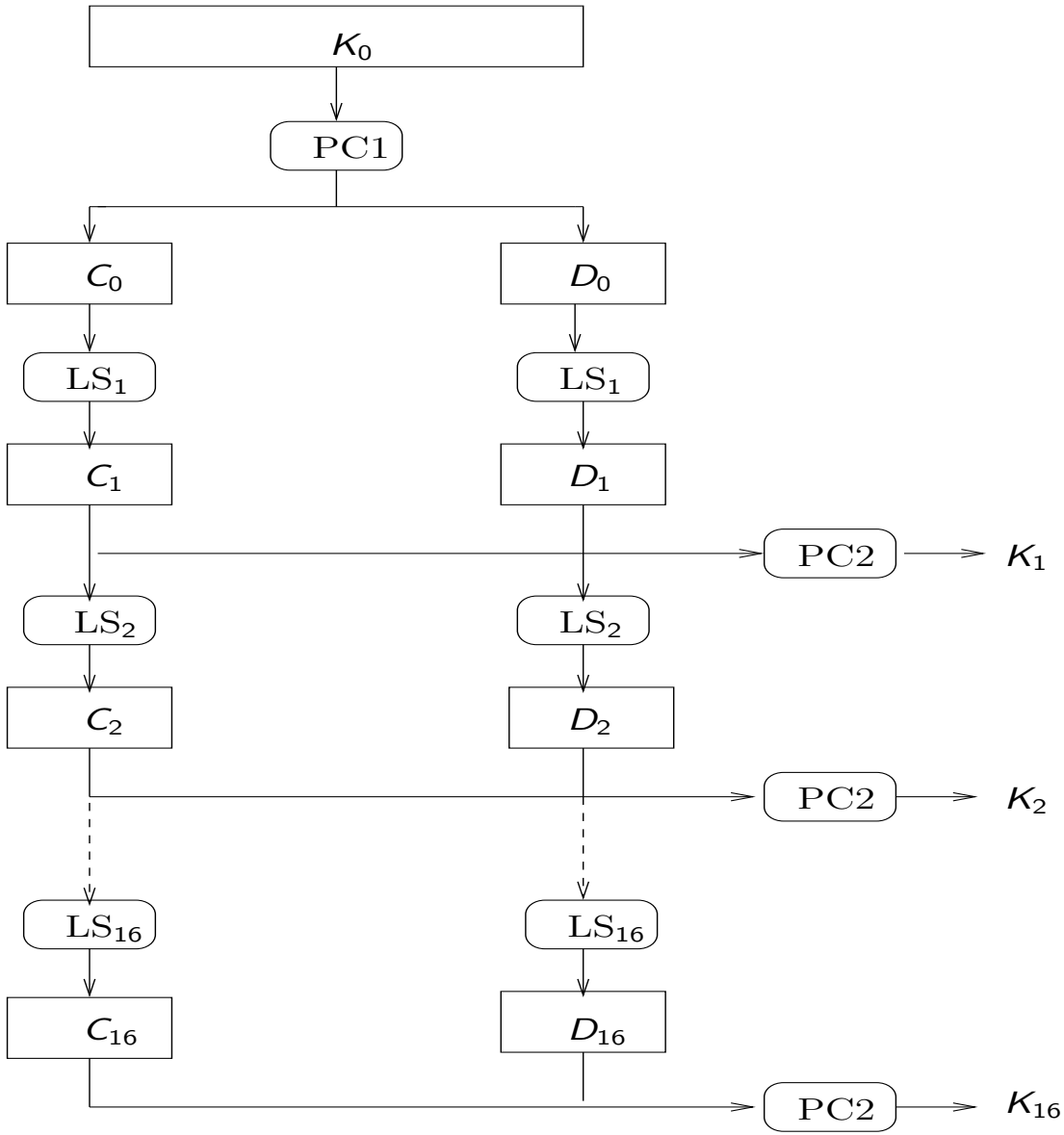


S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Generación de las subclaves

- Clave inicial K_0 de 64 bits.
- Permutación inicial $\mathrm{PC1}$ sobre la K_0 (64 bits $\rightarrow$ 56 bits). Elimina los bits colocados en las posiciones $8, 16, 24, 32, 40, 48, 56, 64$ (bits de paridad).
- Dos mitades de 28 bits.
- Desplazamiento circular a izquierda de cada una de las dos mitades.
- Permutación $\mathrm{PC2}$ (56 bits $\rightarrow$ 48 bits: K_i).

Esquema de generación de subclaves



Permutación PC1

Descarta los bits de la clave K_0 colocados en las posiciones $8, 16, 24, 32, 40, 48, 56, 64$ (bits de paridad).

Permutación PC1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

El resultado se divide en dos mitades C_0 y D_0 de 28 bits.

Desplazamientos circulares LS_i

La subclave K_i se obtiene aplicando la permutación $\mathrm{PC2}$ a la concatenación de dos mitades C_i y D_i. En cada iteración cada una de estas dos mitades se obtiene realizando un desplazamiento circular a la izquierda a la mitad corresponciente de la iteración precedente.

$$C_i = \mathrm{LS}_i(C_{i-1}), \quad D_i = \mathrm{LS}_i(D_{i-1}),$$

El número de posiciones desplazadas viene dado por la tabla:

Iteración	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Desplaz.	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Por ejemplo,

$$C_1 = 10110101011100110111110101 10,$$

$$C_2 = \mathrm{LS}_2(C_1) = 01101010111001101111101011 01,$$

$$C_3 = \mathrm{LS}_3(C_2) = 10101011100110111110101101 01.$$

Permutación $\mathrm{PC2}$

Reduce el tamaño del bloque de 56 a 48 bits, ya que no utiliza los bits de las posiciones $9, 18, 22, 25, 35, 38, 43, 54$.

Permutación $\mathrm{PC2}$					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

La clave K_i se obtiene aplicando esta permutación a la concatenación de las dos mitades C_i, D_i.

$$K_i = \mathrm{PC2}(C_i, D_i).$$

Debilidades y ataques

Debilidades del DES

- *Complementariedad.* Para cualquier mensaje M y cualquier clave K,

$$(\mathrm{DES}_K(M))^c = \mathrm{DES}_{K^c}(M^c)$$

Por ejemplo, en hexadecimal:

$$M = 0123456789abcdef\,, \qquad K = 133457799bbcdff\,1,$$

$$\mathrm{DES}_K(M) = 85e813540f\,0ab405.$$

$$M^c = fedcba9876543210, \qquad K^c = eccba8866443200e,$$

$$\mathrm{DES}_{K^c}(M^c) = 7a17ecabf\,0f\,54bfa.$$

Esta propiedad supone una reducción del trabajo para realizar un ataque por fuerza bruta con texto en claro elegido.

- *Claves débiles.* Son claves K tales que para cualquier mensaje M,

$$\mathrm{DES}_K^2(M) = \mathrm{DES}_K(\mathrm{DES}_K(M)) = M$$

Se conocen 4 claves débiles.

En hexadecimal:

Clave	Clave tras aplicar $\mathrm{PC1}$
0101010101010101	0000000 0000000
fefefefefefefefe	fffffff fffffff
e0e0e0e0f1f1f1f1	fffffff 0000000
1f1f1f1f0e0e0e0e	0000000 fffffff

- *Claves semidébiles*. Son parejas de claves (K_1, K_2) tales que para cualquier mensaje M,

$$\mathrm{DES}_{K_1}(\mathrm{DES}_{K_2}(M)) = M$$

Se conocen 6 parejas semidébiles.

En hexadecimal:

Pareja de claves	Pareja de claves tras aplicar $\mathrm{PC1}$
(01fe01fe01fe01fe, fe01fe01fe01fe01)	(aaaaaaa aaaaaaa, 5555555 5555555)
(1fe01fe00ef10ef1, e01fe01ff10ef10e)	(aaaaaaa 5555555, 5555555 aaaaaaa)
(01e001e001f101f1, e001e001f101f101)	(aaaaaaa 0000000, 5555555 0000000)
(1ffe1ffe0efe0efe, fe1ffe1ffe0efe0e)	(aaaaaaa fffffff, 5555555 fffffff)
(011f011f010e010e, 1f011f010e010e01)	(0000000 aaaaaaa, 0000000 5555555)
(e0fee0fef1fef1fe, fee0fee0fef1fef1)	(fffffff aaaaaaa, fffffff 5555555)

Ataques al DES

- Ataque por fuerza bruta: Consiste en probar de forma exahustiva las 2^{56} claves posibles.

  En 1998, se construyó la máquina "DES-cracker", que descifró un mensaje en menos de 3 días.
- Ataque con texto en claro elegido: El criptoanalista tiene la oportunidad de elegir los textos en claro y obtener sus correspondientes cifrados.
  - En 1990 Eli Biham y Adi Shamir descubren el *criptoanálisis diferencial*, aunque ya era conocido anteriormente por IBM y la NSA.
  - En 1992 Mitsuru Matsui descubre el *criptoanálisis lineal*.

Triple-DES

Un algoritmo de cifrado tiene estructura de *grupo* si para cada par de claves K_1 y K_2 existe otra clave K_3 tal que

$$\mathrm{E}(\mathrm{E}(M, K_1), K_2) = \mathrm{E}(M, K_3).$$

Es decir, si ciframos un mensaje primero con la clave K_1 y el resultado lo ciframos con la clave K_2, obtenemos el mismo criptograma que si ciframos una vez con la clave K_3.

En 1992 se probó que DES **no** posee estructura de grupo, lo que posibilita aplicar varias veces el algoritmo con diferentes claves.

El *Triple-DES* tiene la siguiente estructura:

$$C = \mathrm{DES}_{K_1}(\mathrm{DES}^{-1}_{K_2}(\mathrm{DES}_{K_1}(M))).$$

Es decir, se cifra con la subclave K_1, se descifra con K_2 y se vuelve a cifrar con K_1.

La clave total (K_1, K_2) tiene una longitud de 112 bits.

Fin de la sección