CRIPTOGRAFIA

Sistemas de cifrado simétrico

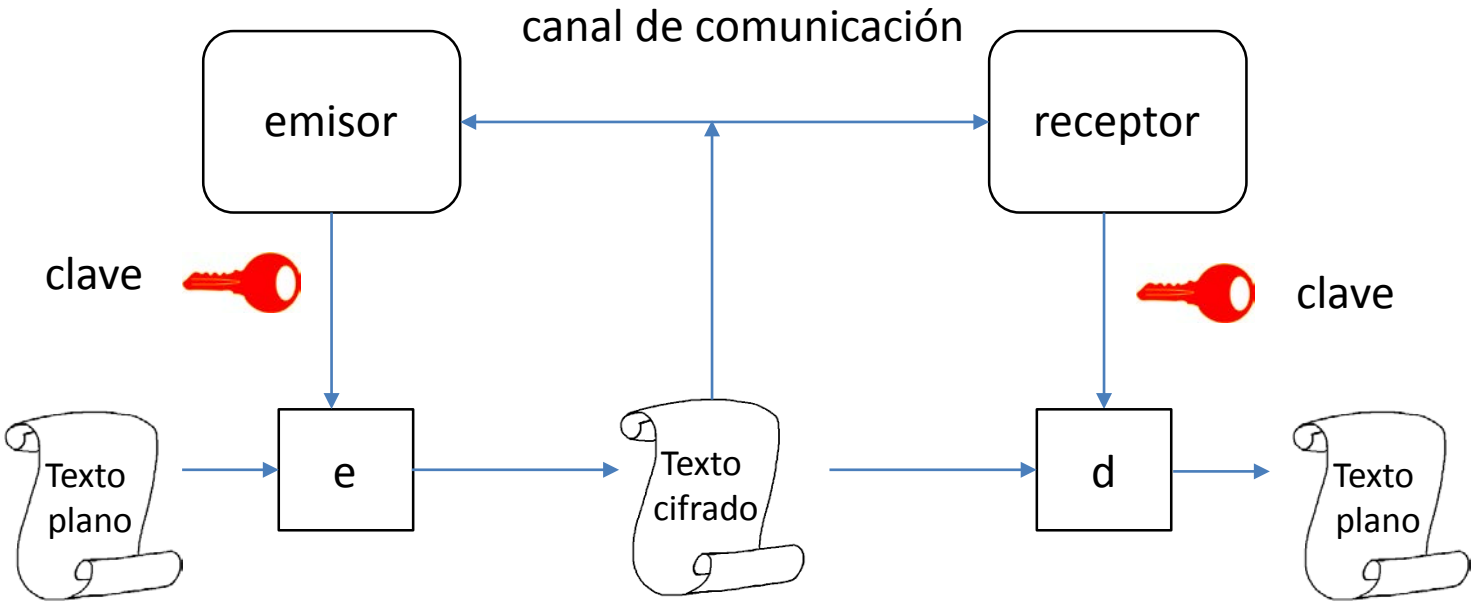
Contenido

1. Esquema básico del cifrado simétrico
2. El sistema DES (*Data Encription Standard*)
   - 2.1 Breve historia del sistema DES.
   - 2.2 Descripción del sistema.
   - 2.3 Propiedades del sistema.
   - 2.4 Modos de aplicación del sistema.
   - 2.5 Ataques exhaustivos: La iniciativa DESCHALL
3. Otros sistemas de cifrado por bloques : FEAL, IDEA, SAFER y RC5.
4. La convocatoria AES (*Advanced Encryption Standard*).
5. El sistema Rijndael

Bibliografía

- The Design of Rijndael. AES - The Advanced Encryption Standard. Joan Daemen Vincent Rijmen. Springer. 2002.
- Cryptography. Theory and Practice. Douglas R. Stinson. Chapman & Hall/CRC. 2006.

Esquema básico del cifrado simétrico



$x = x_1\ x_2\ x_3 \dots x_b\ x_{b+1}\ x_{b+2} \dots x_{2b}\ x_{2b+1} \dots x_{p\cdot b}\ x_{p\cdot b+1} \dots x_n$ en flujo

$x =$ [$x_1\ x_2\ x_3 \dots x_b$] [$x_{b+1}\ x_{b+2} \dots x_{2b}$] [$x_{2b+1}\ x_{2b+2} \dots$] $\dots x_{p\cdot b}$ [$x_{p\cdot b+1} \dots x_n\ 0 \dots 00$] en bloque

Bloque 1 Bloque 2 Bloque 3 Bloque p+1

(cada bloque se puede cifrar por separado utilizando la misma clave)

Breve historia del sistema DES

1973-74 Convocatoria de la Oficina Nacional de Normas y Tecnología (NIST)

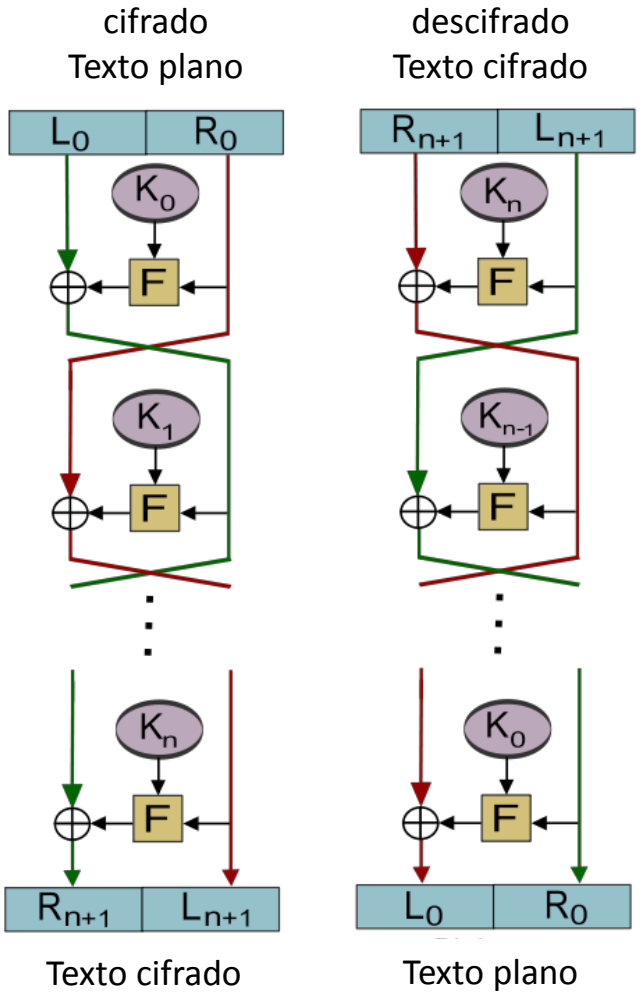
1967 - … H. Feistel & IBM: Sistema DATASEAL, DEMON, LUCIFER

NSA
(la controversia del DES)

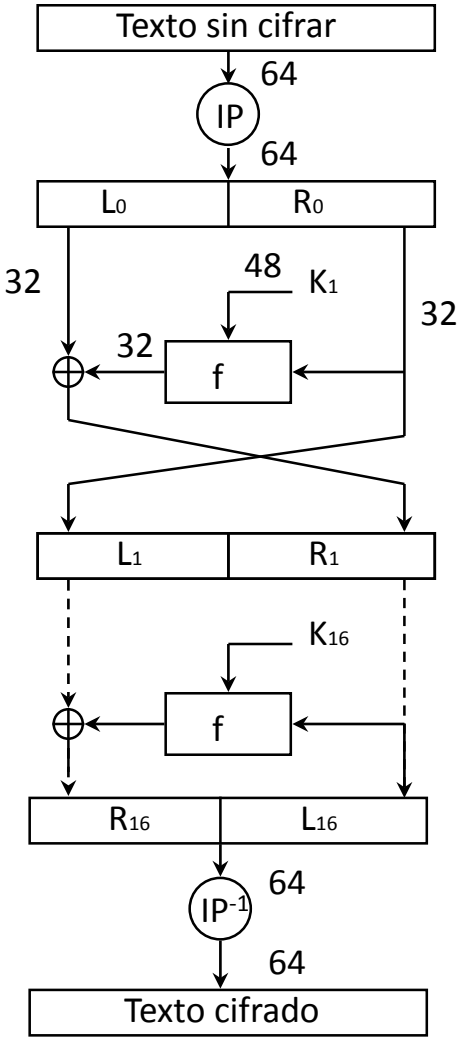
1974-75 Sistema DES

1998: DES deja de ser el estándar. Convocatoria AES.

Redes de Feistel

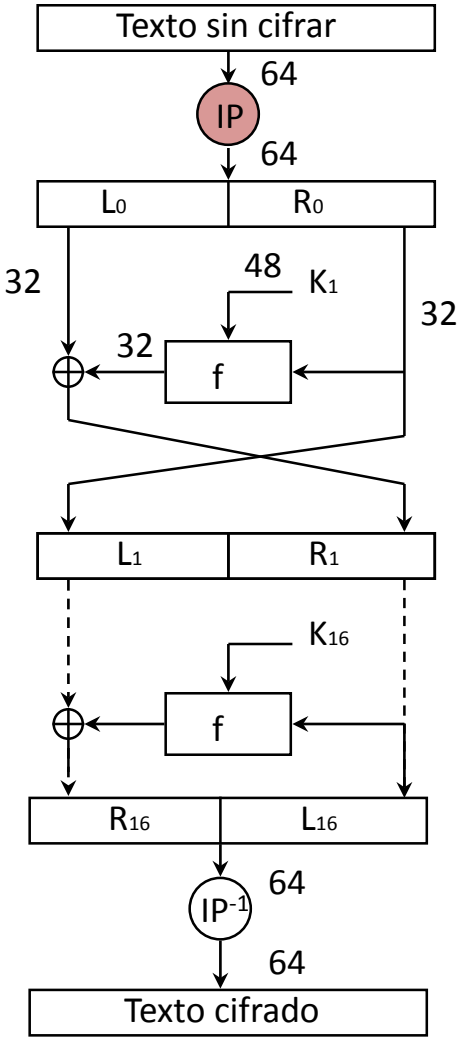


El sistema DES



- 16 iteraciones
- Cifrado Aplicando la lista de claves en el orden $k_1, k_2, \ldots, k_{16}$
- Descifrado Aplicando la lista de claves en el orden $k_{16}, k_{15}, \ldots, k_1$

El sistema DES



La permutación inicial IP

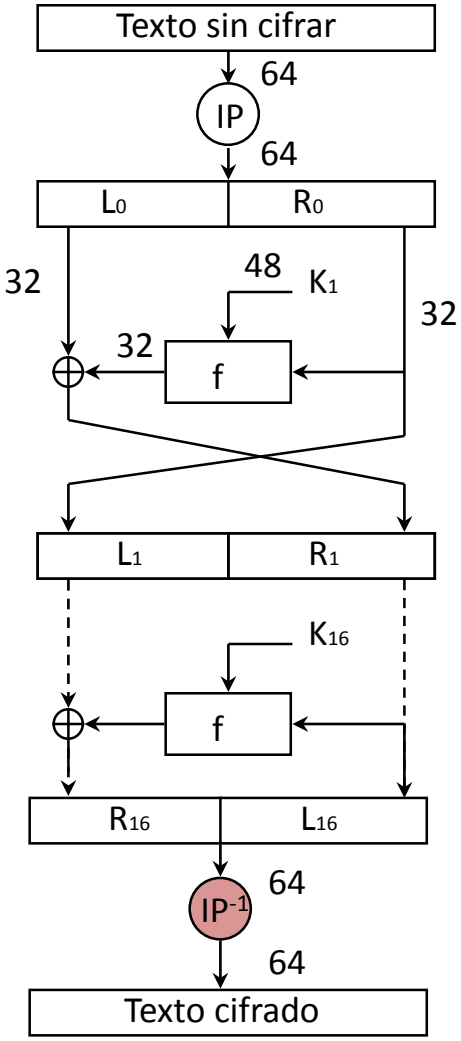
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

#bit en x 12 3 4 5 6 7 … … 50 … 58 64

x = 00101 0 0 . . . 0 . . . 1 . . . 101

x_0 =IP(x)=10 …. ….. …. 0

El sistema DES



La permutación final inversa IP^{-1}

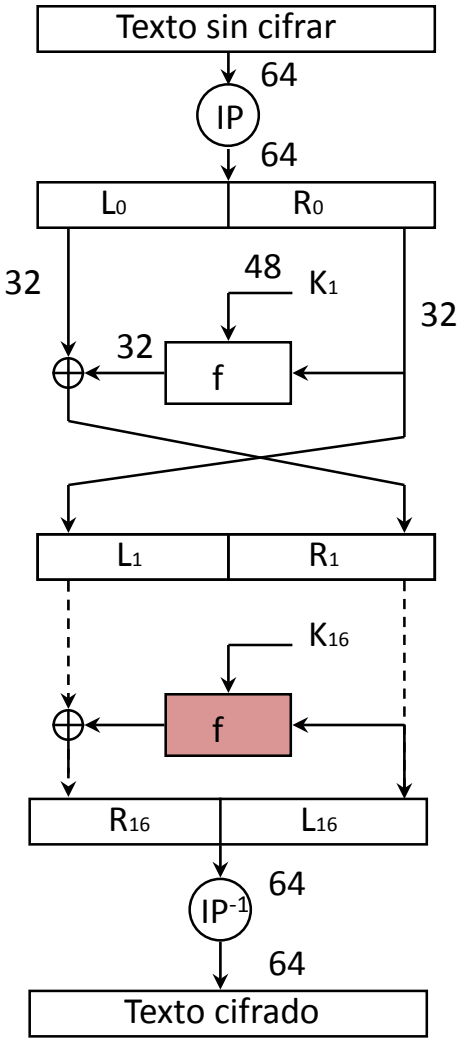
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	23	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

#bit en x 1 … 8 … 25 …40 … 48 … … 64

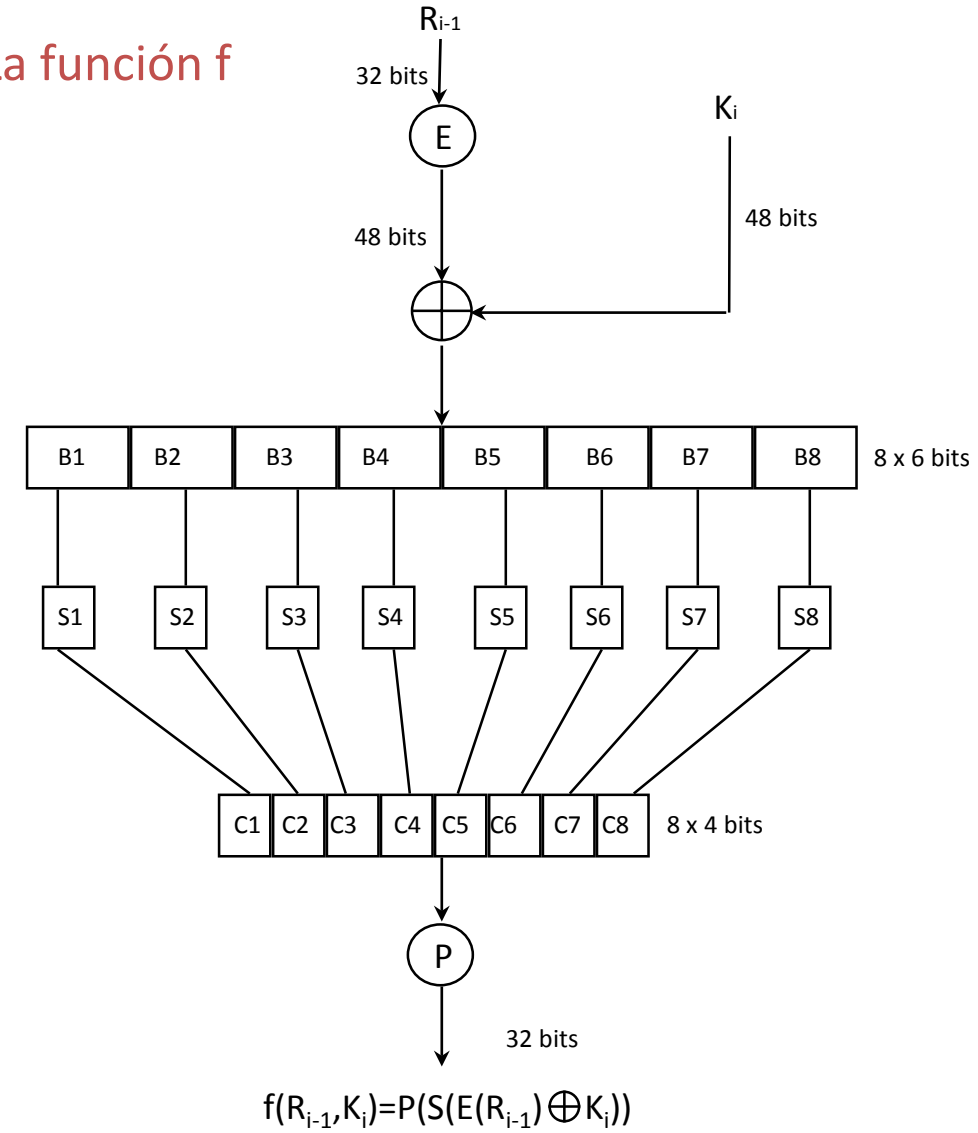
x = 0 … 0 … 1 … 1 … 1 … … 1

x_0 =IP(x)=10 …. ….. …. 1

El sistema DES

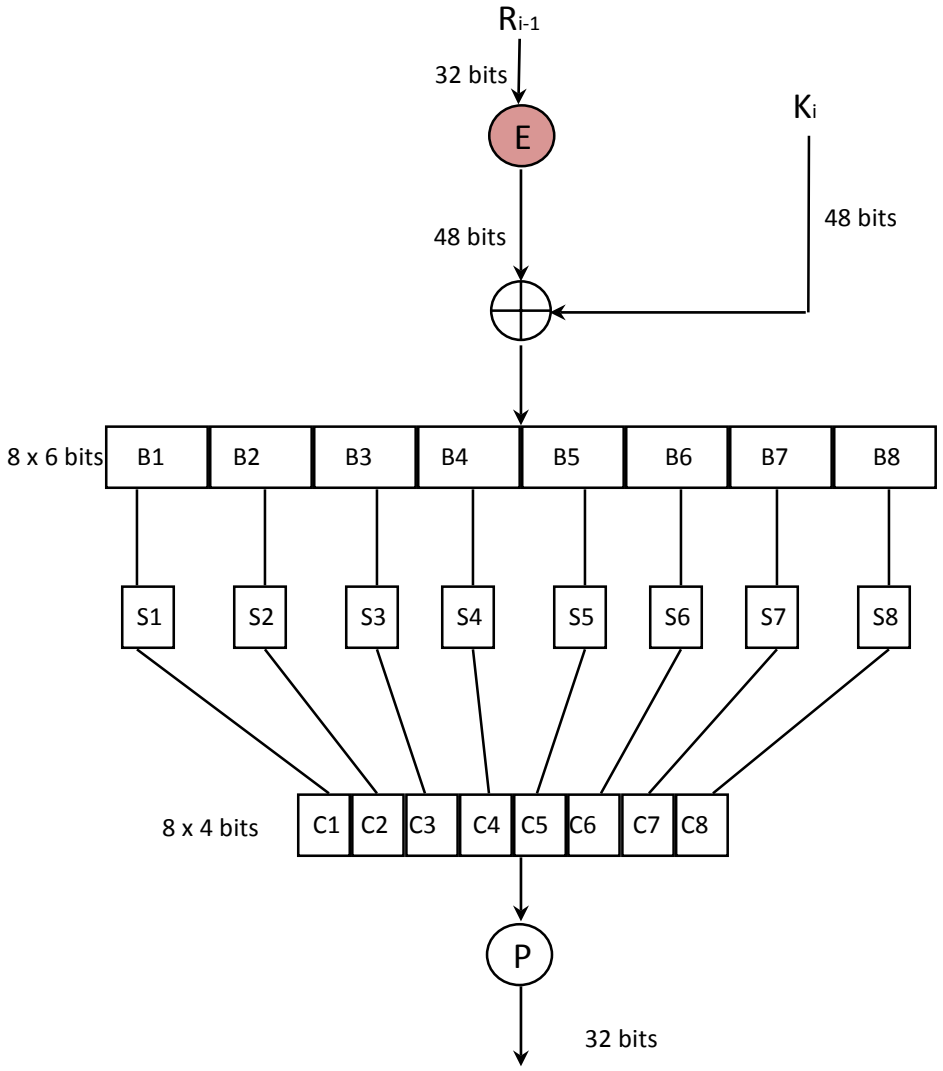


La función f



$$f(R_{i-1},K_i)=P(S(E(R_{i-1})\oplus K_i))$$

El sistema DES



$f(R_{i-1},K_i)=P(S(E(R_{i-1})\oplus K_i))$

La función de expansión de bloque E

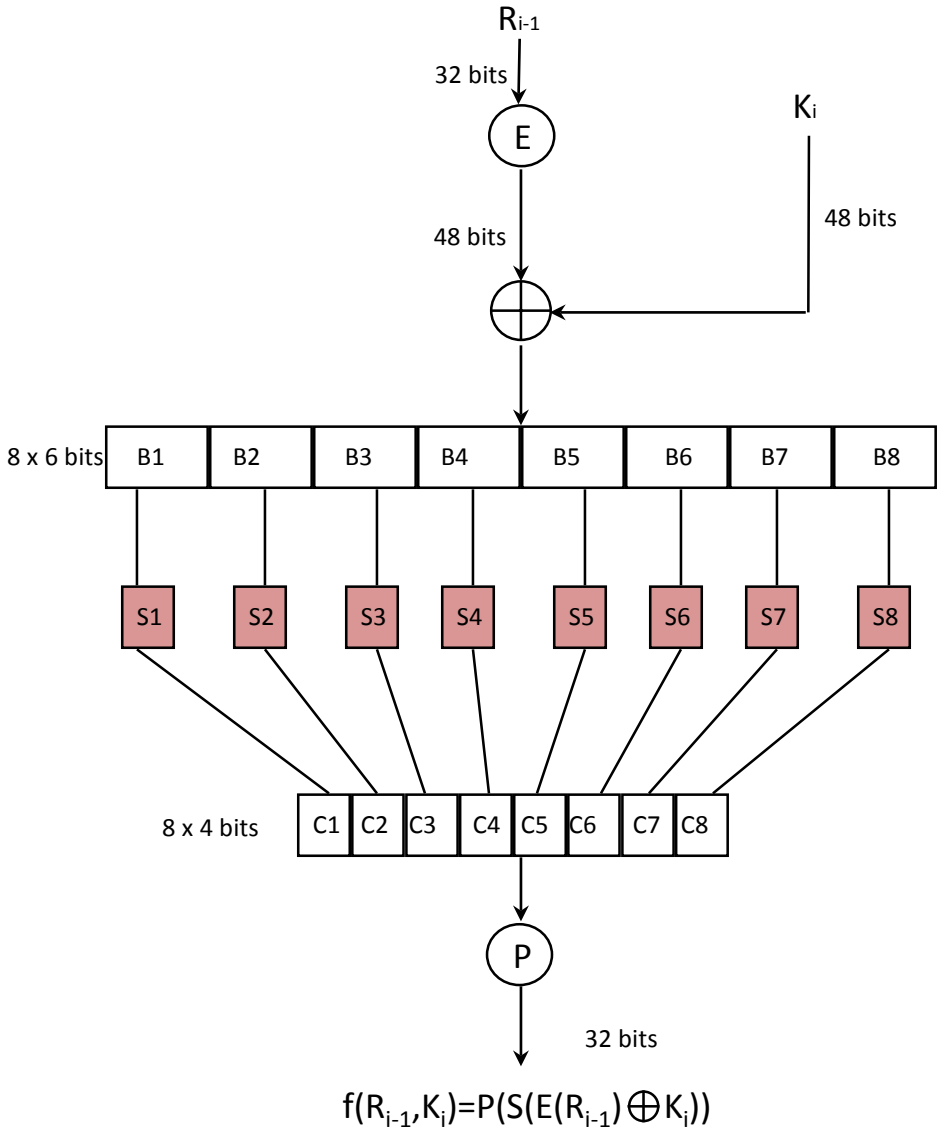
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

bit en R_{i-1} 1 2 3 32

R_{i-1} = 0 0 0 1

$E(R_{i-1})$= 100 0 0

El sistema DES



Las tablas S (*S-boxes*)

El sistema DES — Las tablas S (*S-boxes*)

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

5	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

El sistema DES — Las tablas S (*S-boxes*)

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

S_7

4	11	2	4	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

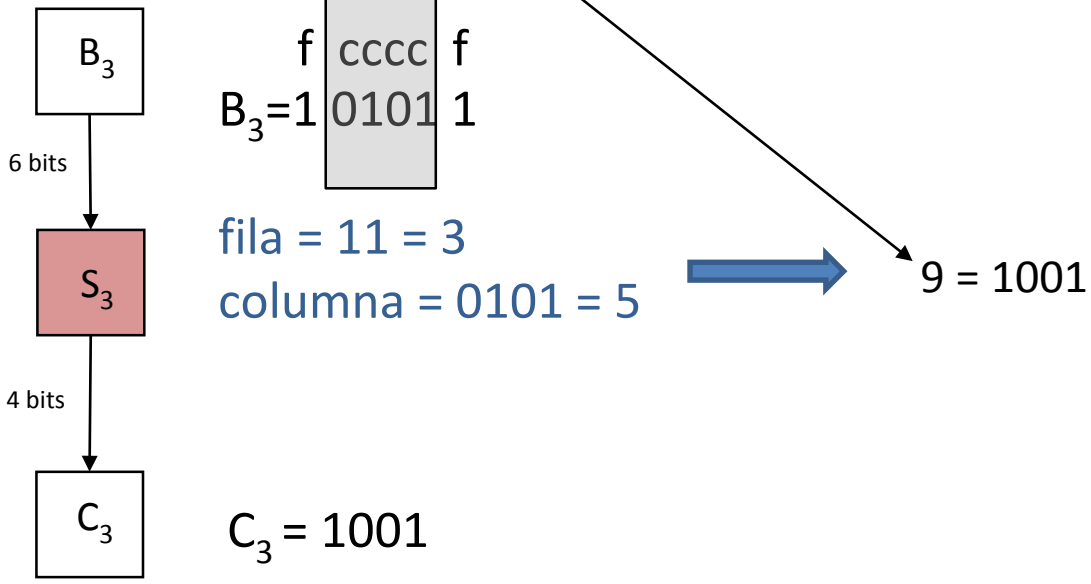
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

El sistema DES

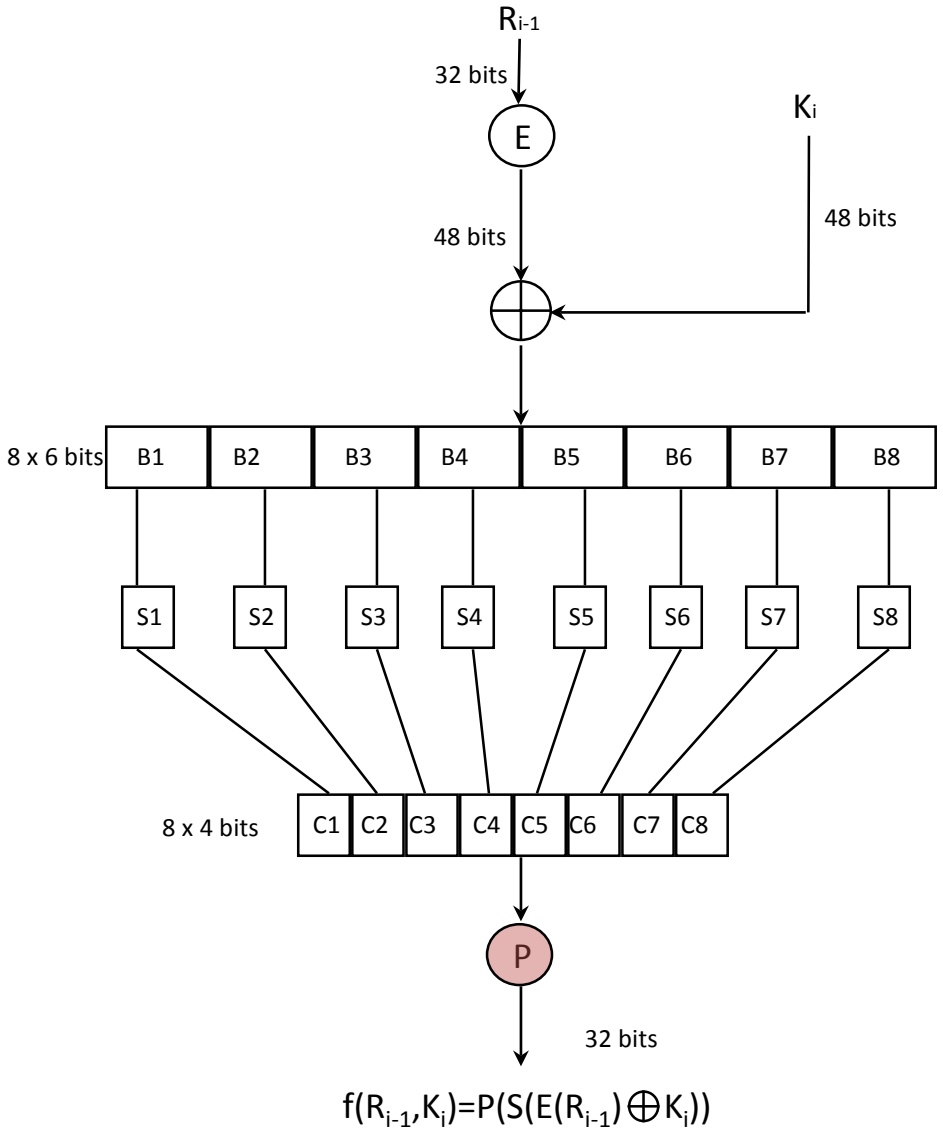
Funcionamiento de las tablas S

S_3

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



El sistema DES

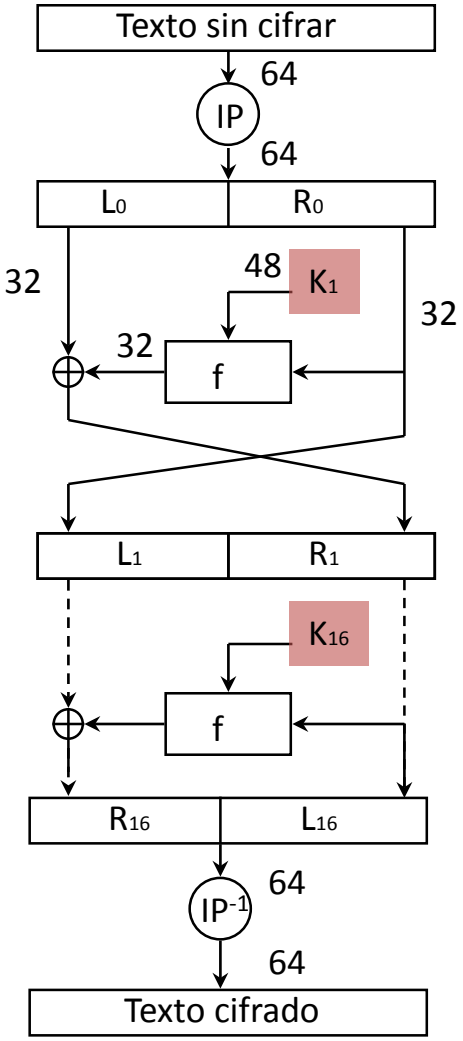


$$f(R_{i-1},K_i)=P(S(E(R_{i-1})\oplus K_i))$$

La permutación P

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

El sistema DES

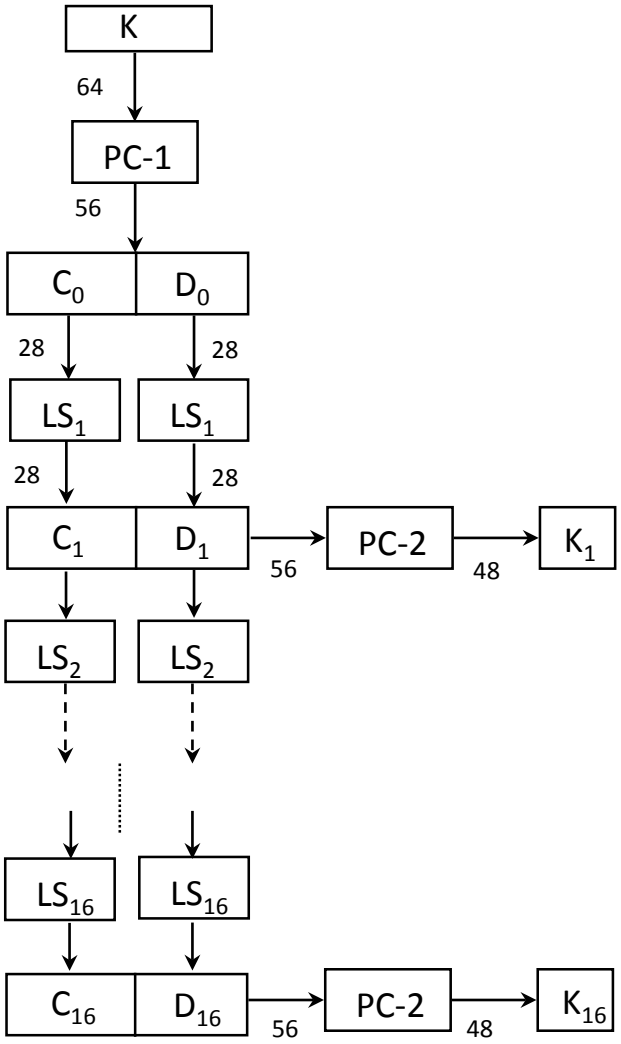


Cálculo de la lista de claves a partir de la clave K

- La clave del DES es única (no se aplican 16 claves)
- La clave tiene 64 bits (8 de ellos de paridad)
- La lista de claves K_1, K_2, …, K_{16} son 16 fragmentos de K que se utilizan en cada una de las iteraciones
- En el triple-DES se utilizan tres claves distintas
- La lista de claves se calcula *off-line* (incluso mediante circuitería hardware)

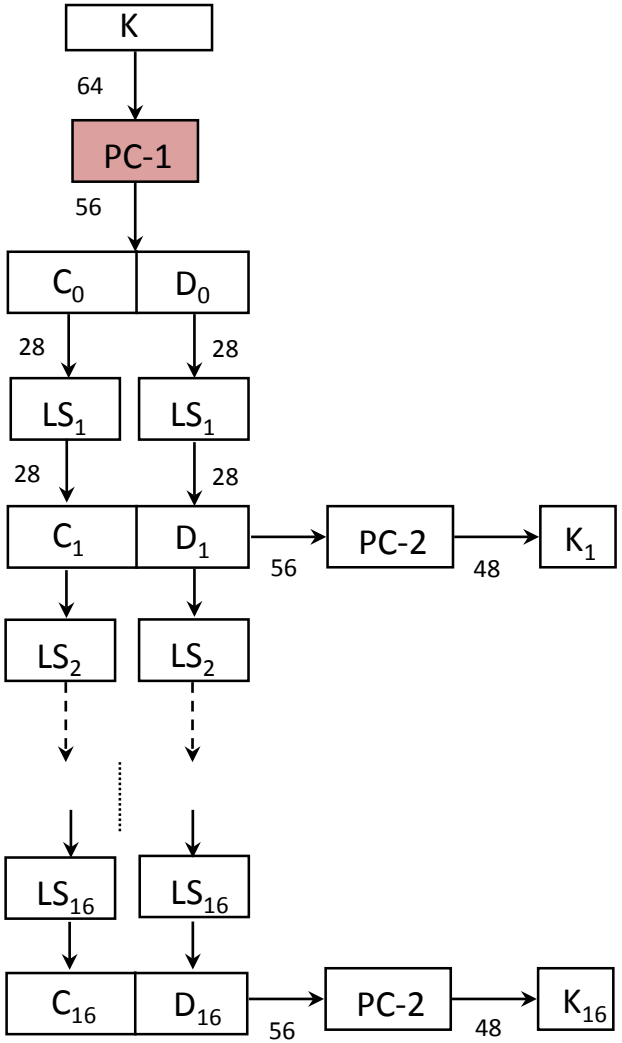
El sistema DES

Cálculo de la lista de claves a partir de la clave K



El sistema DES

Cálculo de la lista de claves a partir de la clave K

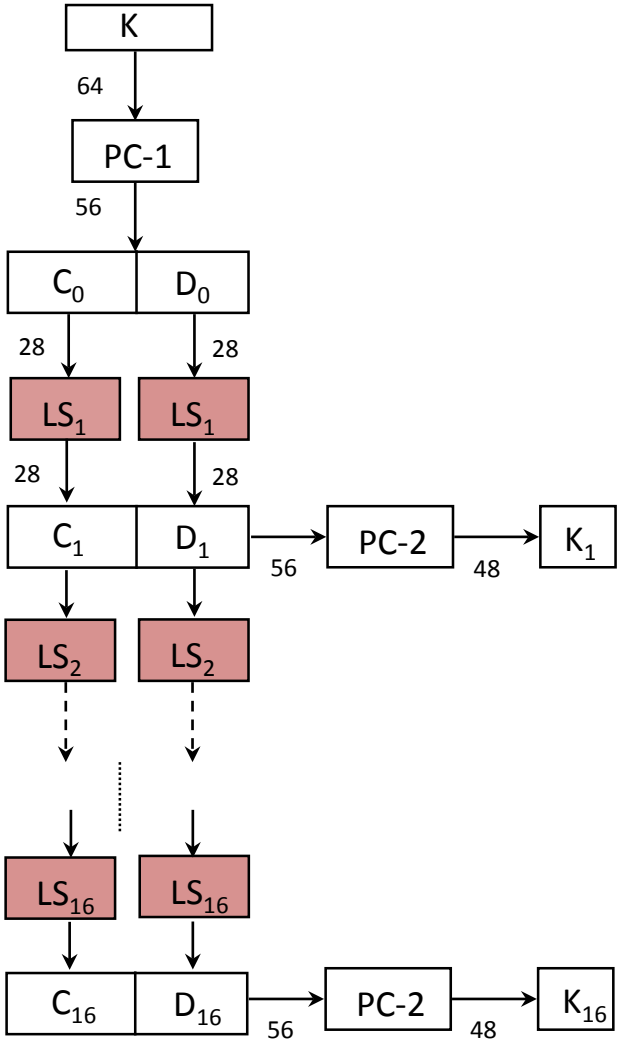


La permutación-compresión inicial PC-1

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

Se evitan los bits de paridad 8,16,24,32,40,48,56 y 64

El sistema DES — Cálculo de la lista de claves a partir de la clave K



Las traslaciones cíclicas LS_i

LS_i es una traslación cíclica a izquierda de:

- una posición (i = 1,2,9 ó 16)

  x= abcde
  $LS_1(x)$ = bcdea

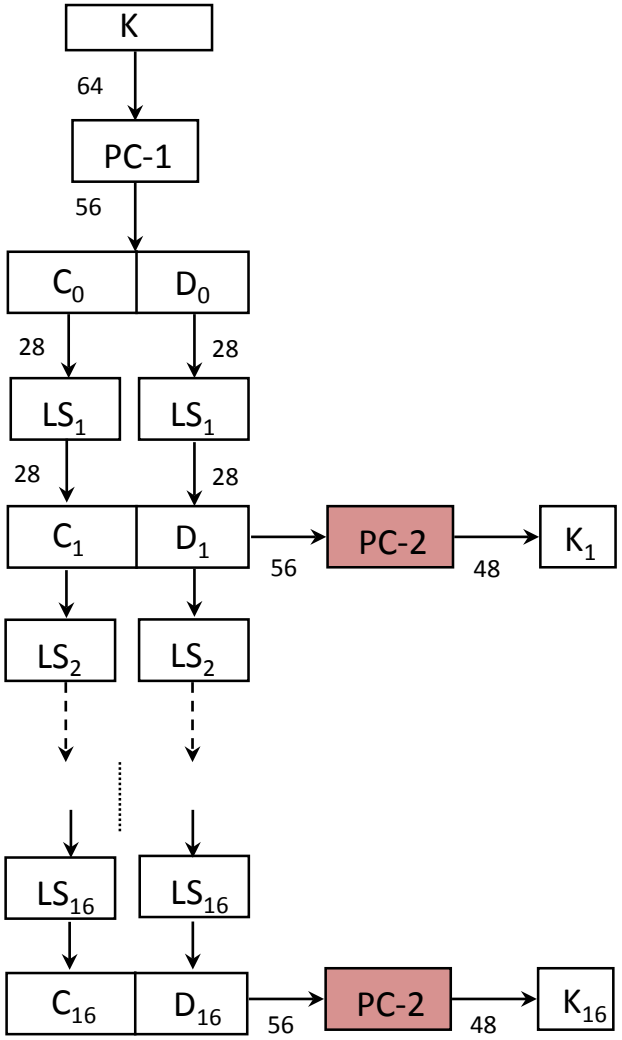
- dos posiciones (resto de valores de i)

  x= abcde
  $LS_3(x)$ = cdeab

Las traslaciones cíclicas permiten utilizar todos los bits significativos de la clave K

El sistema DES

Cálculo de la lista de claves a partir de la clave K



La permutación-compresión PC-2

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

Se evitan los bits 9,18,22,25,35,38,43 y 54

Propiedades del sistema DES

Propiedades de las tablas S

- Cada fila de cada tabla S es una permutación de los enteros del 0 al 15
- Ninguna tabla S computa una función lineal o afín de su entrada
- Para toda tabla S y toda entrada *x* se cumple que S(*x*) y S($x \oplus 001100$) difieren en al menos dos bits
- Para cada tabla S, un cambio de bit en la entrada produce al menos dos cambios de bits en su salida
- Para toda tabla S, toda entrada *x* y todo par de bits *e* y *f* se cumple que S(*x*) ≠ S($x \oplus 11ef00$)
- Para toda tabla S, para cada bit de entrada y cada bit de salida, el número de valores de entrada con salida 0 se aproxima al número de bits de entrada con salida 1

Propiedad de complementación

$$DES_K(x) = y \quad \Rightarrow \quad DES_{\bar{K}}(\bar{x}) = \bar{y}$$

Propiedad de no idempotencia

Para todo trío de claves distintas K_i, K_j y K_p $DES_{K_i}(x) \neq DES_{K_j}(DES_{K_p}(x))$

Propiedades del sistema DES

Propiedades de las claves

C_0	D_0
$\{0\}^{28}$	$\{0\}^{28}$
$\{1\}^{28}$	$\{1\}^{28}$
$\{0\}^{28}$	$\{1\}^{28}$
$\{1\}^{28}$	$\{0\}^{28}$

k

Una clave k es débil si para todo texto x se cumple que

$$DES_k\left(DES_k(x)\right) = x$$

C_0	D_0
$\{01\}^{14}$	$\{01\}^{14}$
$\{01\}^{14}$	$\{10\}^{14}$
$\{01\}^{14}$	$\{0\}^{28}$
$\{01\}^{14}$	$\{1\}^{28}$
$\{0\}^{28}$	$\{01\}^{14}$
$\{1\}^{28}$	$\{01\}^{14}$

k_1

C_0	D_0
$\{10\}^{14}$	$\{10\}^{14}$
$\{10\}^{14}$	$\{01\}^{14}$
$\{10\}^{14}$	$\{0\}^{28}$
$\{10\}^{14}$	$\{1\}^{28}$
$\{0\}^{28}$	$\{01\}^{14}$
$\{1\}^{28}$	$\{10\}^{14}$

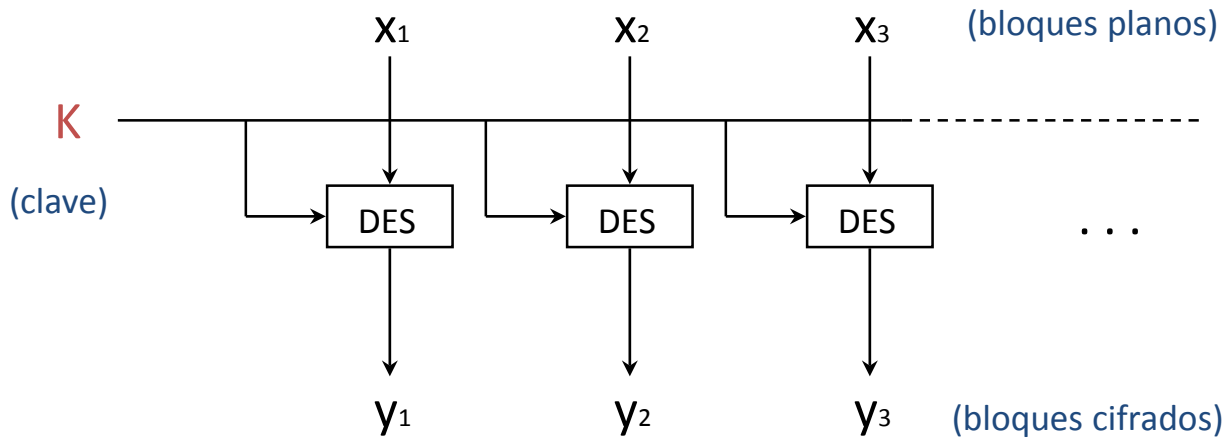
k_2

Un par de claves k_1 y k_2 son semidébiles si para todo texto x se cumple que

$$DES_{k_1}\left(DES_{k_2}(x)\right) = x$$

Modos de aplicación del sistema DES

Modo ECB (Electronic CodeBook)

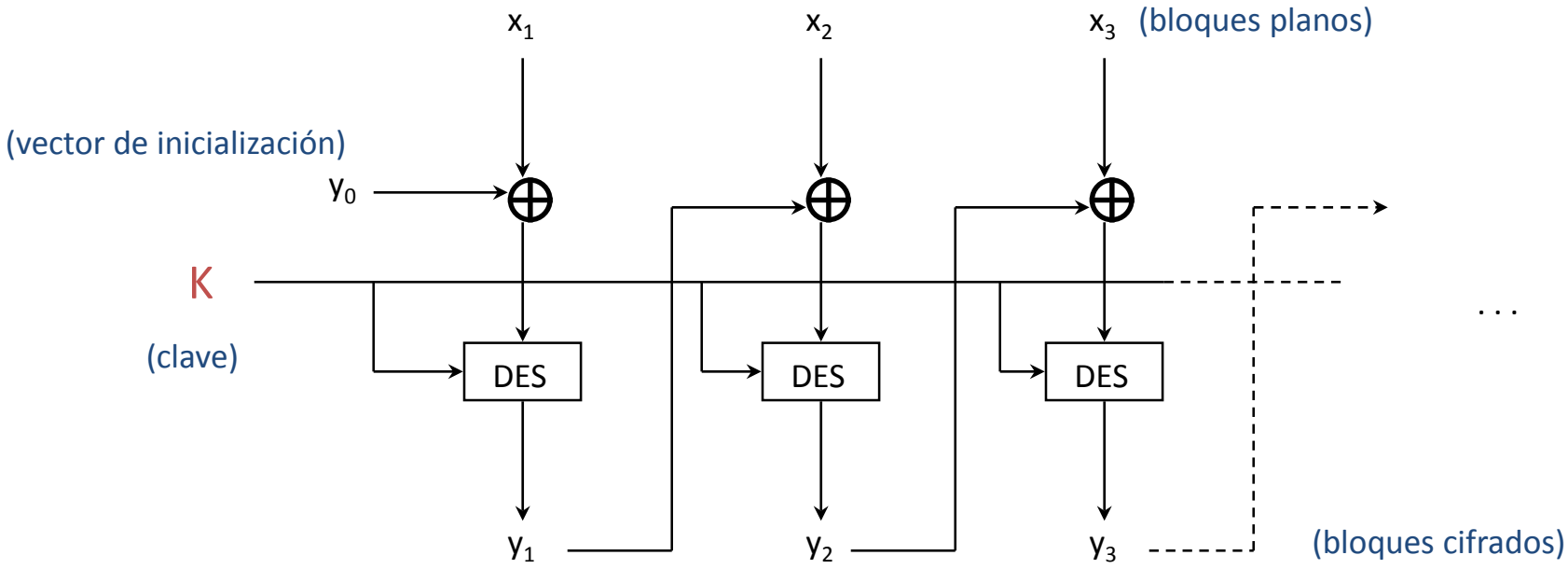


- Modo no encadenado
- Extremadamente rápido (cifrado paralelo hardware)
- Resistente a ruido y errores de transmisión
- Débil ante ataques de sustitución
- Descifrado aplicando la lista de claves en sentido inverso

$$DES_{K^{-1}}(y_j) = x_j$$

Modos de aplicación del sistema DES

Modo CBC (Cipher Block Chaining)

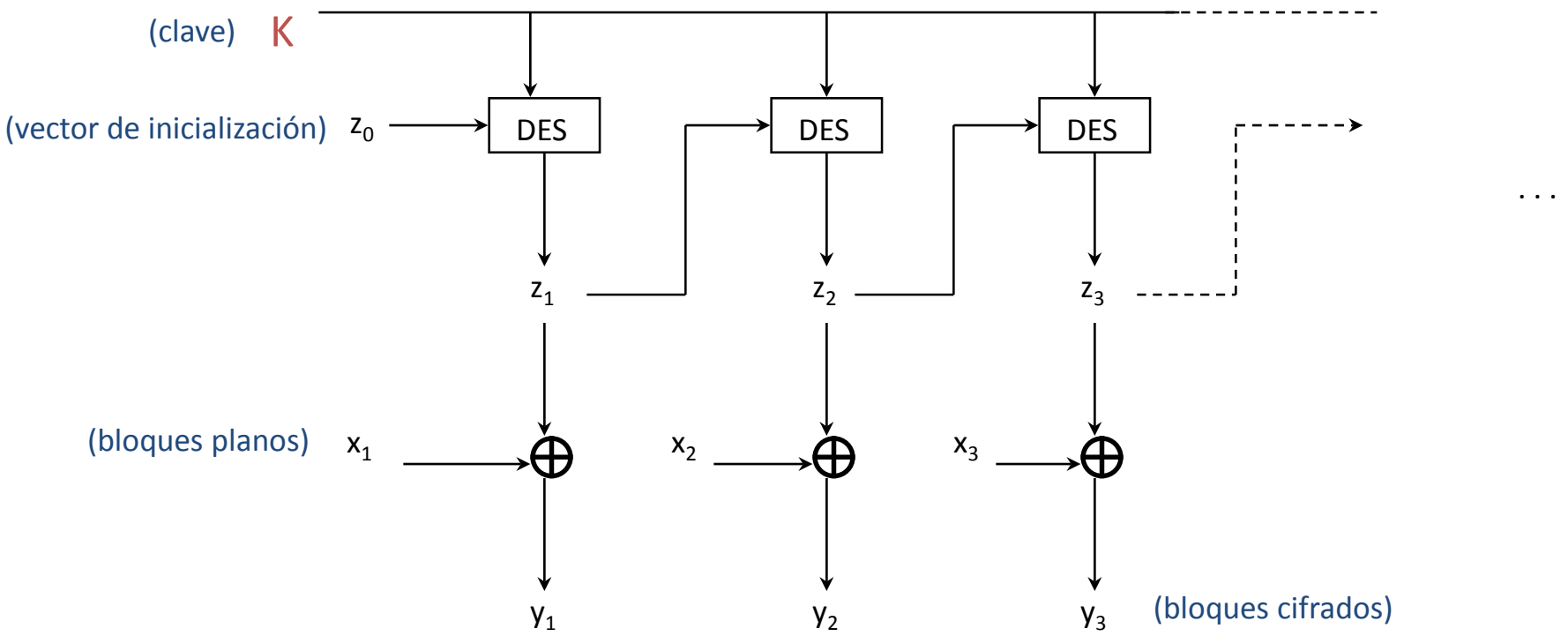


- Modo encadenado
- Cifrado secuencial
- Débil ante ruido y errores de transmisión
- Resistente ante ataques de sustitución
- Descifrado aplicando la lista de claves en sentido inverso sobre y_j y, al resultado, aplicar X-OR con y_{j-1}

$$DES_{K^{-1}}(y_j) \oplus y_{j-1} = x_j$$

Modos de aplicación del sistema DES

Modo OFB (Output FeedBack)

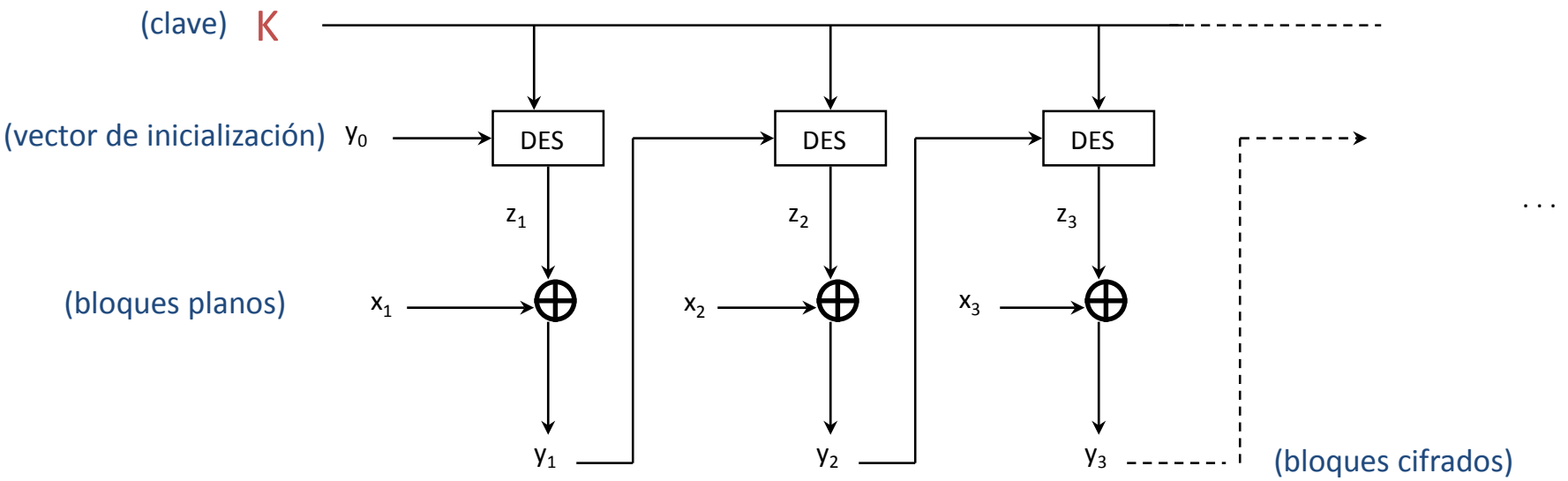


- Modo pseudo-encadenado (sólo a nivel de clave)
- Cifrado paralelo (preparación de la clave *off-line*)
- Resistente ante ruido y errores de transmisión
- Débil ante ataques de sustitución
- Descifrado aplicando X-OR con los vectores z_j (obtenidos a partir de la clave)

$$DES_K(z_{j-1}) = z_j$$
$$z_j \oplus y_j = x_j$$

Modos de aplicación del sistema DES

Modo CFB (Cipher FeedBack)

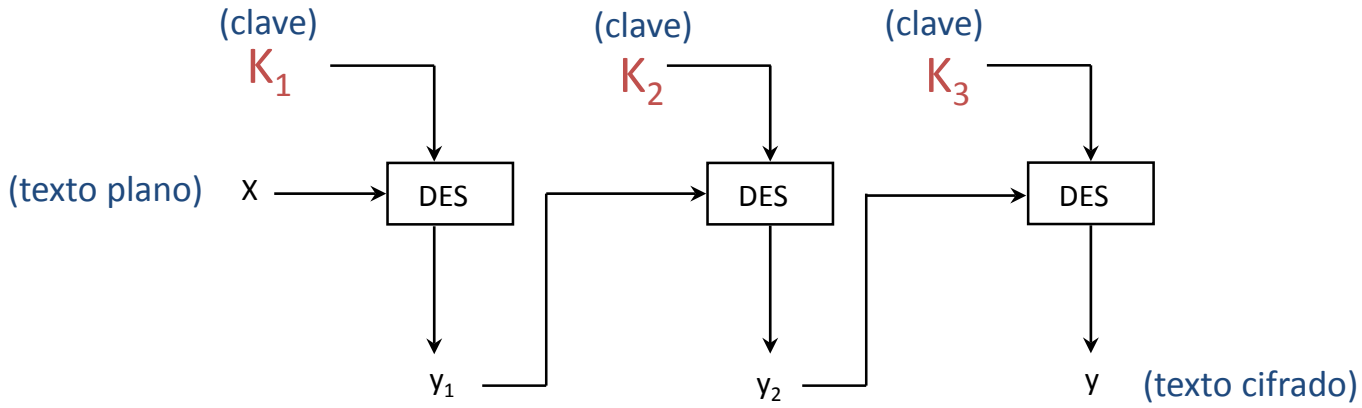


- Modo encadenado
- Cifrado secuencial
- Débil ante ruido y errores de transmisión
- Resistente ante ataques de sustitución
- Descifrado aplicando X-OR con los vectores z_j obtenidos al cifrar el bloque cifrado anterior

$$z_j \oplus y_j = x_j \qquad DES_K(y_{j-1}) = z_j$$

Modos de aplicación del sistema DES

Triple-DES



- Las claves K_1, K_2 y K_3 deben ser distintas y evitar que sean débiles o semidébiles
- Se permite reducir el número de iteraciones (como mínimo a 8) y se mantiene la seguridad
- El espacio de claves pasa de 2^{56} a 2^{168}
- Descifrado aplicando las claves en orden inverso

$$DES_{K_1}{}^{-1}\left(DES_{K_2}{}^{-1}\left(DES_{K_3}{}^{-1}(y)\right)\right) = x$$

Tipos de ataques al sistema DES

Ataque	Cantidad de texto		Memoria	Tiempo
	conocido	elegido		
Búsqueda exhaustiva con precálculo	.....	1	2^{56}	1 consulta
Búsqueda exhaustiva sin precálculo	1	.....	insignificante	2^{55}
Análisis lineal	2^{43} (85 %) 2^{38} (10%)	.....	Ocupación del texto	2^{43} 2^{50}
Análisis diferencial	..... 2^{55}	2^{47}	Ocupación del texto	2^{47} 2^{55}

Tipos de ataques al sistema DES

Ataques exhaustivos coordinados: la iniciativa DESCHALL

- La iniciativa parte de un desafío lanzado en enero de 1997 por RSA Labs. que solicitaba la ruptura de una clave DES de 56 bits mediante ataques del tipo sólo texto cifrado.

- Imposibilidad de colaboración entre equipos USA y fuera de USA

- La arquitectura utilizada se basaba en cliente-servidor con clientes disgregados por todo el dominio internet de USA mediante una filosofía parecida al proyecto SETI@home.

- La búsqueda de claves mediante fuerza bruta (un total de 2^{56} claves) se inició en la misma fecha del desafío y concluyó el 18 de junio de 1997.

- Se emplearon diferentes plataformas y arquitecturas y aproximadamente 78.000 direcciones IP de clientes distintas.

http://www.interhack.net/projects/deschall/

Otros sistemas de cifrado simétrico

FEAL (Fast data Encipher Algorithm)

- Desarrollado en 1987 por NTT (Nippon Telegraph and Telephone Co.)
- Permite parametrizar el número de iteraciones FEAL-n (aconsejable n=32)
- Sustituye las tablas S por operaciones X-OR, sustituciones y traslaciones relativas

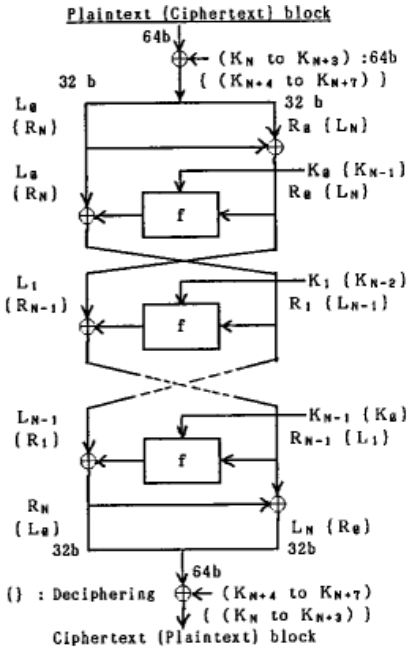
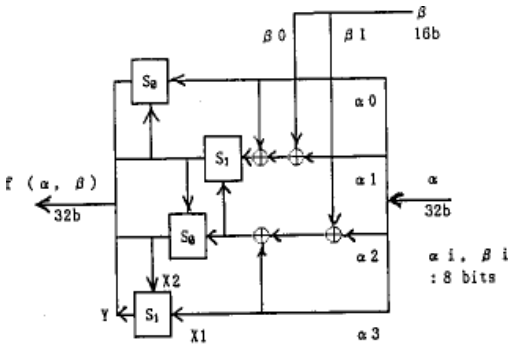


Fig. 1 Data Randomization



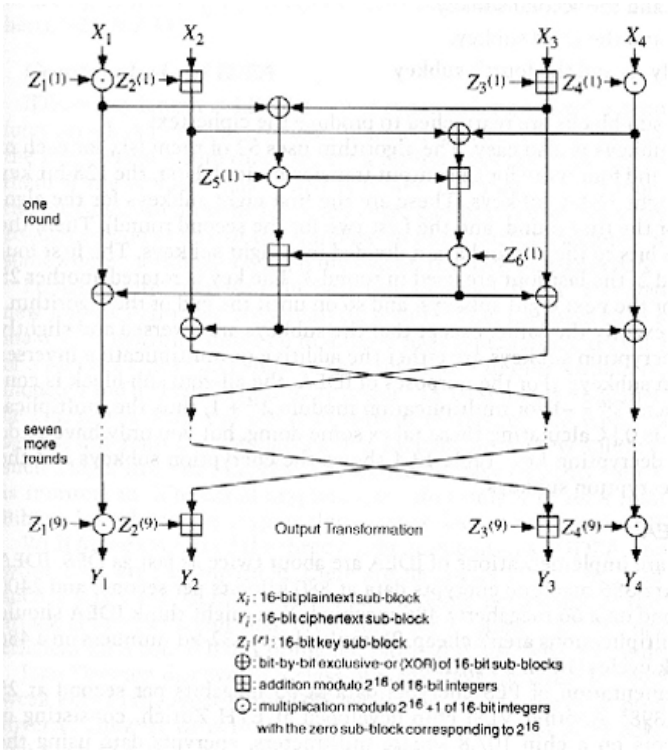
Y = S0(X1,X2) = Rot2((X1+X2) mod256)
Y = S1(X1,X2) = Rot2((X1+X2+1) mod256)
Y: output(8 bits), X1/X2(8 bits): inputs,
Rot2(Y): a 2-bit left rotation on 8-bit data Y

Fig. 3 f-function

Otros sistemas de cifrado simétrico

IDEA (International Data Encryption Algorithm)

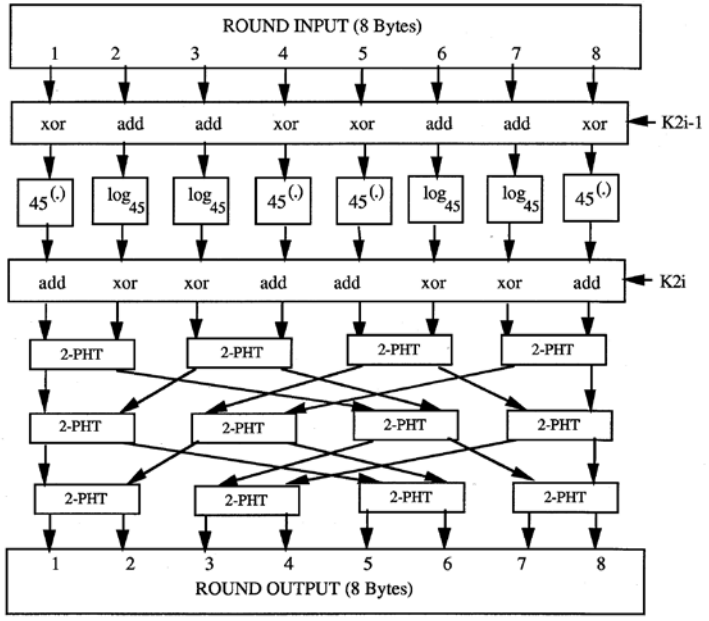
- Diseñado por X. Lai y J.L. Massey de la Escuela Politécnica Federal de Zúrich y descrito por primera vez en 1991.
- Utiliza 8 iteraciones. Operaciones x-or, sumas lógicas y sumas y multiplicaciones modulares con módulos exponenciales



Otros sistemas de cifrado simétrico

SAFER (Secure And Fast Encryption Routine)

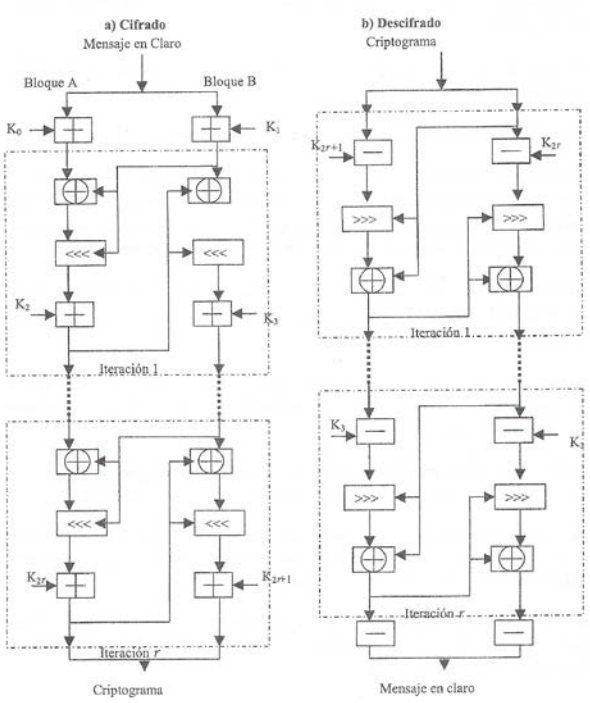
- Es una familia de cifrados por bloques diseñado inicialmente en 1993 por J.L. Massey para Cylink Corporation.
- Hay variaciones según la clave k-64, k-128 y sk-128
- Similar al IDEA. La función f se establece mediante una malla de operaciones.



Otros sistemas de cifrado simétrico

RC5 (Rivest Cypher) ó (Ron's Code)

- Diseñado por Ronald Rivest en 1994.
- Arquitectura orientada a palabras
- Parametrizado (w/r/b)
  (w = longitud de la palabra, r = número de iteraciones y b = longitud de las claves).
- Operaciones de sumas lógicas y modulares, rotaciones a izquierda y derecha.
- Constantes "mágicas"



Breve historia de la convocatoria AES (Advanced Encryption Standard)

1997 La NIST convoca la propuesta de sistemas para un nuevo sistema de cifrado estándar.

1998 La NIST anuncia la preselección de 15 candidatos: *CAST-256 CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, RIJNDAEL, SAFER+, SERPENT, TWOFISH*.

1999 La NIST depura el número de candidatos a tan sólo cinco:

MARS : IBM
RC6TM : RSA Laboratories
Rijndael : Joan Daemen, Vincent Rijmen
Serpent : Ross Anderson, Eli Biham, Lars Knudsen
Twofish : Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niel Ferguson

2000 Se suceden las reuniones, pruebas y debates. En Octubre de 2000, el sistema Rijndael se adopta como el nuevo estándar.

2001- … Puesta en marcha del nuevo estándar: documentación, especificaciones, etc.

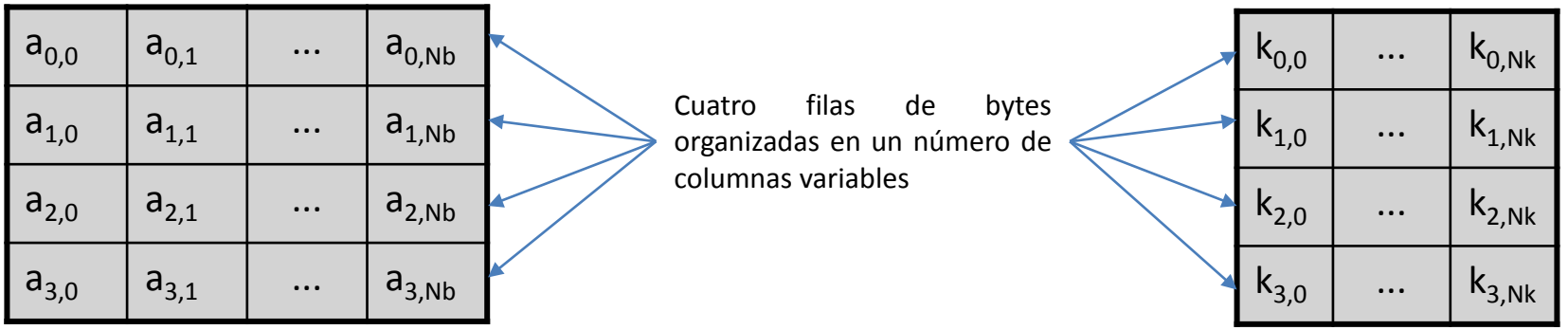
La selección de candidatos se ha realizado mediante consulta pública a expertos y diversos congresos específicos sobre el tema (*AES Candidate Conferences*)

http://csrc.nist.gov/archive/aes/

El algoritmo Rijndael (sistema AES)

Principales características

- Fundamento teórico: álgebra de cuerpos finitos GF(2^8) (operaciones a nivel de byte)
- Las claves y los textos (planos y cifrados, denominados "*estados*") se organizan en arrays múltiplos de 32 (4 bytes)



Estado
(texto plano, cifrado ó intermedio)

Clave

- El acceso se realiza por columnas y los tamaños de texto y clave son múltiplo de 32 bits:

En Rijndael: tamaño de clave y estado múltiplo de 32 con un rango desde 128 hasta 256 bits

En AES: tamaño de estado de 128 bits y tamaño de clave de 128, 192 ó 256 bits

El algoritmo Rijndael (sistema AES) (cifrado)

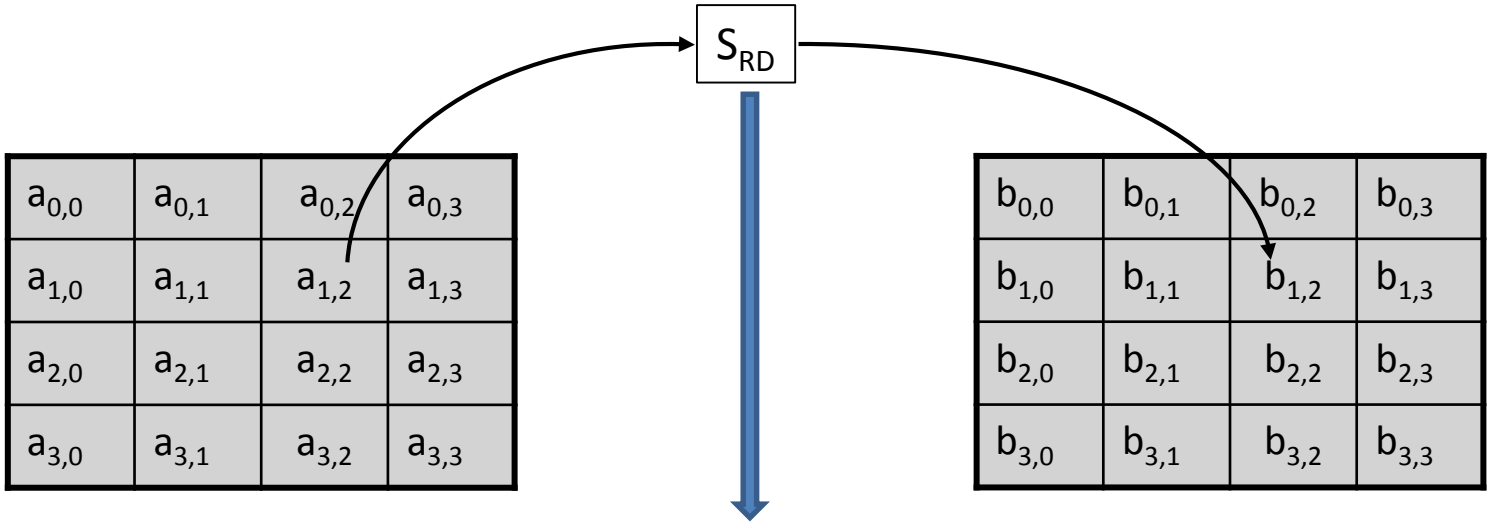
```
Rijndael(estado,clave)
{
  KeyExpansion(clave, clave_extendida);
  AddRoundKey(estado,clave_extendida[0]);
  for(i=1; i < Nr; i++)
        Round(estado,clave_extendida[i]);
  FinalRound(Estado,clave_extendida[Nr]
}
```

```
FinalRound(estado,clave_extendida[Nr])
{
  SubBytes(estado);
  ShiftRows(estado);
  AddRoundKey(estado,clave_extendida[Nr]);
}
```

```
Round(estado,clave_extendida[i])
{
  SubBytes(estado);
  Shiftrows(estado);
  MixColumns(estado);
  AddRoundKey(estado,clave_extendida[i]);
}
```

El algoritmo Rijndael (sistema AES)

SubBytes(estado)



$$\begin{bmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \\ 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \end{bmatrix} \times \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

- Transformación no lineal
- Equivalente a una tabla S

El algoritmo Rijndael (sistema AES)

ShiftRows(estado)

- Transposición del estado por filas
- Desplazamiento cíclico a izquierda de cada fila en función del tamaño de bloque

N_b	C_0	C_1	C_2	C_3
4	0	1	2	3
5	0	1	2	3
6	0	1	2	3
7	0	1	2	4
8	0	1	3	4

fila

desplazamiento

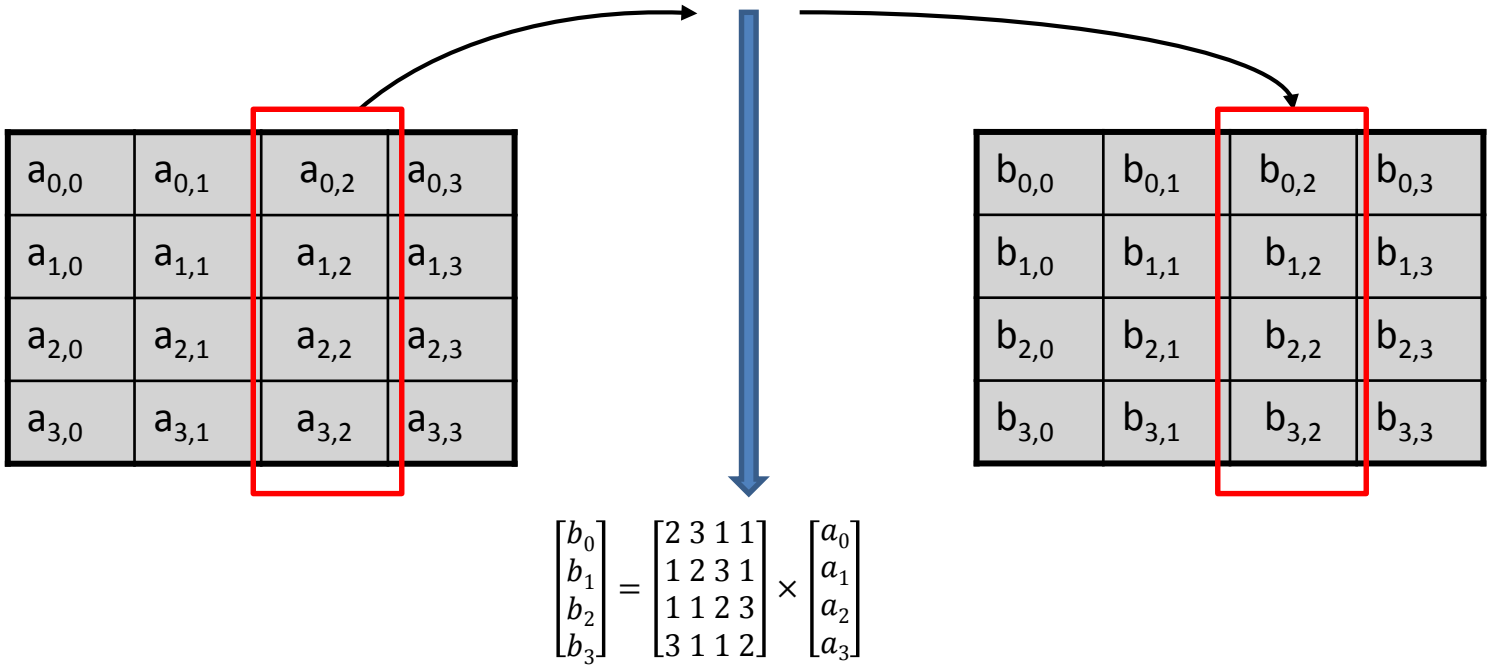
número de columnas (tamaño de bloque)

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

a	b	c	d
f	g	h	e
k	l	i	j
p	m	n	o

El algoritmo Rijndael (sistema AES)

MixColumns(estado)



- Multiplicación de polinomios representativos de cada byte
- Permutación operando a nivel de columna

El algoritmo Rijndael (sistema AES)

AddRoundKey(estado,clave_extendida[i])

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

$\oplus$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

=

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

El algoritmo Rijndael (sistema AES)

Determinación del número de iteraciones (rounds) N_r

Depende del tamaño del bloque de estado y del tamaño de la clave

$$N_k = \frac{tamaño\ de\ clave}{32}$$

N_k	N_b = 4	5	6	7	8
4	10	11	12	13	14
5	11	11	12	13	14
6	12	12	12	13	14
7	13	13	13	13	14
8	14	14	14	14	14

$$N_b = \frac{tamaño\ del\ bloque\ de\ estado}{32}$$

El algoritmo Rijndael (sistema AES)

Expansión de clave y lista de claves

KeyExpansion(clave,clave_extendida)

- La clave extendida toma la forma de un array $W[4][N_b(N_r+1)]$
- La clave a emplear para la *i*-ésima iteración, clave_extendida[i], se forma por las columnas de W que van desde $N_b \cdot i$ hasta $N_b \cdot (i+1) - 1$
- Se emplea la transformación SRD utilizada en la operación SubBytes(estado), rotaciones cíclicas a nivel de columna y adición de constantes en cada iteración para la eliminación de simetrías.
- Los algoritmos de expansión son distintos en función del valor N_k (hasta 6 ó por encima de 6)
- Definición de las constantes RC

$RC[1] = x^0$ (i.e. 01)
$RC[2] = x$ (i.e. 02)
$RC[j] = x \cdot RC[j-1] = x^{j-1}$, $j > 2$

El algoritmo Rijndael (sistema AES)

Expansión de clave y lista de claves para $N_k \leq 6$

```
KeyExpansion(byte clave[4][Nk], byte W[4][Nb(Nr+1)])
{
for(j=0; j< Nk; j++)
       for(i=0; i < 4; i++) W[i][j] = clave[i][j];
for(j=Nk; j < Nb(Nr+1; j++)
    {
     if (j mod Nk == 0)
         {
          W[0][j] = W[0][j-Nk] ⊕ SRD[W[1][j-1]] ⊕ RC[j/Nk];
          for(i=1; i < 4; i++)
                W[i][j] = W[i][j-Nk] ⊕ SRD[W[i+1 mod 4][j-1]
          }
     else
         {
          for(i=0; i < 4; i++)
                W[i][j]= W[i][j-Nk] ⊕ W[i][j-1]
         }
     }
}
```

El algoritmo Rijndael (sistema AES)

Expansión de clave y lista de claves para $N_k > 6$

```
KeyExpansion(byte clave[4][Nk], byte W[4][Nb(Nr+1)])
{
for(j=0; j< Nk; j++)
       for(i=0; i < 4; i++) W[i][j] = clave[i][j];
for(j=Nk; j < Nb(Nr+1); j++)
    {
     if (j mod Nk == 0)
         {
          W[0][j] = W[0][j-Nk] ⊕ SRD[W[1][j-1]] ⊕ RC[j/Nk];
          for(i=1; i < 4; i++)
                W[i][j] = W[i][j-Nk] ⊕ SRD[W[i+1 mod 4][j-1]
          }
     else if (j mod Nk == 4)
         {
          for(i=0; i < 4; i++)
                 W[i][j]= W[i][j-Nk] ⊕ SRD[W[i][j-1]];
         }
         else
         {
          for(i=0; i < 4; i++)
                 W[i][j]= W[i][j-Nk] ⊕ W[i][j-1];
         }
     }
}
```

El algoritmo Rijndael (sistema AES) (descifrado)

```
InvRijndael(estado,clave)
{
  EqKeyExpansion(clave, EqClave_extendida);
  AddRoundKey(estado,EqClave_extendida[Nr]);
  for(i=Nr-1; i > 0, i--)
            EqRound(estado,EqClave_extendida[i]);
  EqFinalRound(Estado,EqClave_extendida[0]
}
```

```
EqFinalRound(estado,EqClave_extendida[0])
{
  InvSubBytes(estado);
  InvShiftRows(estado);
  AddRoundKey(estado,EqClave_extendida[Nr]);
}
```

```
Round(estado,EqClave_extendida[i])
{
  InvSubBytes(estado);
  InvShiftrows(estado);
  InvMixColumns(estado);

AddRoundKey(estado,EqClave_extendida[i])
;
}
```

El algoritmo Rijndael (sistema AES)

Expansión de clave y lista de claves para descifrado

```
EqKeyExpansion(clave,EqClave_extendida)
{
  KeyExpansion(clave,EqClave_extendida);
  for(i=1; i < Nr; i++)
      InvMixColumns(EqClave_extendida[i]);
}
```

¿ preguntas ?