SEGURIDAD DE LA INFORMACIÓN

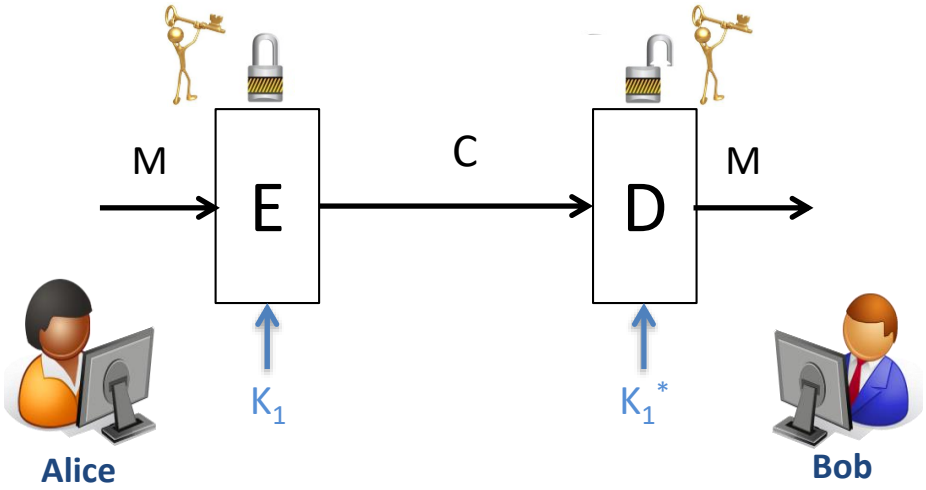
TEMA 2

TÉCNICAS CRIPTOGRÁFICAS BÁSICAS

(Y SERVICIOS DE SEGURIDAD ASOCIADOS)

Recordatorio…

- En esta situación, el mismo algoritmo de descifrado *D* será usado por todos los receptores, pero cada uno necesitará la clave correspondiente de descifrado (K_1^*, K_2^*, K_3^* ...)
- En resumen, para la comunicación específica entre *Alice* y *Bob*:



$$D_{K1^*}(E_{K1} (M)) = M$$

- Por lo tanto, en las nuevas condiciones anteriores, **es posible hacer públicos los algoritmos *E* y *D***
  - De hecho, se pueden evaluar públicamente para detectar posibles fallos
    - En caso de no tener fallos, entonces se pueden introducir en herramientas comerciales, etc.
  - Esto se formaliza en el **segundo principio de *Kerckhoffs***:
    - *"The system must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience"*
- Por lo tanto, la seguridad del sistema dependerá finalmente de que *Alice* y *Bob* mantengan en secreto las claves secretas *K* y *K**
  - Los **algoritmos simétricos** son aquellos en los que *K* y *K** son la misma clave, y se denomina **clave de sesión**
  - En los **algoritmos asimétricos**, las claves *K* y *K** son distintas

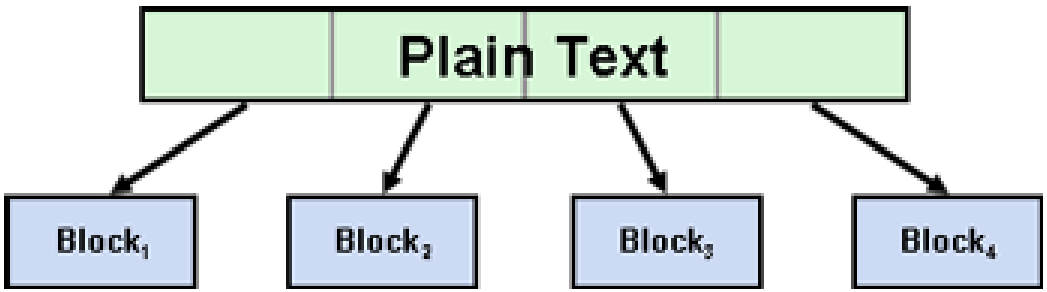
- A partir de la expresión

$$E\,(M) = C$$

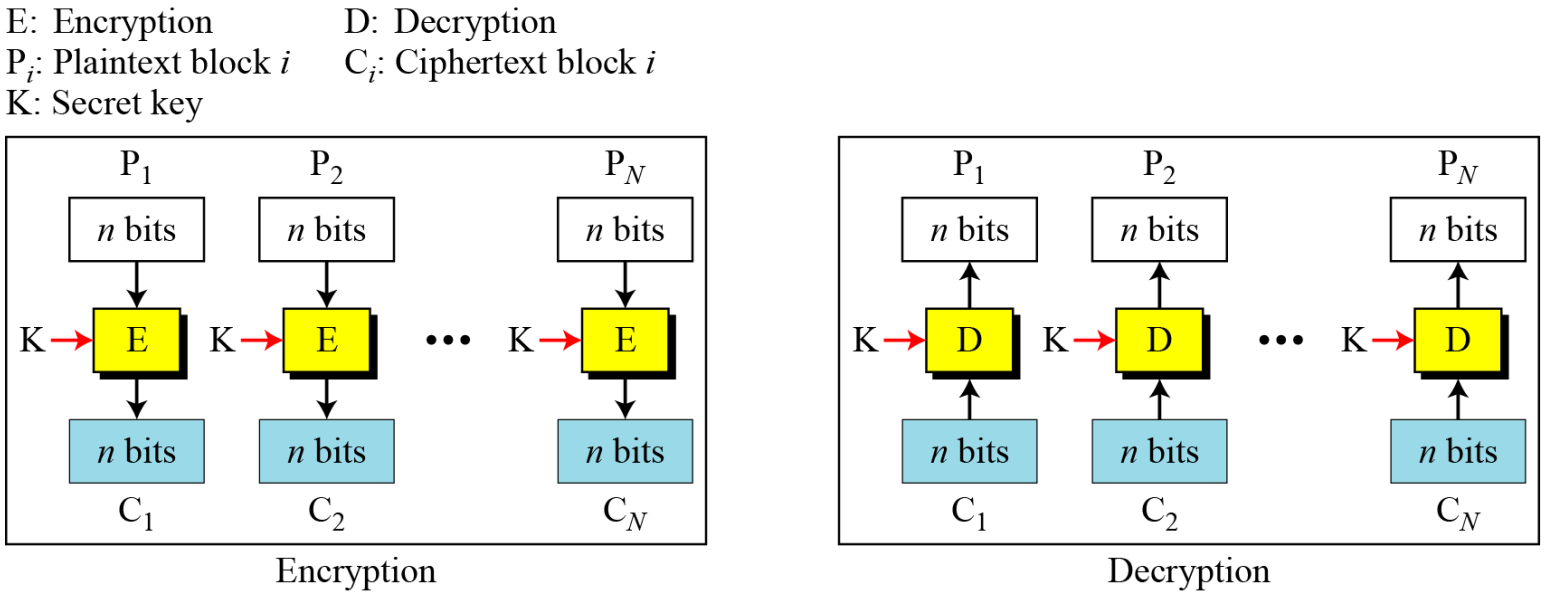
se podría pensar que el algoritmo de cifrado procesa todo el mensaje de una sola vez

  - Sin embargo, por cuestiones de diseño, es raro que ocurra eso

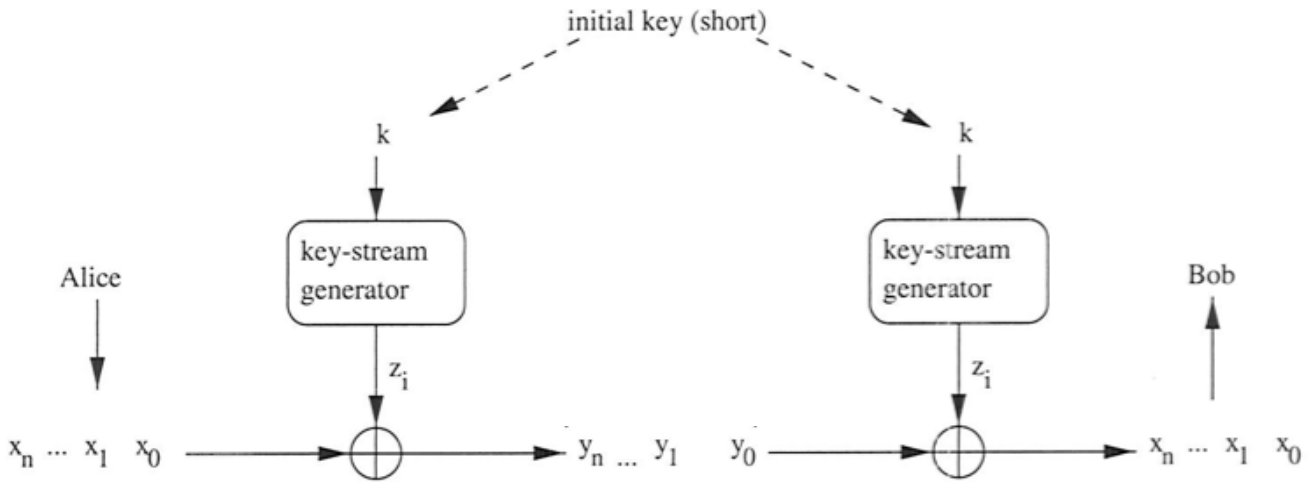
- De hecho, son muchos los algoritmos que necesitan procesar el mensaje *M* en bloques de *n* bits, denominándose entonces **cifrados en bloque**



- La longitud específica *n* de los bloques viene determinada por el propio diseño interno del algoritmo
- Cada uno de ellos se cifra de la misma forma, como se observa en la figura



- Otros algoritmos, en lugar de procesar *M* particionándolo en bloques, necesitan procesarlo bit a bit, denominándose entonces **cifrados en flujo**
- Para ello, se opera en XOR cada bit del mensaje con el bit correspondiente del flujo de clave
  - El flujo de clave depende de la clave inicial *K*



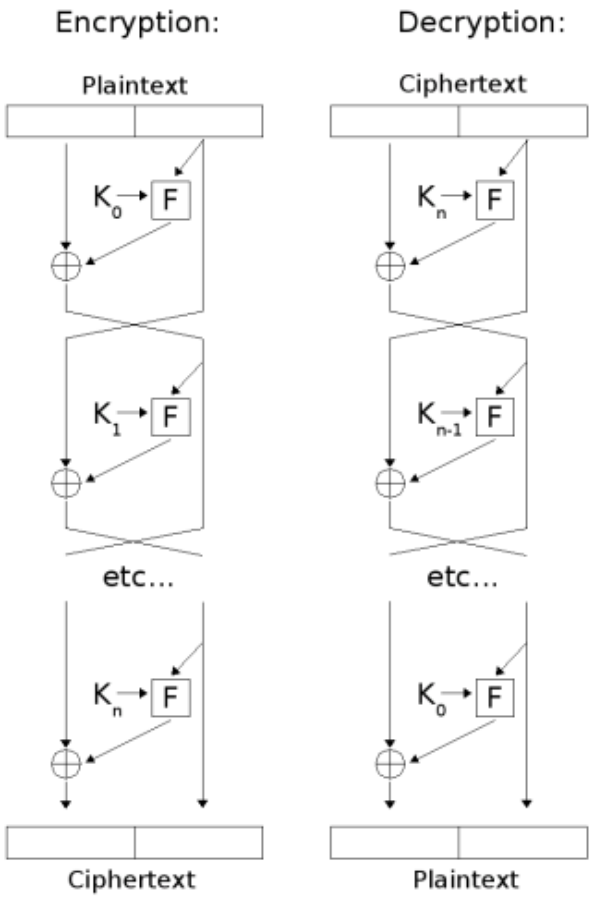
Algoritmo DES (Data Encryption Standard)

- *DES* es un algoritmo de cifrado simétrico que:
  - Usa bloques de **texto en claro de 64 bits**, y produce **bloques cifrados de igual tamaño**
  - Usa una **clave** que también es **de 64 bits** (8 octetos) de longitud
    - El último bit de cada octeto de la clave se usa como bit de paridad, por lo que la longitud efectiva de la clave (a efectos de seguridad) es de, en realidad, **56 bits**
- Diseñado por *IBM* para la competición del *National Bureau of Standards* (ahora *NIST*), en la que se solicitaban propuestas de algoritmos que pudiesen usarse como estándares para:
  - cifrado de datos en transmisión
  - cifrado de datos en almacenamiento

  por parte de el Gobierno americano, las empresas privadas y, en general, de cualquier tipo de usuario

Algoritmo DES

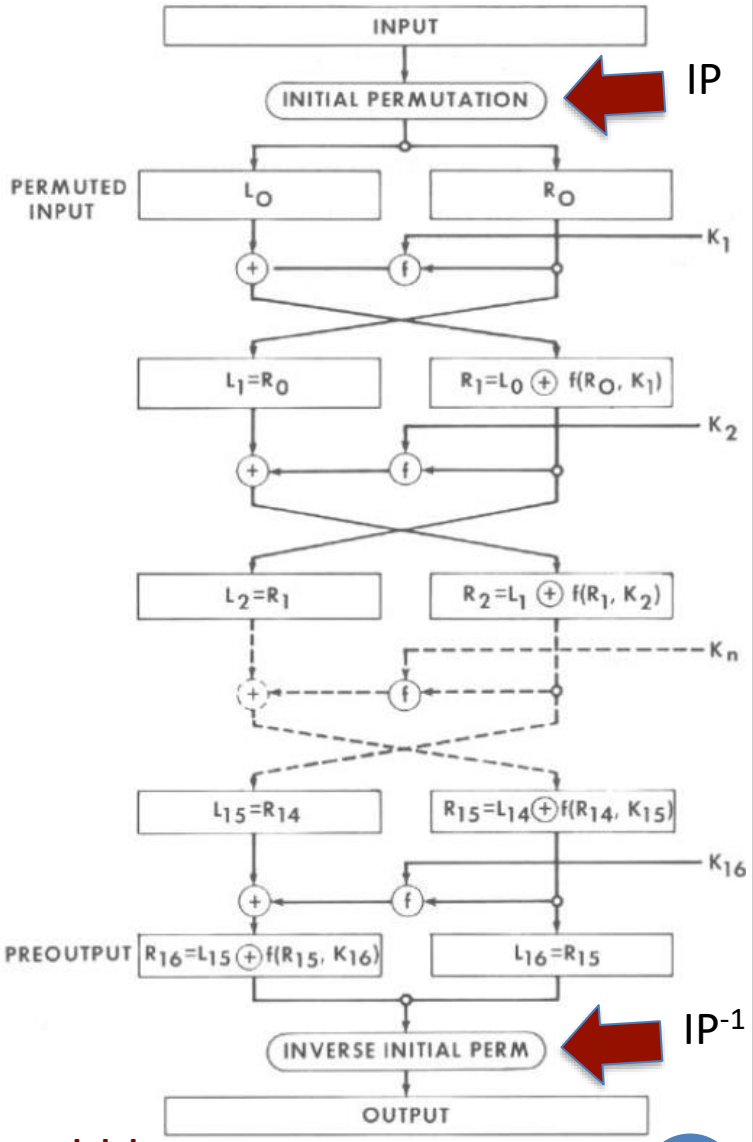
- Para el desarrollo del *DES*, *IBM* partió de *Lucifer*, un algoritmo propio desarrollado con anterioridad y usado principalmente en entornos bancarios
- Lucifer se basaba en el uso de una técnica denominada **red de Feistel**, y usaba una longitud de clave de 128 bits



Red de Feistel

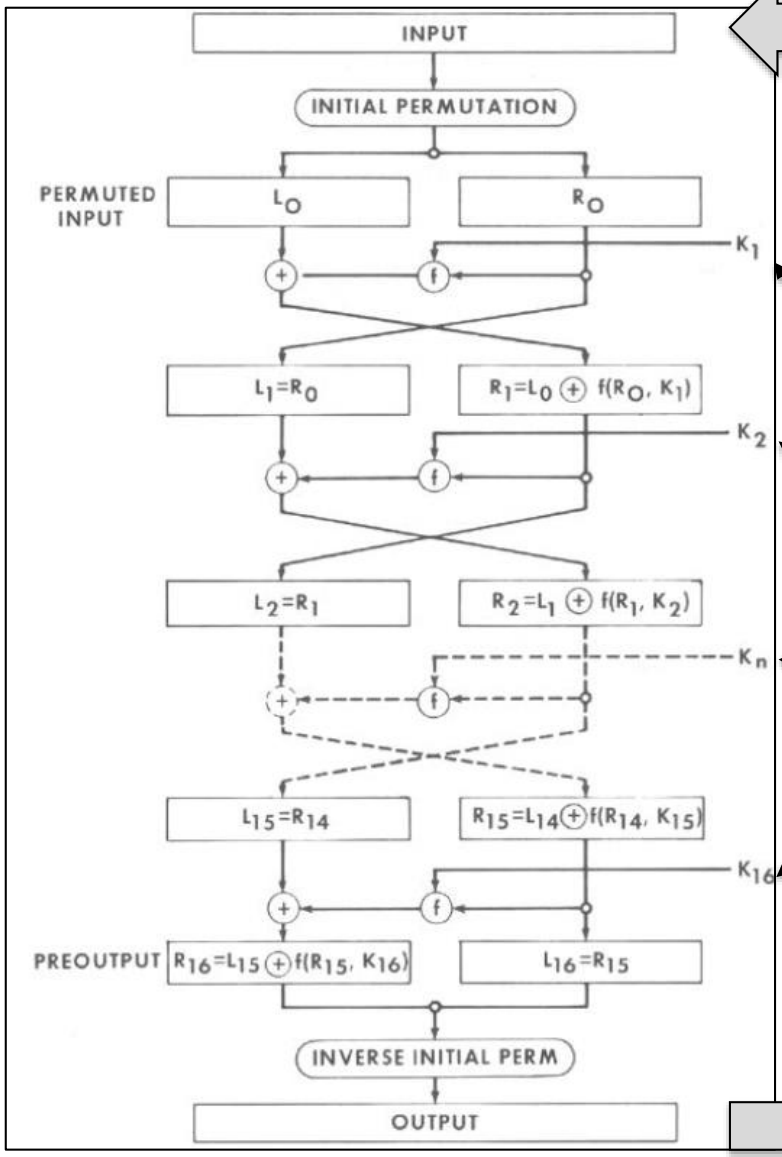
Algoritmo DES

- La idea de la red de Feistel queda reflejada en el propio *DES* como muestra la figura
- El esquema corresponde a la operación de cifrado (**16 etapas**) que se realiza para cada uno de los bloques del texto en claro
  - Usando 16 claves en cada etapa



Esquema general del algoritmo DES

Algoritmo DES



64 bits ... 64 bits

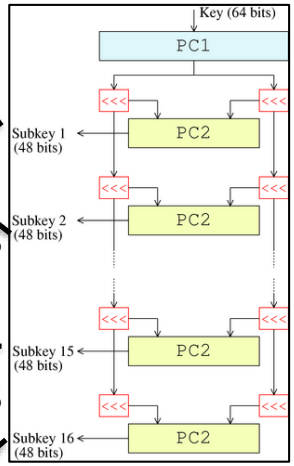
Texto a Cifrar

48 bits

48 bits

48 bits

48 bits



Semilla

Clave Privada

64 bits

64 bits ... 64 bits

Texto Cifrado

Algoritmo DES

- En el diagrama anterior se observan dos permutaciones (antes y después de las 16 etapas correspondientemente) que simplemente cambian los bits de lugar

  a) Initial Permutation (IP)

  b) Inverse Initial Permutation (IP^{-1})

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

Permutación inicial 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	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

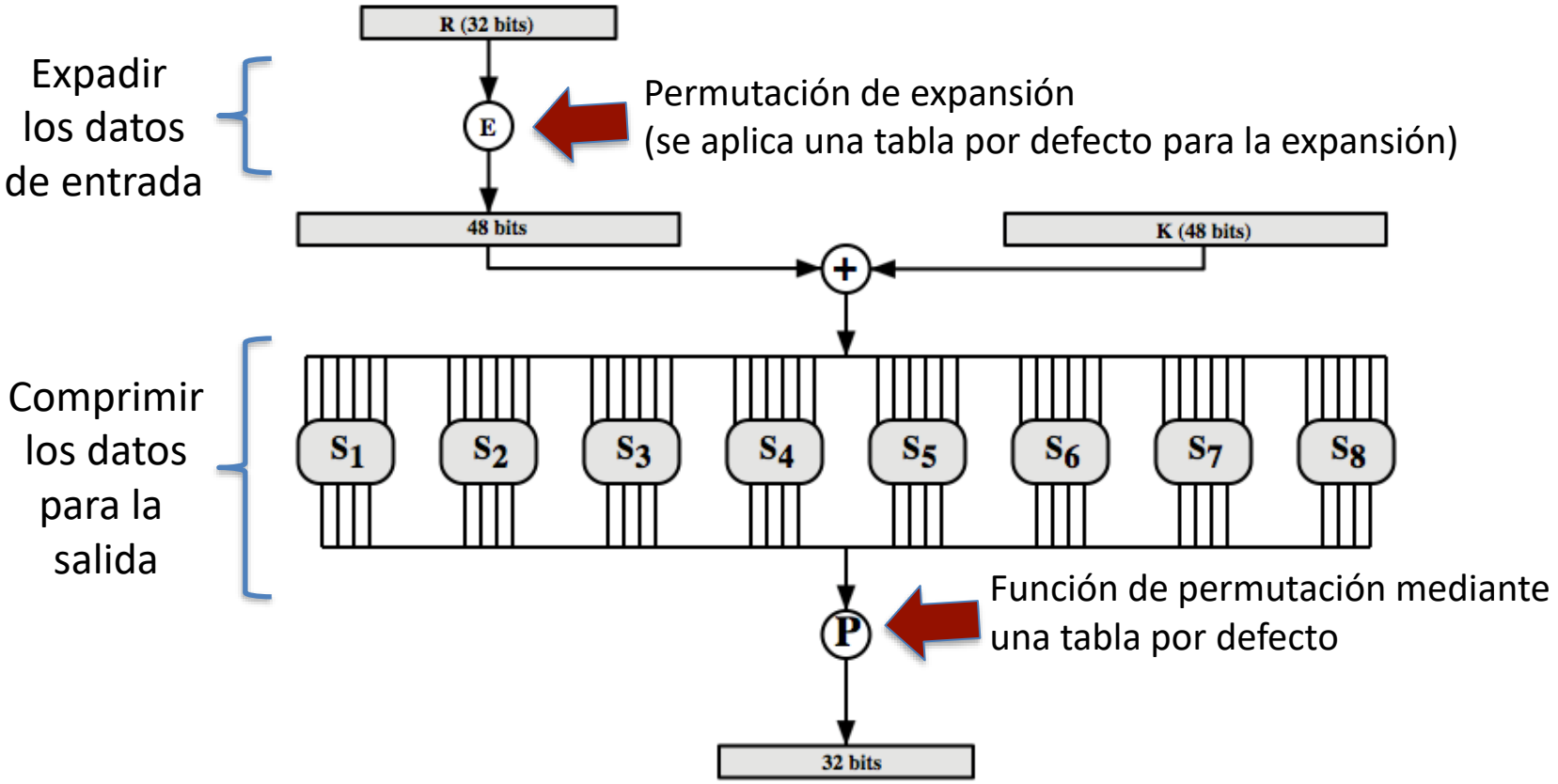
Las tablas están conteniendo valores enumerados entre 1 y 64, y la entrada a cada tabla indica la posición de un bit de entrada enumerada en la salida, siendo la matriz de entrada, y general:

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

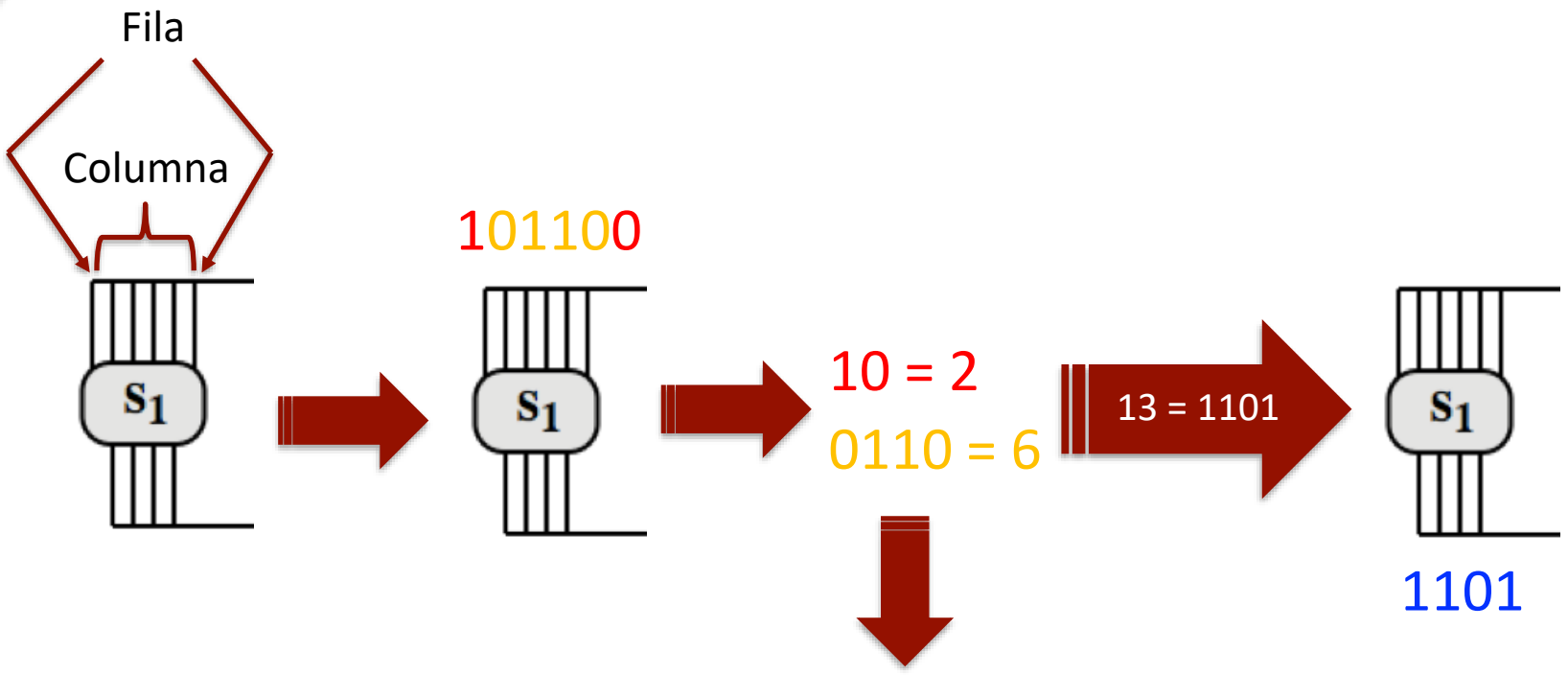
8x8 = 64 bits

Algoritmo DES

- En el esquema general mostrado con anterioridad aparece la función $f(R_{n-1}, K_n)$, que es el núcleo de *DES*, y que internamente funciona como sigue:



Algoritmo DES - cajas negras



S_2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Algoritmo DES – Tablas de permutaciones

S1

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

S2

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

S3

Fila	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

S4

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

S5

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

S6

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

S7

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

S8

Fila	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

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

E

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

P

Ejemplo 1: DES – cajas negras

- Si en una de las vueltas del algoritmo DES, la entrada a las cajas negras (S_i) corresponde a D_{hex} = BB7A 0742 8DCF, computar el valor de salida teniendo en cuanta las tablas correspondientes:

S1

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

S2

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

S3

Fila	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

S4

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

S5

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

S6

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

S7

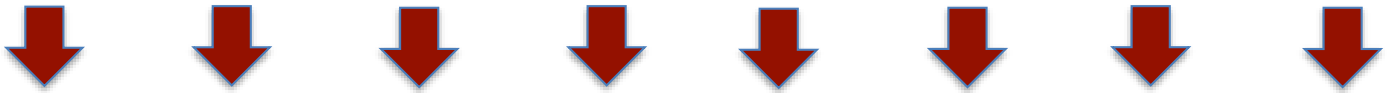
Fila	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8

Fila	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

Ejemplo 1: DES – cajas negras

- D_{hex} = BB7A 0742 8DCF
- D_{bin} = 1011 1011 0111 1010 0000 0111 0100 0010 1000 1101 1100 1111
- D_{bin} = 101110 110111 101000 000111 010000 101000 110111 001111



S1 2-7 | S2 3-11 | S3 2-4 | S4 1-3 | S5 0-8 | S6 2-4 | S7 3-11 | S8 1-7

S1

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

S2

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

S3

Fila	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

S4

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

S5

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

S6

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

Ejemplo 1: DES – cajas negras

- D_{hex} = BB7A 0742 8DCF
- D_{bin} = 1011 1011 0111 1010 0000 0111 0100 0010 1000 1101 1100 1111
- D_{bin} = 101110 110111 101000 000111 010000 101000 110111 001111

S7

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

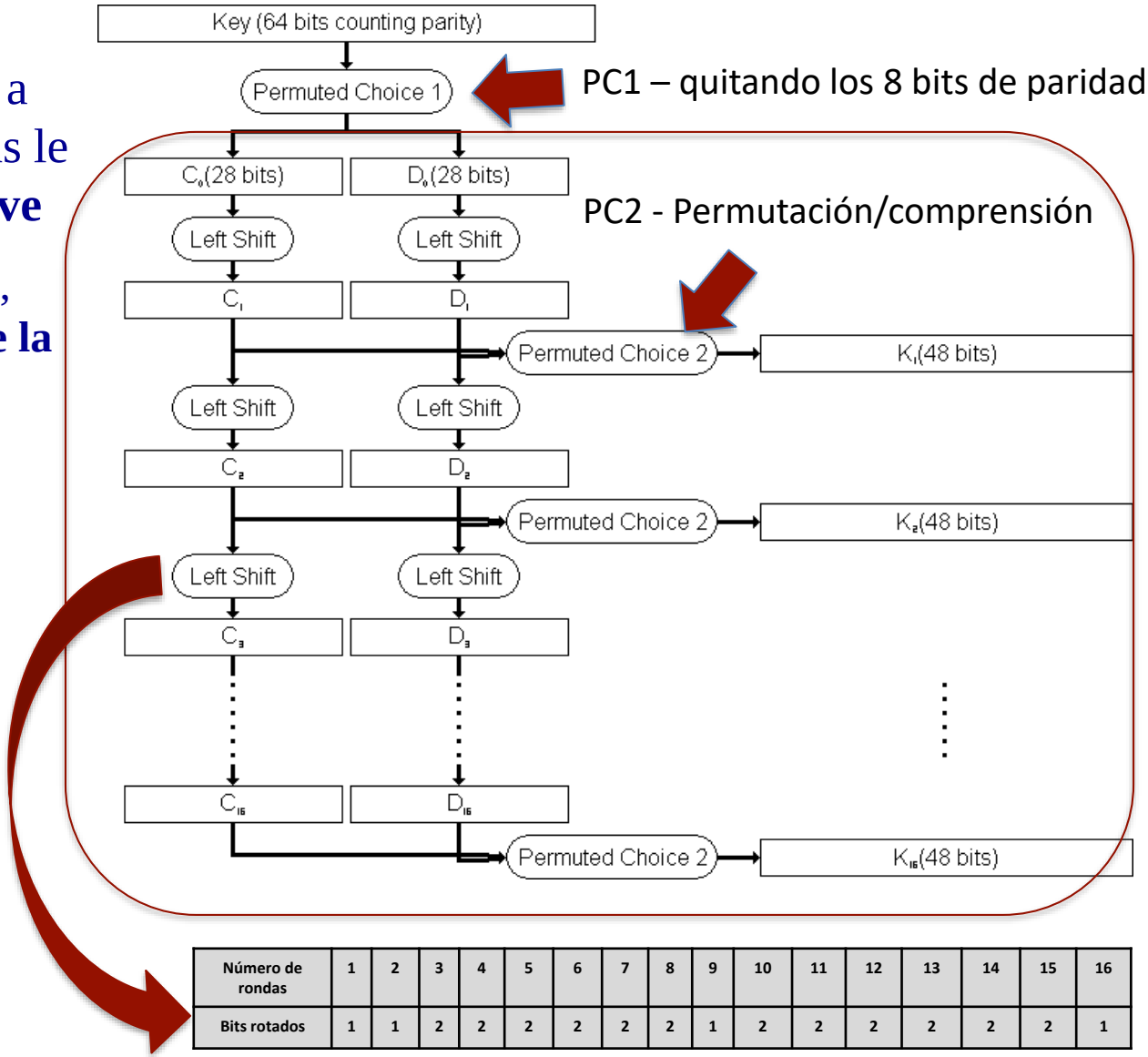
S8

Fila	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

Por consiguiente, la salida sería: 1011 1100 1000 0101 1000 1100 1111 0100 que corresponde a BC858CF4

Algoritmo DES – subclaves

- Se observa también que a cada una de las 16 etapas le corresponde una **subclave**
  - En total, **16 subclaves**, **generadas a partir de la clave inicial *K***, como muestra la figura



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	32
14	6	61	53	45	37	29
21	13	5	28	20	12	4

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

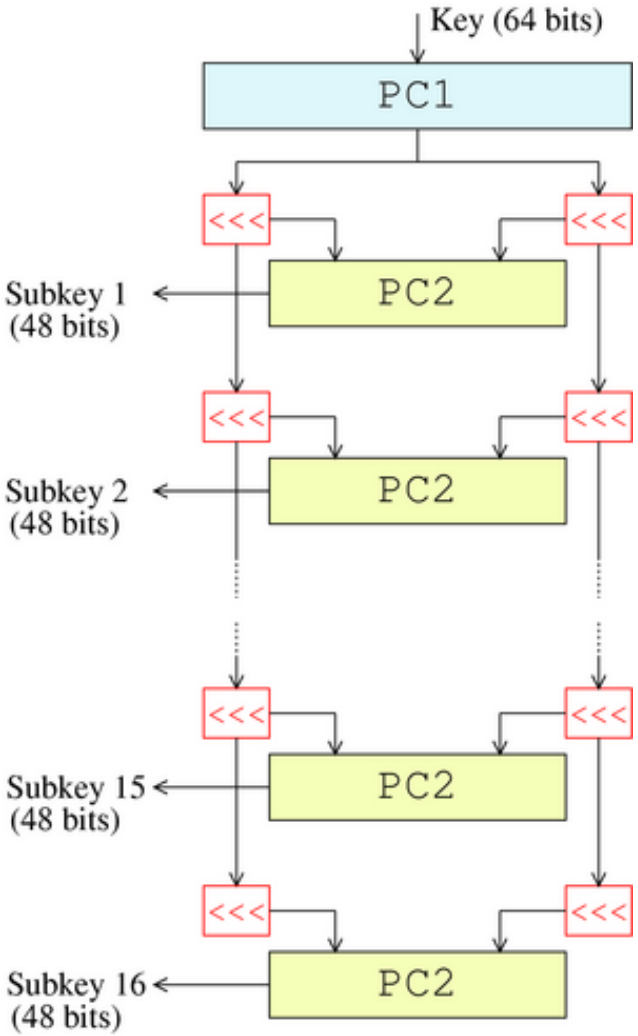
PC2

Número de rondas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotados	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Número de desplazamientos

Algoritmo DES – subclaves

- De forma simplificada, generación subclaves K_i:



Algoritmo DES – subclaves

- PC-1

Key:

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

PC1

C:

D:

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

PC-1 es equivalente a decir:

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

8x8 = 64 bits

Algoritmo DES – subclaves

- PC-1

Key:

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

De arriba a derecha

De arriba a izquierda

PC1

C:

D:

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 de permutar las posiciones siguiendo las posiciones de la matriz:

C:

D:

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

C_i = 0111011 0011100 0111001 1101111

D_i = 0010001 0000110 1101110 1111000

Algoritmo DES – subclaves

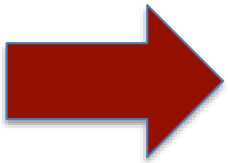
- Desplazamiento a la izquierda. Si i=1, entonces la rotación es de 1 bit:
  - C_1 = 111011 0011100 0111001 1101111**0**
  - D_1 = 010001 0000110 1101110 1111000**0**

- PC-2: C_1D_1 = 1110110 0111000 1110011 1011110 0100010 0001101 1011101 1110000

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

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



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

K_1 = 01111110 11111000 10101101 ….
hasta tener los 48 bits

8x7 = 56 bits

Algoritmo DES

- La siguiente animación repasa los conceptos básicos de DES y muestra sus funcionalidades:
  - http://kathrynneugent.com/animation.html

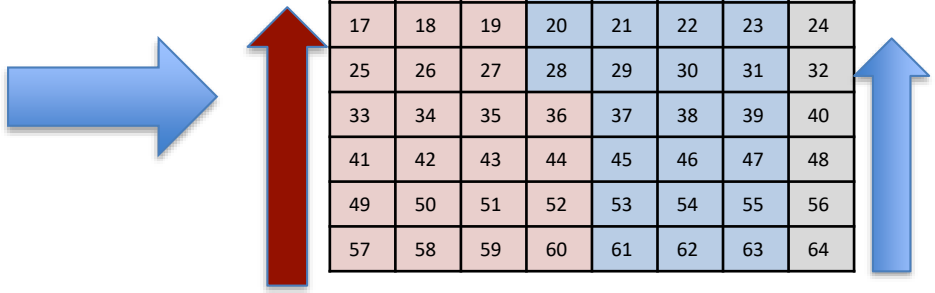
Algoritmo DES – subclaves (ejercicio 1)

- Si tenemos como clave:
  - K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001
- Calcular la primera subclave (K_1) teniendo en cuenta:

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

8x7 = 56 bits



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

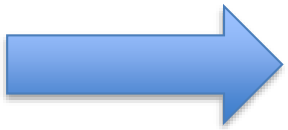
8x8 = 64 bits

Número de rondas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotados	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

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

6x8 = 48 bits



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

8x7 = 56 bits

Algoritmo DES – subclaves (ejercicio 1)

- Si tenemos como clave:
  - K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001
- Calcular la primera subclave (K_1) teniendo en cuenta:

SOLUCIÓN: **000110 110000 001011 101111 111111**
000111 000001 110010

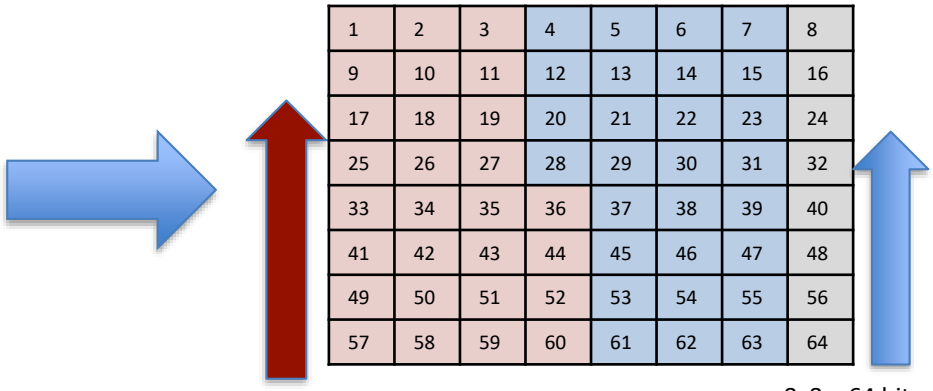
Algoritmo DES – subclaves (ejercicio 2)

- Considerando el enunciado anterior, calcular la K_7 teniendo en cuenta:

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

8x7 = 56 bits



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

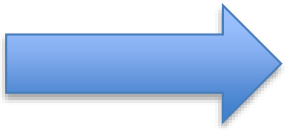
8x8 = 64 bits

Número de rondas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotados	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

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

6x8 = 48 bits



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

8x7 = 56 bits

Algoritmo DES – subclaves (solución)

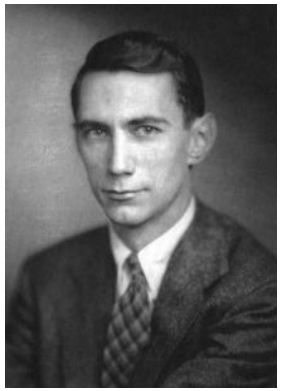
- Primero aplicamos PC-1:
  - K_{PC-1} = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111
  - Tal que:
    - C = 1111000 0110011 0010101 0101111
    - D = 0101010 1011001 1001111 0001111
- Se obtiene la lista de rotaciones para cada C_i y D_i:
  - C_1 = 11100001100110010101011111 D_1 = 10101010110011001111100011110
  - C_2 = 11000011001100101010111111 D_2 = 01010101100110011111000111101
  - ….
- Se concatena C_1D_1 y se aplica PC-2:
  - C_1D_1 = 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110
  - Tal que:
    - K_1 = 000110 110000 001011 101111 111111 000111 000001 110010
- ¿Y para las claves K_2, k_7?
  - K_2 = 011110 011010 111011 011001 110110 111100 100111 100101
  - K_7 = **111011 001000 010010 110111 111101 100001 100010 111100**

Algoritmo DES

- Una característica eficiente de *DES*, y deseable en cualquier algoritmo de cifrado, es el **efecto avalancha**
  - Es decir, **un cambio pequeño en el texto en claro, o en la clave**, produce un cambio significativo en el texto cifrado
    - Si, por el contrario, el cambio fuera pequeño, el criptoanalista tendría mucha ventaja, porque se reduciría el número de posibles textos en claro o de posible claves
    - En otras palabras: impide reducir el espacio de búsqueda de claves para un ataque de fuerza bruta
- Ejemplo de efecto avalancha en *DES*:

Plaintext: 0000000000000000 Ciphertext: 4789FD476E82A5F1	Key: 22234512987ABB23
Plaintext: 0000000000000001 Ciphertext: 0A4ED5C15A63FEA3	Key: 22234512987ABB23

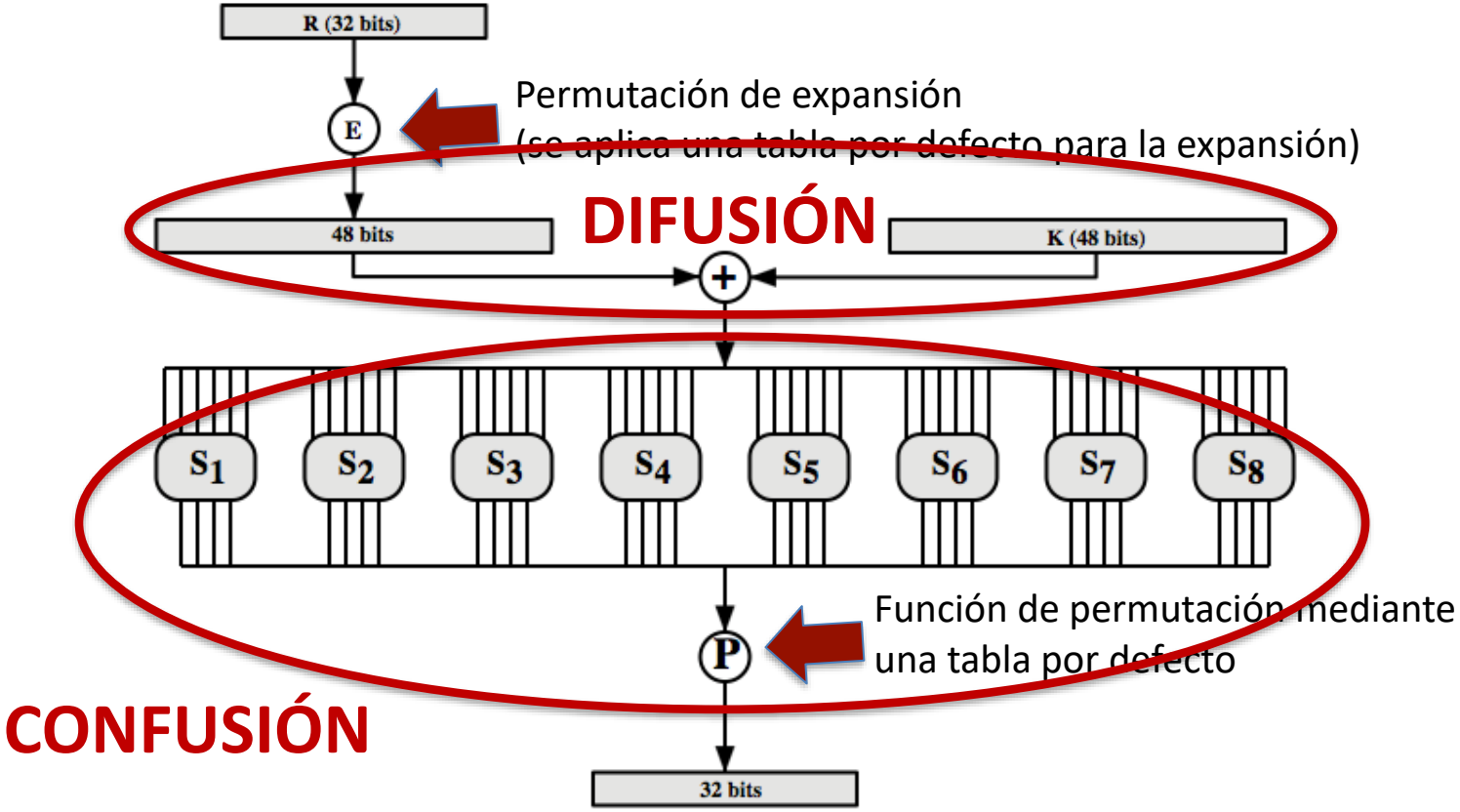
Efecto avalancha



- El concepto de efecto avalancha se deriva de las tesis de Claude Shannon, el padre de la **Teoría de la Información** en su artículo:
  - *Communication Theory of Secrecy Systems*, 1949
- En ese artículo definía, entre otros conceptos, las propiedades de **difusión** y **confusión** para evitar (o dificultar) los ataques basados en análisis estadísticos
  - **difusión**: cada carácter del texto cifrado ha de depender de diferentes partes de la clave
  - **confusión**: la relación entre el texto cifrado y la clave ha de ser tan complicada como sea posible
- Nota: es un criterio que se aplica a cualquier algoritmo de cifrado (DES, 3DES, IDEA, Camellia, etc.)

Algoritmo DES

- En el esquema general mostrado con anterioridad aparece la función $f(R_{n-1}, K_n)$, que es el núcleo de *DES*, y que internamente funciona como sigue:



Algoritmo DES

- Hasta el momento no se conoce ningún ataque al algoritmo *DES* en sí mismo, y que haya sido completamente efectivo
- Por otro lado, un **ataque exhaustivo a la clave** (***brute-force attack***) podría parecer impracticable si suponemos, por ejemplo, una operación de descifrado por microsegundo
  - con longitud de clave de 56 bits, existen 2^{56} posibles claves (= 7.2×10^{16})

Algoritmo DES

- Sin embargo, se puede suponer que el criptoanalista va a disponer de capacidad de descrifado con microprocesadores en paralelo y, por lo tanto, la situación cambia drásticamente:

Key size (bits)	Number of alternative keys	Time required at 1 decryption/µs	Time required at 10^6 decryptions/µs
32	$2^{32} = 4.3 \times 10^9$	2^{31} µs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} µs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} µs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} µs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} µs = 6.4×10^{12} years	6.4×10^{6} years

- Por lo anterior, se deduce que la longitud de clave del *DES* resulta demasiado corta si el criptoanalista dispone del hardware adecuado

DES Key Search Machine

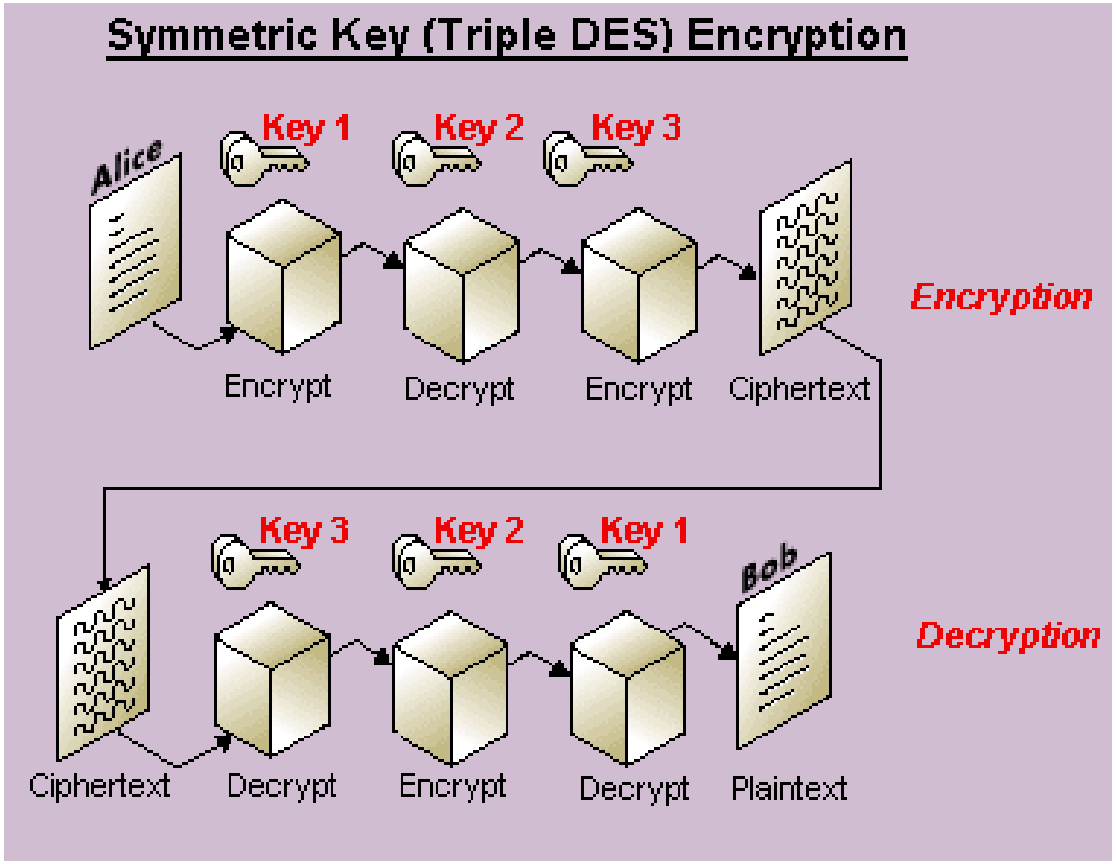
6 estaciones SUN-2, 27 placas de circuitos, 1800 chips customizados,

24 unidades de búsqueda por cada chip

Algoritmo Triple-DES

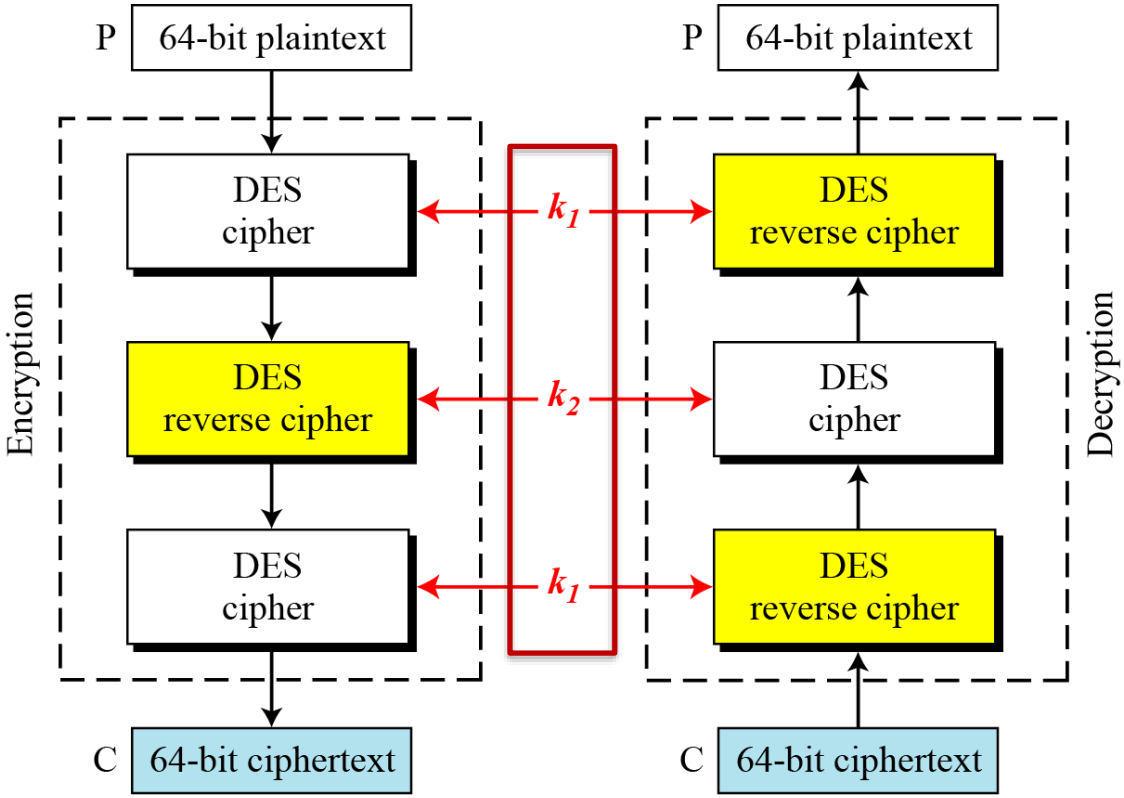
- Para aprovechar las ventajas del *DES*, y a la vez contrarrestar su escasa longitud de clave, los organismos de estandarización han adoptado el criptosistema ***Triple-DES*** (ó ***3DES***)

- Consiste en usar **una secuencia de tres operaciones *DES*, con 3 claves distintas (168 bits)**



Algoritmo Triple-DES (variante de 2 keys)

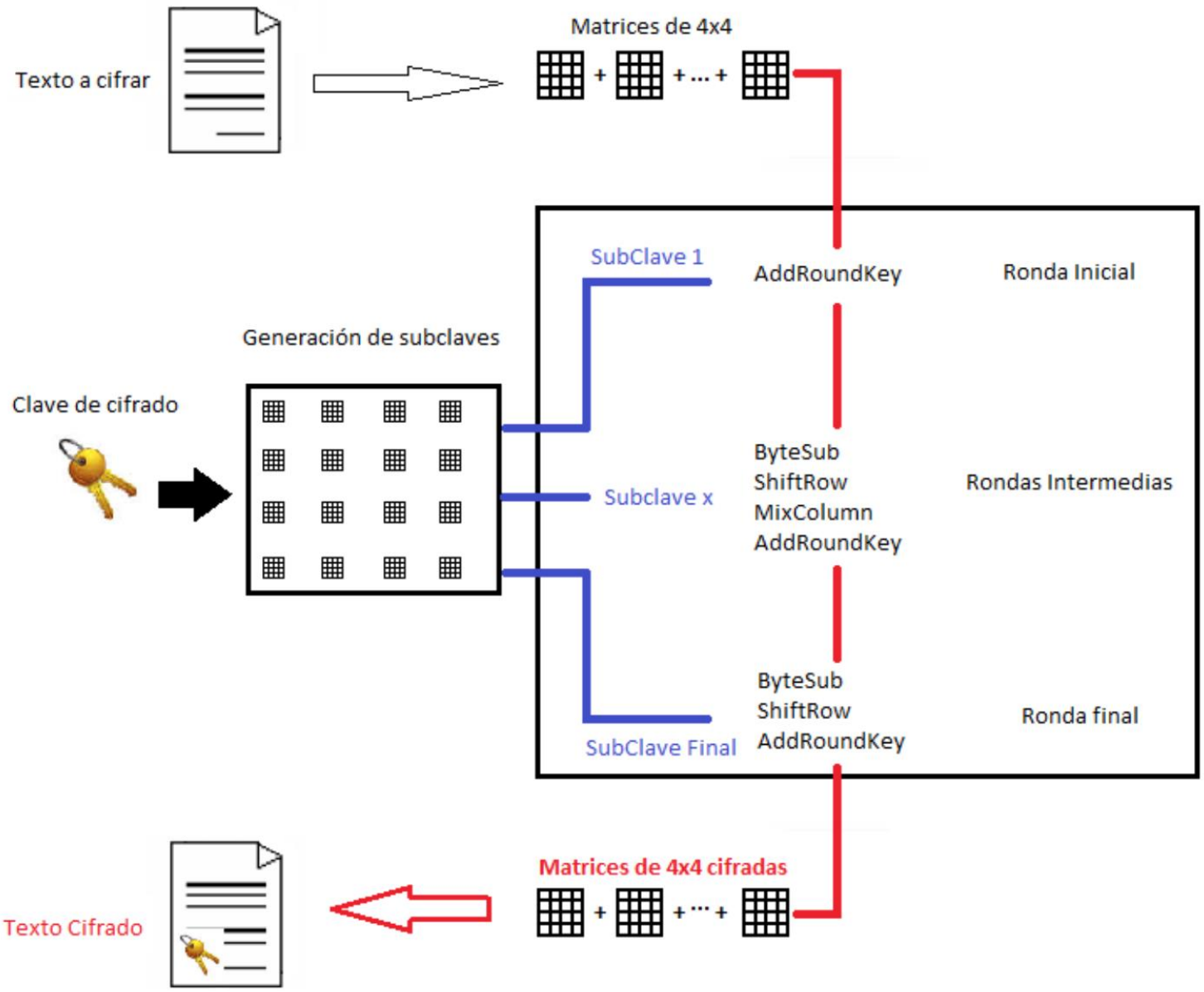
- Existe una variante en la que se utilizan sólo 2 claves, pero este esquema ha sido objeto de ataques, de forma que la **K1 = K3 (112 bits)**



Algoritmo AES (Advanced Encryption Standard)

- *AES* fue publicado por *NIST* en 2001 como estándar de **cifrado simétrico en bloque** para sustituir a *DES*, especialmente en aplicaciones comerciales
  - su nombre original es *Rijndael*, por sus autores *Rijmen* y *Daemen*
  - utiliza **una clave de 128**, **192** o **256 bits**
  - **la longitud *n* de cada bloque** de datos en los que se subdivide *M* **es 128 bits**
- *AES* no utiliza una red de Feistel, sino una **red de sustitución-permutación**
- Cada etapa de *AES* se compone de cuatro funciones distintas:
  - **sustitución de byte**
  - **permutación**
  - **operaciones aritméticas en campo finito**
  - **XOR**

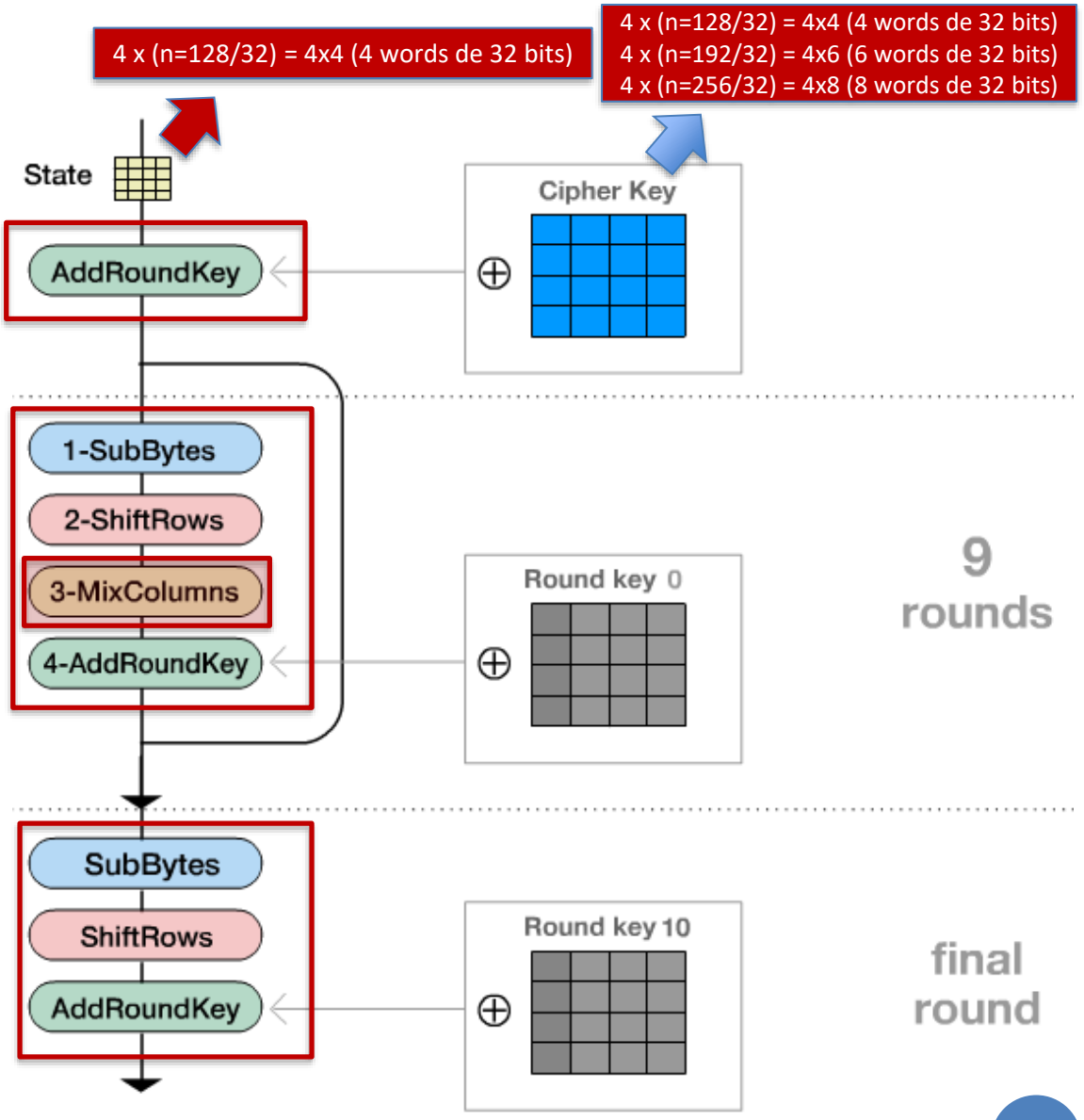
Algoritmo AES



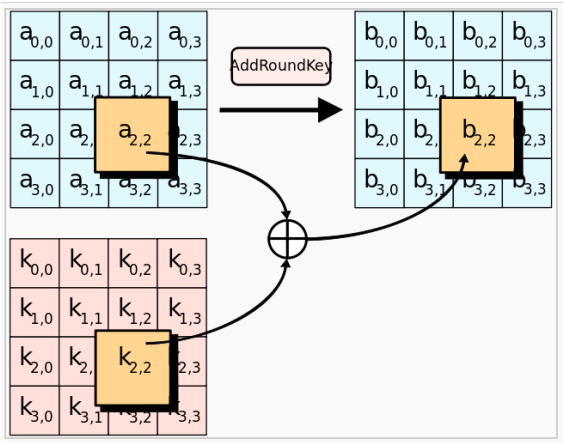
Algoritmo AES

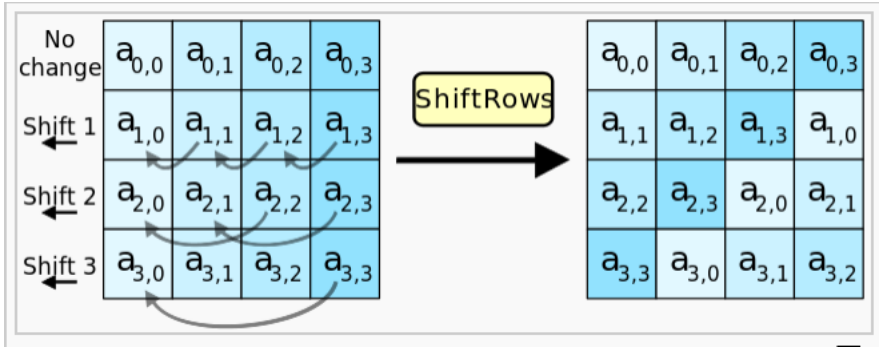
- La figura muestra el proceso de cifrado en ***AES* para 128 bits** de clave (**10 etapas**)

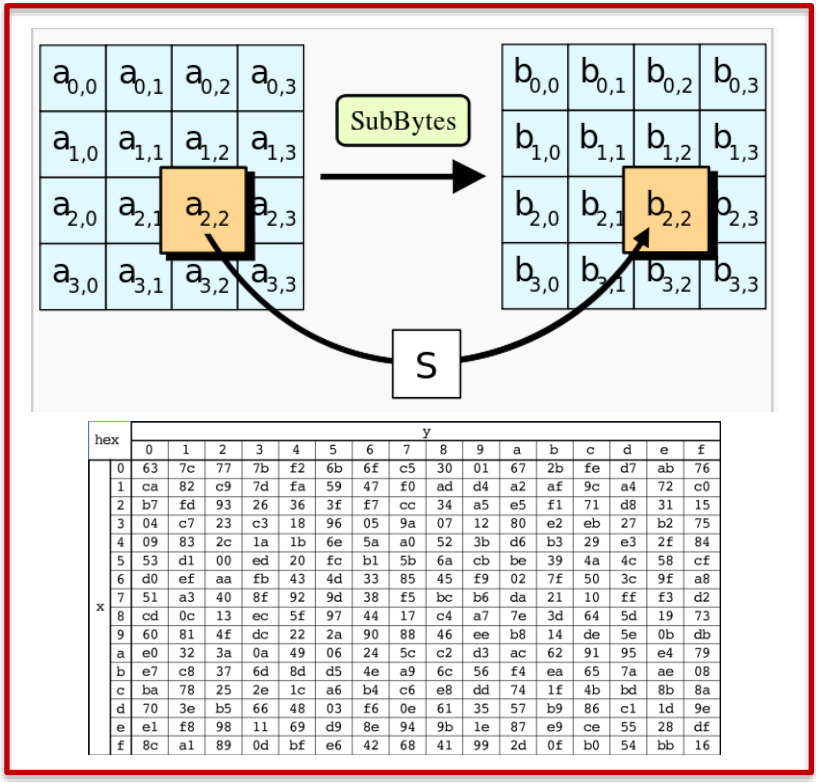
- Otras posibilidades:
  - **192 bits (12 etapas)**
  - **256 bits (14 etapas)**



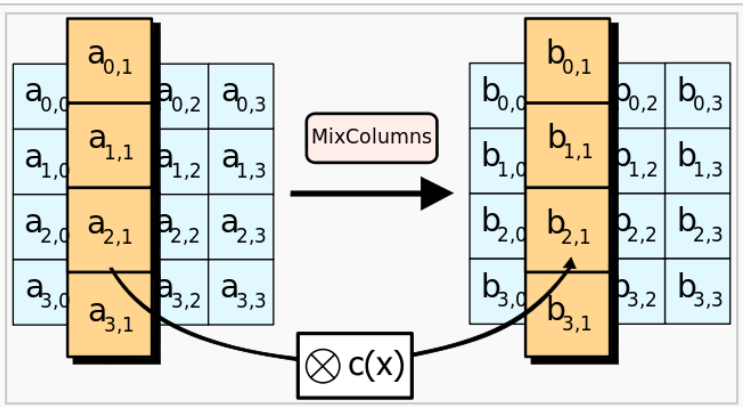
Algoritmo AES





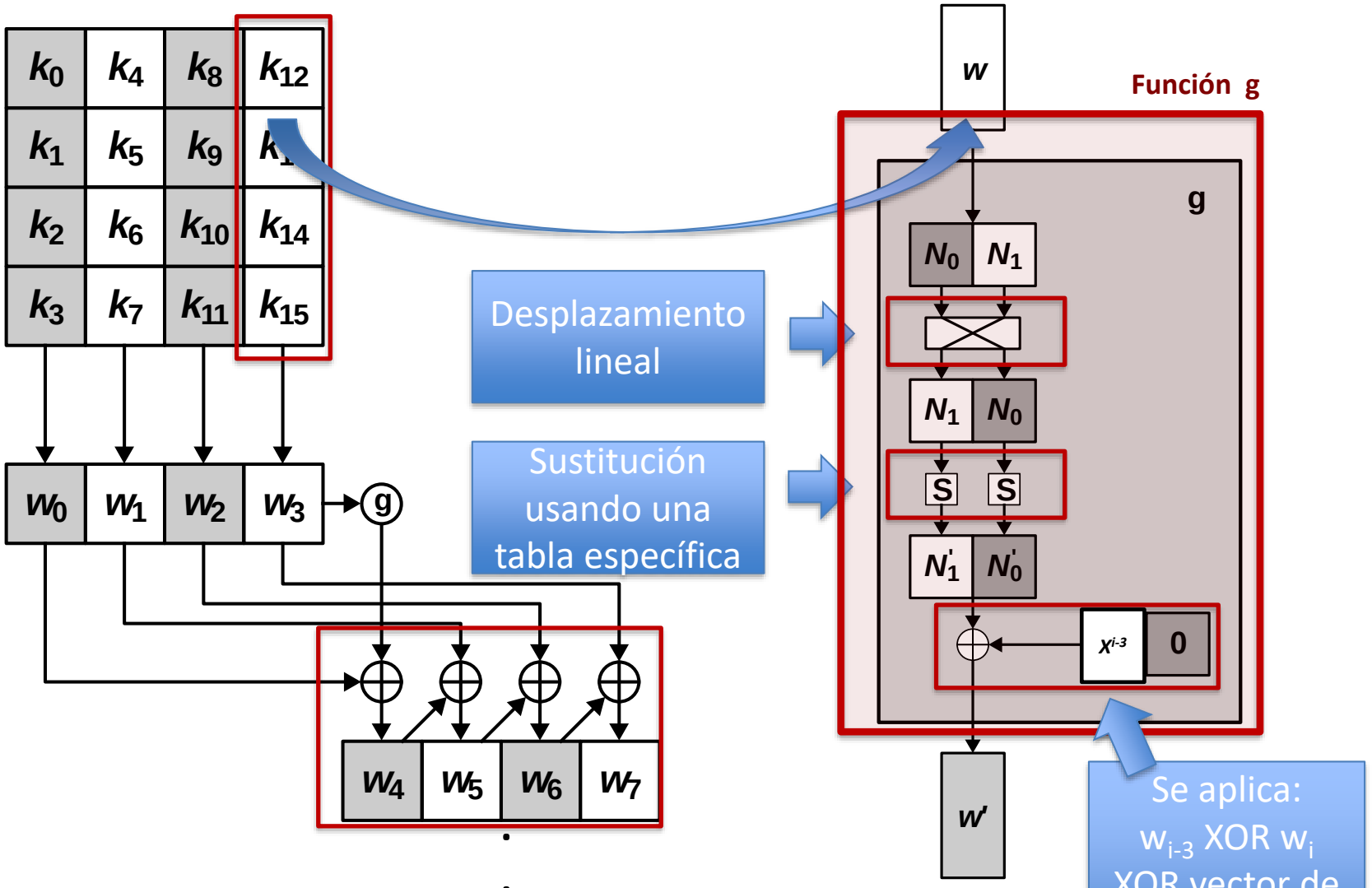


hex	y															
x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	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



MixColumn mutiplica la matriz que se está computando con matrices preestablecidas

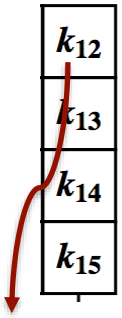
Algoritmo AES – Expansión de las claves AES



Algoritmo de expansión de Claves en AES = Rijndael key schedule

Algoritmo AES – caja negra

Rotword



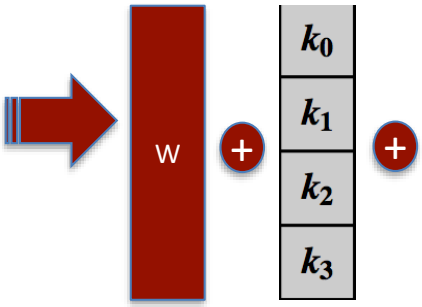
Sólo se mueve K_{12} al final, resultando en:

k_{13},k_{14},k_{15},k_{12}

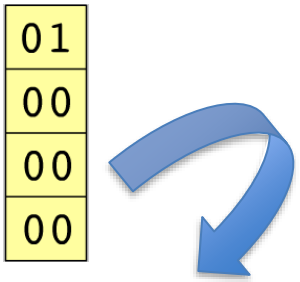
SubBytes

hex	y															
x	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

XOR (w_{i-3})



Rcon (XOR)



02	04	08	10	20	40	80	1b	36	
00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	Rcon

Vector Rcon (vector round constant): se aplica una columna en cada ronda

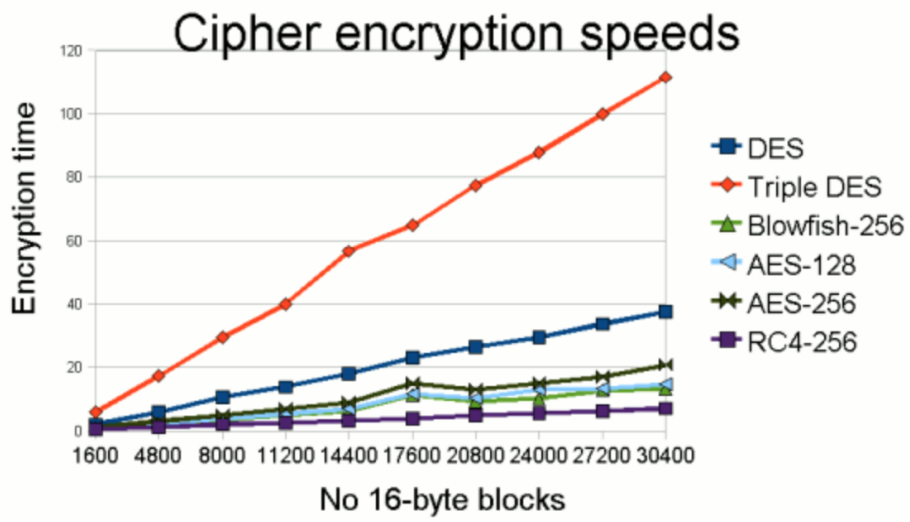
Rondas	Long de las claves
10	16
12	24
14	32

Algoritmo AES

- La siguiente animación repasa los conceptos básicos de AES y muestra sus funcionalidades:
  - http://www.formaestudio.com/rijndaelinspector/
  - https://www.youtube.com/watch?v=gP4PqVGudtg

Algoritmo AES

- Rendimiento de AES comparado con otros algoritmos simétricos



Date	Minimum of Strength	Symmetric Algorithms
2010 (Legacy)	80	2TDEA*
2011 - 2030	112	3TDEA
> 2030	128	AES-128
>> 2030	192	AES-192
>>> 2030	256	AES-256