

# SEGURIDAD DE LA INFORMACIÓN

## TEMA 2

**TÉCNICAS CRIPTOGRÁFICAS BÁSICAS  
(Y SERVICIOS DE SEGURIDAD ASOCIADOS)**

# Indice del tema (I)

- Introducción a la criptografía clásica
  - Cifrados por sustitución y transposición. Ejemplos
  - Cifrado producto
  - Cifrado Vernam (one-time pad)
- Algoritmos simétricos
  - Fundamentos
  - Algoritmo DES
  - Algoritmo triple-DES
  - Algoritmo AES
  - Otros algoritmos simétricos
  - Modos de operación para algoritmos simétricos
  - Ventajas y desventajas de los algoritmos simétricos

## Indice del tema (II)

- Algoritmos asimétricos (o de clave pública)
  - Cifrado/descifrado
  - Firma Digital
  - Intercambio de Claves
  - Algoritmo de Diffie-Hellman
  - Algoritmo RSA
- Otras primitivas criptográficas
  - Funciones hash
  - Códigos de autenticación de mensajes
- Referencias bibliográficas

## Introducción a la criptografía clásica



## Cifrados por sustitución y transposición. Ejemplos

- Como se ha visto en la última tabla del capítulo anterior, un algoritmo de cifrado es uno de los mecanismos para la implementación de servicios de seguridad
- Criptografía: ciencia que estudia cómo mantener la seguridad en los mensajes ( $M$ )
  - usando, entre otros mecanismos, los algoritmos de cifrado
- Criptoanálisis: ciencia que estudia cómo romper los textos cifrados
- Criptología: Criptografía + Criptoanálisis

- El algoritmo de cifrado es un mecanismo que transforma un texto en claro en texto ininteligible
  - Su objetivo es dar cobertura al servicio de Confidencialidad
  - El algoritmo de cifrado se denota por **E** (del inglés “encrypt”) y opera sobre el **texto en claro  $M$**  (mensaje) para producir el **texto cifrado  $C$**  (criptograma)
- La transformación inversa, o sea, de un texto cifrado en un texto en claro, se denomina algoritmo de descifrado
  - El algoritmo de descifrado se denota por **D** (“decrypt”) opera sobre  $C$  para producir el mensaje  $M$
- Se cumple que:

$$E(M) = C$$

$$D(C) = M$$

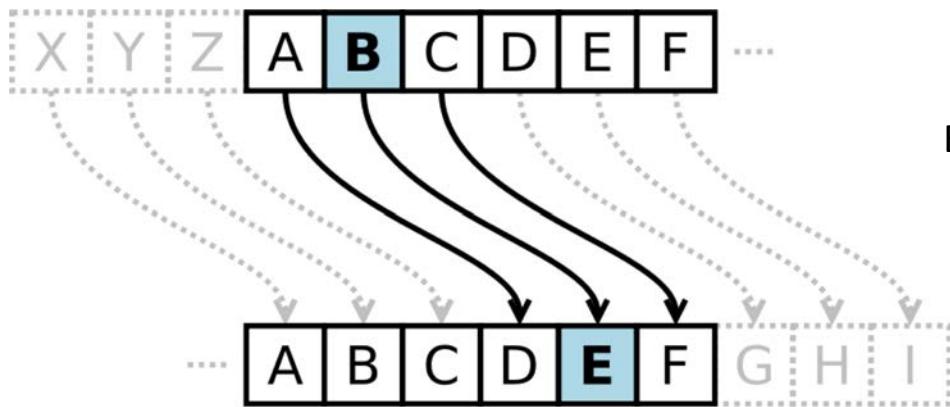
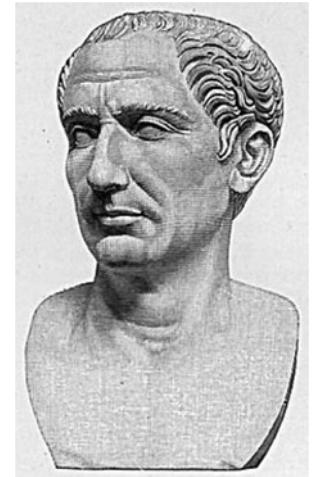
$$D(E(M)) = M$$

- Antes de la existencia de ordenadores, la criptografía clásica consistía en algoritmos basados en caracteres
- Los algoritmos criptográficos clásicos o bien sustituían caracteres o bien los transponían
  - Cifrado por sustitución:
    - cifrado en el que cada carácter del texto en claro se sustituye por otro carácter en el texto cifrado
      - $A \rightarrow V$
      - $V \rightarrow W$
      - ...
  - Cifrado por transposición:
    - consiste en realizar una permutación de las posiciones que ocupan los símbolos en el mensaje en claro
      - HOLA  $\rightarrow$  ALHO

# Ejemplo: cifrado por sustitución César

- Consiste en una transformación única. Cada carácter de texto en claro se reemplaza por el carácter tercero a la derecha, módulo 27

$$C: M \rightarrow M + 3 \ (\text{mod. } 27)$$



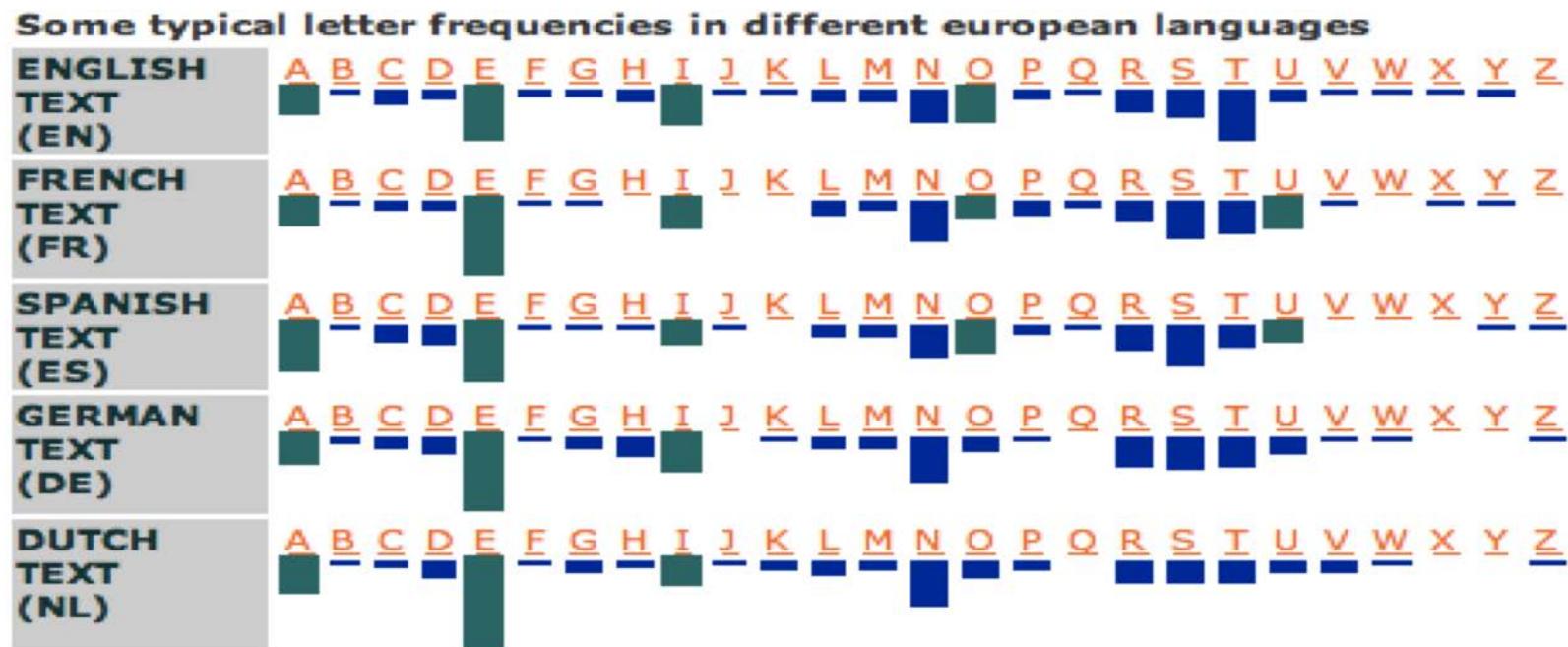
Ejemplo texto cifrado: WX WDPELHQ, EUXWR, KLMR PLR

¿Cómo sería el descifrado de este texto cifrado?

- Generalizado después a un sistema de cifrado con 27 posibles combinaciones

$$C: M \rightarrow M + i \ (\text{mod. } 27) \quad 1 \leq i \leq 27$$

- Ese algoritmo da ventaja al criptoanalista, porque la frecuencia de aparición de las letras es bien conocida. Así:

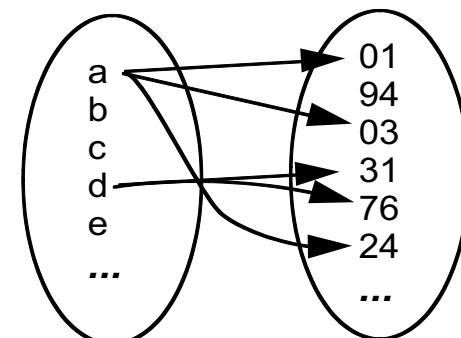


English											
E	12.4%	H	6.5%	U	2.7%	G	2.0%	K	0.7%		
T	8.9%	S	6.2%	M	2.5%	Y	2.0%	Q	0.1%		
A	8.0%	R	6.1%	W	2.3%	P	1.6%	X	0.1%		
O	7.6%	D	4.6%	C	2.2%	B	1.3%	J	0.1%		
N	7.0%	L	3.6%	F	2.2%	V	0.8%	Z	0.0%		
I	6.7%										

Spanish											
E	13.0%	S	6.9%	U	3.6%	V	1.0%	J	0.3%		
A	11.1%	T	5.3%	P	3.0%	F	0.8%	Z	0.3%		
O	9.7%	C	5.2%	M	2.9%	Y	0.7%	X	0.2%		
I	8.2%	D	4.5%	G	1.4%	H	0.6%	W	0.1%		
N	8.0%	L	3.6%	B	1.3%	Q	0.6%	K	0.0%		
R	7.7%										

## Ejemplo: cifrado por sustitución homofónico

- Se basa en la idea de asignar a un símbolo del alfabeto fuente varios del alfabeto cifrado, solventando el problema de la frecuencia de letras
- Correspondencia uno a muchos  $\Rightarrow$  al cifrar un mensaje podemos obtener varios criptogramas
- Ejemplo:



Letra	% (redondeado)	Símbolos asignados
A	8	10, 11, 23, 45, 76, 79, 87, 98
L	6	02, 15, 21, 25, 56, 60
N	3	44, 63, 71
O	8	04, 16, 28, 29, 37, 52, 69, 90
P	2	30, 88
T	2	24, 77

“PLATON” se cifra como “882110772963”

## Ejemplo: cifrado por sustitución polialfabética

- Alfabeto para posiciones impares:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
V	K	L	s	w	e	M	N	U	f	a	b	Q	r	S	t	o	j	I	P	x	s	Ñ	h	d	Z	W

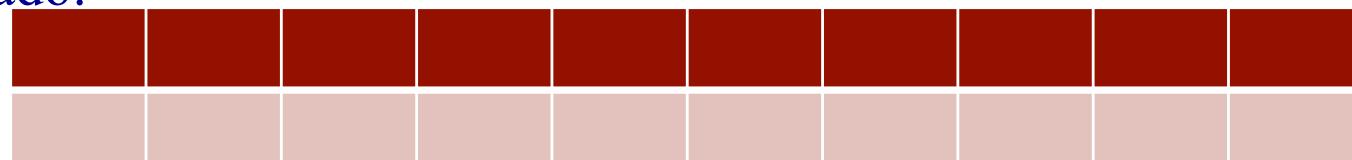
- Alfabeto para posiciones pares:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
z	g	X	Y	a	b	D	K	L	P	q	s	t	U	O	Ñ	Q	k	e	c	H	W	M	N	f	g	i

- Cifrado del texto: “*HOLA A TODOS*”

# H O L A A T O D O S

- Descifrado:



# Ejemplo: cifrado por sustitución polialfabética

- Alfabeto para posiciones impares:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
V	K	L	s	w	e	M	N	U	f	a	b	Q	r	S	t	o	j	I	P	x	s	Ñ	h	d	Z	W

- Alfabeto para posiciones pares:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
z	g	X	Y	a	b	D	K	L	P	q	s	t	U	O	Ñ	Q	k	e	c	H	W	M	N	f	g	i

- Cifrado del texto: “**HOLA A TODOS**”

H	O	L	A	A	T	O	D	O	S
N	Ñ	b	z	v	H	t	Y	t	c

- Descifrado:

N	Ñ	b	z	v	H	t	Y	t	c
H	O	L	A	A	T	O	D	O	S

## Ejemplo: cifrado por transposición

- La forma más simple de transposición: el texto en claro se escribe como secuencia de filas (con una cierta profundidad) y se lee como secuencia de columnas
- Ejemplo: “EN ANDALUCIA, EL MULHACEN Y EL VELETA,  
SON LAS MONTAÑAS MAS ALTAS”

ENANDALUCIAELMULHACENYELVE

LETASONLASMONTAÑASMASALTAS

- Mensaje cifrado:

ELNEATNADSAOLNULCAISAMEOLNMTUALÑHAASCMEANSYAELLTVAES

## Ejemplo: cifrado por transposición con clave

- Se podría complicar el procedimiento anterior estableciendo una restricción en el número de columnas cuyo valor va a depender del tamaño que tenga una **clave**
- Ejemplo:
  - Texto en claro: “**HOLA A TODOS, QUE TENGÁIS UN BUEN DÍA**”
  - Clave: ”**SECRETO**” con un tamaño de 7

S	E	C	R	E	T	O
H	O	L	A	A	T	O
D	O	S	Q	U	E	T
E	N	G	A	I	S	U
N	B	U	N	D	I	A

- Para el cifrado se puede poner la condición siguiente: se va a ir cogiendo las letras de aquellas columnas por orden alfabético del secreto, es decir: **C, E, E, O, R, S, T**, resultando en: “ \_\_\_\_\_ ”

## Ejemplo: cifrado por transposición con clave

- Se podría complicar el procedimiento anterior estableciendo una restricción en el número de columnas cuyo valor va a depender del tamaño que tenga una **clave**
- Ejemplo:
  - Texto en claro: “**HOLA A TODOS, QUE TENGÁIS UN BUEN DÍA**”
  - Clave: ”**SECRETO**” con un tamaño de 7

S	E	C	R	E	T	O
H	O	L	A	A	T	O
D	O	S	Q	U	E	T
E	N	G	A	I	S	U
N	B	U	N	D	I	A

- Para el cifrado se puede poner la condición siguiente: se va a ir cogiendo las letras de aquellas columnas por orden alfabético del secreto, es decir: **C, E, E, O, R, S, T**, resultando en: “**LSGUOONBAUIDOTUAQANHDENTESI**”

## Ejemplo: cifrado por transposición Railfence

- El cifrado consiste
  - en escribir diagonalmente el texto en claro con una profundidad  $P$  específica
  - el criptograma se escribe leyendo las filas
- Ejemplo:  $M = \text{"Hola a todos"}$ , con una profundidad de  $P=4$ , entonces el criptograma es: **Hoot d laoas**  
por simplemente computar:

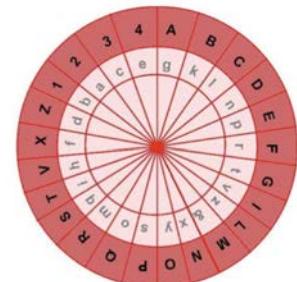
H			O	
O		T	D	
L	A		O	
A			S	

# Ejemplo: métodos polialfabéticos y nomenclátores

- Para complicar el proceso de cifrado, se puede hacer uso del disco de Alberti junto con nomenclátores, los cuales consisten en asociar a determinados palabras, códigos específicos

Felipe II	123
Rey	124
Walshingan	122

- Se desea descifrar el siguiente texto: “*baa&hpmiyvsvoylrlxckngkl*”
- Uso del disco:
- Cada diez letras descifradas, se ha de girar el disco externo (de las mayúsculas) dos posiciones en el sentido de las agujas del reloj
- En el disco de Alberti, la **u** se identifica con la **v** al cifrar. Al descifrar, por el sentido de la frase, se puede conocer si se ha de escribir una u otra letra

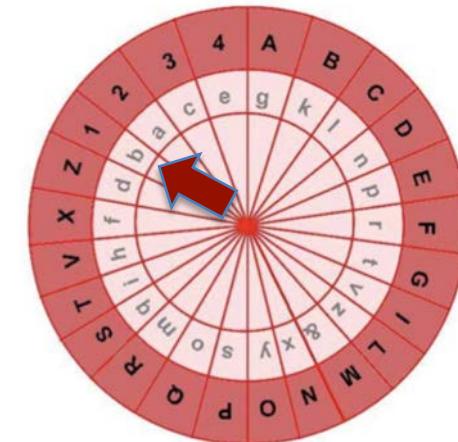


# Ejemplo: métodos polialfabéticos y nomenclátores

- Funcionamiento para cifrar:
  - Posicionar los disco en el estado inicial

b	a	a	&	H	p	m	i	Y	v
1	2								

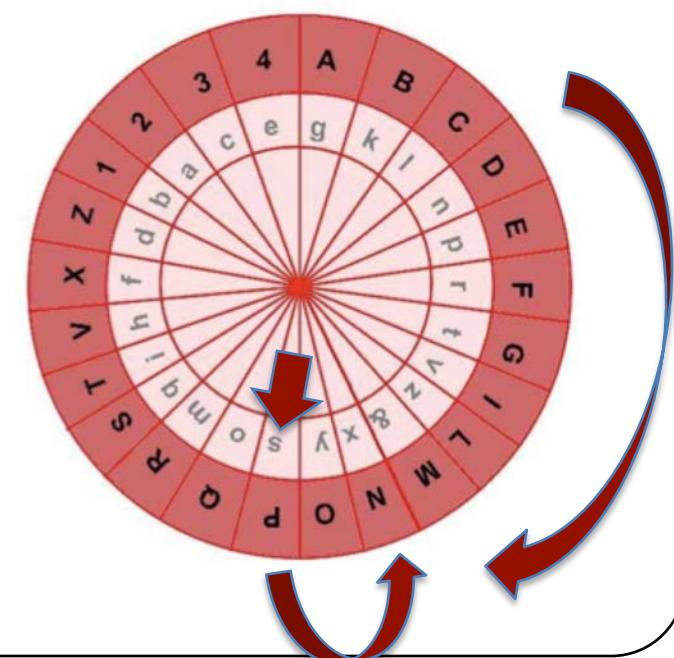
**"baa&hpmiyvsvoiylrlxckngkl"**



- Con el disco externo girar 2 posiciones en el sentido de las agujas del reloj (sólo en cada diez letras descifrada):

s	v	o	l	y	l	r	l	x	c
N	F								

**"baa&hpmiyvsvoiylrlxckngkl"**



# Ejemplo: métodos polialfabéticos y nomenclátores

- Funcionamiento:
  - Con el disco externo volver a girar 2 posiciones en el sentido de las agujas del reloj:

k	n	g	k	L
2	4	1	2	3

***"baa&hpmiyvsvoiylrlxckngkl"***

- Por consiguiente, el texto en claro es:

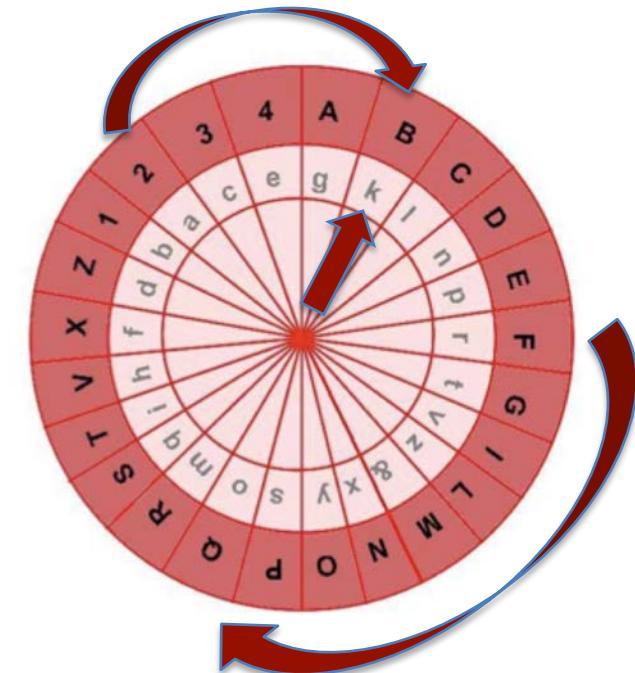
b	a	a	&	H	p	m	i	Y	v
1	2	2	M	V	E	R	T	O	I
s	v	o	I	Y	I	r	I	X	c
N	F	O	R	M	A	D	A	L	1
k	n	g	k	L					
2	4	1	2	3					

**"1 2 2 M V E R T O I N F O R M A D A L 1 2 4 1 2 3"**

- Si, además, añadimos los nomenclátores + la restricción de la V → U:

Felipe II	123
Rey	124
Walshingan	122

**"WALSHINGAN MUERTO INFORMAD AL REY FELIPE II"**



## Cifrado Producto

- Combina sustitución y transposición
- Se pueden considerar como la aplicación sucesiva de varios cifrados  $E_i$

$$E = E_1 \cdot E_2 \cdot \cdots \cdot E_r$$

$$E(M) = E_1(E_2(\cdots(E_r(M))))$$

- La composición de funciones de descifrado  $D_i$  se realiza en orden inverso

$$D = D_r \cdot D_{r-1} \cdots D_1$$

$$M = D(C) = D_r(D_{r-1}(\dots(D_1(C))))$$

- Es un esquema utilizado para obtener un alto grado de seguridad con sistemas relativamente sencillos aplicados reiterativamente
- Dan lugar a sistemas de cifrado complejos, seguros, difíciles de atacar, fácilmente trasladables a un ordenador

## Cifrado Vernam

- Variante del cifrado llamado one-time pad (OTP)
- Un one-time pad es un *conjunto infinito y no repetitivo* de letras aleatorias
- Cada letra del pad se usa para cifrar una única letra del texto en claro, en módulo  $n$  (longitud del alfabeto)



One-time pad booklet and microdot reader, concealed in a toy truck and used by an illegal agent that operated in Canada. © Canadian Security Intelligence Service

Texto : T H I S   I S   S E C R E T  
OTP: X V H E   U W   N O P G C Z

-----  
Cifrado : Q C P W   C O   F S R X H S

- Otro dos ejemplos:

- Aquí se observan grupos de tres filas, que se corresponden con texto en claro (en decimal), clave y criptograma

0 3 3 1 6	8 7 6 7	0 8 7 6 2	6 3 1 2 3	7 6 4 8 7	0 2 0 7	
1 1 8 6 4	6 8 6 3 2	4 6 0 5 1	8 7 9 3 1	3 8 2 7 2	0 3 0 2 3	6 7 0 6 3
6 9 1 4 0	1 0 3 9 9	4 4 7 1 3	4 0 0 1 4	4 4 6 7 7	0 9 2 8 0	0 1 7 7 6
2 3 7 7 7	6 8 2 7 9	6 5 8 6 7	0 6 7 0 9	6 8 3 9 5	7 4 5 8 8	7 2 3 9 7
6 2 7 7 3	4 1 1 6 9	4 2 1 5 7	4 7 4 3 5	4 2 1 3 3	7 1 3 7 0	4 6 5 1 1
8 5 6 8 0	0 9 3 3 8	0 7 7 1 4	4 5 8 5 4	1 0 4 2 2	7 7 7 8	7 7 2 3
6 3 0 2 5	8 7 0 8 9	5 8 6 7 2	7 1 5 7 1	7 2 8 4 3	9 3 7 0 7	4 9 8 7 6
4 8 7 7 4	0 7 0 9 8	4 9 1 2 6	6 0 0 9 8	6 2 9 6 2	4 8 6 5 6	8 7 9 6
6 1 9 8 9	8 4 8 6 9	7 6 9 9 7	5 1 5 1 4	3 4 3 2 2	7 1 3 7 5	2 8 7 8 6
3 1 7 2 6	5 0 8 3 1	8 2 0 5 8	2 8 7 2 7	6 6 6 2 6	3 1 8 3 3	7 1 1 1 1
4 4 7 4 0	1 9 4 7 1	7 8 2 1 3	7 6 6 3 9	2 1 8 3 0	4 2 3 9 0	6 2 3 3 0
1 6 2 7 6	6 9 2 0 4	5 0 2 9 1	9 4 3 1 1	5 6 9 5 6	7 3 3 7 3	3 5 7 4 1
7 7 7 7 7	2 8 3 6 6	5 8 1 7 6	4 6 7 0 2	7 7 6 1 3	0 5 8 6 7	6 3 2 3 5
1 2 6 4 4	3 5 6 0 1	7 4 5 0 8	5 2 0 6 0	5 7 7 2 1	5 2 5 0 9	7 8 6 4 3
1 9 9 2 1	5 3 5 6 7	4 2 4 7 4	9 8 7 2 0	4 4 4 8 4	5 7 3 6 1	3 1 8 7 1
2 0 7 7 3	7 8 2 0 8	7 6 9 2 6	3 8 3 9 6	3 2 6 7 6	0 3 7 4 6	4 1 6 6 3
6 7 8 1 8	0 0 6 2 1	0 7 4 0 8	7 8 5 7 3	6 7 2 3 0	6 7 8 0 8	8 1 7 7 2
8 0 0 0 1	7 8 8 2 9	7 3 3 2 0	0 3 8 8 1	9 9 8 0 6	6 0 7 4 9	2 3 1 7 5
1 9 4 3 9	7 6 9 5 6	9 8 7 6 7	2 6 7 9 6	3 9 3 7 7	7 3 9 8 7	6 2 9 4 6
2 2 7 7 2	3 0 6 4 2	3 8 0 9 1	4 0 1 6 9	4 8 4 3 3	4 6 8 2 5	7 3 1 7 1
3 1 2 2 1	0 6 3 1 0	2 6 7 5 8	6 7 8 9 5	9 7 7 1 0	3 9 7 0 2	3 5 0 6 7
5 2 7 2 8	7 3 3 3 3	0 0 0 7 7	1 5 8 3 2	6 5 8 5 0	6 5 8 7 1	8 8 7 2 8
0 0 6 3 9	2 5 0 6 1	3 2 2 4 7	8 0 0 1 1	8 8 7 2 3	3 2 3 7 1	2 2 7 9 1
2 1 5 4 0 2	9 8 3 3 2	3 2 2 1 4	9 3 2 9 3	7 7 3 3	7 7 7 5 3	0 0 5 1 3

Fuente: <http://www.caslab.cl/che.php>

B A R R O Y C A Ñ A B R A V A	MClá
1 0 1 8 1 8 1 5 2 5 2 0 1 4 0 1 1 8 0 2 2 0	Clave (tan larga como el mensaje)
E D S A S A C E T N I E V E D	MClá + Clave
4 3 1 9 0 1 9 0 2 4 2 0 1 3 8 4 2 2 4 3	Criptograma
5 3 1 0 1 8 7 2 5 4 4 7 1 3 9 2 2 2 2 2 6 3	
F D K R H Y E E H N J V V Z D	

Fuente: <http://bit.ly/2cqBu8D>

Cifrado: (carácter del texto en claro + key ) + mod 27

Descifrado: (carácter del criptograma - key ) + mod 27

- En los ordenadores, el OTP aleatorio de longitud infinita se combina mediante XOR con el texto en claro. Ejemplo:

Texto en claro	1	1	0	0	1	0	1	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	$\oplus$
OTP	1	0	0	1	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	=		
Criptograma	0	1	0	1	0	0	0	1	1	0	1	0	1	1	1	0	1	0	1	0	0	$\oplus$	
OTP	1	0	0	1	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	=		
Texto en claro	1	1	0	0	1	0	1	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	

- Inconvenientes del cifrado Vernam:
  - las letras del OTP (o bits si se usa en ordenador) han de generarse aleatoriamente
  - el OTP no se vuelve a usar

## Relación de ejercicios

1. Considerando el alfabeto común (incluyendo la ñ en el alfabeto) y un desplazamiento de 3 posiciones para el proceso de cifrado o descifrado, aplicar la técnica de sustitución Caesar para cifrar el siguiente texto:

“EL PATIO DE MI CASA ES PARTICULAR”

Nota: se empieza a contar desde la posición 1 (A del alfabeto)

**SOLUCIÓN:** HÑ SDWLR GH OL FDVD HV SDUWLFXÑDU

## Relación de ejercicios

2. Dado el criptograma  $C = \text{"FMIRZIRMHS E PE EW$   
 $MKREXYVE HI WIKYVMHEH HI PE MRJSVQEGMKR"}$  descifrar el contenido del mismo, sabiendo, además, que hay que usar la técnica de sustitución Caesar con un desplazamiento de 4 posiciones modulo  $n=26$

Nota: se empieza a contar desde la posición 1 (A del alfabeto)

**SOLUCIÓN:** BIENVENIDO A LA ASIGNATURA DE SEGURIDAD  
DE LA INFORMACIÓN

## Relación de ejercicios

3. El siguiente algoritmo aplicará una sustitución monoalfabética, pero esta vez teniendo en cuenta la siguiente regla:  $C_i = M_i + K_i \text{ mod } 26$  donde  $K$  representa una clave de longitud  $L$ . El objetivo es cifrar el texto original usando el alfabeto inglés

¿Cuál sería el criptograma del mensaje  $M = \text{"HOLA AMIGOS"}$  usando una clave  $K = \text{CIFRA}$ ?

Nota: se empieza a contar desde la posición 1 (A del alfabeto)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
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

H	O	L	A	A	M	I	G	O	S
8	16	12	1	1	13	9	7	15	19
C	I	F	R	A	C	I	F	R	A
+3	+8	+6	+18	+1	+3	+9	+6	+18	+1
K	X	R	S	B	P	R	M	G	T
11	24	17	19	2	16	18	13	32→7	20

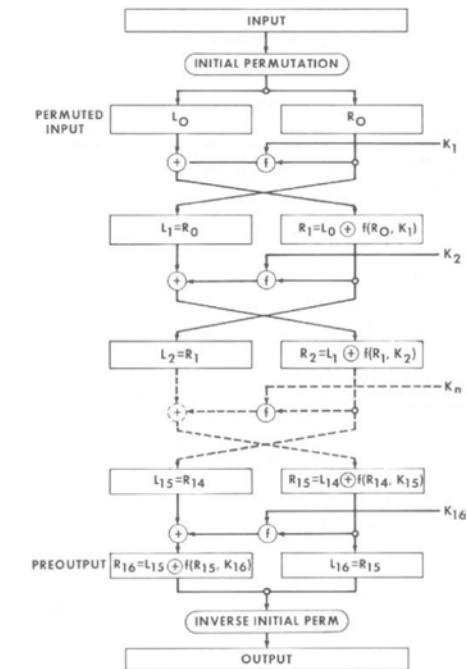
**SOLUCIÓN:** KXRS BPRMGT

## Relación de ejercicios

4. Mediante la técnica Railfence, determinar el criptograma correspondiente al mensaje “*El perro de San Roque no tiene rabo porque Ramón Ramírez se lo ha robado*” con una profundidad P=7 (alfabeto inglés)

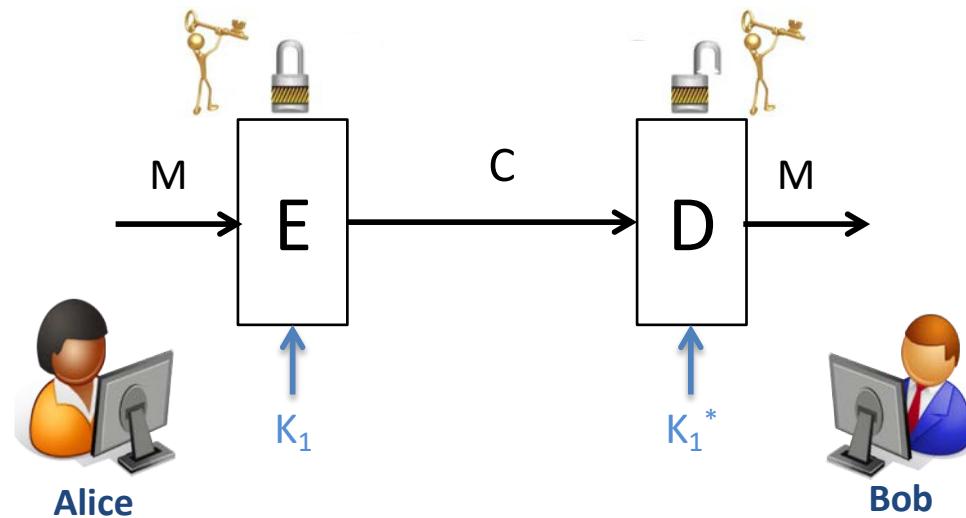
**SOLUCIÓN:** ER rmhln oe aan oapaq nb RRIre Sueo eaeore eipóu msbírdn to qr za ooored

# Algoritmos simétricos



## Recordatorio...

- 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

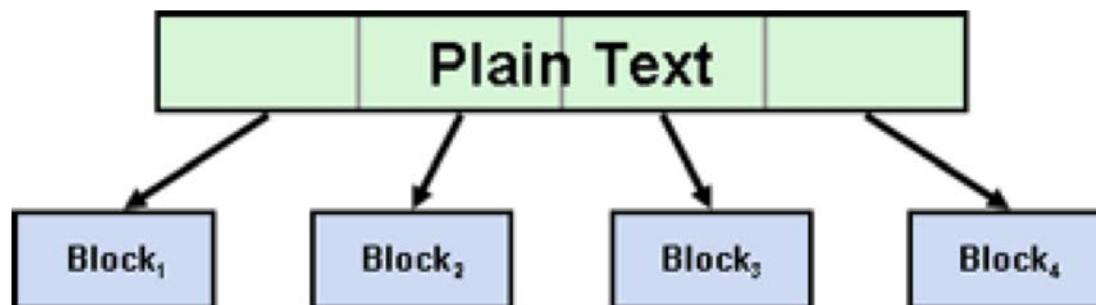


- A partir de la expresión

$$E(M) = C$$

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

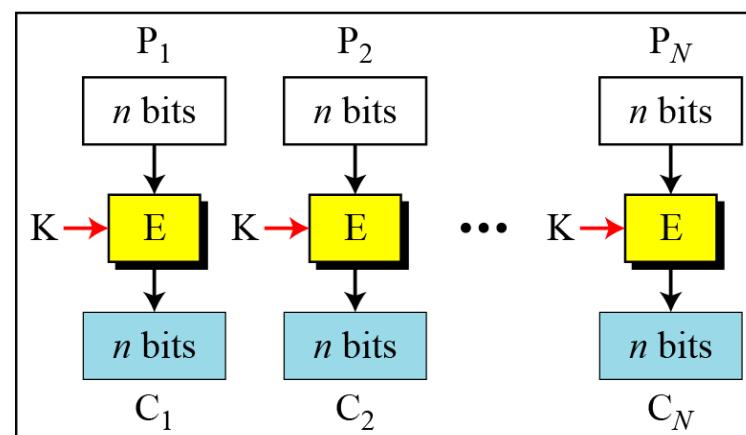
E: Encryption

$P_i$ : Plaintext block  $i$

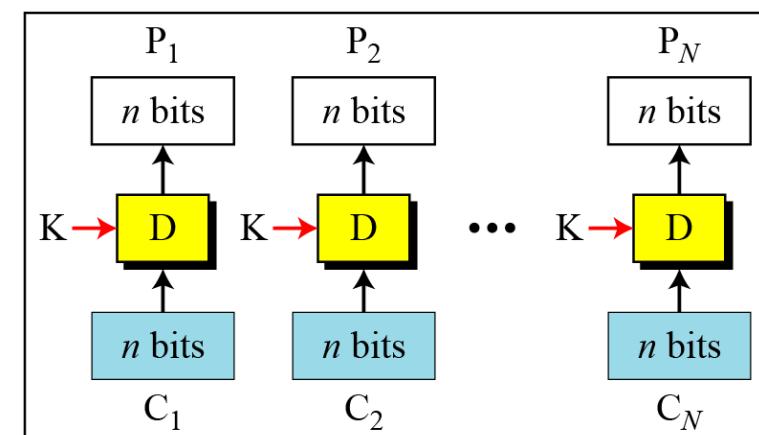
K: Secret key

D: Decryption

$C_i$ : Ciphertext block  $i$

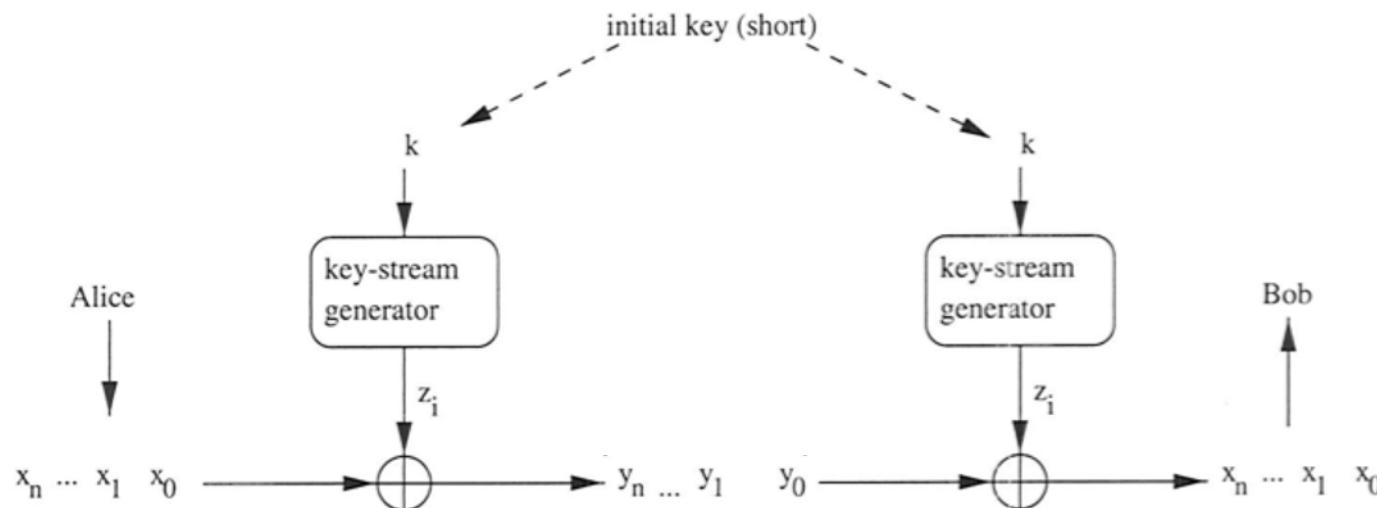


Encryption



Decryption

- 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  $K$

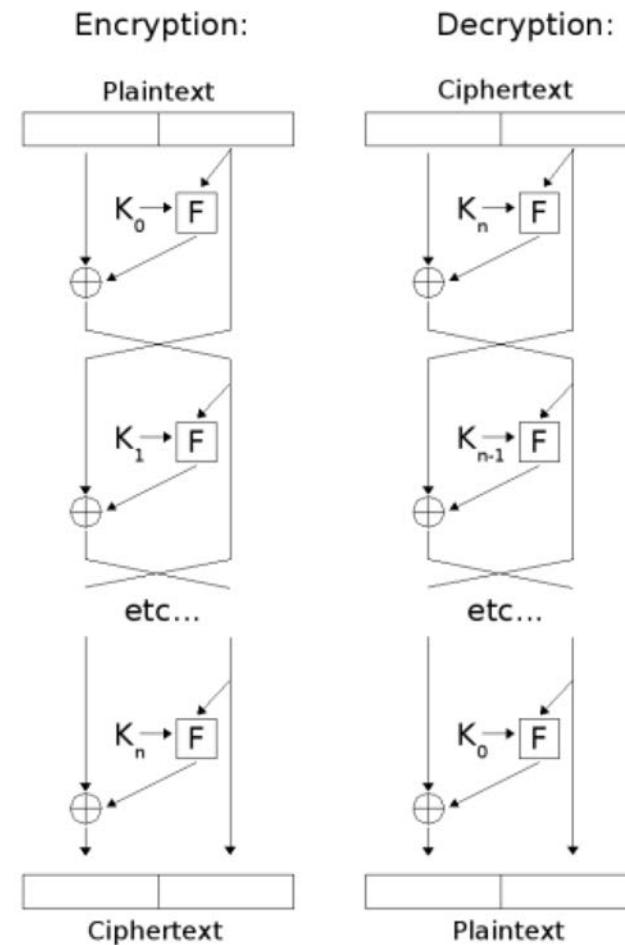


## Algoritmo DES (Data Encryption Standard)

- *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 almacenamientopor parte de el Gobierno americano, las empresas privadas y, en general, de cualquier tipo de usuario

# Algoritmo DES

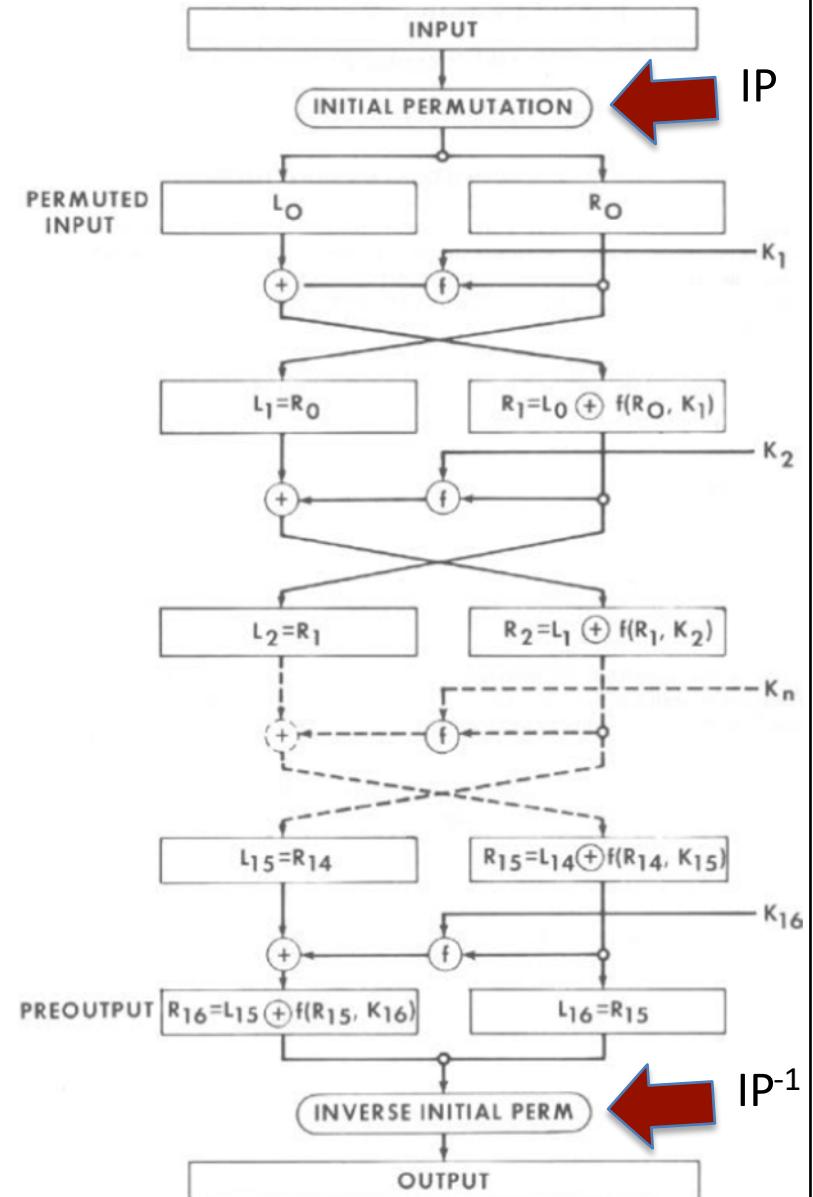
- Para el desarrollo del *DES*, IBM partió de *Lucifer*, un algoritmo propio desarrollado con anterioridad y usado principalmente en entornos bancarios
- Lucifer se basaba en el uso de una técnica denominada **red de Feistel**, y usaba una longitud de clave de 128 bits



Red de Feistel

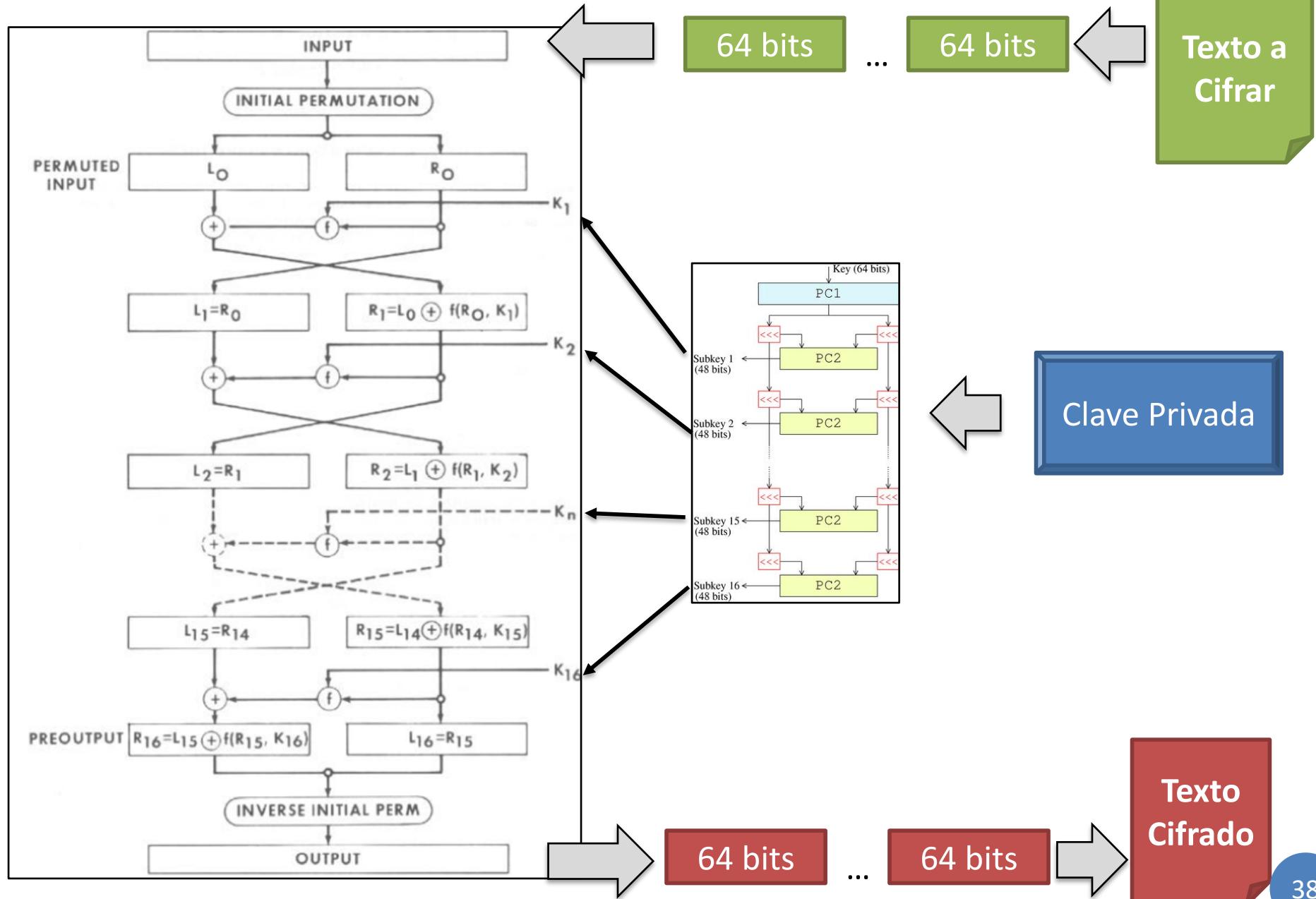
# Algoritmo DES

- 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



# 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

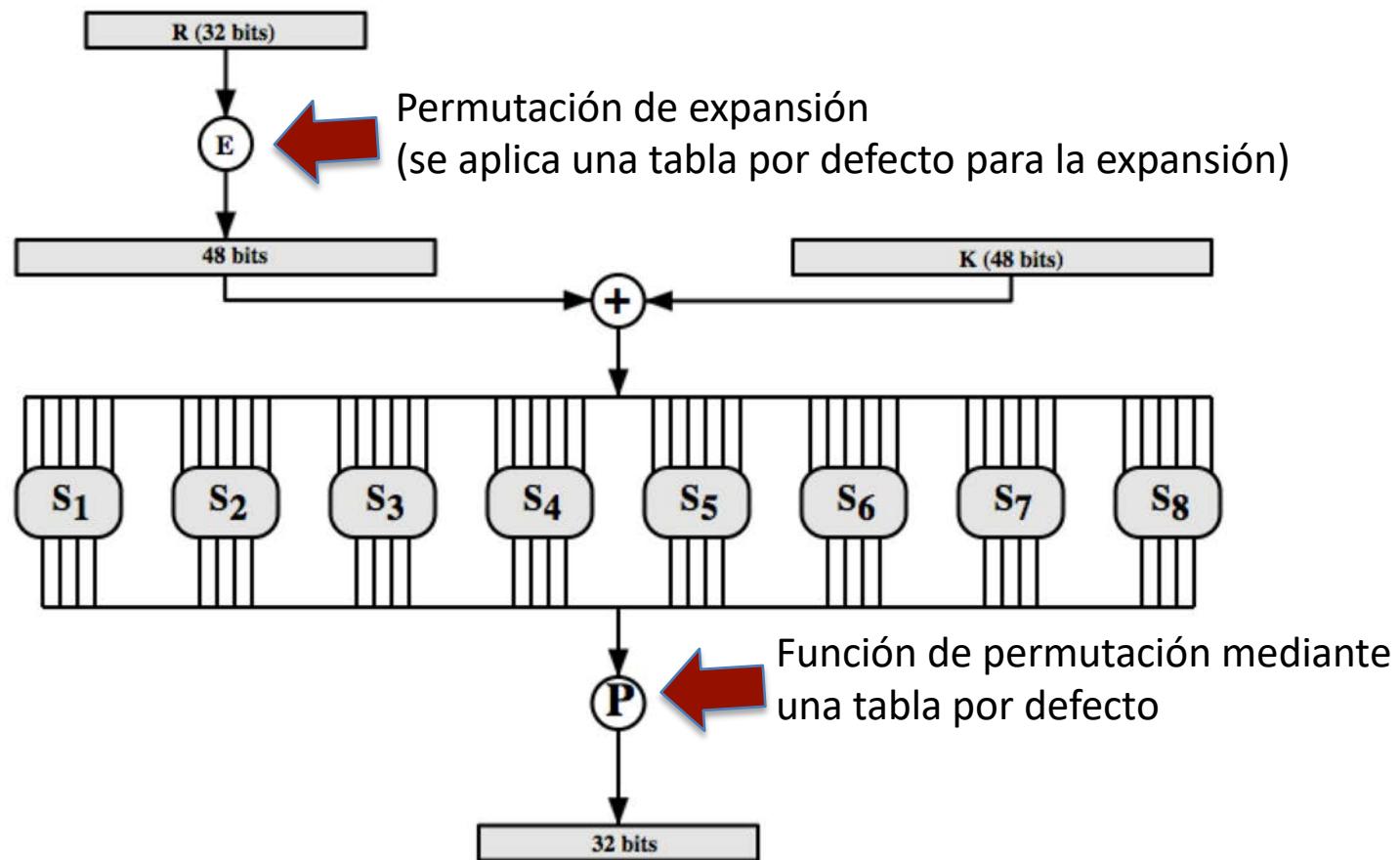
Las tablas están conteniendo valores enumerados entre 1 y 64, y la entrada a cada tabla indica la posición de un bit de entrada enumerada en la salida, siendo la matriz de entrada, y general:

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

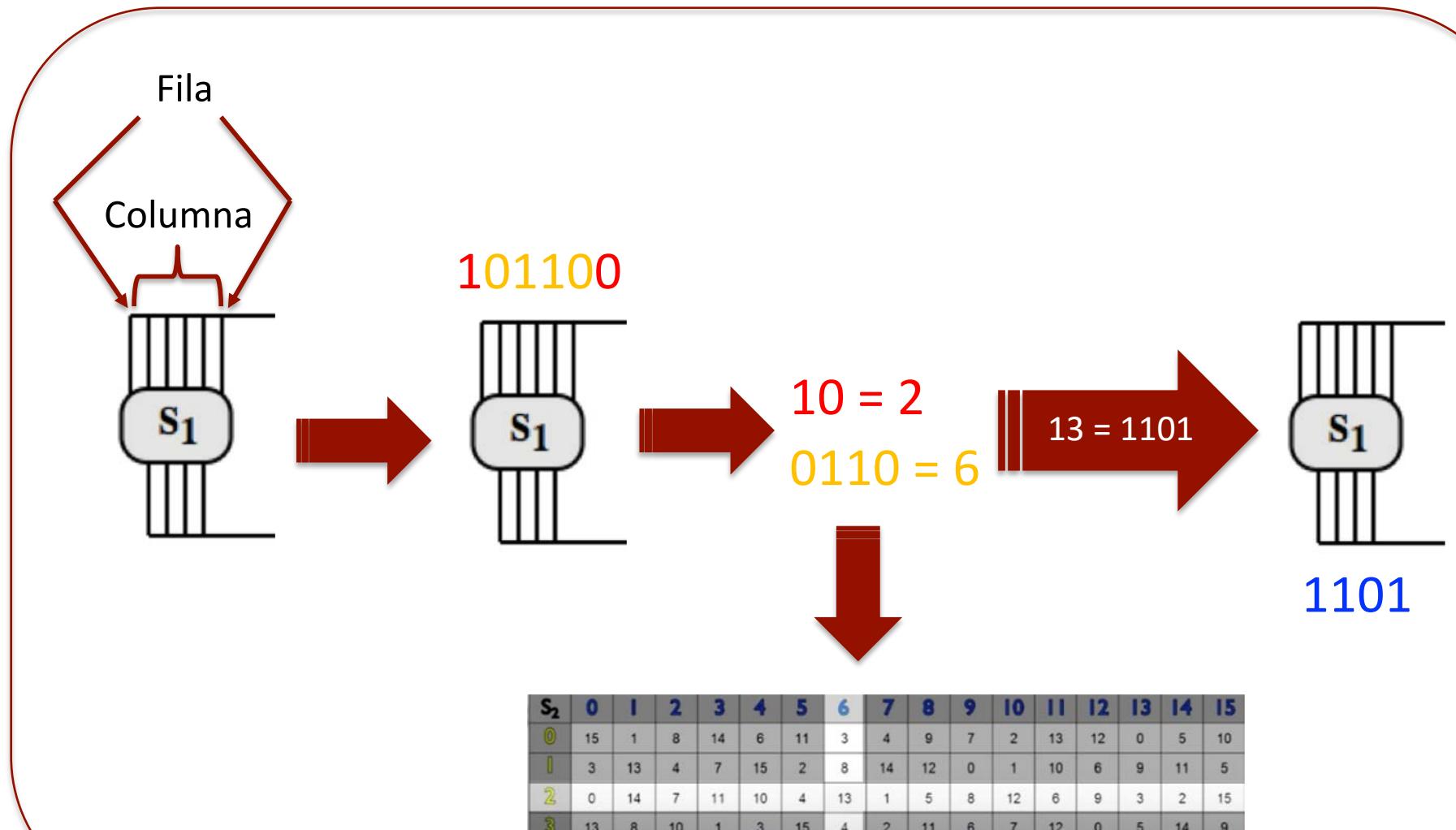
8x8 = 64 bits

# 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



# 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 cuenta 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

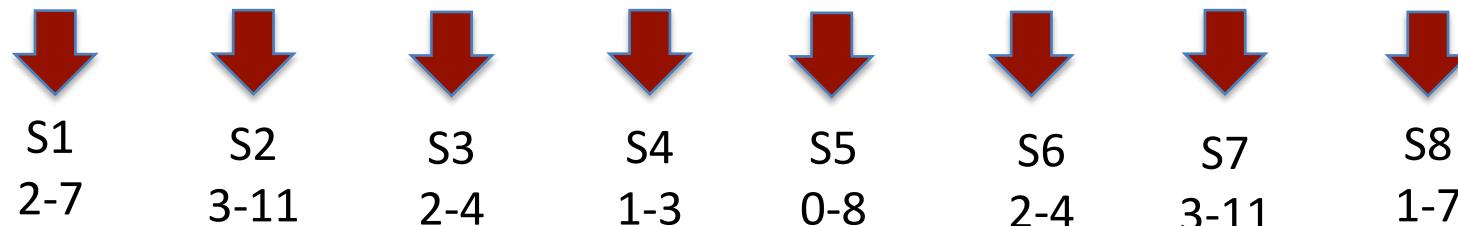
Fila	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

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																
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

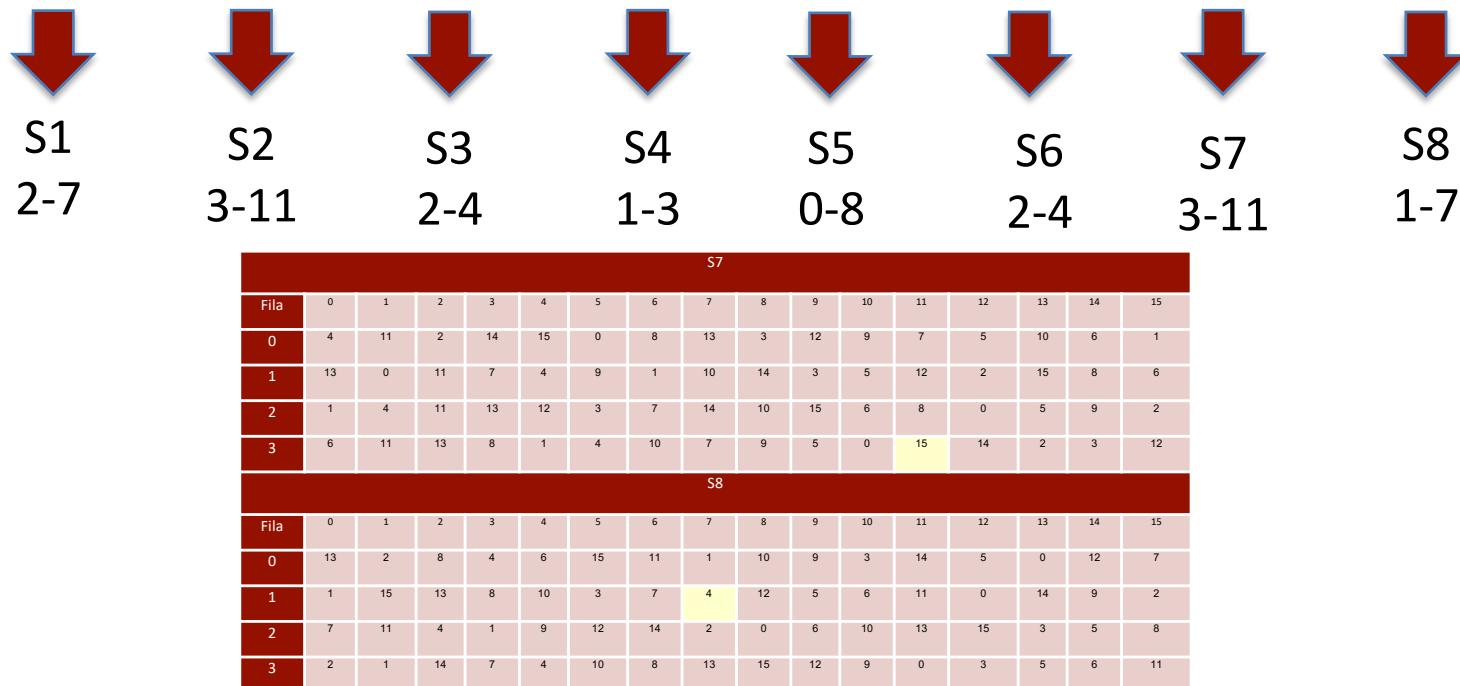
S5																
Fila	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

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

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



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

# Algoritmo DES - subclaves

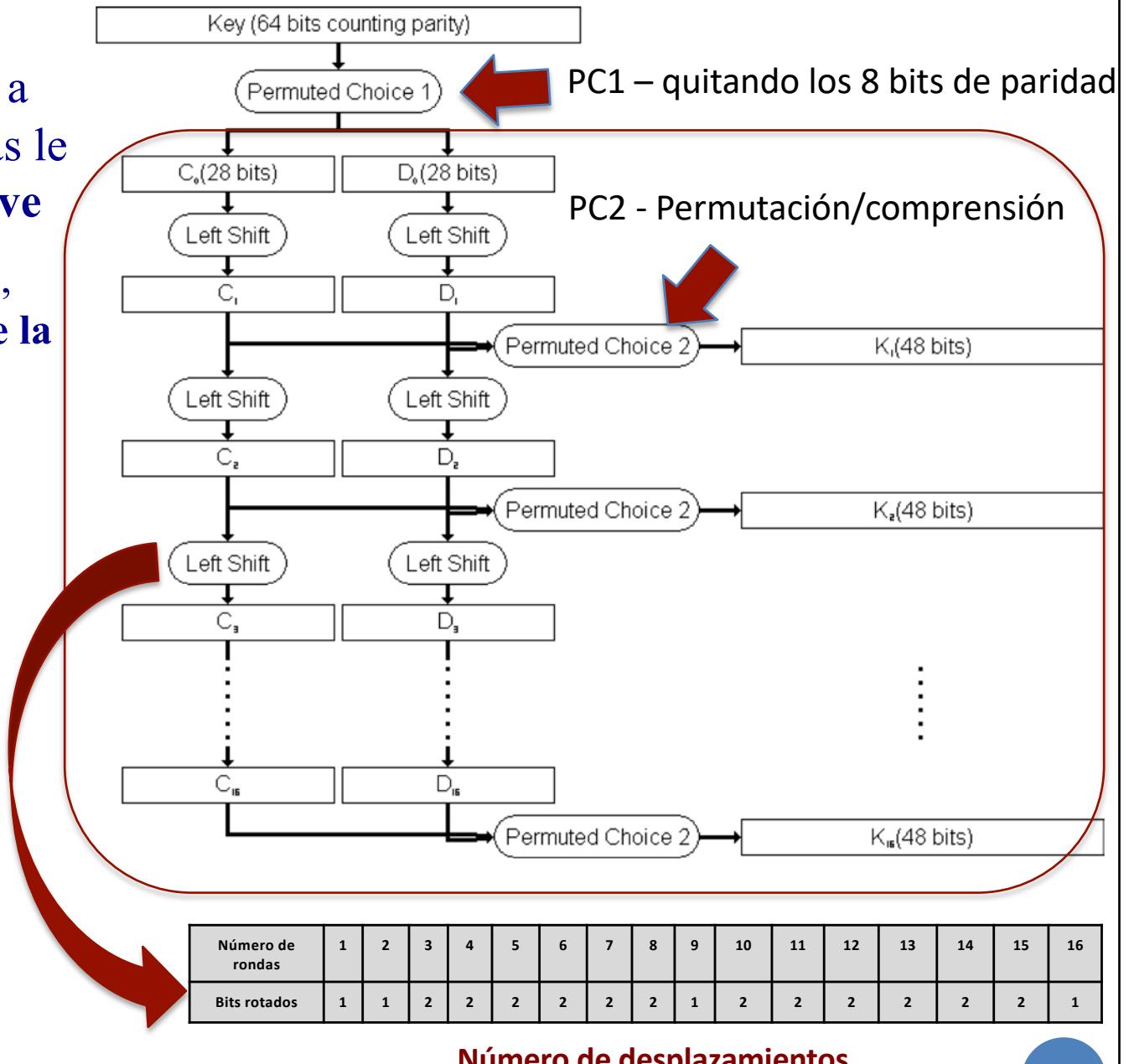
- Se observa también que a cada una de las 16 etapas le corresponde una **subclave**
  - En total, **16 subclaves**, generadas a partir de la clave inicial  $K$ , como muestra la figura

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

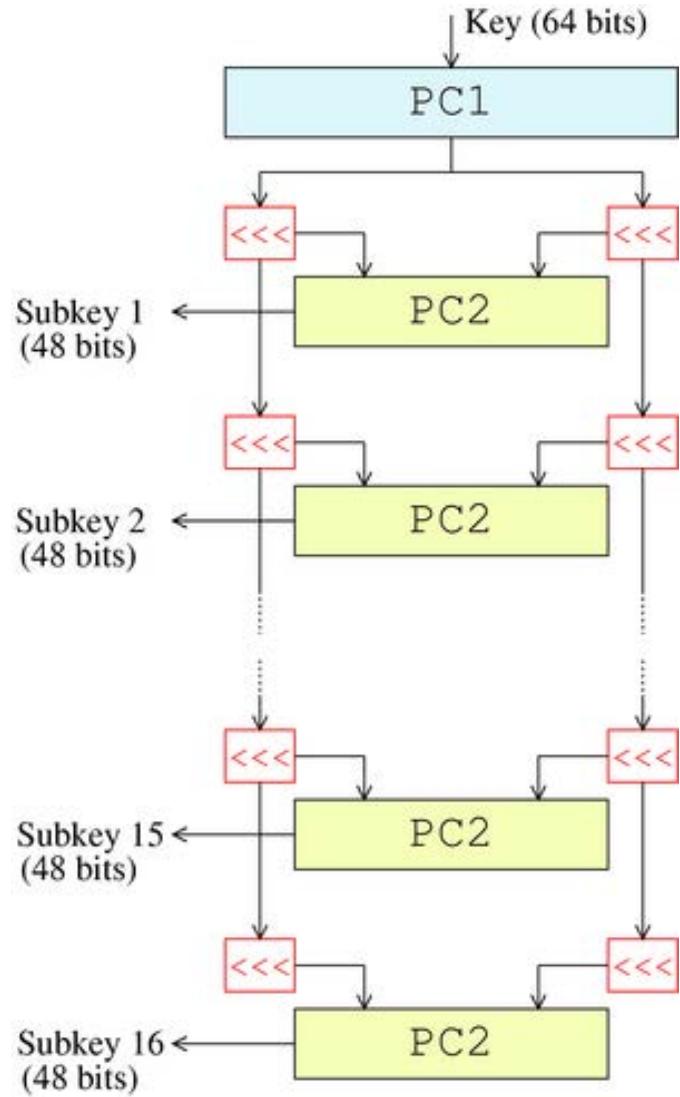
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

PC2



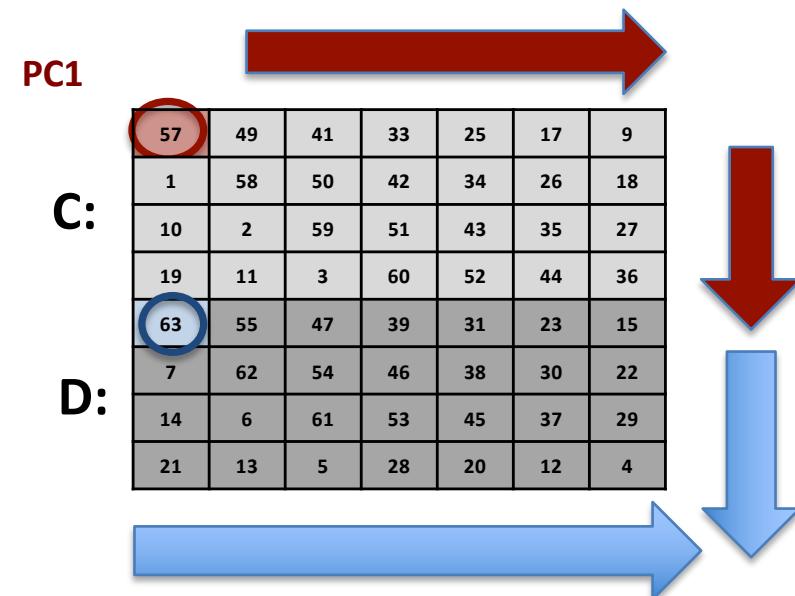
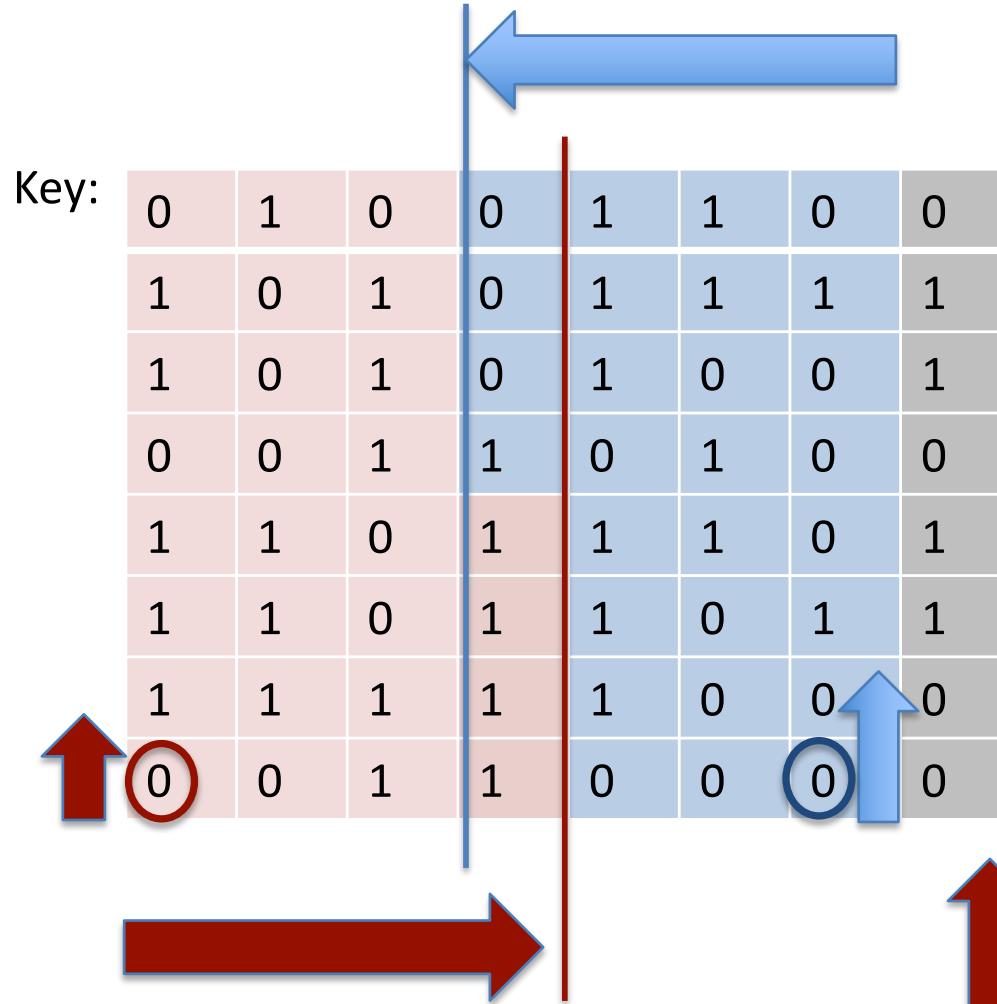
## Algoritmo DES - subclaves

- De forma simplificada, generación subclaves  $K_i$ :



# Algoritmo DES - subclaves

- PC-1

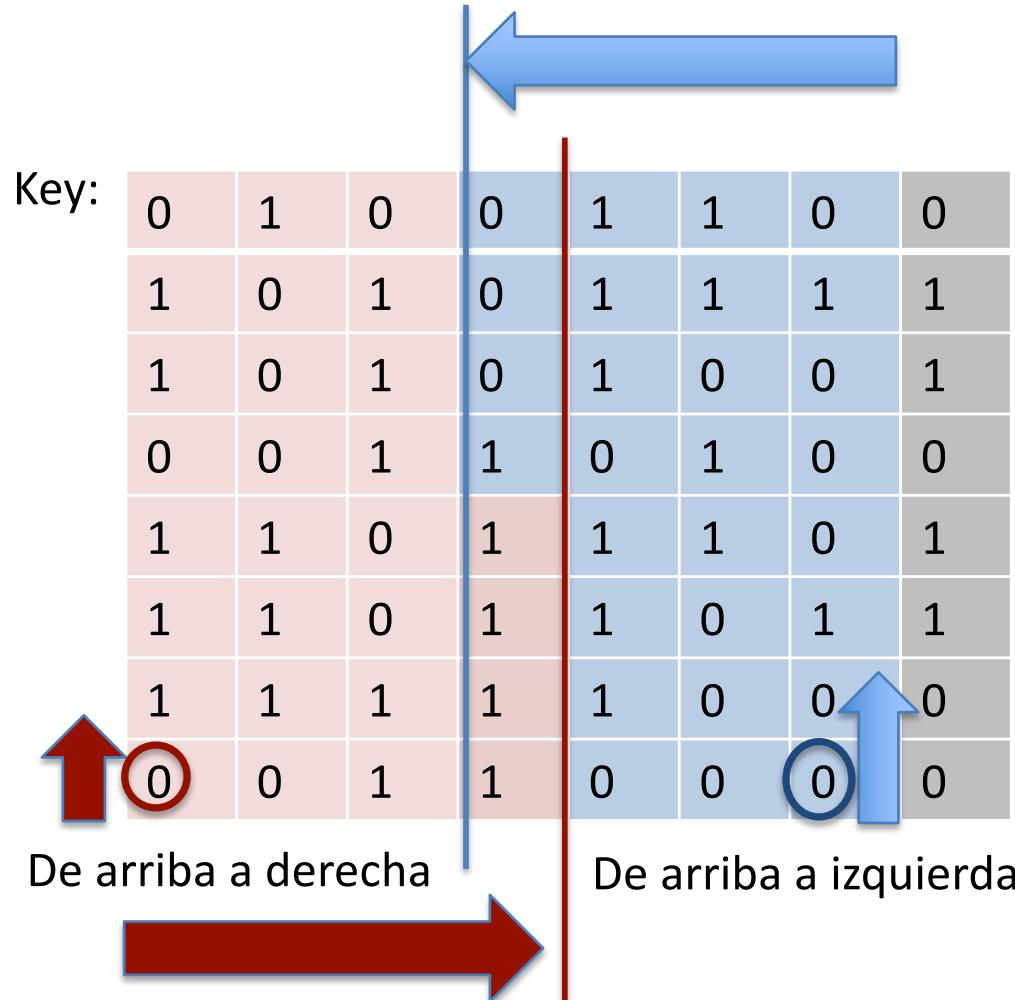


1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

8x8 = 64 bits

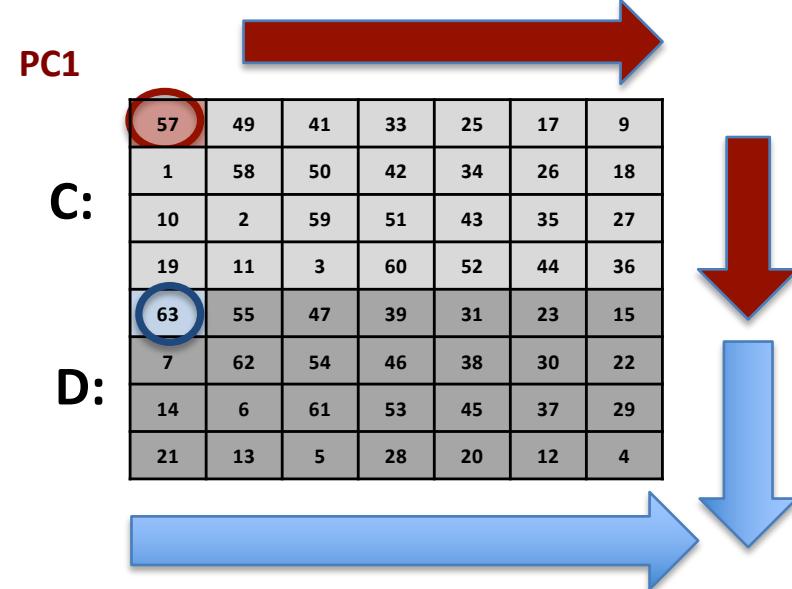
# Algoritmo DES - subclaves

- PC-1



$$C_i = 01110111 00111000 01110011 11011111$$

$$D_i = 00100011 00001101 11011110 11110000$$



El resultado de permutar las posiciones  
siguiendo las posiciones de la matriz:

C:

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

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

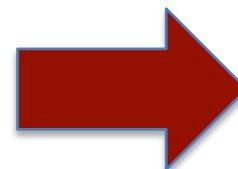
## Algoritmo DES - subclaves

- Desplazamiento a la izquierda. Si  $i=1$ , entonces la rotación es de 1 bit:
  - $C_1 = 1110111 0011100 0111001 11011110$
  - $D_1 = 0100011 0000110 1101110 11110000$
- PC-2:  $C_1 D_1 = 1110110 0111000 1110011 1011110 0100010 0001101 1011101 1110000$

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

$K_1 = 01111110 11111000 10101101 \dots$   
hasta tener los 48 bits

$8 \times 7 = 56$  bits

## Algoritmo DES

- La siguiente animación repasa los conceptos básicos de DES y muestra sus funcionalidades:
  - <http://kathrynnneugent.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  $D_i$ :
  - $C_1 = 111000011001100101010101111\ D_1 = 1010101011001100111100011110$
  - $C_2 = 110000110011001010101011111\ D_2 = 0101010110011001111000111101$
  - ....
- 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  $K_2, k_7$ ?
  - $K_2 = 011110\ 011010\ 111011\ 011001\ 110110\ 111100\ 100111\ 100101$
  - $K_7 = \mathbf{111011\ 001000\ 010010\ 110111\ 111101\ 100001\ 100010\ 111100}$

## 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

Key: 22234512987ABB23

Ciphertext: 4789FD476E82A5F1

Plaintext: 0000000000000001

Key: 22234512987ABB23

Ciphertext: 0A4ED5C15A63FEA3

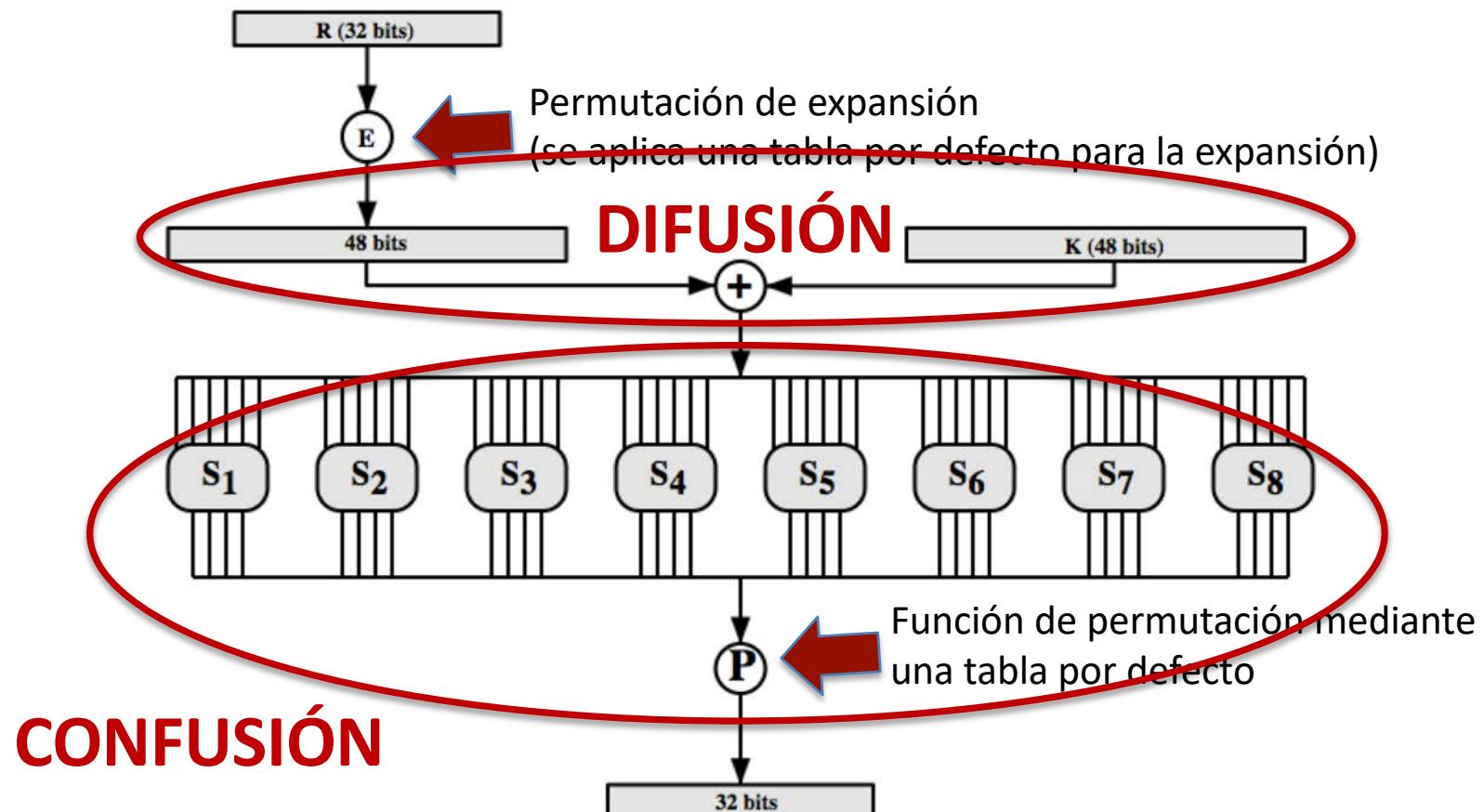
## Efecto avalancha

- 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 (=  $7.2 \times 10^{16}$ )



# Algoritmo DES

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

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

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

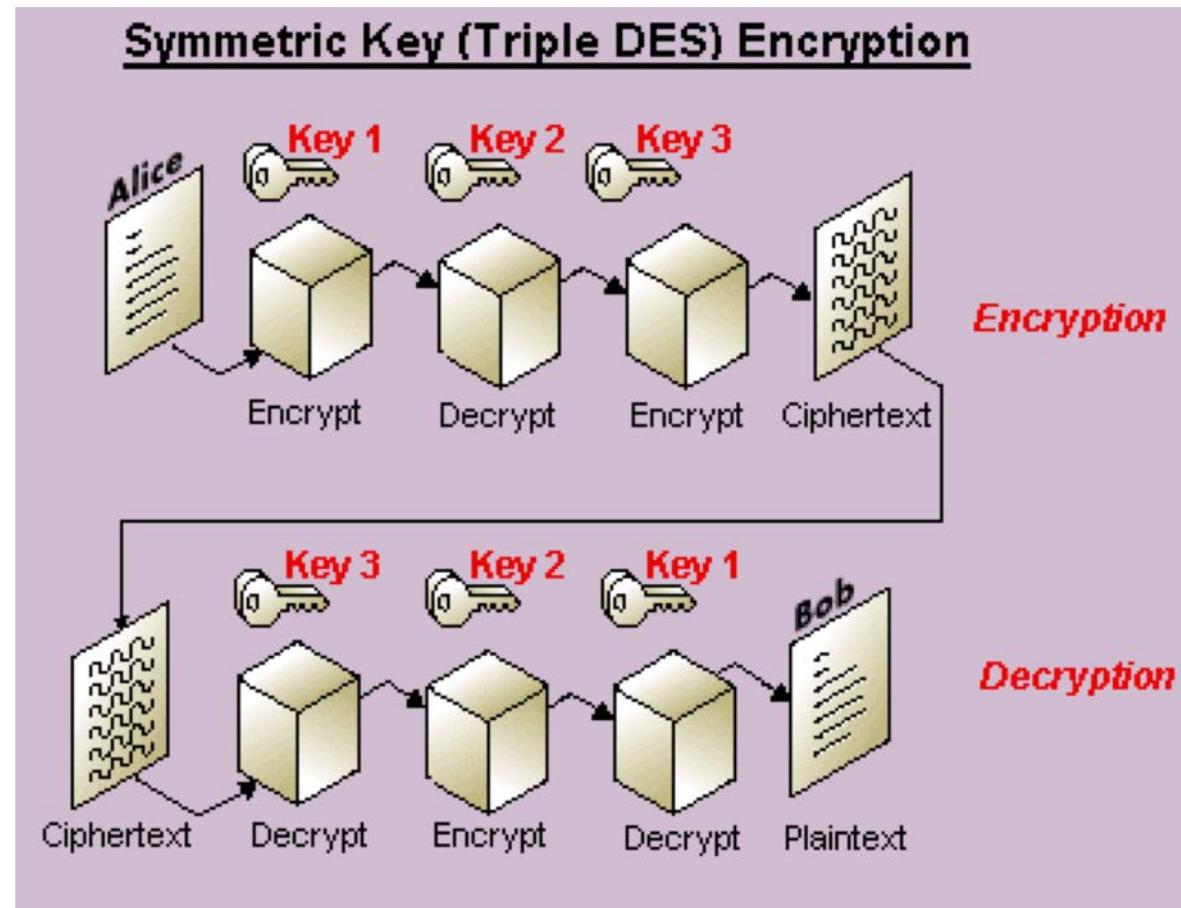


*DES Key Search Machine*  
6 estaciones SUN-2, 27 placas de circuitos, 1800 chips customizados,  
24 unidades de búsqueda por cada chip

# Algoritmo Triple-DES

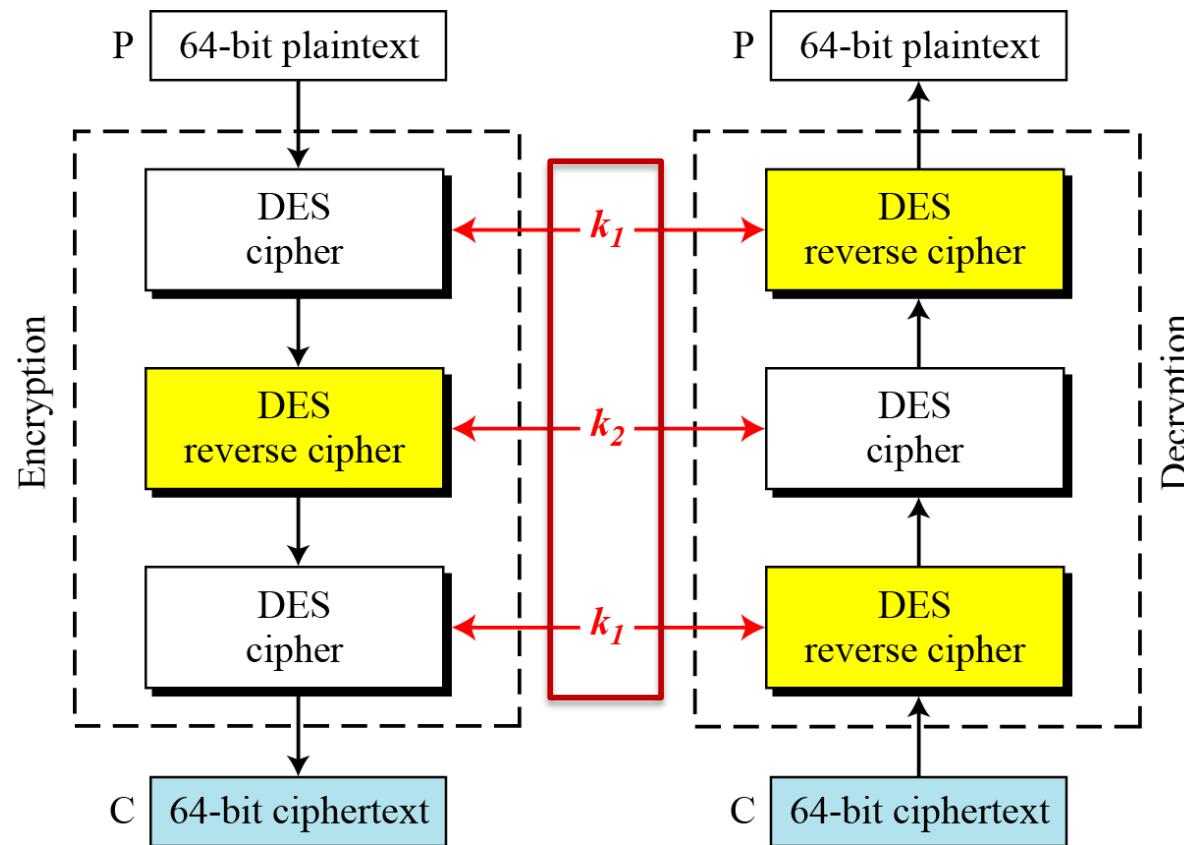
- Para aprovechar las ventajas del *DES*, y a la vez contrarrestar su escasa longitud de clave, los organismos de estandarización han adoptado el criptosistema ***Triple-DES*** (ó *3DES*)

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



## Algoritmo Triple-DES (variante de 2 keys)

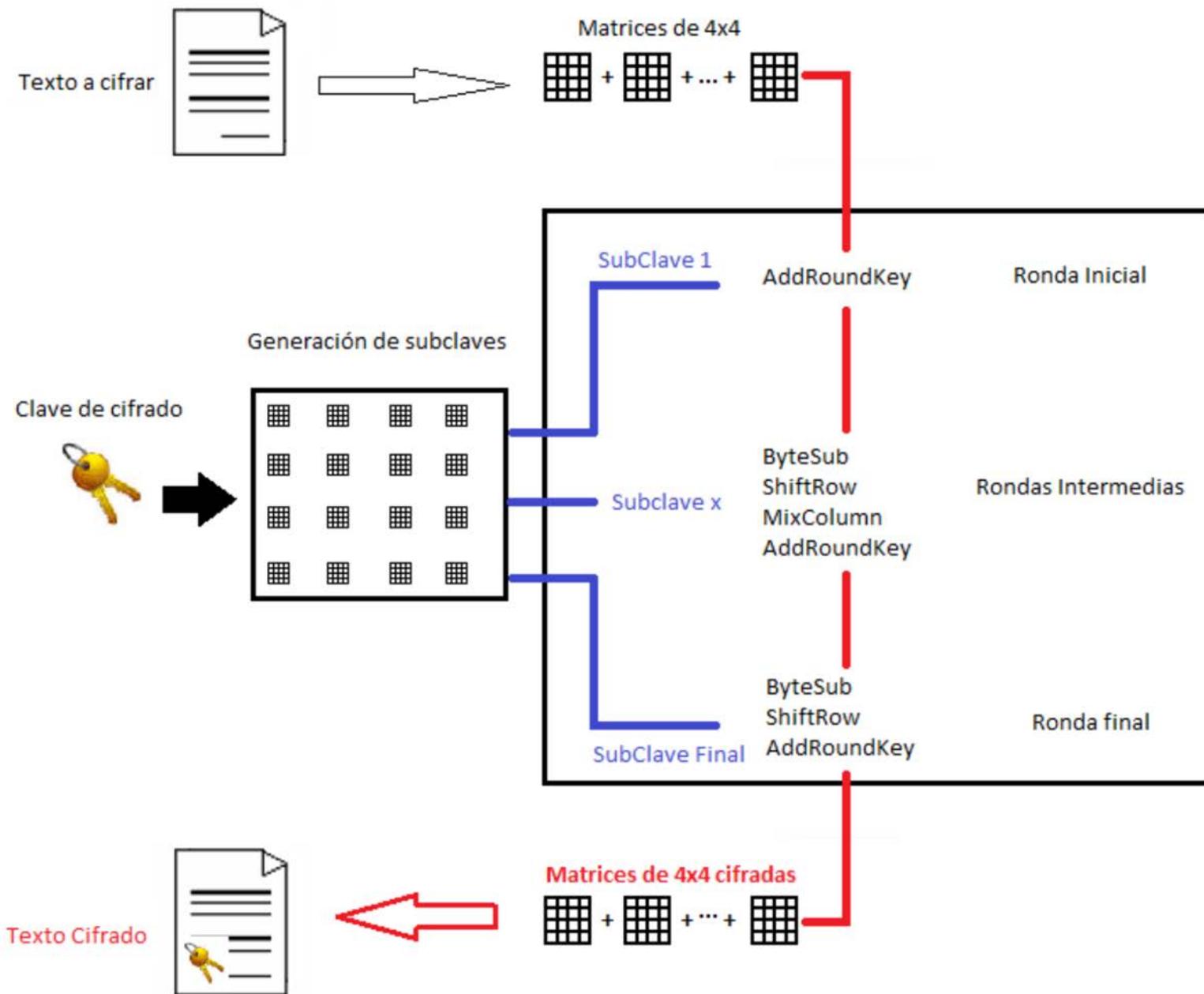
- Existe una variante en la que se utilizan sólo 2 claves, pero este esquema ha sido objeto de ataques, de forma que la **K1 = K3 (112 bits)**



# 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**

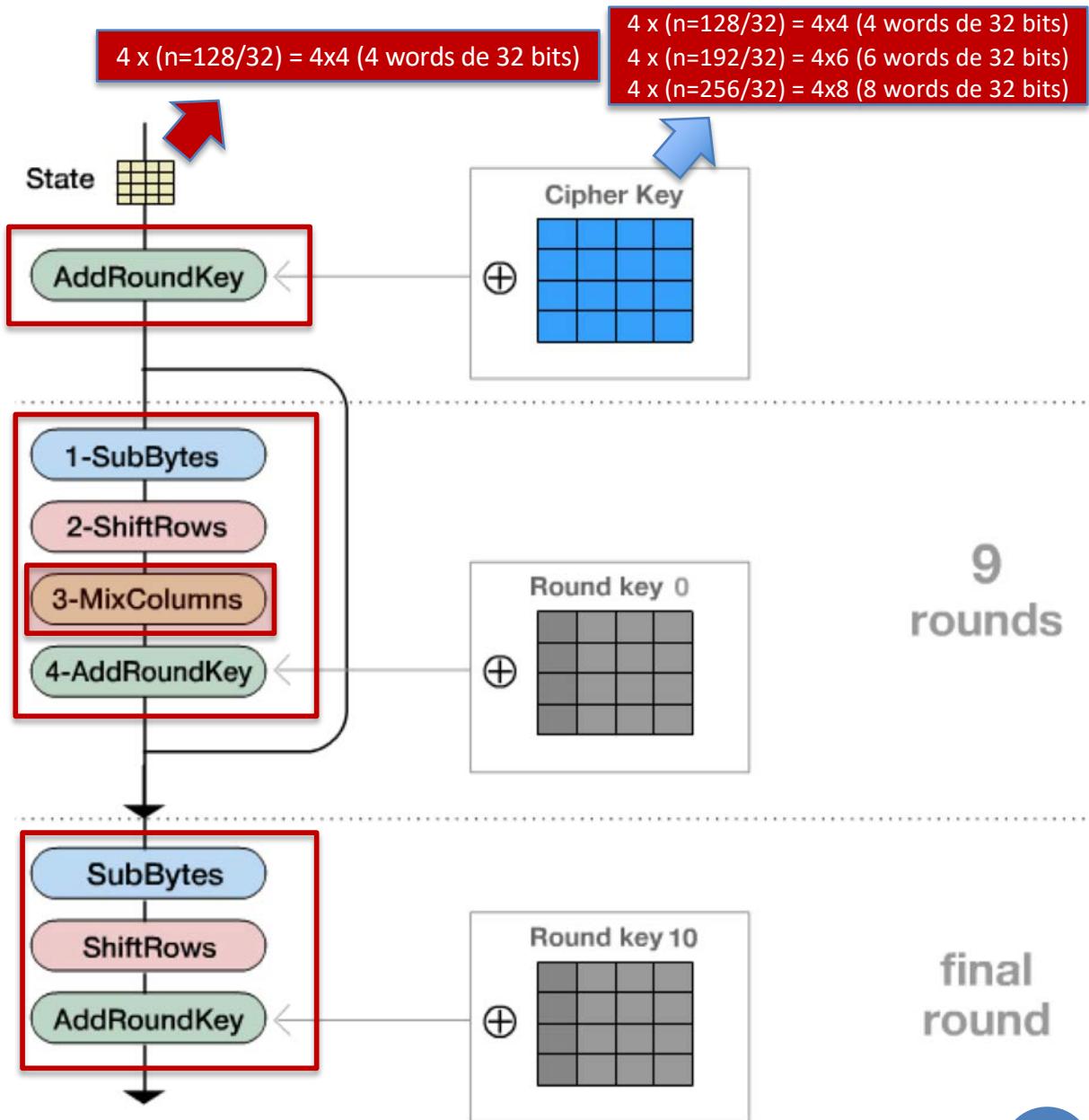
# Algoritmo AES



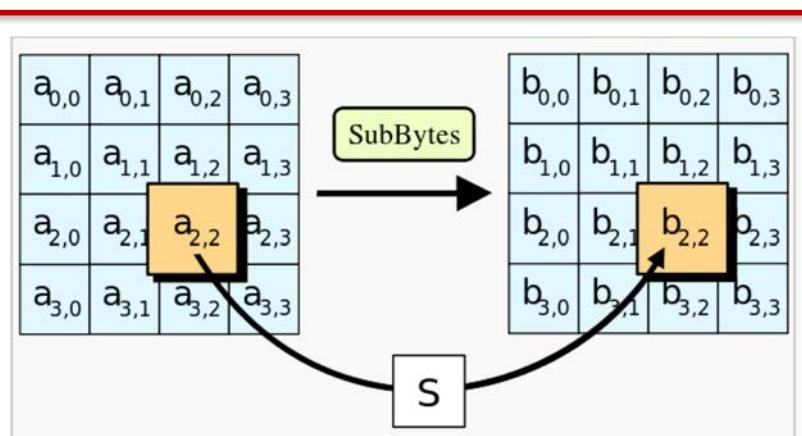
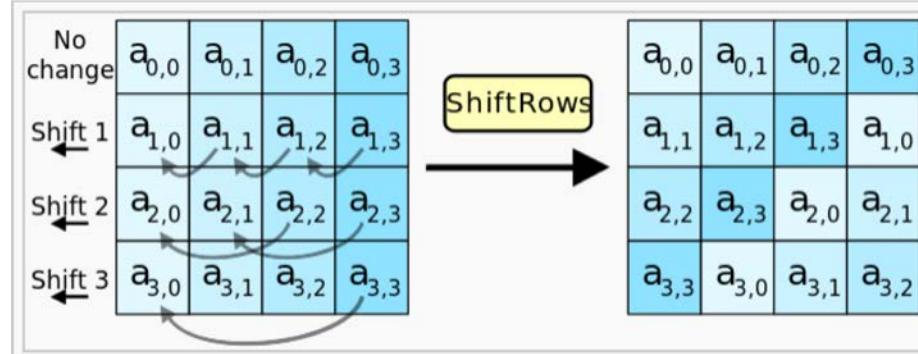
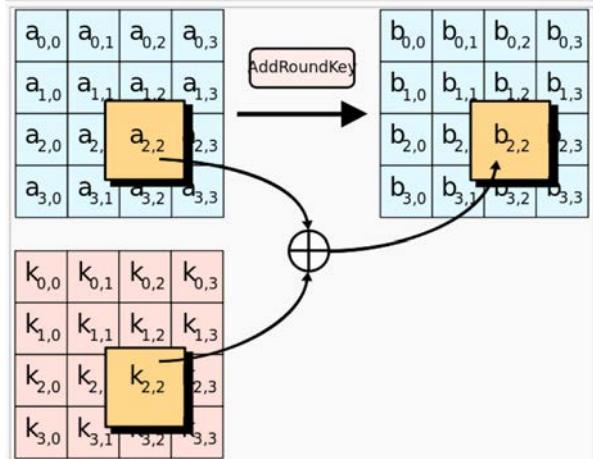
# Algoritmo AES

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

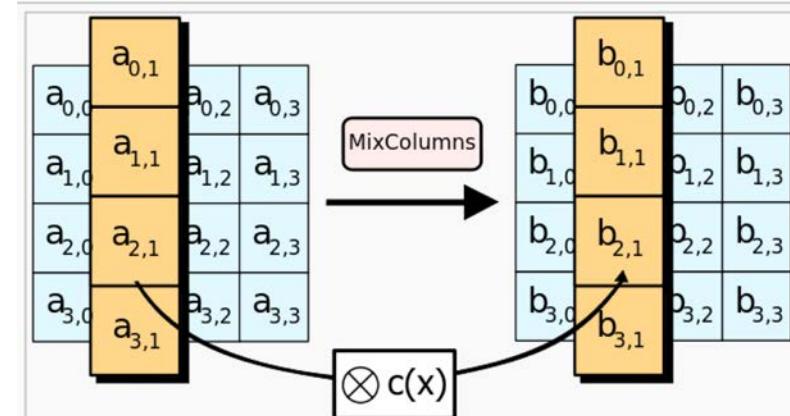
- Otras posibilidades:
  - 192 bits (12 etapas)
  - 256 bits (14 etapas)



# Algoritmo AES

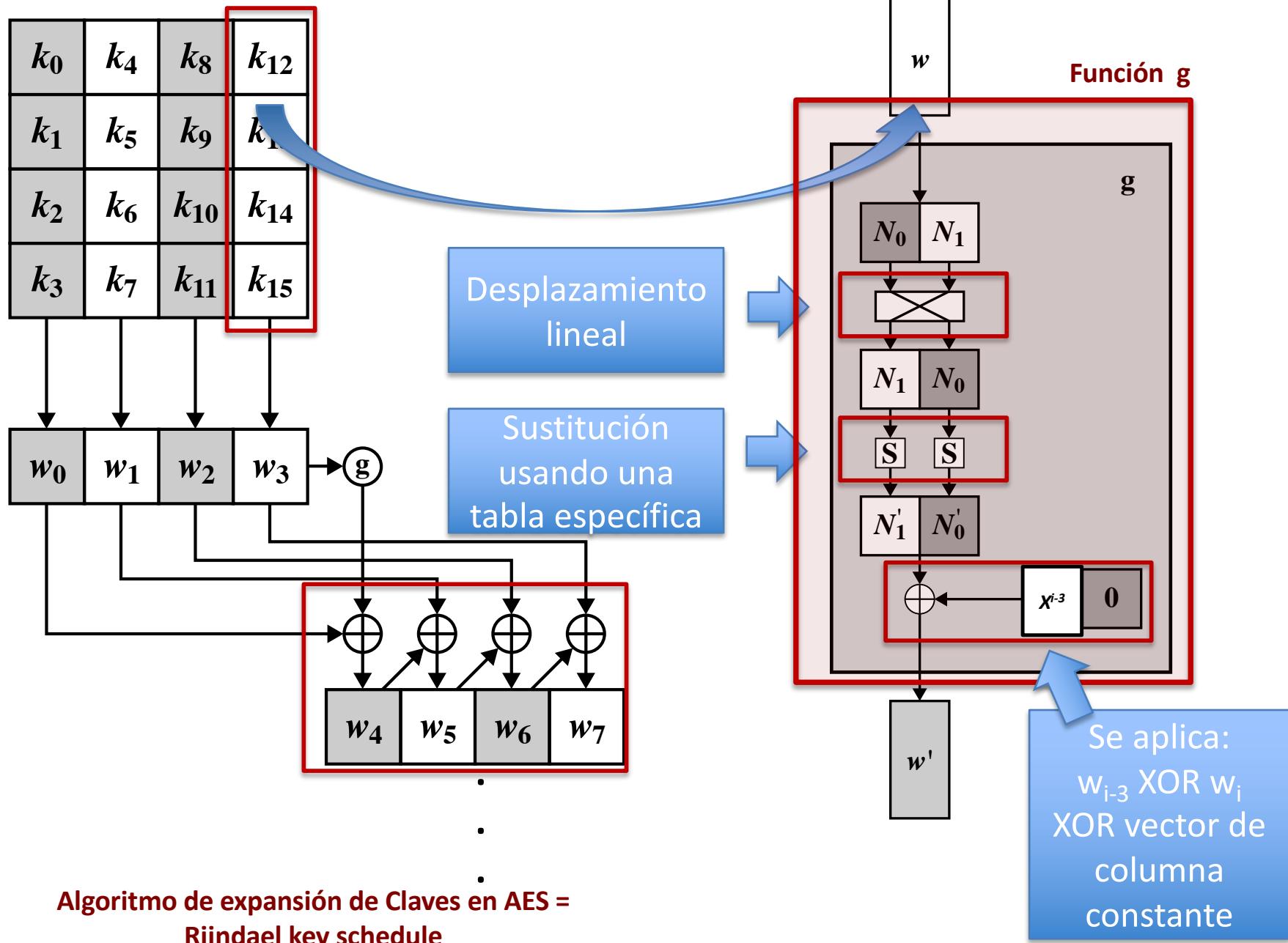


	hex	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76	
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0	
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15	
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75	
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3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



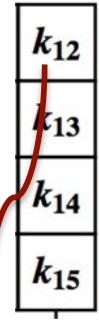
MixColumn multiplica la matriz que se está computando con matrices preestablecidas

# Algoritmo AES - Expansión de las claves AES



# Algoritmo AES - caja negra

Rotword



Sólo se mueve

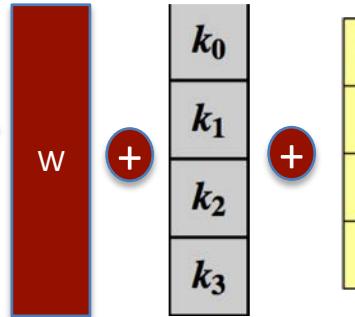
$K_{12}$  al final, resultando en:

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

SubBytes

x	y	hex
0	0	00
1	1	01
2	2	02
3	3	03
4	4	04
5	5	05
6	6	06
7	7	07
8	8	08
9	9	09
a	a	0a
b	b	0b
c	c	0c
d	d	0d
e	e	0e
f	f	0f

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

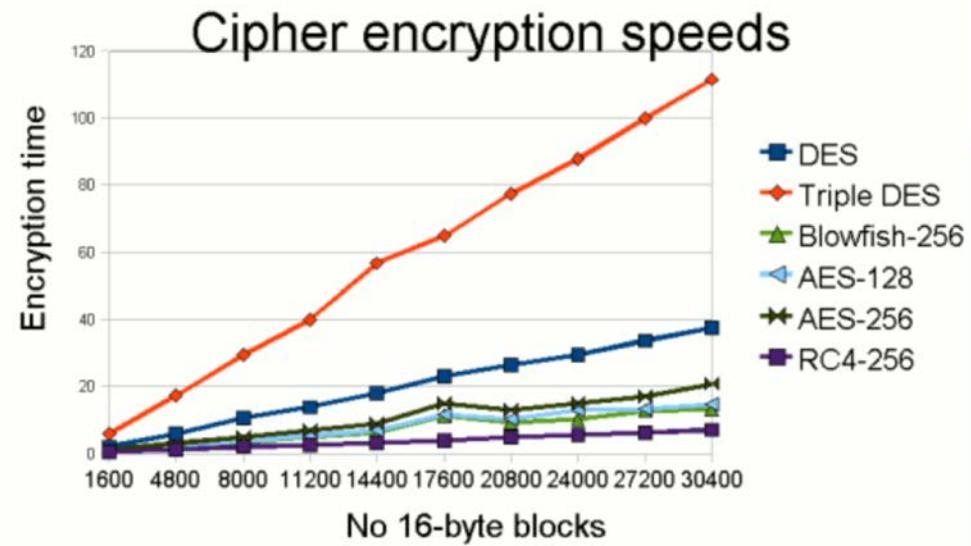
32

## 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

# Otros algoritmos simétricos relevantes

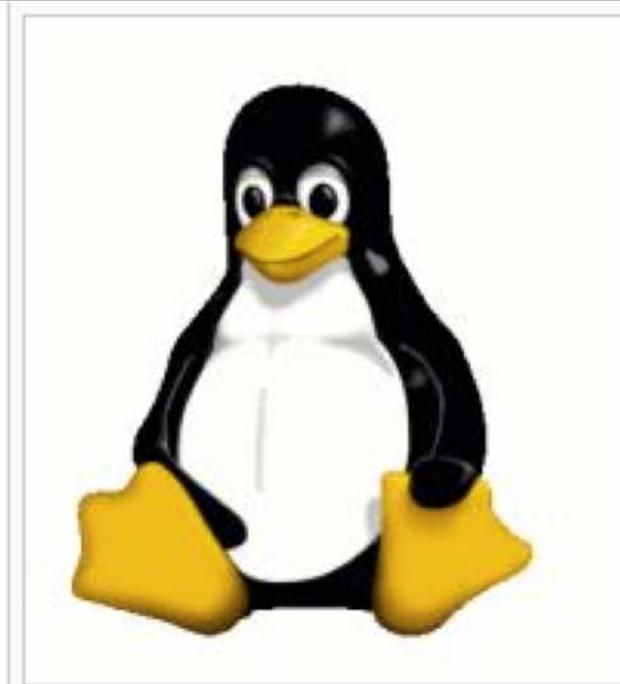
- Blowfish
  - Requiere una **clave de entre 32 y 448 bits** (pero sólo se recomienda su uso con **más de 80 bits**)
  - Se utiliza en algunas configuraciones de IPSEC
  - La longitud de cada **bloque de datos es de 64 bits**, demasiado pequeño para algunas aplicaciones
    - Por ese motivo, sólo se recomienda en uso heredado (*legacy use*)
- Kasumi
  - Requiere una **clave de 128 bits** de longitud, y la longitud de los **bloque de datos es de 64 bits**
  - Se usa en UMTS (con el nombre UIA1) y en GSM (con el nombre A5/3)
  - Presenta una serie de problemas que no afectan a su uso práctico en esas aplicaciones
    - sin embargo, no se recomienda para aplicaciones futuras
- Camellia
  - Requiere una **clave de 128 bits**, pero también da soporte a claves **de 192 y 256 bits**
    - Las versiones con 192 o 256 bits de clave son un 33% más lentas que la de 128 bits
  - Se utiliza como **uno de los posibles cifrados en TLS**
  - Por el momento no se han encontrado ataques efectivos a este algoritmo

- Recomendaciones generales:
  - En general, la **longitud mínima de clave** para un cifrado simétrico en bloque debería ser **128 bits**
  - **El tamaño mínimo de los bloques de datos** dependerá de la aplicación específica en la que se use el algoritmo, pero en la mayoría de ocasiones el mínimo **debería ser 128 bits**
  - La **cantidad máxima de información a cifrar con una misma clave** debería limitarse a  $2^{n/2}$ , donde  $n$  es el tamaño del bloque de datos

Primitive	Recommendation	
	Legacy	Future
AES	✓	✓
Camellia	✓	✓
Three-Key-3DES	✓	✗
Two-Key-3DES	✓	✗
Kasumi	✓	✗
Blowfish $\geq$ 80-bit keys	✓	✗
DES	✗	✗

## Modos de operación para algoritmos simétricos

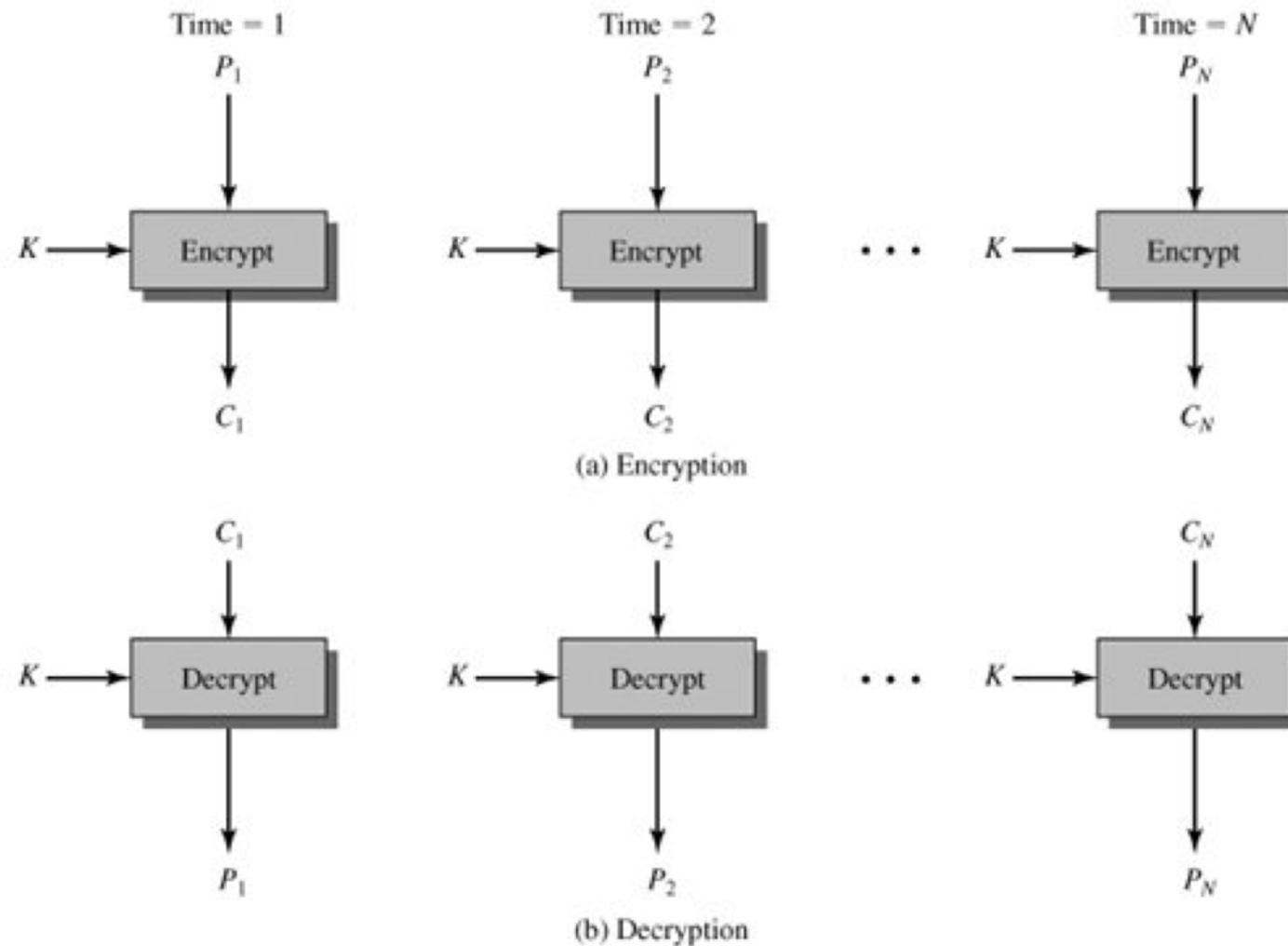
- Un modo de operación es una técnica para mejorar el efecto final de un algoritmo criptográfico
  - También se usa para adaptar el algoritmo a un tipo de aplicación concreta
- En ningún caso supone la modificación del algoritmo de cifrado en sí, sino de la forma en que se opera con los bloques de datos

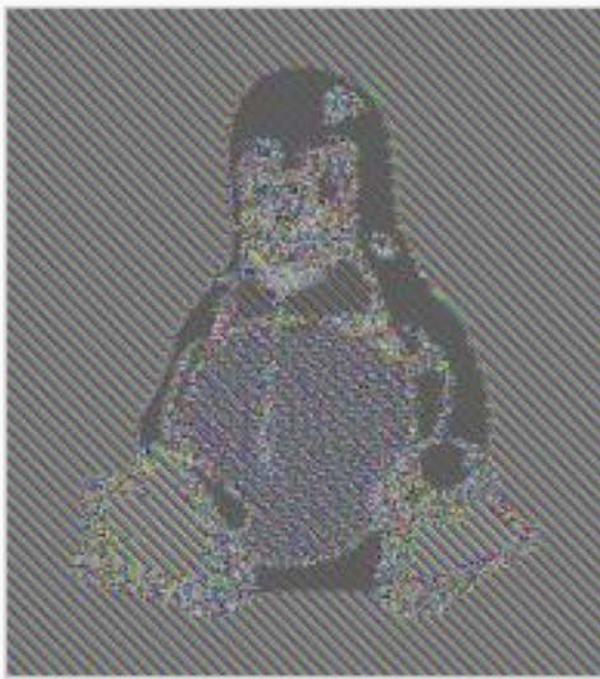


## Modos de operación para algoritmos simétricos

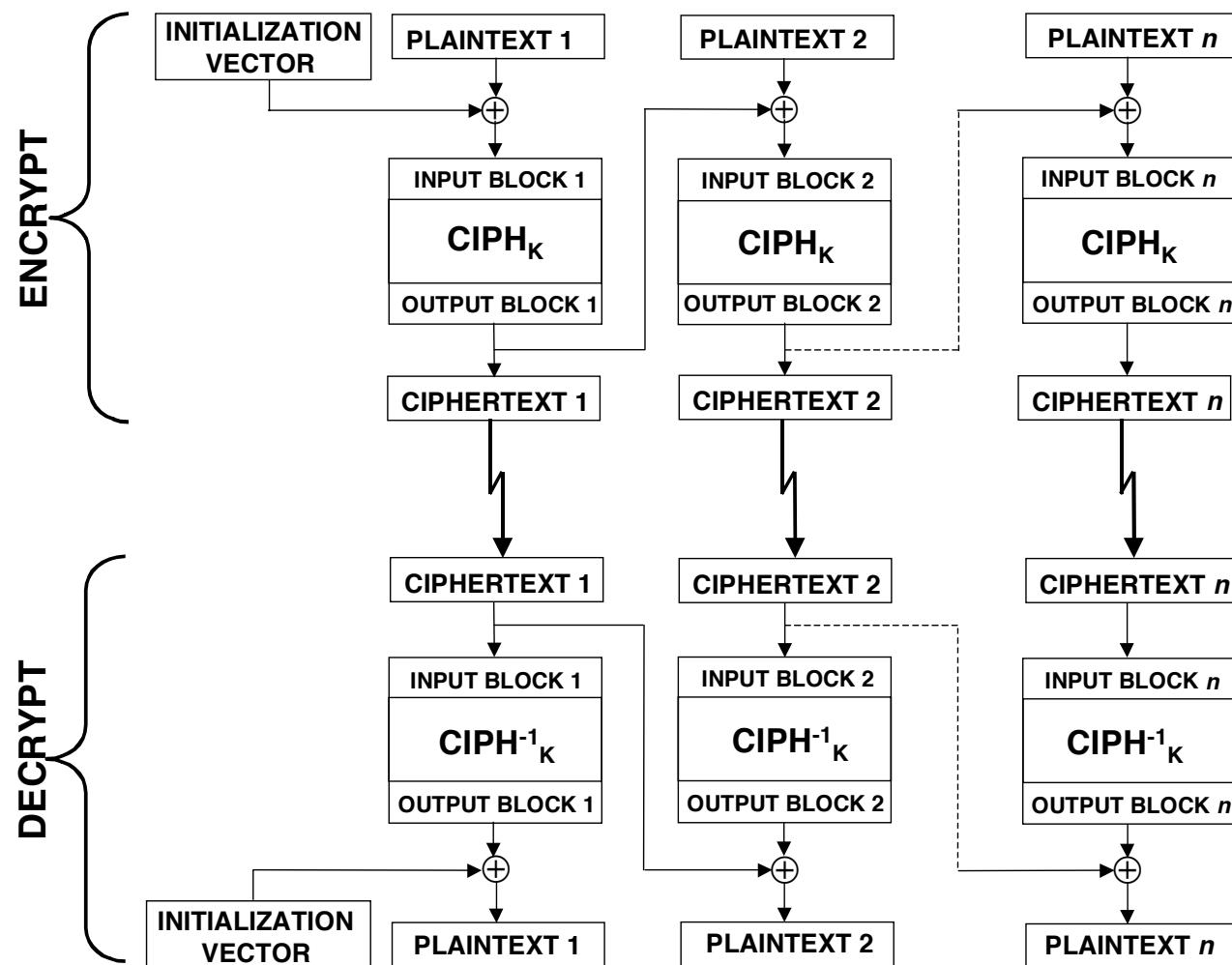
- Hay diferentes modos de operación, o lo que es lo mismo, diferentes formas de aplicar un mensaje  $M$  a un algoritmo de cifrado en bloque
  - cada una de ellas con sus ventajas y desventajas
- NIST ha definido cinco modos de operación, y cualquiera de ellos se puede utilizar con cualquier algoritmo simétrico ( $DES$ ,  $3DES$ ,  $AES$ , ...)
  - Electronic Codebook (ECB)
  - Cipher Block Chaining (CBC)
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
  - Counter (CTR)
  - Galois-CTR (GCM)

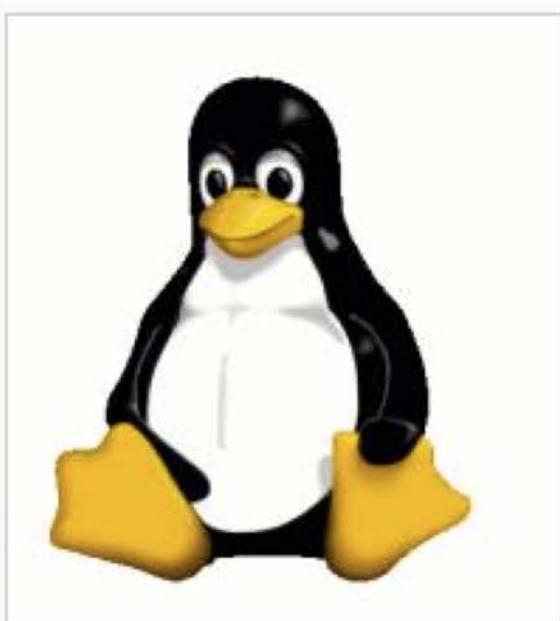
- Electronic Codebook (ECB)



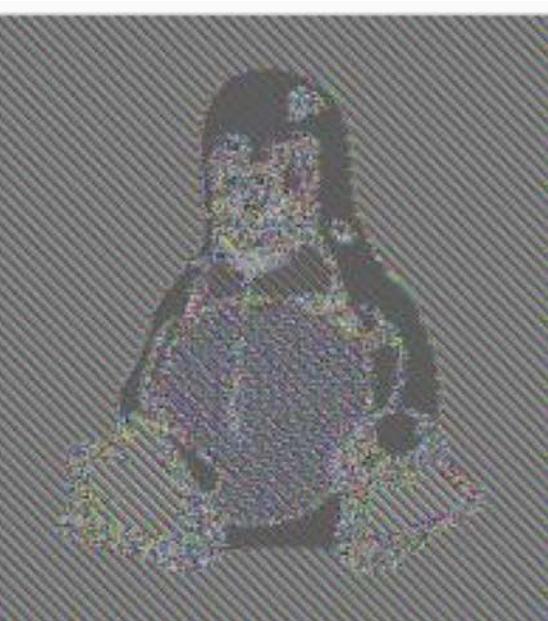


- Cipher Block Chaining (CBC)

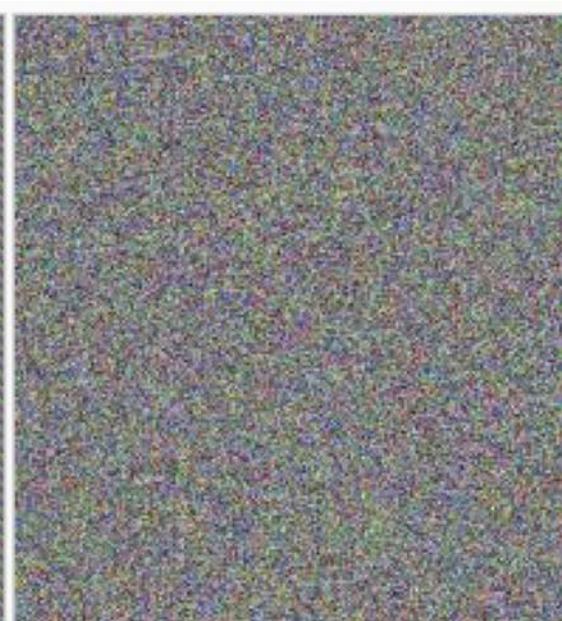




Original image

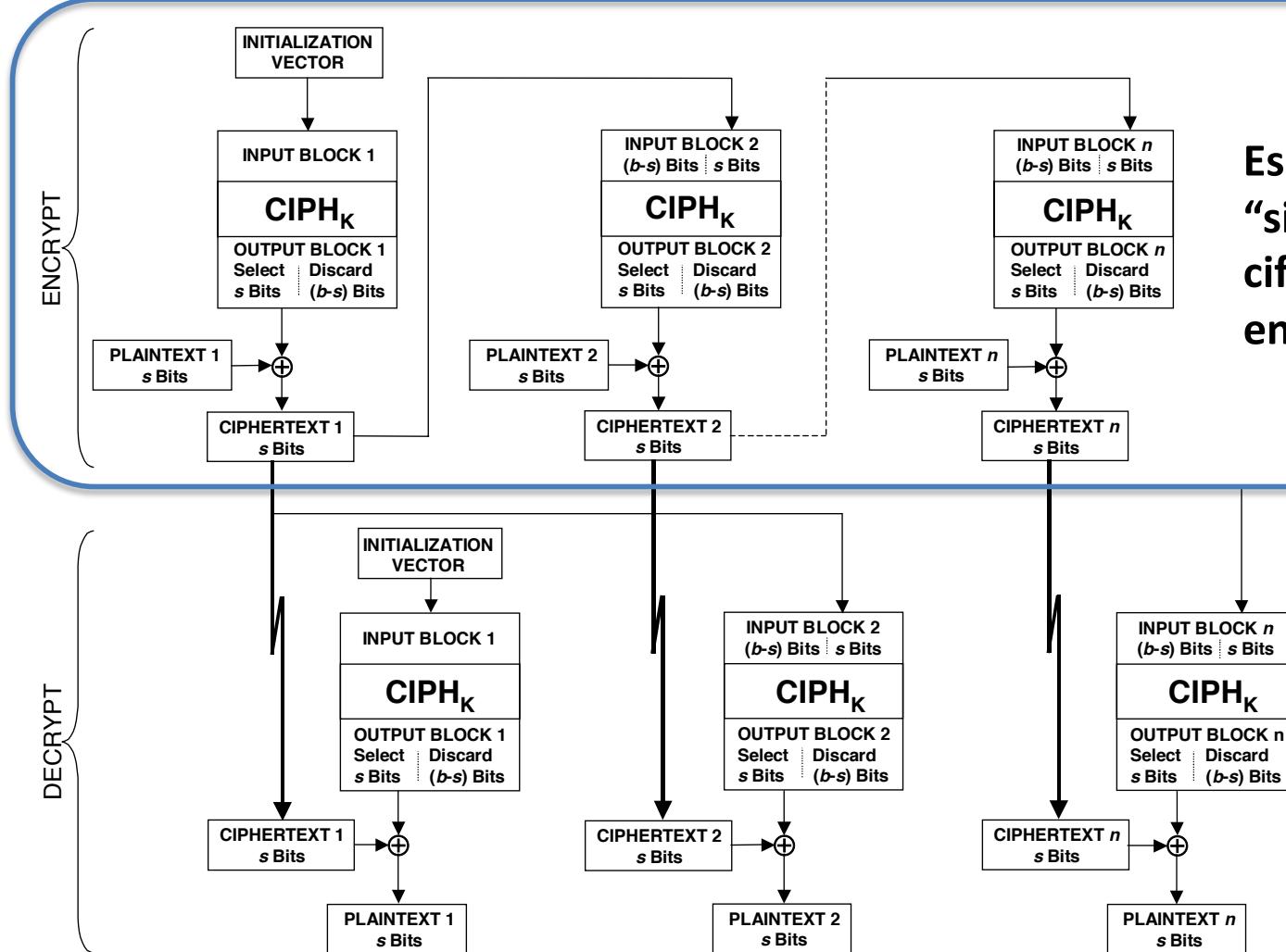


Encrypted using ECB mode



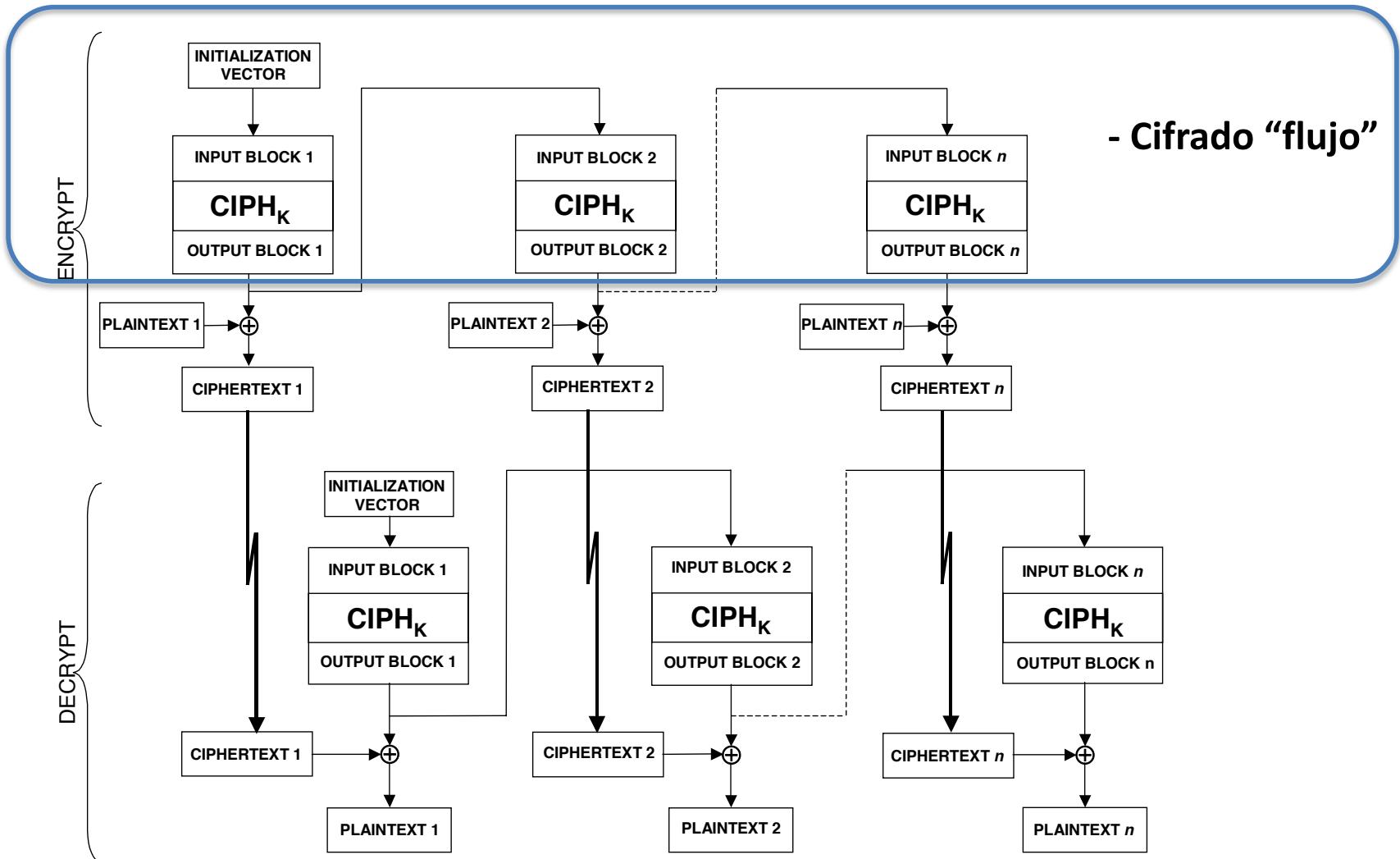
Modes other than ECB result in  
pseudo-randomness

- Cipher Feedback (CFB)



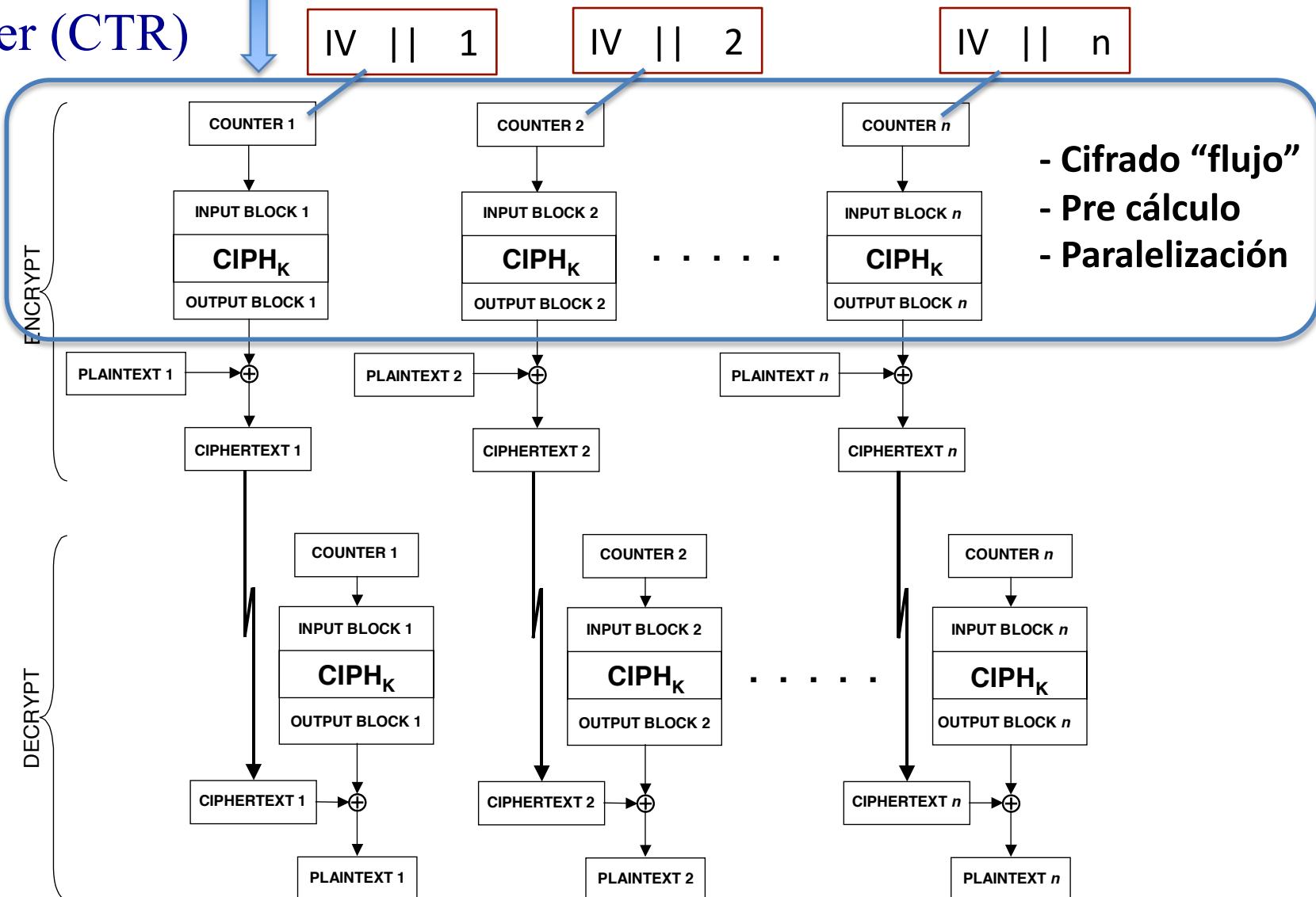
Es posible  
“simular” un  
cifrado basado  
en flujo

- Output Feedback (OFB)



- Counter (CTR)

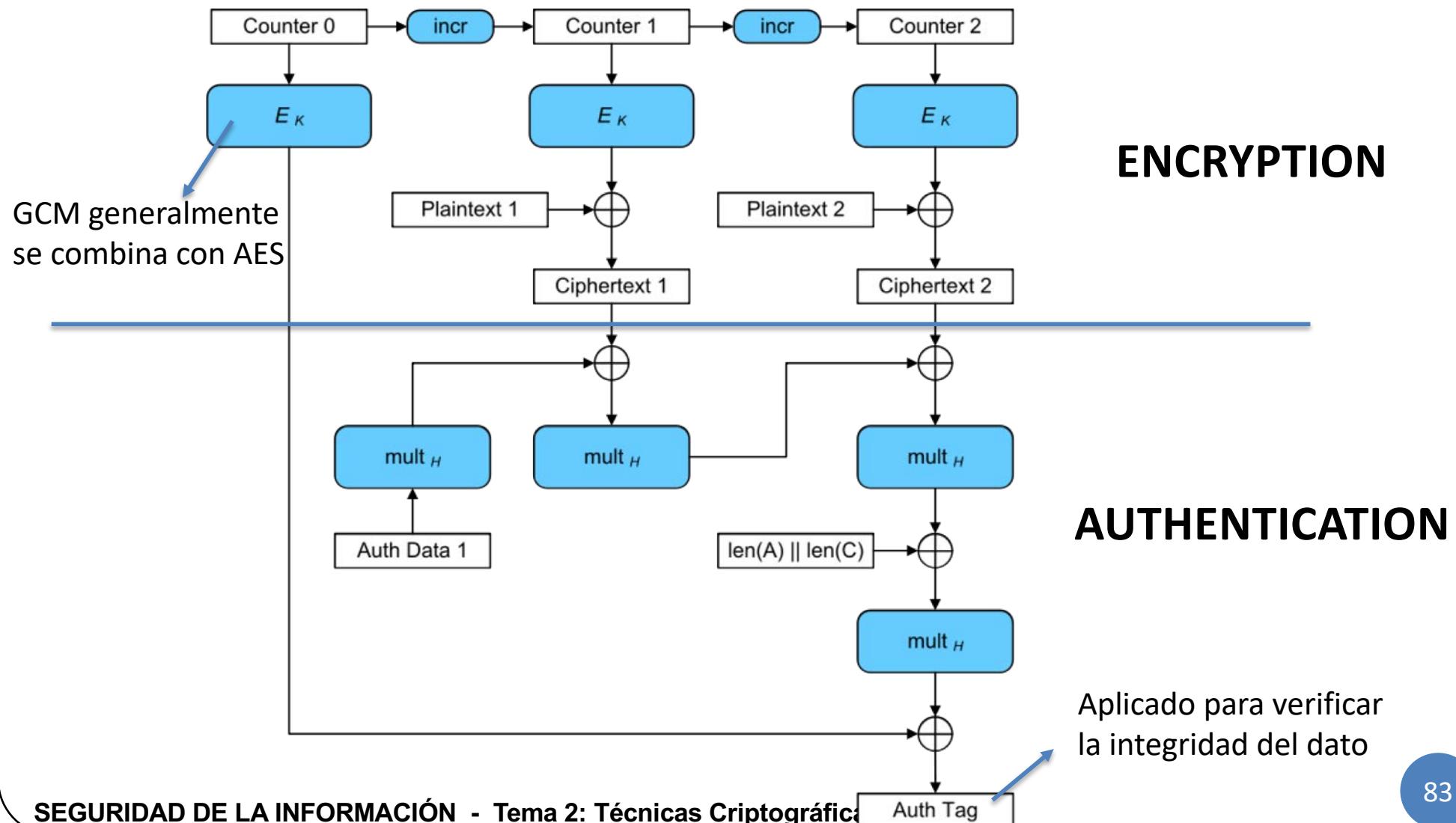
Normalmente se concatena el IV al contador



## Extra: Modos de operación con integridad

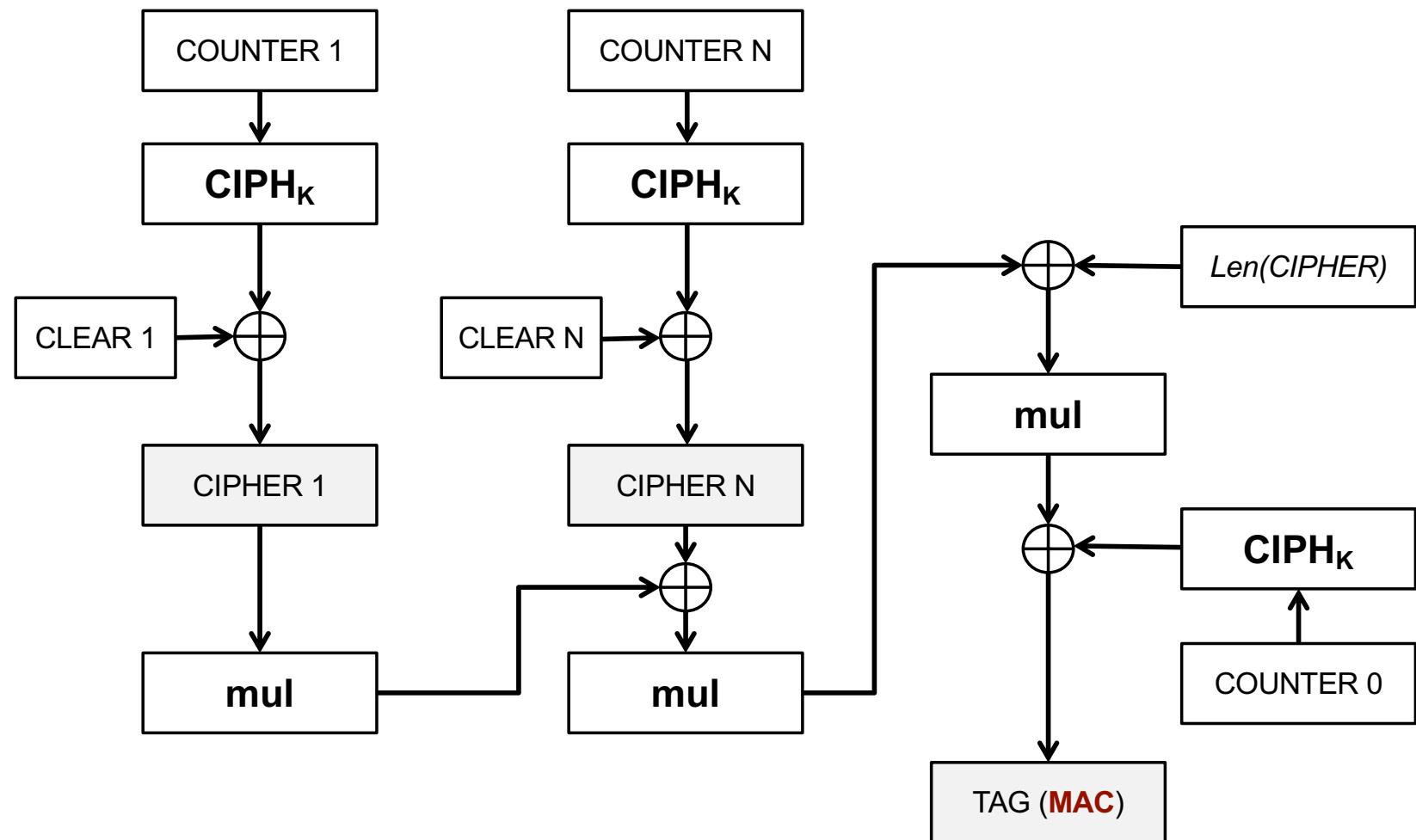
### • Galois-CTR (GCM)

- Funciona de forma similar que el modo CTR pero usa Carter-Wegman MAC en un campo de Galois
- Es rápido y eficiente, y está soportado por el suite de cifrado dado por TLS



## Extra: Modos de operación con integridad

- Galois-CTR (GCM)



# Ventajas y desventajas de los algoritmos simétricos

- **Ventajas:**
  - Los algoritmos simétricos se pueden diseñar para alcanzar un **alto rendimiento** (alto caudal de información cifrada)
    - En HW se pueden alcanzar del orden de cientos de Mbytes/sec
    - En SW se pueden alcanzar del orden Mbytes/sec
  - Se pueden **componer** para producir cifrados más fuertes
  - Se pueden utilizar como base para **construir otros mecanismos** criptográficos, como **funciones hash** y **generadores pseudoaleatorios de números**
  - Los algoritmos simétricos necesitan **claves K relativamente cortas**

- **Desventajas:**
  - En una comunicación entre dos usuarios, estos han de **acordar, a priori, la clave K** con la que cifrarán/descifrarán sus comunicaciones
    - la clave ha de permanecer estrictamente en secreto, por lo que sólo la han conocer esos dos usuarios que se comunican
  - Si los dos usuarios están **físicamente lejanos** entre sí, acordar la clave puede convertirse en una tarea difícil
    - ¿qué medio suficientemente seguro habrán de utilizar si no es posible una reunión presencial entre ambos?
    - además, a efectos de seguridad, es recomendable que la clave  $K$  entre dos usuarios se cambie con cierta frecuencia, lo que complica el problema
  - En una red grande (de muchos usuarios) habrá demasiadas claves que administrar
    - Para una comunidad de  $n$  usuarios, el número de claves en el sistema será de  **$(n * (n-1)) / 2$** 
      - Ej: 100 usuarios → 4950 claves
    - Probablemente hará falta una **tercera parte confiable** para ayudar a los usuarios en las tareas de administración de claves

- Desventajas:
  - En una comunicación entre dos usuarios, estos han de **acordar, a priori, la clave K** con la que cifrarán sus mensajes.
  - la clave ha de permanecer secreta para los usuarios que se comunican
  - Si los dos usuarios están **fisicamente separados**, la tarea de acordar la clave K puede convertirse en una tarea difícil
    - ¿qué medio suficientemente seguro existe entre ambos?
    - además, a efectos de seguridad, la clave K debe cambiarse con cierta frecuencia, lo que dificulta la tarea
  - En una red grande (de muchos usuarios) es difícil administrar las claves
    - Para una comunidad de  $n$  usuarios, el número de claves en el sistema será de  $(n * (n-1)) / 2$ 
      - Ej: 100 usuarios → 4950 claves
    - Probablemente hará falta una **tercera parte confiable** para ayudar a los usuarios en las tareas de administración de claves

