



# CRIPTOGRAFÍA PARA INGENIER@S

## Class4crypt

© Jorgeramió 2022

Aula virtual de  
criptografía  
aplicada

Diapositivas  
utilizadas en las  
clases grabadas  
de Class4crypt

Módulo 8 Criptografía simétrica en bloque

Dr. Jorge Ramió Aguirre © 2022



Attribution-NonCommercial-  
NoDerivatives 4.0 International  
(CC BY-NC-ND 4.0)



*El ingenio es intrínseco al ser humano,  
solo hay que darle una oportunidad  
para que se manifieste.*

<https://www.criptored.es/cvJorge/index.html>

# Tu aula virtual de criptografía aplicada

## Módulo 10. Criptografía asimétrica

# Class4crypt

## Módulo 8. Criptografía simétrica en bloque

8.1 Fundamentos de la cifra simétrica en bloque

8.2 Algoritmo DES: redes de Feistel y cajas S

8.3 Algoritmo DES: expansión de clave, cifra y rellenos

8.4a ECB y CBC: modos de cifra en bloque con confidencialidad

8.4b CFB, OFB y CTR: modos de cifra en bloque con confidencialidad

8.5 Ataques al DES, DES Challenge y 3DES

8.6a Algoritmo AES parte 1: visión general y fortaleza

8.6b Algoritmo AES parte 2: Campos de Galois y expansión de clave

8.6c Algoritmo AES parte 3: SubBytes, ShiftRows, MixColumns, AddRoundKey

Lista de reproducción del módulo 8 en el canal Class4crypt

[https://www.youtube.com/playlist?list=PLq6etZPDh0kvTwlIHHeaq\\_HQKgJ90AIPx](https://www.youtube.com/playlist?list=PLq6etZPDh0kvTwlIHHeaq_HQKgJ90AIPx)

# Class4crypt c4c8.1

## Módulo 8. Criptografía simétrica en bloque

### Lección 8.1. Fundamentos de la cifra simétrica en bloque

8.1.1. Formando bloques para cifrar

8.1.2. Esquema de la cifra simétrica en bloque

8.1.3. Características de la cifra simétrica en bloque

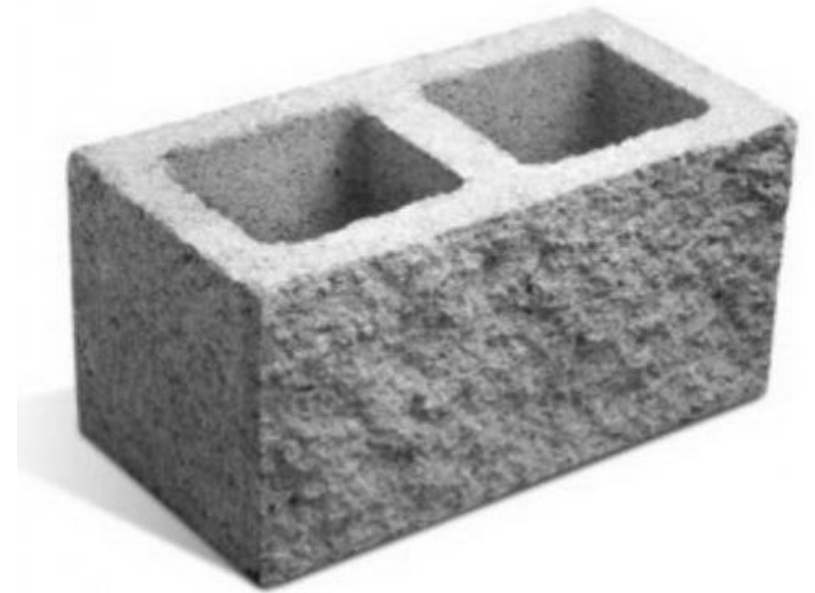
8.1.4. Recorrido histórico por los algoritmos de cifra en bloque más populares

8.1.5. Algoritmos de cifra en bloque que deberíamos conocer y estudiar

Class4crypt c4c8.1 Fundamentos de la cifra simétrica en bloque  
<https://www.youtube.com/watch?v=85tpwBNWnN0>

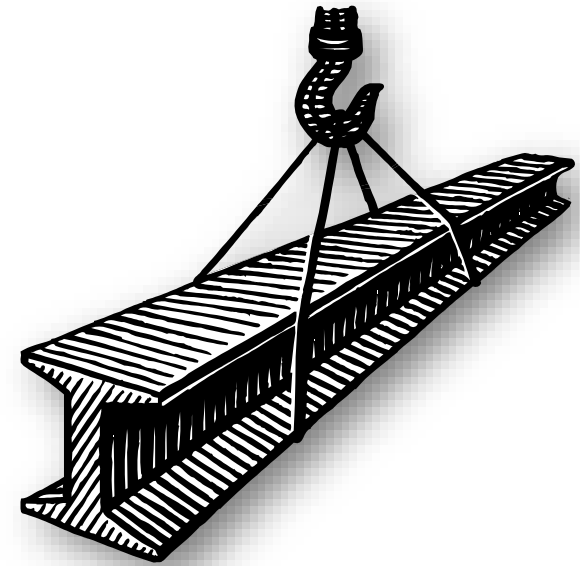
# Armando bloques para la cifra

- En criptografía clásica ya se formaban bloques de texto en claro para cifrar la información
- Cifra poligráfica: por ejemplo cifradores de Playfair (1854) y de Hill (1929)
- En criptografía moderna se hace lo mismo, aunque obviamente no se forman bloques de letras sino de bytes
- Entre los algoritmos más conocidos de la cifra en bloque moderna están el DES, el IDEA y el AES, si bien existen muchos más



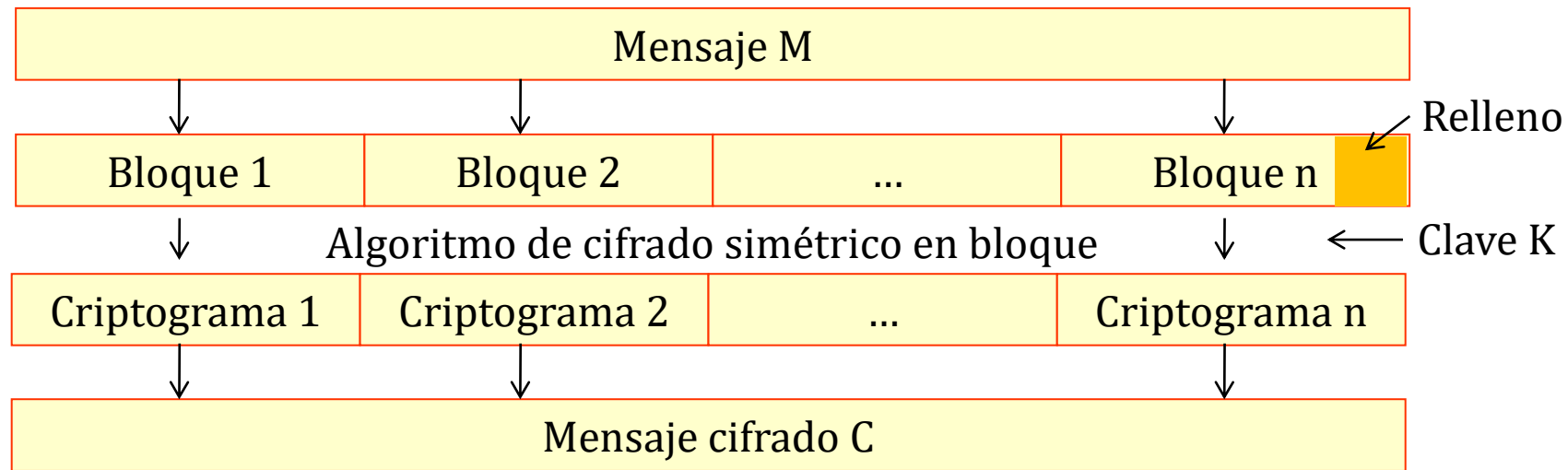
# Cifrado y tamaño típico del bloque

- Se denomina cifrado en bloque a aquella cifra en la que el mensaje original se agrupa en bloques de “x” bytes antes de proceder a aplicar el proceso de cifrado
- Un bloque pequeño (1, 2, 3 bytes) facilitaría un ataque por estadísticas del lenguaje
- Un bloque grande (por ejemplo miles de bytes) supondría un tratamiento no adecuado del texto en claro a cifrar
- Lo típico es utilizar bloques de 8 o 16 bytes, 64 o 128 bits. Actualmente se usan 16 bytes





# Esquema de la cifra simétrica en bloque




- El mensaje  $M$  se divide en bloques  $M_1, M_2, M_3, \dots, M_{n-1}, M_n$ . Es posible que el último bloque necesite relleno; aquí dependerá del algoritmo usado
- El criptograma  $C$  será igual a la concatenación  $C_1 + C_2 + C_3 \dots + C_{n-1} + C_n$
- Problema: que sean bloques independientes forzará a modos de cifra

# Características de la cifra en bloque

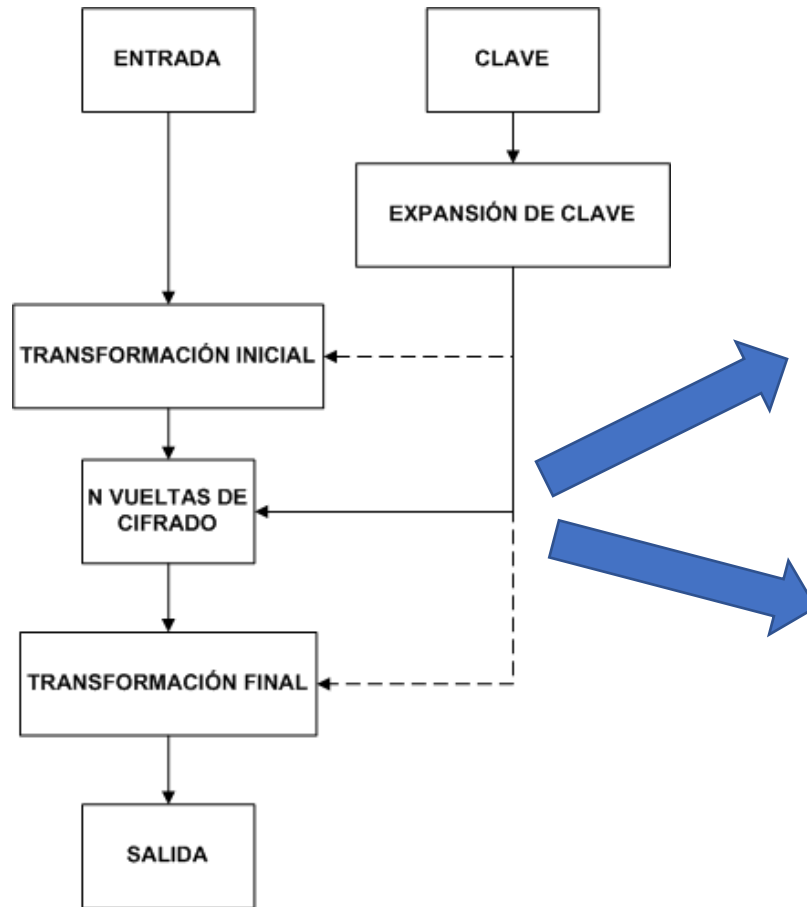
- **Dependencias entre símbolos:** en cada bloque cifrado, cada bit del criptograma es una función compleja de todos los bits de la clave y de todos los bits del bloque del texto en claro
- **Cambio en los bits de entrada:** el cambio de un bit en el bloque del texto en claro produce un cambio de aproximadamente el 50% de los bits del criptograma
- **Cambio en los bits de la clave:** el cambio en un bit de la clave produce aproximadamente un cambio del 50% de los bits del criptograma
- **Errores de transmisión:** un error en la transmisión de un texto cifrado, se propaga a todo el bloque del que forma parte, produciendo un conjunto de errores después del proceso de descifrado del 50% de los bits del bloque afectado, lo que equivale a que en media todo el bloque sea ininteligible, y potencialmente otros bloques



# Estructura de la cifra en bloque (1)

1. Una transformación inicial en algunos cifradores
    1. Que carece de significado criptográfico
    2. Y cambia la posición de los datos de entrada, oculta bloques de datos
  2. El algoritmo de expansión de la clave. Convierte la clave introducida por el usuario en un conjunto de subclaves
    - a) Diferentes para cada vuelta
    - b) El conocimiento de una o varias subclaves intermedias, no debe permitir deducir las subclaves anteriores o siguientes
  3. Las vueltas o rondas intermedias consisten en una función no lineal y compleja entre los datos de entrada y de la clave
  4. Una transformación final en algunos cifradores
    - a) Que carece de significado criptográfico
    - b) Y su función consiste en invertir la transformación inicial
- Gráficamente...
- 

# Estructura de la cifra en bloque (2)



## Analogía mayonesa



- Aquí se realizan operaciones de sustitución (confusión) y de permutación (difusión)
- Son cifradores de producto

# Tamaño de bloque óptimo

- DES, IDEA y Blowfish usan bloques de 8 bytes, 64 bits ( $2^6$  bits)
- AES, Twofish y Serpent usan bloques de 16 bytes, 128 bits ( $2^7$  bits)
- Bloques no muy cortos para evitar:
  - Ataque Sweet32
- Bloques no muy largos para reducir:
  - El tamaño del texto cifrado implica mayores requisitos de memoria
- Además, un bloque de 128 bits (de 64 a 512 bits) puede ser implementado en hardware de manera muy eficiente

# Número de vueltas o rondas

- Dos opciones: algoritmo con una única operación compleja versus otro algoritmo con una secuencia de operaciones sencillas que se repiten  $n$  veces ( $n$  vueltas o rondas)
- Una vuelta es una operación de transformación
  - Se itera múltiples veces ( $n$  veces) sobre dicha operación para diseñar el algoritmo de cifrado simétrico en bloque
- Más sencillo de analizar desde el punto de vista de seguridad
- Todas las vueltas emplean la misma operación
  - Se diferencian en la clave de vuelta (diferente en cada vuelta)
  - Evitar ataques de desplazamiento (slide attacks)
- Claves de vuelta derivadas de la clave original
  - Algoritmo de expansión de clave (key schedule)

# Redes de sustitución y permutación

- Esquemas de Feistel (DES y 3DES)
- Redes de sustitución y permutación (AES)
  - SPNs: Substitution-Permutation Networks
  - Transformaciones eficientes de bits
- Aplicación de los conceptos de confusión y difusión
  - Operaciones de sustitución (confusión)
    - S-boxes (Substitution boxes) o Cajas-S
    - Transformación de un grupo pequeño de bits (4 u 8)
    - Ecuaciones no lineales y robustas criptográficamente
  - Operaciones de permutación (difusión)
    - P-boxes (Permutation boxes) o Cajas-P
    - Reordenación o mezcla de los bits
    - Operaciones de álgebra lineal y multiplicaciones de matrices

# Barrido histórico de algoritmos en bloque

	Algoritmo	Tamaño de Bloque (bits)	Tamaño de Clave (bits)	Número de Vueltas
1971	Lucifer	128	128	16
1976	DES	64	56	16
1989	Loki	64	64	16
1987	RC2	64	variable	-
1996	CAST	64	64	8
1993	Blowfish	64	variable	16
1991	IDEA	64	128	8
1998	Skipjack	64	80	32
2001	AES	128	128 o más	flexible

La lista de algoritmos de cifra simétrica en bloque es mayor. Entre otros, Khufu, Khafre, Gost, RC5, SAFER 64, Akelarre, FEAL, Camellia, CAST-128, SEED, ARIA, TEA, XTEA






# Algoritmos simétricos de cifra en bloque 1

- **Lucifer**: tipo Feistel usado a comienzos de los años 70 por en el Reino Unido y que posteriormente dará lugar al DES
- **DES**: tipo Feistel que se convirtió en estándar durante casi 25 años. Hoy es vulnerable por su pequeña longitud de clave y ha dejado de ser estándar
- **Loki**: australiano similar al DES, también de tipo Feistel
- **RC2**: de Ron Rivest, y que se incluye en navegadores de Internet desde 1999
- **CAST**: canadiense tipo Feistel, que se ofrece como uno de los algoritmos de cifra en algunas versiones de PGP
- **Blowfish**: algoritmo de tipo Feistel propuesto por Bruce Schneier
- **IDEA**: algoritmo europeo usado en el correo electrónico de PGP
- **Skipjack**: una propuesta de la NSA a finales de los 90 para comunicaciones oficiales (tiene puerta trasera)
- **Rijndael**: nuevo estándar mundial desde finales de 2001, conocido como AES, Advanced Encryption Standard

# Algoritmos simétricos de cifra en bloque 2

- **Khufu**: de Ralph Merkle con una clave generada con un sistema de cajas S
  - **Khafre**: de Ralph Merkle en que la clave ya no depende de las cajas S
  - **Gost**: similar al DES con cajas S secretas de la Unión Soviética
  - **RC5**: de Ron Rivest, realiza operaciones or exclusivo, suma modular y desplazamiento de bits
  - **SAFER 64**: propuesto por James Massey
  - **Akelarre**: español propuesto en 1996 por el Consejo Superior de Investigaciones Científicas CSIC
  - **FEAL**: algoritmo propuesto en Japón
  - Además de los 15 algoritmos que participaron desde 1997 en el concurso del NIST para elegir el nuevo estándar mundial AES, Advanced Encryption Estándar, y que finalmente gana Rijndael
- 

# 15 algoritmos admitidos al concurso AES

- **CAST-256**: Entrust Technologies, Inc.
  - **CRYPTON**: Future Systems, Inc.
  - **DEAL**: Richard Outerbridge, Lars Knudsen
  - **DFC**: CNRS – Centre National pour la Recherche Scientifique – École Normale Supérieure
  - **E2**: NTT – Nippon Telegraph and Telephone Corporation
  - **FROG**: TecApro International, S.A.
  - **HPC**: Rich Schroepel
  - **LOKI97**: Lawrie Brown, Josef Pieprzyk, Jennifer Seberry
  - **MAGENTA**: Deutsche Telekom AG
  - **MARS**: IBM
  - **RC6**: RSA Laboratories
  - **Rijndael**: John Daemen, Vincent Rijmen
  - **SAFER+**: Cylink Corporation
  - **Serpent**: Ross Anderson, Eli Biham, Lars Knudsen
  - **Twofish**: Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson
- <https://nvlpubs.nist.gov/nistpubs/jres/104/5/j45nec.pdf>

# Finalistas y ganador del concurso AES

- El NIST llama a concurso AES, Advanced Encryption Standard, el 2 de enero de 1997
- Con 5 candidatos en la última ronda, el NIST decide el 2 de octubre de 2000 que el algoritmo ganador del concurso es Rijndael
- Oficialmente Rijndael es desde noviembre de 2001 el nuevo estándar de cifra simétrica en bloque AES

Algoritmo	Votos a favor	Votos en contra
Rijndael	86	10
Serpent	59	7
Twofish	31	21
RC6	23	37
MARS	13	84

# ¿Por qué estudiamos solo DES, IDEA y AES?

- DES es un cifrador tipo Feistel que ha sido un estándar desde 1976 hasta 1999, y además el primer algoritmo digital de uso civil. El algoritmo es de muy fácil comprensión y usa unas cajas S de sustitución, similar a algoritmos modernos y actuales como AES
- El 3DES se sigue utilizando en cifra local o convencional
- IDEA (1991) hace uso de los inversos multiplicativos, aditivos y xor para las claves de cifrado y descifrado dentro de un módulo de cifra, por lo que además tiene un importante interés didáctico
- AES (Rijndael) es el nuevo estándar de cifra simétrica en bloque desde 2001 y seguirá vigente durante varios años más

# Audiovisual complementario: intypedia



Lección 2: Sistemas de cifra con clave secreta

<https://www.youtube.com/watch?v=46Pwz2V-t8Q>



# Conclusiones de la lección 8.1

- Tanto en cifra clásica como en moderna existe el modo de cifra en bloque
- Los tamaños típicos de bloques eran 64 bits (8 bytes) y actualmente son 128 bits (16 bytes)
- Al formar bloques habrá que añadir rellenos y forzar a que exista unos modos de cifra en bloque, que lo veremos en próximas clases
- Las características de un esquema de cifra en bloques son:
  - El tamaño del bloque de texto en claro
  - Las vueltas o rondas con su algoritmo de expansión de claves
  - Las redes de sustitución y permutación
- Entre los algoritmos de cifra simétrica en bloque más usados y conocidos se encuentran DES, 3DES, IDEA, Serpent, Twofish y AES

# Lectura recomendada

- SWEET32: Ataques de cumpleaños contra las cifras de TLS de bloques de 64 bits (CVE-2016-2183), Red Hat Customer Portal, 2019
  - <https://access.redhat.com/es/articles/2621311>
- Slide Attacks, David Wagner & Alex Biryukov, 1999
  - <http://www.cs.haifa.ac.il/~orrd/BlockCipherSeminar/NadavGreenberg.pdf>
- Finalistas de AES, NIST
  - <https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development>
- Clipper chip (Skipjack), Wikipedia
  - [https://en.wikipedia.org/wiki/Clipper\\_chip](https://en.wikipedia.org/wiki/Clipper_chip)
- Lección 2: Sistemas de cifra con clave secreta, intypedia, Fausto Montoya, 2010
  - <https://www.criptored.es/intypedia/docs/es/video2/GuionIntypedia002.pdf>

# Class4crypt c4c8.2

## Módulo 8. Criptografía simétrica en bloque

### Lección 8.2. Algoritmo DES: redes de Feistel y cajas S

8.2.1. Estudio cronológico del Data Encryption Standard DES

8.2.2. Limitaciones de la NSA a los tamaños del bloque y de la clave

8.2.3. La clave real del DES y el código hexadecimal

8.2.4. Redes de Feistel: creación de bloques izquierdo y derecho del texto en claro

8.2.5. Operaciones de permutación en el texto en claro y en resultados

8.2.6. Operaciones de sustitución con cajas S

Class4crypt c4c8.2 Algoritmo DES: redes de Feistel y cajas S  
<https://www.youtube.com/watch?v=2NtR5giK04c>

# Orígenes y cronología del DES

- **1973**: la NBS National Bureau of Standards (actual NIST) llama a concurso público para un algoritmo criptográfico estándar con fines no militares
- **1974**: la NSA National Security Agency declara desierto el primer concurso. Se publican unas segundas especificaciones y se elige como ganador a Lucifer, un algoritmo original de IBM de los años 70, pero con algunas variaciones
- **1976**: el DES se adopta como estándar y se autoriza para ser utilizado en las comunicaciones no clasificadas del gobierno
- **1976-1999**: DES fue utilizado como estándar mundial durante casi 25 años
- **1999**: después de 4 ataques distribuidos por fuerza bruta, conocidos como DES Challenge I, II-1, II-2 y III, propuestos y liderados por RSA, en 1999 el algoritmo DES claudica definitivamente al ser roto en menos de un día

# Confirmaciones como estándar mundial

- 1976: el DES se adopta como estándar de cifra simétrica en bloque
- 1983: se confirma al DES como estándar por primera vez
- 1988: se confirma al DES como estándar por segunda vez
- En 1988 la NBS pasa a llamarse NIST National Institute of Standards and Technology
- 1993: se confirma al DES como estándar por tercera vez
- 1999: se confirma al DES como estándar por cuarta vez pero se indica que se use preferentemente la variante denominada 3DES o Triple DES
- 2002: el DES es reemplazado definitivamente como estándar por el AES, Advanced Encryption Standard

# Limitaciones en tamaños de bloque y clave

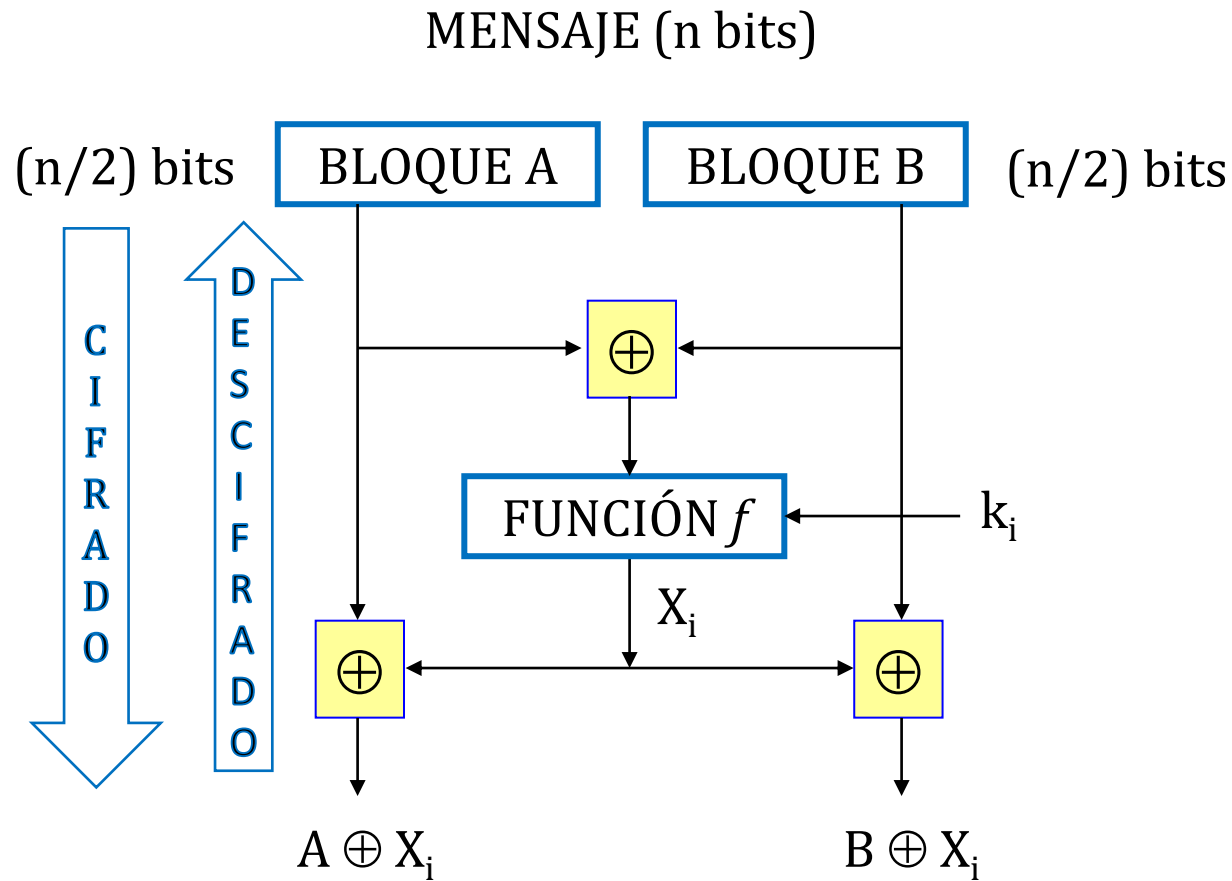
- La NSA impone una limitación en la longitud del bloque y de la clave
- De los 128 bits originales del algoritmo Lucifer, los deja en 64 bits
- Pero, la clave efectiva sólo son 56 bits puesto que son números de 8 bits (byte), donde el octavo bit es de paridad y por tanto conocido
- Espacio de claves:  $2^{56} = 7,2 \cdot 10^{16} = 72.057.594.037.927.936$
- Un valor de tan solo setenta y dos mil billones (bajo para criptografía)
- Versiones sobre esta reducción del espacio de claves
  - Dificultad para diseñar chips capaces de operar de forma eficiente (consumo y coste) con una clave de 128 bits en esa década de los 70
  - Política de seguridad interna para proteger información sensible ante ataques externos y poder realizar fuerza bruta en un tiempo razonable



# Clave real del DES y el código hexadecimal

- ASCII American Standard Code for Information Interchange, se crea en 1963 por ASA American Standards Association, hoy ANSI American National Standards Institute
- Tiene 7 bits para representar 128 caracteres, dejando el octavo bit de cada byte como bit de paridad para detectar errores de transmisión
- ASCII fue publicado como estándar en 1967
- Actualmente usamos como estándar ISO/IEC 8859-1, también conocido como ASCII extendido, de 8 bits sin bit de paridad (IBM, 1981)
- Pero... el código hexadecimal fue introducido en computación por IBM en 1963 y no tiene código de paridad... ¿Por qué no se usó en DES?

# Redes de Feistel



- Puesto que  $A \oplus X_i \oplus X_i = A$  y  $B \oplus X_i \oplus X_i = B$  por la característica involutiva del xor, haciendo ahora el recorrido de abajo hacia arriba, para descifrar, se recupera el texto A y B
- Si la función  $f$  (difícil de invertir) se aplicase sólo a una mitad del texto en claro (A o B), el modelo sigue siendo válido

# Ejemplo de red de Feistel con cifra clásica

Sustitución:  $c = m + 1 \bmod 27$  Permutación:  $\Pi_{3241}$



STAR WARS, LA MISIÓN CONTINÚA

	M = STAR WARS LAMI SIÓN CONT INÚA					
$M_1$	STAR	WARS	LAMI	SION	CONT	INUA
$S_1$	TUBS	WARS	MBNJ	SION	DPÑU	INUA
$P_1$	BUST	WARS	NBJM	SION	ÑPUD	INUA
$M_2$	WARS	BUST	SION	NBJM	INUA	ÑPUD
$S_2$	XBST	BUST	TJPÑ	NBJM	JÑVB	ÑPUD
$P_2$	SBTX	BUST	PJÑT	NBJM	VÑBJ	ÑPUD

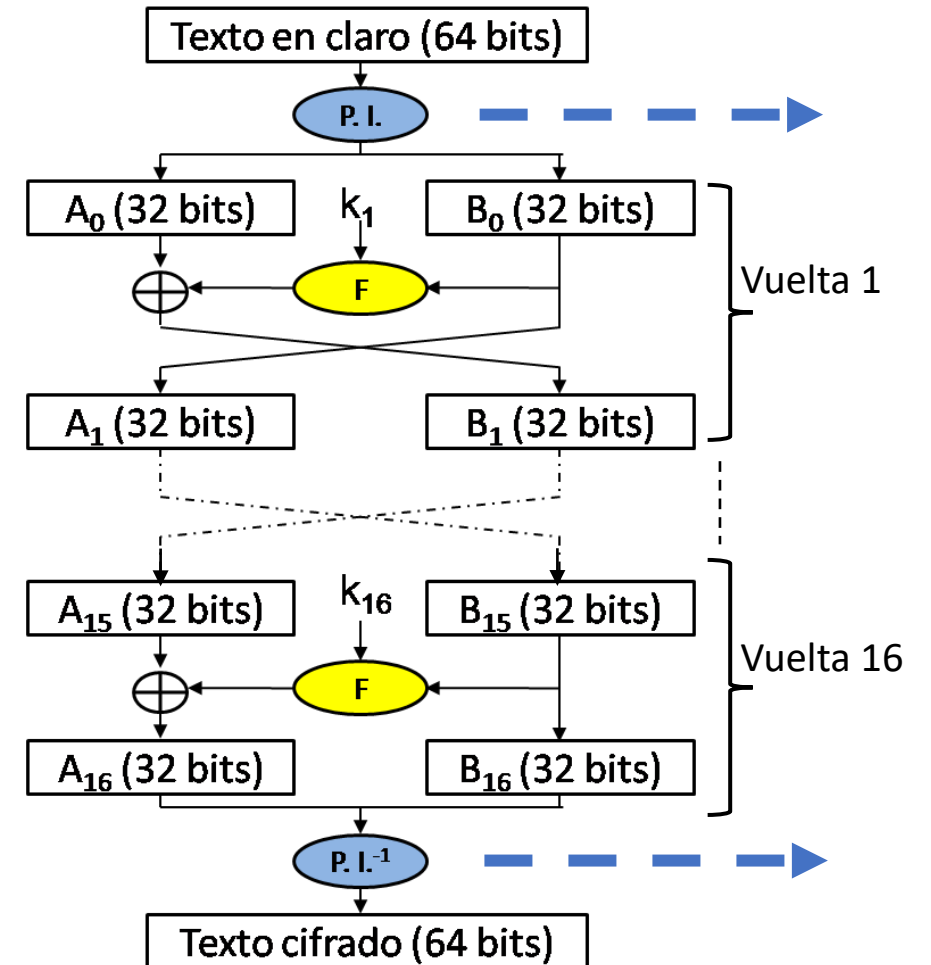
Primera vuelta

Segunda vuelta

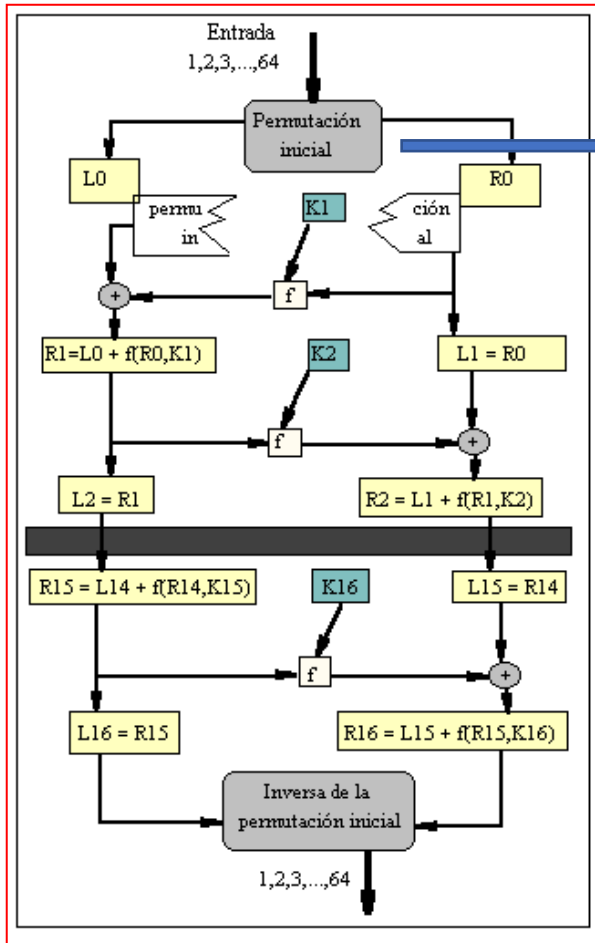
- $C = \text{SBTXB USTPJ } \tilde{\text{NTNBJ}} \text{ MVÑBJ } \tilde{\text{NPUD}}$
- El algoritmo DES hará exactamente lo mismo, añadiendo una operación xor antes de intercambiar los bloques A y B en las vueltas siguientes

# Red de Feistel en el DES

- Cifra bloques de texto en claro de 64 bits
- Usa una clave de 64 bits, que se ve reducida a 56 bits por el bit de paridad
- Usa 16 vueltas con claves  $k_1$  a  $k_{16}$  para la operación de cifrado
- Como se fuerza a que  $k_{16} = k_1$  entonces para el descifrado se recorre el algoritmo en sentido inverso, desde  $k_{16}$  a  $k_1$
- Para poder realizar las operaciones xor, se usarán permutaciones con expansión y permutaciones con compresión, con en fin de igualar la cantidad de bits de cada operando
- La seguridad del DES reside en la función no lineal  $f$  basada en las cajas  $S$  (sustitución)



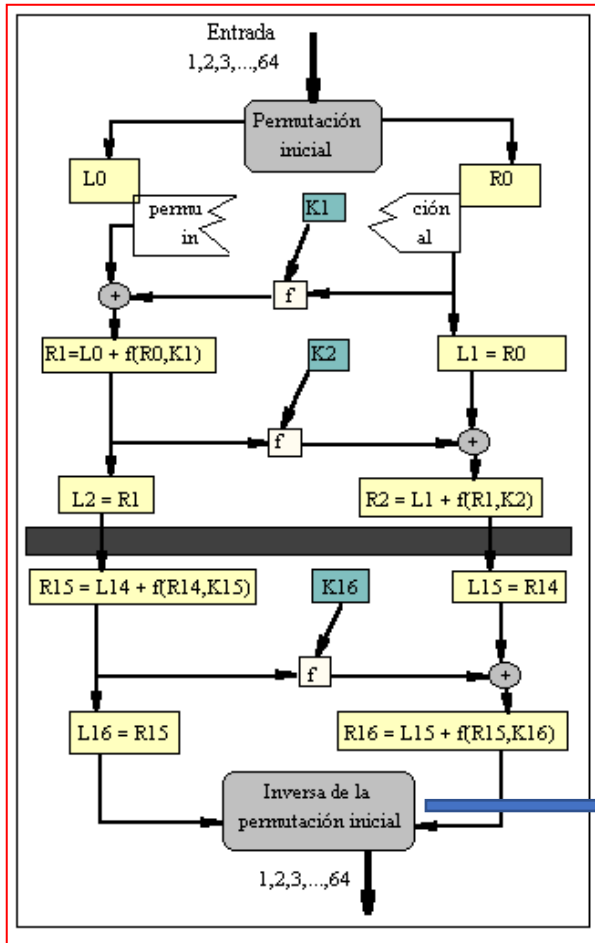
# Tabla IP Initial Permutation



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

- El bit 1 se envía a la posición 40 de la matriz. Operación sin interés criptográfico: separar bits contiguos del texto

# Tabla $IP^{-1}$ Inverse Initial Permutation



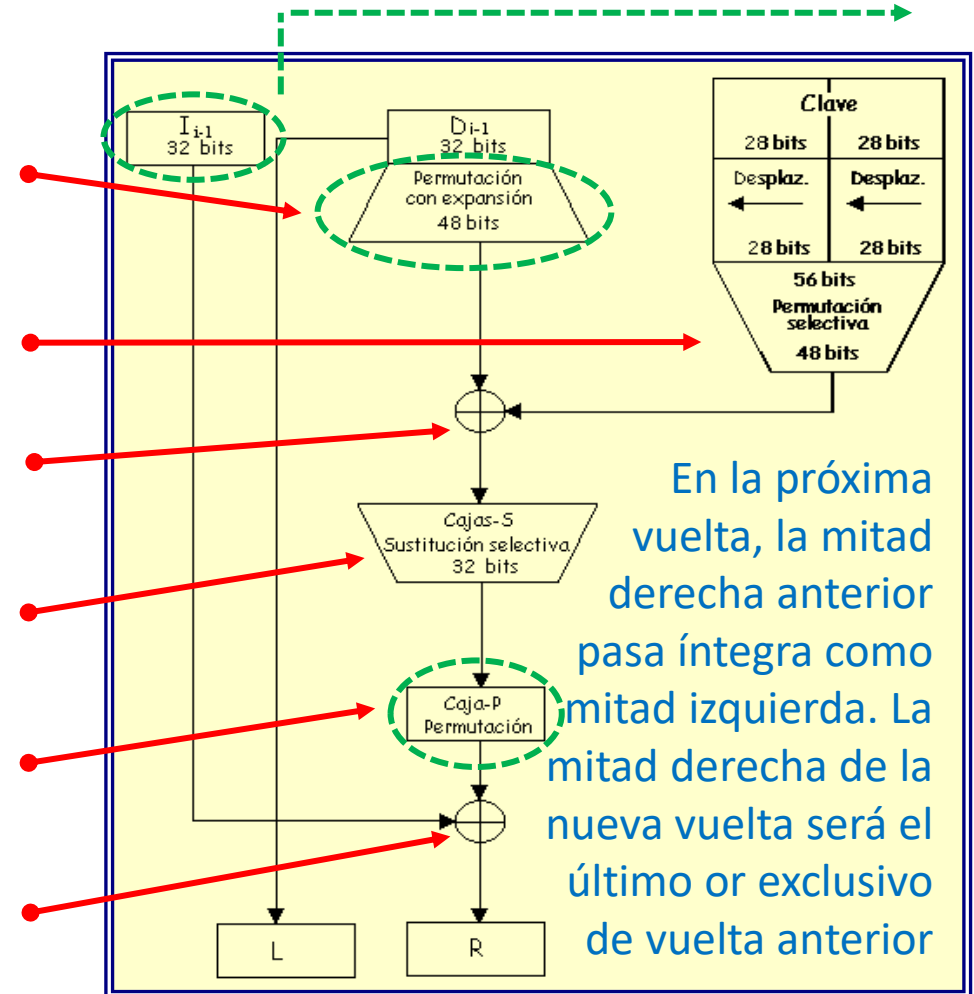
- El bit en la posición 40 de la tabla, vuelve a la posición 1, al igual que los restantes 63 bits de la matriz

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



# Operaciones en cada vuelta

- Se permuta la mitad derecha del texto en claro  $R_i$  con expansión a 48 bits
- La clave real de 56 bits se desplaza y permuta, seleccionando los 48 bits de  $K_i$  de cada una de las 16 vueltas
- La nueva mitad derecha  $R_i$  y la clave  $K_i$  se suman or exclusivo
- Se reducen los 48 bits de entrada a 32 bits de salida mediante las Cajas S
- Se permuta el resultado bit a bit sin modificar el tamaño de la matriz
- El resultado anterior se suma or exclusivo con la mitad izquierda  $L_i$



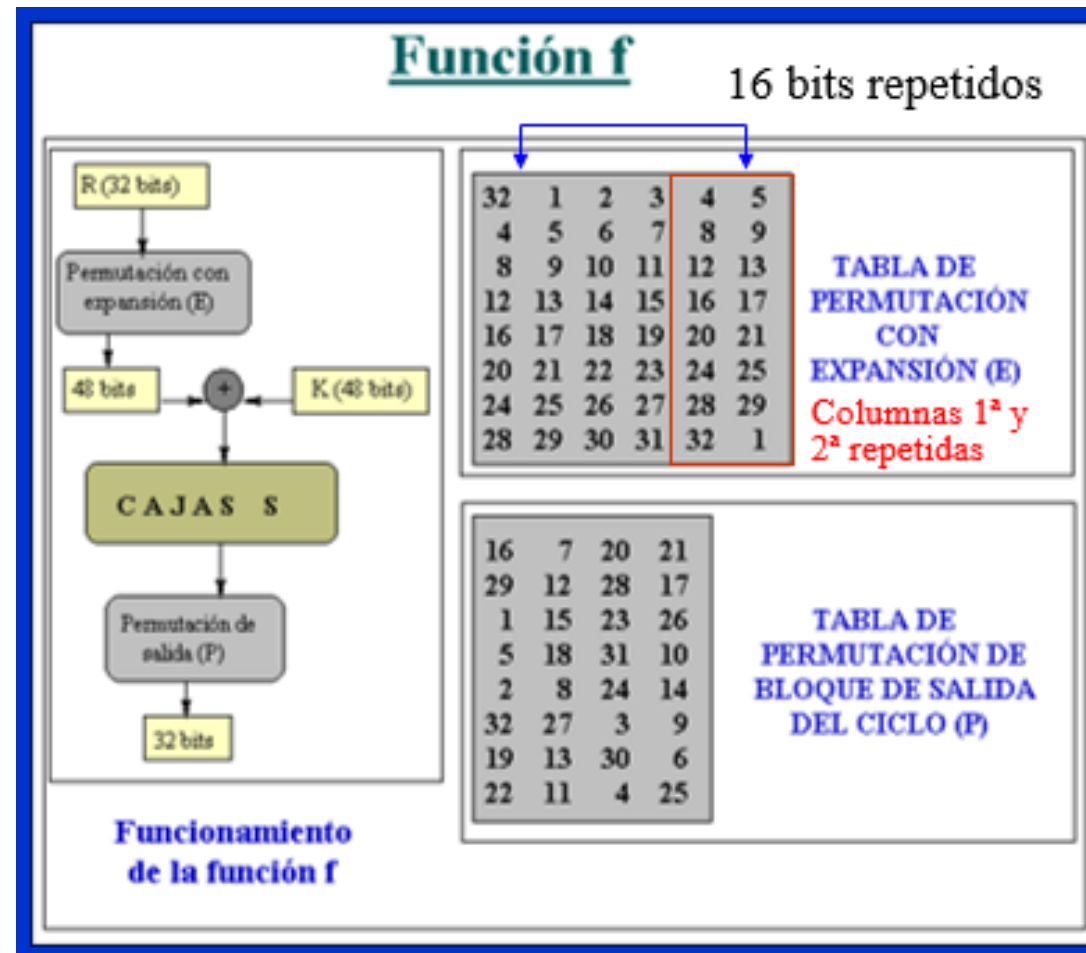
# Bloques de texto izquierdo y derecho

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

$L_0 = 58\ 50\ 42\ 34\ 26\ 18\ 10\ 02\ 60\ 52$   
 $44\ 36\ 28\ 20\ 12\ 04\ 62\ 54\ 46\ 38\ 30$   
 $22\ 14\ 06\ 64\ 56\ 48\ 40\ 32\ 24\ 16\ 08$

$R_0 = 57\ 49\ 41\ 33\ 25\ 17\ 09\ 01\ 59\ 51$   
 $43\ 35\ 27\ 19\ 11\ 03\ 61\ 53\ 45\ 37\ 29$   
 $21\ 13\ 05\ 63\ 55\ 47\ 39\ 31\ 23\ 15\ 07$

# Permutación con expansión E y Tabla P



# Cajas $S_1$ y $S_2$ del DES

COLUMNAS

$S_1$     F I L A S		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

COLUMNAS

$S_2$     F I L A S		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

# Cajas $S_3$ y $S_4$ del DES

COLUMNAS

$S_3$   F I L A S		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

COLUMNAS

$S_4$   F I L A S		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	7	13	14	3	0	6	9	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

# Cajas $S_5$ y $S_6$ del DES

		COLUMNAS															
$S_5$		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

		COLUMNAS															
$S_6$		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

# Cajas $S_7$ y $S_8$ del DES

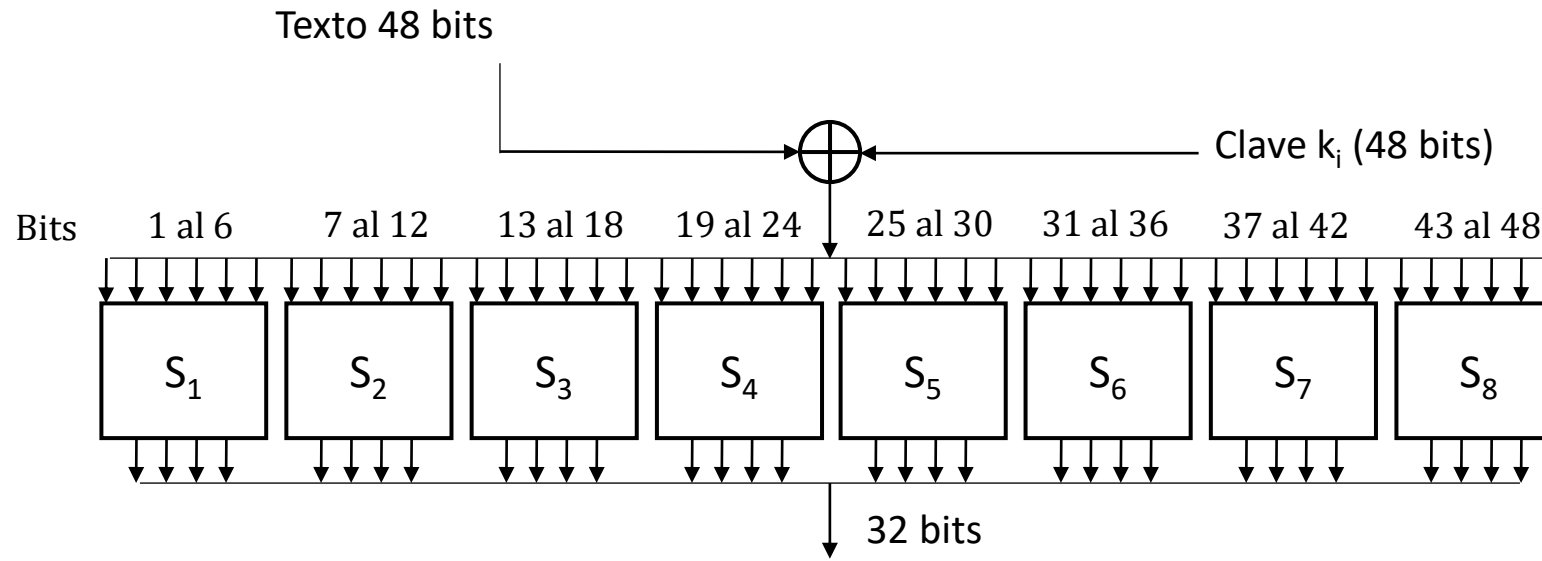
COLUMNAS

$S_7$     F I L A S		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

COLUMNAS

$S_8$     F I L A S		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

# Operaciones con las cajas S



- Cada caja tiene 16 columnas y 4 filas. Cada celda de una fila tiene números distintos del 0 al 15
- En cada Caja-S se obtienen 4 bits de salida por cada 6 bits de entrada
- Se trata de una operación no lineal y unidireccional, ya que habrá cuatro soluciones posibles de entrada para cada una de las salidas
- Como hay 8 cajas S y tenemos 16 vueltas, en cada bloque de cifra, hacer el camino de vuelta significaría  $(4^8)^{16} = 2^{256}$  operaciones, muchísimo más grande que las  $2^{56}$  de un ataque por fuerza bruta a la clave



# Ejemplo de operación de una cajas S

- Sean **101100** los bits 7 al 12 de la cadena de 48 bits
- Por lo tanto debemos leer en la caja  $S_2$



COLUMNAS

$S_2$		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
I	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
L	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
A	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S																	

- Se selecciona la fila de la caja con los bits extremos: **10**<sub>2</sub> = 2
- Los cuatro bits centrales sirven para determinar la columna: **0110**<sub>2</sub> = 6
- La celda intersección entre la fila 2 y la columna 6 de la caja  $S_2$  es el número de salida de cuatro bits buscado: 13 = 1101 = 0xD

# Fortaleza de las cajas S

- Aunque se especuló mucho sobre la posible intervención de la NSA en el diseño de las cajas S, se ha demostrado (Biham – Shamir) que no fue así
- Son funciones no lineales, donde es muy fácil pasar de 48 bits a 32 bits, pero computacionalmente es muy difícil hacer el recorrido inverso
- Para cada salida en una caja S, existen 4 entradas posibles. En el ejemplo anterior, las 4 entradas para la misma salida 13 en la caja  $S_2$  son:
  - Fila 0 Columna 11 Bits de entrada **010110** Salida 13 = 1101
  - Fila 1, Columna 1 Bits de entrada **000011** Salida 13 = 1101
  - Fila 2, Columna 6 Bits de entrada **101100** Salida 13 = 1101
  - Fila 3, Columna 0 Bits de entrada **100001** Salida 13 = 1101
- Romper las 8 cajas S en 16 vueltas de un bloque, significa  $2^{256}$  cálculos. Es más fácil romper la clave del DES por fuerza bruta con solo  $2^{56}$  cálculos

# Más información en píldoras Thoth



<https://www.youtube.com/watch?v=XwUOwqSHzyo>

# Conclusiones de la lección 8.2

- El algoritmo DES fue el estándar de cifra simétrica en bloque por el NIST desde 1976 hasta 1999, cuando sucumbe ante del DES Challenge
- Es una variante del algoritmo Lucifer, del grupo de investigación de IBM liderado por Horst Feistel
- La NSA bajó el tamaño del bloque de 128 a 64 bits y la clave de 128 a 64 bits, que se queda en 56 bits reales al usar bytes en que el último bit es de paridad
- Usa una red de Feistel para la cifra, mezclando con la clave en cada una de sus 16 vueltas a la mitad del texto en claro, que además va cambiando de posición
- Utiliza operaciones de permutación con expansión y permutación con compresión para adaptar el texto de entrada y la clave de cada vuelta a las cajas S de sustitución, que es donde reside la seguridad del algoritmo

# Lectura recomendada

- The DES Algorithm Illustrated, J. Orlin Grabbe
  - <http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm>
- Guion píldora formativa Thoth nº 28, ¿Cómo funcionan los algoritmos DES y 3DES?, proyecto Thoth, Jorge Ramió, 2015
  - <https://www.criptored.es/thoth/material/texto/pildora028.pdf>
- CLCRIPT 00: Códigos y tablas de uso frecuente en criptografía (cajas S y códigos ASCII)
  - [https://www.criptored.es/descarga/Codigos\\_y\\_tablas\\_de\\_uso\\_frecuente\\_en\\_criptografia.pdf](https://www.criptored.es/descarga/Codigos_y_tablas_de_uso_frecuente_en_criptografia.pdf)

# Class4crypt c4c8.3

## Módulo 8. Criptografía simétrica en bloque

### Lección 8.3. Algoritmo DES: expansión de clave, cifra y rellenos

8.3.1. Los bits de paridad en la clave del DES

8.3.2. Generación de las 16 subclaves de ronda  $k_1$  a  $k_{16}$  en cifrado y descifrado

8.3.3. Operaciones de cifrado y descifrado

8.3.4. Relleno zero padding

8.3.5. Claves débiles y semidébiles

Class4crypt c4c8.3 Algoritmo DES: expansión de clave, cifra y rellenos  
<https://www.youtube.com/watch?v=lubHEnuupis>

# Los bits de paridad en el algoritmo DES

- DES elimina el octavo bit de cada byte de la clave de 64 bits, al suponer que se trata de un bit de paridad: quita los bits 8, 16, 24, 32, 40, 48, 56 y 64
- Quita 8 bits en total y la clave real se queda en 56 bits. Por lo tanto, el espacio de claves real del algoritmo DES es igual a  $2^{56} = 72.057.594.037.927.936$

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

- Esto lo hace con la tabla de PC-1 de permutación y selección
- Los 56 bits de la clave del DES y que se dividirán en dos mitades, izquierda y derecha de 28 bits cada una, para generar las 16 subclaves  $k_1$  a  $k_{16}$ , una para cada vuelta

# Tabla PC-1 Permuted Choice en clave DES

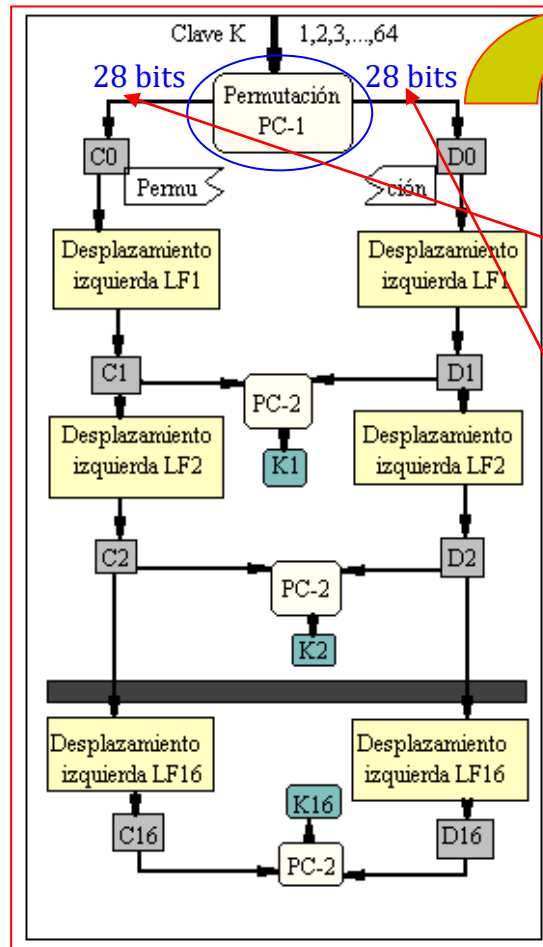


Tabla PC-1 (56 bits)

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

Desaparecen los 8 bits de paridad:

8, 16, 24, 32, 40, 48, 56 y 64



# Tabla PC-2 Permuted Choice en clave DES

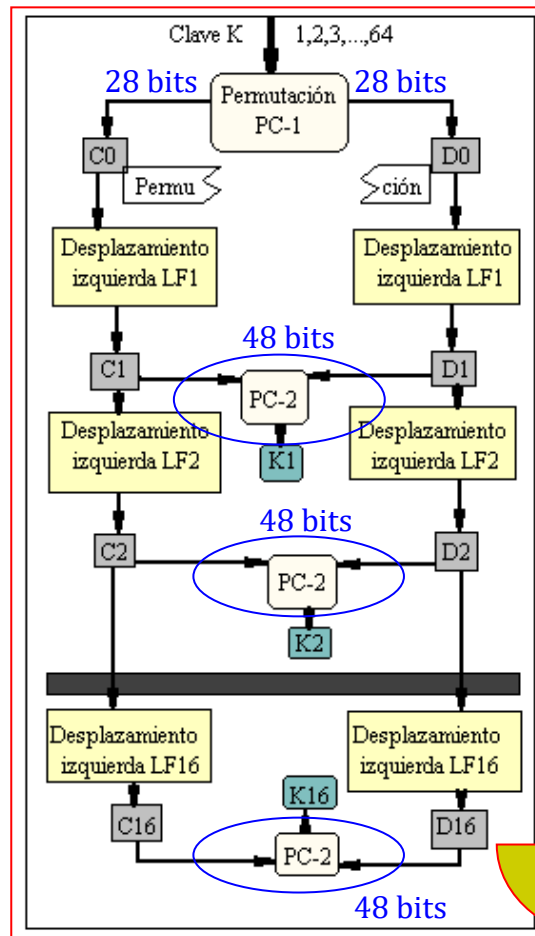
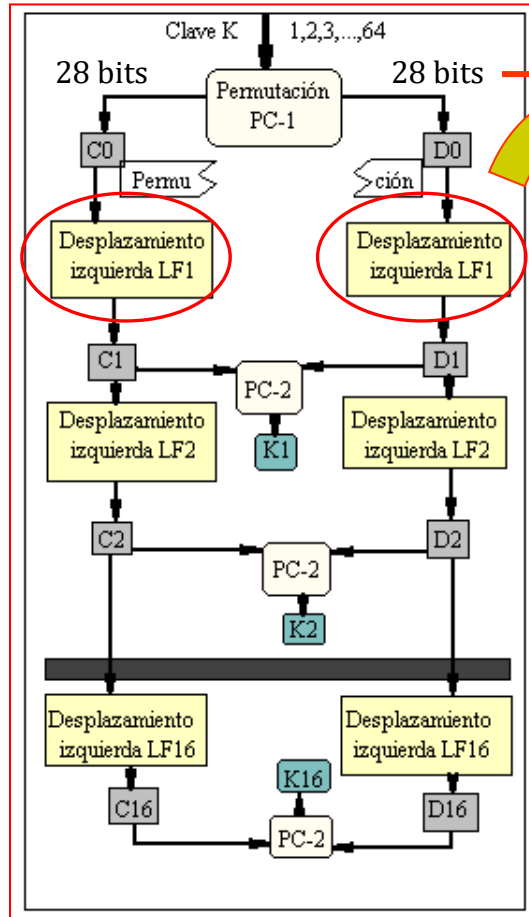


Tabla PC-2 (48 bits)  $\Rightarrow k_1, k_2, \dots k_{15}, k_{16}$

4	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 han eliminado 8 bits:  
9, 18, 22, 25, 35, 38, 43 y 54

# Desplazamientos para cálculo de subclaves



Se produce un desplazamiento total igual a 28, todos los bits de cada bloque  $C_i$  (izquierda) y  $D_i$  (derecha)

Left Shift:  $LF_1, LF_2, \dots, LF_{15}, LF_{16}$

Vuelta $i$	Bits Despl. Izda.
$k_1$	-1
$k_2$	-1
$k_3$	-2
$k_4$	-2
$k_5$	-2
$k_6$	-2
$k_7$	-2
$k_8$	-2

Vuelta $i$	Bits Despl. Izda.
$k_9$	-1
$k_{10}$	-2
$k_{11}$	-2
$k_{12}$	-2
$k_{13}$	-2
$k_{14}$	-2
$k_{15}$	-2
$k_{16}$	-1

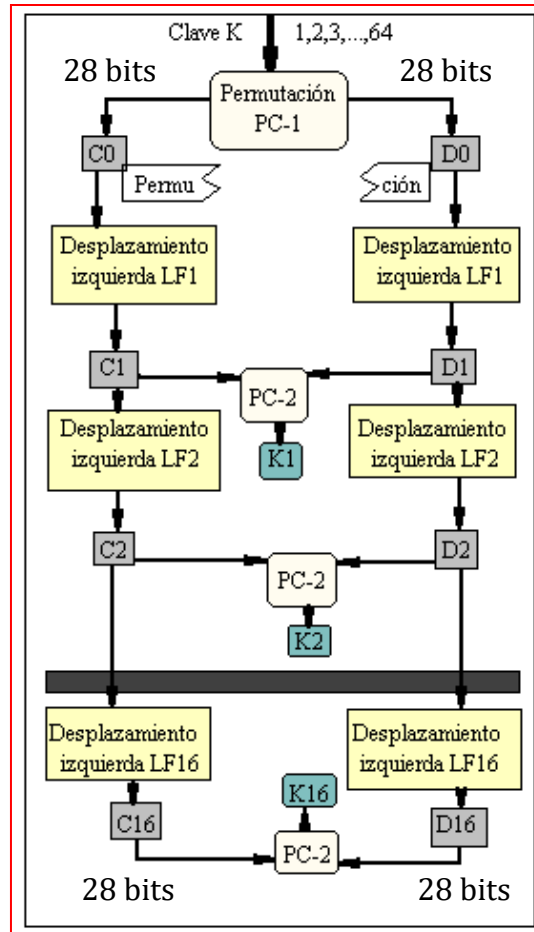
Como en la cifra se ha aplicado un desplazamiento de 28 bits en cada uno de los dos bloques de la clave:

- $D_{16} = D_0$
- $C_{16} = C_0$

$\Sigma = 28$

# Descifrado en DES con claves $k_{16}$ a $k_1$

Sabemos que  $C_{16} = C_0$  y  $D_{16} = D_0$ . Ahora se desplaza hacia la derecha y en sentido inverso para volver desde  $k_{16}$  a  $k_1$



Claves para descifrado

Right Shift:  $RH_1, RH_2, \dots, RH_{15}, RH_{16}$

Vuelta i	Bits Despl. Dcha.
$k_{16}$	+1
$k_{15}$	+2
$k_{14}$	+2
$k_{13}$	+2
$k_{12}$	+2
$k_{11}$	+2
$k_{10}$	+2
$k_9$	+1

Vuelta i	Bits Despl. Dcha.
$k_8$	+2
$k_7$	+2
$k_6$	+2
$k_5$	+2
$k_4$	+2
$k_3$	+2
$k_2$	+1
$k_1$	+1



Charles Perrault

# Cifrado, descifrado y rellenos en DES

- En DES se cifran bloques de 8 bytes, 64 bits
- Si el último bloque de texto en claro no tiene 8 bytes, se rellenará con tantos ceros como sea necesario
- Este relleno se conoce como Zero padding (hexadecimal)
- Por ejemplo  $M_1$  = Buenas tardes serán dos bloques a cifrar con 3 bytes de relleno y  $M_2$  = ¡Eureka! un solo bloque sin relleno:
  - $M_1$  = 42 75 65 6E 61 73 20 74    61 72 64 65 73 00 00 00
  - $M_2$  = A1 45 75 72 65 6B 61 21

# Claves débiles y semidébiles

- Por teoría de la información, si una cifra es  $C = E_k(M)$ , donde  $E_k$  significa cifrar con la clave  $k$ , el único descifrado válido será  $M = D_k(C)$ , donde  $D_k$  significa descifrar con la clave  $k$
- $M = D_k(E_k(M))$  Cualquier otra opción, es una **Solución Falsa**
- Una clave es débil si se verifica que  $M = E_k(E_k(M))$ , es decir, se recupera el texto en claro volviendo a cifrar el criptograma
- Una clave es semidébil si se verifica que  $M = E_{k_y}(E_{k_x}(M))$ , es decir, se recupera el texto en claro si se cifra el criptograma con una clave  $k$  diferente a la usada en la primera cifra

# Claves débiles en DES

- En DES habrá 6 claves débiles:  $M = E_k(E_k(M))$
- Los bloques C y D de 28 bits tienen todo ceros o todo unos
  - Caso 1a: 0x 00000000000000000000 Caso 1b: 0x FFFFFFFF0000000000000000
  - Caso 2a: 0x 01010101010101010101 Caso 2b: 0x FEFEFEFEFEFEFEFEFEF
  - Caso 3a: 0x E0E0E0E0F1F1F1F1F1F1 Caso 3b: 0x 1F1F1F1F0E0E0E0E0E0E
- Los bits de C y D en 1a serán todo 0s y en 1b serán todo 1s
- Lo mismo sucede en 2a y 2b al eliminar el bit de paridad
- Por la permutación PC-1, en 3a la cadena C serán 1s y la cadena D serán 0s, y en 3b la cadena C serán 0s y la cadena D serán 1s

# Claves semidébiles en DES

- En DES habrá 6 claves semidébiles:  $M = E_{k_y}(E_{k_x}(M))$
- Estas son las 6 claves  $k_x$  y  $k_y$  semidébiles
  - $k_x = 0x\ 01FE01FE01FE01FE \Rightarrow k_y = 0x\ FE01FE01FE01FE01$
  - $k_x = 0x\ 1FE01FE00EF10EF1 \Rightarrow k_y = 0x\ E01FE01FF10EF10E$
  - $k_x = 0x\ 01E001E001F101F1 \Rightarrow k_y = 0x\ E001E001F101F101$
  - $k_x = 0x\ 1FFE1FFE0EFE0EFE \Rightarrow k_y = 0x\ FE1FFE1FFE0EFE0E$
  - $k_x = 0x\ 011F011F010E010E \Rightarrow k_y = 0x\ 1F011F010E010E01$
  - $k_x = 0x\ E0FEE0FEF1FEF1FE \Rightarrow k_y = 0x\ FEE0FEE0FEF1FEF1$
- Siempre vienen en parejas x, y

# Prácticas con software safeDES



- Cifrado y descifrado de textos en modo ECB
- Observación de rellenos
- Cifrado y descifrado de archivos en modo ECB
- Comprobación de claves débiles y semidébiles



# Más información en píldoras Thoth



<https://www.youtube.com/watch?v=XwUOwqSHzyo>

# Conclusiones de la lección 8.3

- Expansión de clave: una vez se han eliminado los (supuestos) bits de paridad, los 56 bits de la clave se dividen en dos mitades C y D de 28 bits cada una
- Se aplican desplazamientos hacia la izquierda a cada cadena de 28 bits C y D para obtener las claves  $k_1$  a  $k_{16}$  necesarias para cada una de las 16 vueltas del DES. Como esos desplazamientos suman 28, entonces las cadenas de bits  $C_0$  y  $C_{16}$  así como  $D_0$  y  $D_{16}$  estarán en fase, serán las mismas
- Por lo tanto, en el descifrado de una red Feistel (hacer el recorrido inverso), las claves  $k_{16}$  a  $k_1$  se obtienen realizando ahora los desplazamientos hacia la derecha: cuento Pulgarcito y sus guijarros (migas de pan) para volver a casa
- El relleno en DES es del tipo Zero padding
- Existen 6 claves débiles y 6 claves semidébiles en DES

# Lectura recomendada

- The DES Algorithm Illustrated, J. Orlin Grabbe
  - <http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm>
- Key Expansion Function and Key Schedule of DES (Data Encryption Standard) Algorithm, Ritul (2019)
  - <https://medium.com/@artistritul1995/key-expansion-function-and-key-schedule-of-des-data-encryption-standard-algorithm-1bfc7476157>
- Padding (cryptography), Wikipedia
  - [https://en.wikipedia.org/wiki/Padding\\_\(cryptography\)](https://en.wikipedia.org/wiki/Padding_(cryptography))
- Guion píldora formativa Thoth nº 28, ¿Cómo funcionan los algoritmos DES y 3DES?, proyecto Thoth, Jorge Ramió, 2015
  - <https://www.criptored.es/thoth/material/texto/pildora028.pdf>
- ASCII Text to Hex Code Converter, Rapid Tables
  - <https://www.rapidtables.com/convert/number/ascii-to-hex.html>

# Class4crypt c4c8.4a

## Módulo 8. Criptografía simétrica en bloque

### Lección 8.4a. ECB y CBC, modos de cifra con confidencialidad

8.4a.1 Necesidad de los modos de cifra en bloque

8.4a.2. Modos de cifra aprobados por el NIST

8.4a.3. Modo de cifra ECB Electronic codebook

8.4a.4 Modo de cifra CBC Cipher block chaining

Class4crypt c4c8.4a ECB y CBC, modos de cifra con confidencialidad  
[https://www.youtube.com/watch?v=8LvA\\_P-zR\\_o](https://www.youtube.com/watch?v=8LvA_P-zR_o)

# Necesidad de los modos de cifra en bloque

- Como el mensaje  $M$  se divide en bloques  $M_1, M_2, \dots, M_{n-1}, M_n$ , el criptograma es la concatenación  $C = C_1 + C_2 + \dots + C_{n-1} + C_n$
- Pero no se debe hacer una cifra de bloques independientes de esta manera, porque ello permitiría realizar ataques del tipo:
  - **Inicios y finales iguales.** Si se cifran dos mensajes diferentes con la misma clave, con inicios y/o finales iguales del texto en claro, el conocimiento del texto en claro del primer criptograma por parte del atacante, le permitiría deducir partes del segundo criptograma
  - **Reenvío de bloques.** Como los bloques  $C_i$  son independientes, un atacante podría retener una cifra, cambiar bloques y reenviarlos

# Modos de cifra aprobados por el NIST

- *Currently, NIST has approved fourteen modes of the approved block ciphers in a series of special publications.*
- *As summarized on the Current Modes page, there are:*
  - *Eight confidentiality modes (ECB, CBC, OFB, CFB, CTR, XTS-AES, FF1, and FF3)*
  - *One authentication mode (CMAC)*
  - *Five combined modes for confidentiality and authentication (CCM, GCM, KW, KWP, and TKW)*
- *Ref: Block Cipher Modes, Computer Security Resource Center, NIST*
  - <https://csrc.nist.gov/Projects/block-cipher-techniques/BCM>

# Modos de cifra bloque con confidencialidad

- Modos de cifra en bloque con confidencialidad a estudiar:
  - ECB Electronic codebook ✓
  - CBC Cipher block chaining ✓
  - CFB Cipher feedback
  - OFB Output feedback
  - CTR Counter
- No veremos aquí estos otros modos
  - XTS-AES, FF1, FF3
  - Tampoco veremos PCBC Propagating cipher block chaining (que se usa entre otros en Kerberos v4 y WASTE) no incluido por el NIST

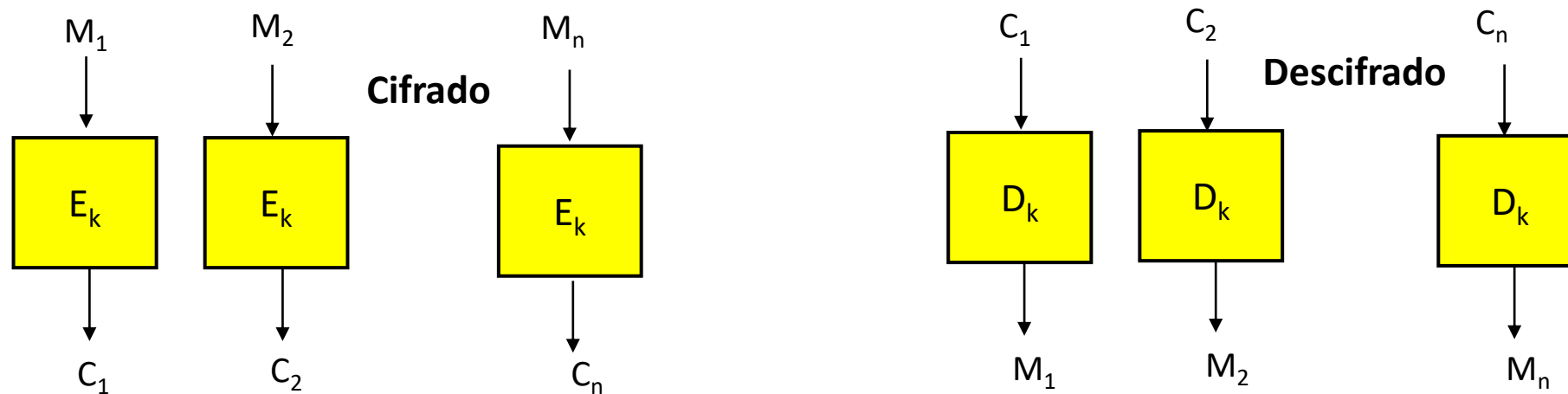
# 1) Modo ECB

- El modo ECB Electronic codebook o Libro electrónico de códigos, cifrará cada bloque con la clave  $k$  de forma independiente
- Por lo tanto, el resultado es como si se codificase mediante un gran libro de códigos
- Recuerda que codificar no es lo mismo que cifrar...
- Se podría reconstruir ese gran libro electrónico de códigos, sin necesidad de conocer la clave
- Se aplicará relleno a  $b$  bits en último bloque





# Esquema modo ECB en cifrado y descifrado



- Para romper la cifra, se podría reconstruir ese gran libro electrónico de códigos, sin necesidad de conocer la clave
- Y para textos muy formateados, como sería por ejemplo una imagen, la cifra en modo ECB no oculta el perfil de esa imagen

# Ventajas y desventajas del modo ECB

- Ventajas

- La cifra es muy simple y se puede realizar cifrado y descifrado en paralelo
- Un error se propaga solamente dentro del bloque en el que éste ocurre

- Desventajas

- Mensajes con inicios y finales iguales y cifrados con la misma clave, darán lugar al mismo criptograma, lo que entrega información útil al atacante
- Para textos muy formateados, como una imagen, la cifra no oculta su perfil
- Y, al menos en teoría, se podría vulnerar la integridad de un mensaje, capturando dos criptogramas donde se haya usado la misma clave de cifra, reorganizando los bloques de texto cifrado y el reenviando al destinatario uno o más criptogramas diferentes de otros supuestos textos en claro

# Inicios y finales iguales en modo ECB

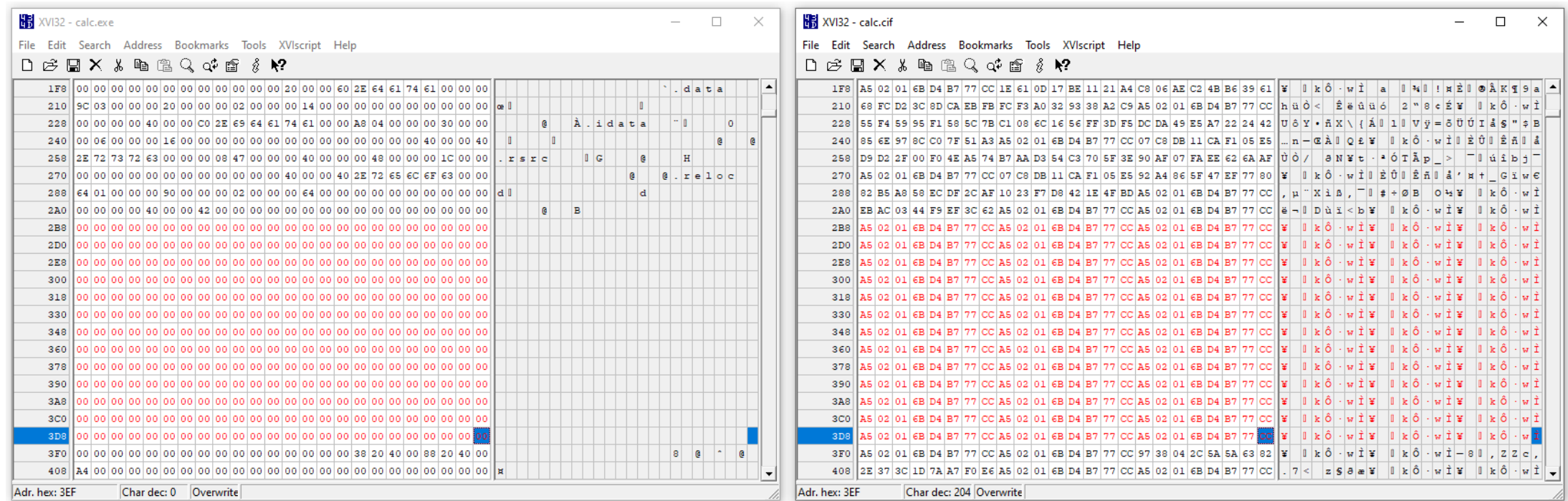
- Sea la clave  $K = 0x \text{ FABADAACABADA007}$
- Si el texto en claro es  $M_1 = \text{Misil V2 Informe secreto del Alto Mando ...}$
- El criptograma será  $C_1 = \text{D1A4012BAC837852 87041908AD9202F5 ...}$
- Si ahora el texto en claro es  $M_2 = \text{Misil V2 Nuevos datos de diseño mejorado ...}$
- El nuevo criptograma será  $C_2 = \text{D1A4012BAC837852 354366AE6DFFF489 ...}$
- Si se había roto la cifra de  $C_1$  y en la cifra de  $M_2$  (un cifrado posterior) se ha usado la misma clave  $K$ , entonces el enemigo al observar el criptograma  $C_2$  puede sospechar (y está en lo cierto) que el nuevo criptograma trata sobre el misil V2 porque el primer bloque de ambos criptogramas es el mismo
- Este conocimiento previo de un secreto no está permitido en la criptografía

# Reenvío de bloques falsos en modo ECB

- Se cifra con clave  $K = 0x\text{FF007FABADA007FF}$  ambos textos  $M_1$  y  $M_2$ 
  - $M_1$  = Ingresar 100.000 euros en cuenta ES32007
  - $C_1 = 8C449BC618F92AD9\ E191554EEF4BB810\ 2107623A0CF2B45F$   
 $62D2BB2E4B2933FE\ 30061A2A44FD85D3\ E87AF6FBCEAB3F82$
  - $M_2$  = Ingresar 999.999 euros en cuenta ES32000
  - $C_2 = 8C449BC618F92AD9\ 056B628F9840FEC0\ 2107623A0CF2B45F$   
 $62D2BB2E4B2933FE\ 30061A2A44FD85D3\ 5E619C7911CA5E5F$
- Un atacante, que conoce que  $M_1$  y  $M_2$  han sido cifrados con la misma clave  $K$ , podría por ejemplo cambiar bloques ya cifrados y asignar los 999.999 euros a la cuenta terminada en 007 enviando al banco el siguiente criptograma
  - $C_3 = 8C449BC618F92AD9\ 056B628F9840FEC0\ 2107623A0CF2B45F$   
 $62D2BB2E4B2933FE\ 30061A2A44FD85D3\ E87AF6FBCEAB3F82$
  - $M_3$  = Ingresar 999.999 euros en cuenta ES32007 (que es un mensaje falso)

# Modo ECB en archivos muy formateados

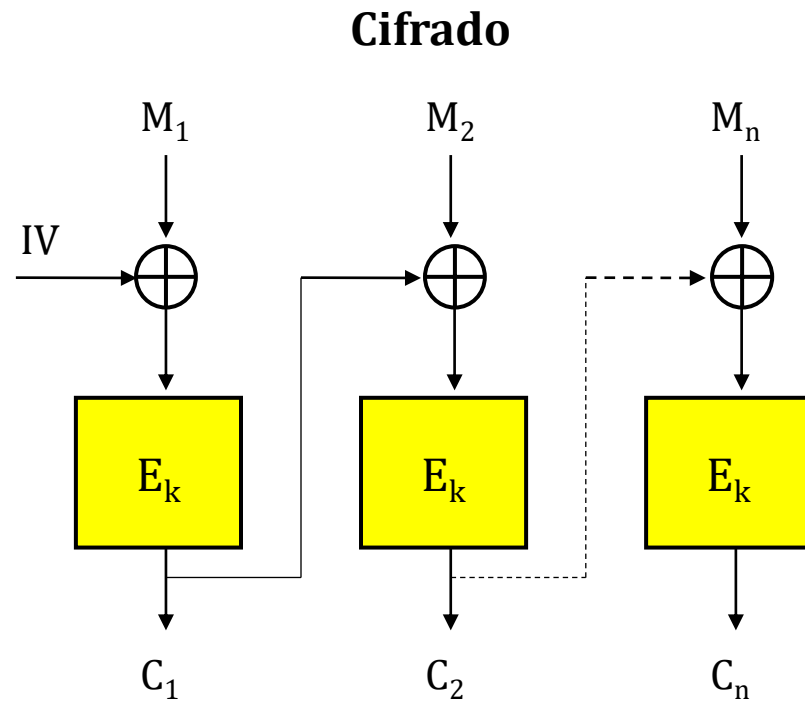
- Cifra DES ECB de la calculadora de Windows con K = 0x 1234567890ABCDEF



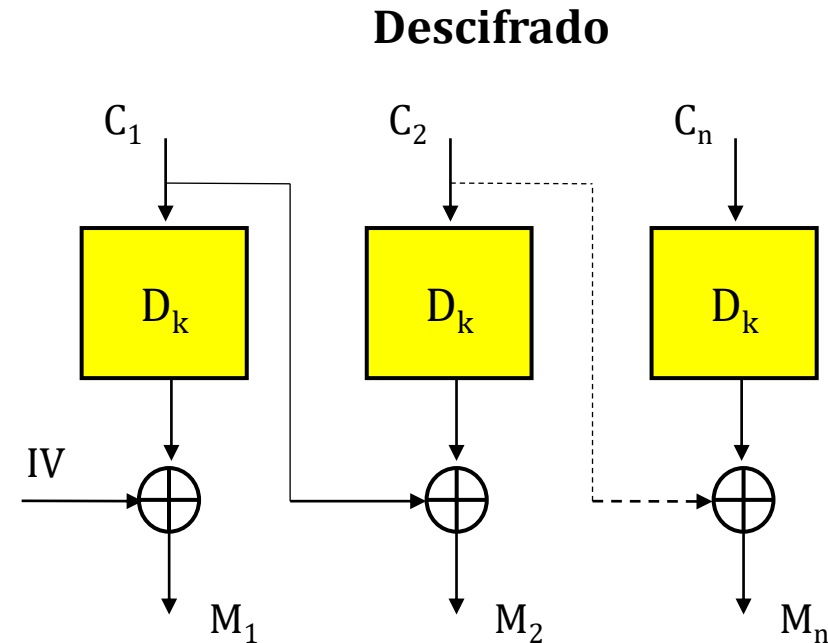
## 2) Modo CBC

- El modo CBC Cipher block chaining o Encadenamiento de bloques cifrados, fue muy utilizado en SSL/TLS hasta el año 2015
- Se realiza una operación xor entre el bloque del texto en claro y un bloque de  $b$  bits de igual tamaño que el bloque de texto, que actuará como una segunda clave y que no es necesario que sea secreta
- Para el primer bloque se usa un vector de inicialización IV
- En los siguientes bloques se usa como vector IV el criptograma anterior
- El vector IV es aleatorio, lo asigna el software del programa y por ello depende del tiempo; actúa como sello de tiempo. La probabilidad que dos vectores IV sean iguales es mínima ( $1/2^{64}$  en DES y  $1/2^{128}$  en AES)
- Se aplicará relleno a  $b$  bits en último bloque

# Esquema modo CBC en cifrado y descifrado



- El vector IV debe ser del mismo tamaño que el bloque de texto en claro
- $C_1 = E_k(M_1 \text{ xor } IV)$ ,  $C_{i>1} = E_k(M_i \text{ xor } C_{i-1})$



- Se descifra con IV y luego usando los criptogramas recibidos  $C_1, C_2, \dots$
- $M_1 = D_k(C_1) \text{ xor } IV$ ,  $M_{i>1} = D_k(C_i) \text{ xor } C_{i-1}$

# Ventajas y desventajas del modo CBC

- Ventajas

- Se puede descifrar más de un bloque usando los criptogramas anteriores
- Si el vector IV es aleatorio y cambia en cada cifra, al cifrar dos veces el mismo documento con la misma clave, obtenemos criptogramas distintos
- La cifra de archivos muy formateados, como por ejemplo una imagen, no mostrará su perfil
- No será posible sustituir un bloque en el criptograma sin que se detecte

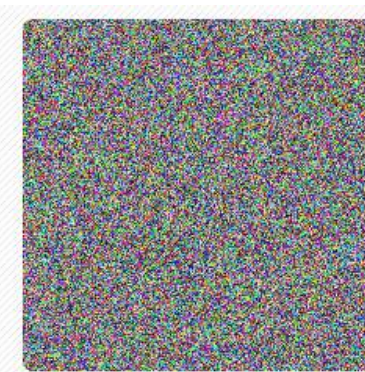
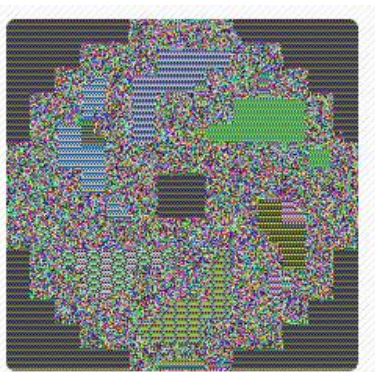
- Desventajas

- Un error se propaga en todo ese bloque y en los siguientes
- No es posible realizar una cifra simultánea de varios bloques pues la cifra desde el segundo bloque de texto depende del criptograma anterior

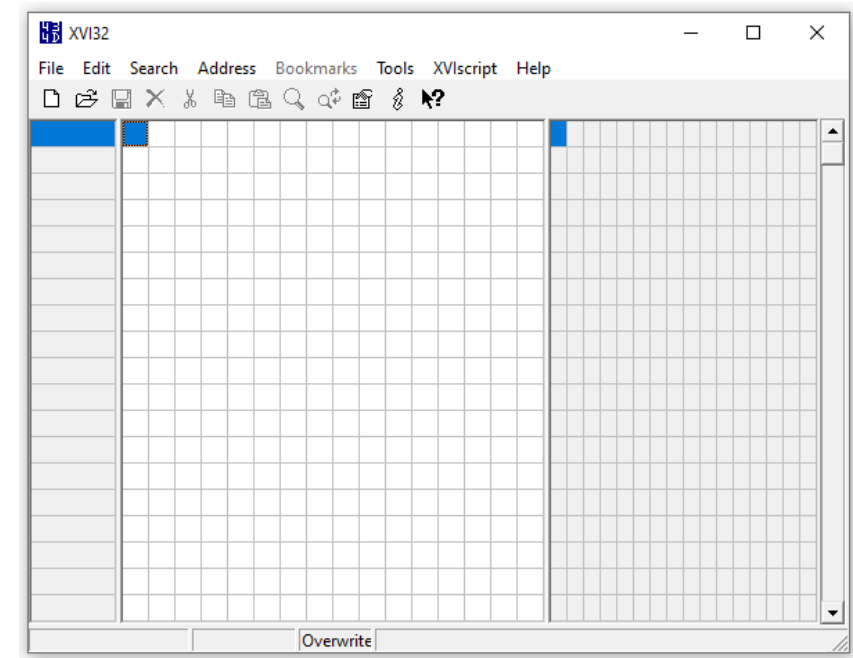


# Cifrado modo ECB versus modo CBC

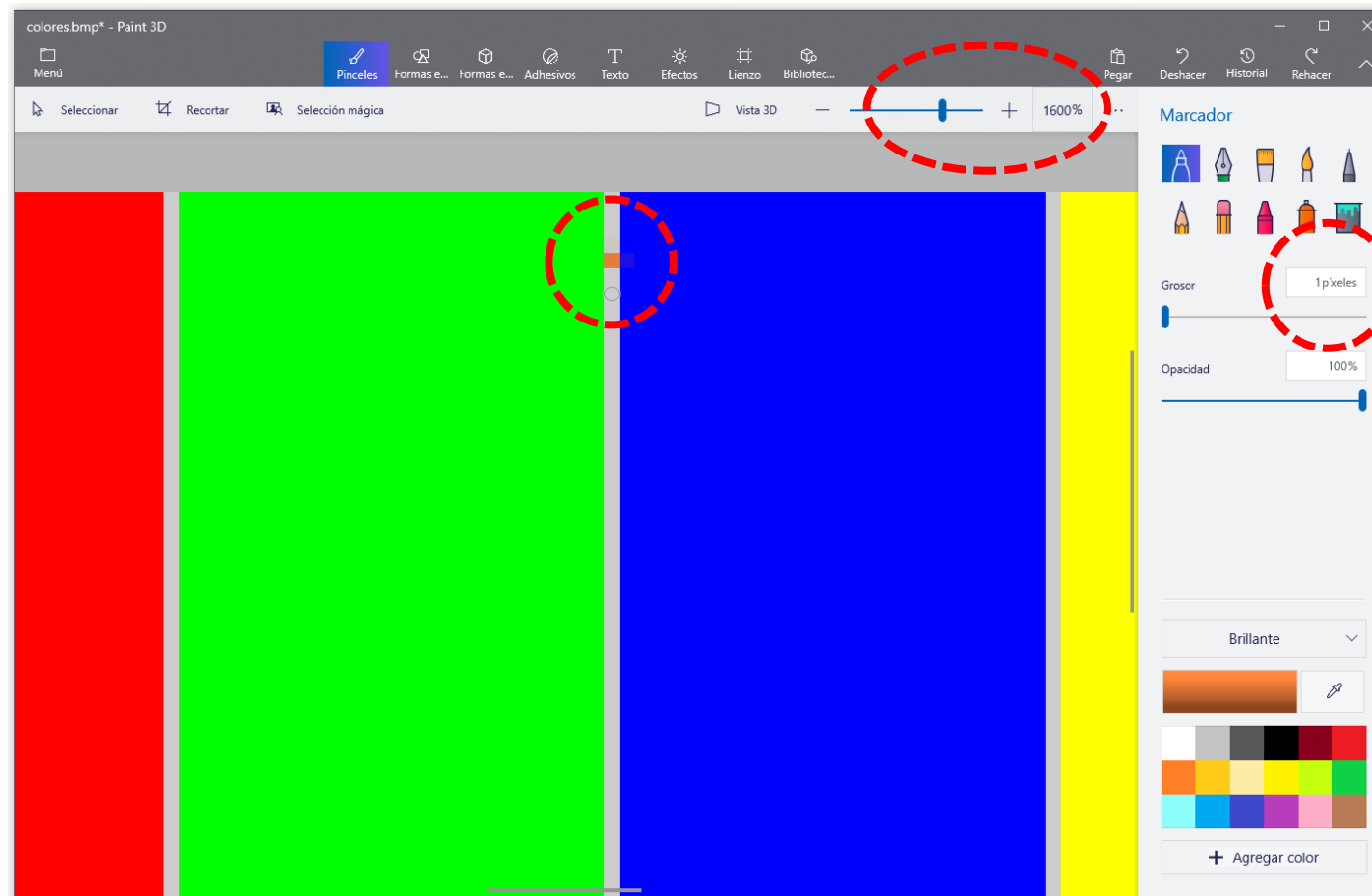
Imagen original - Cifrado ECB - Cifrado CBC (y otros)



Práctica con XVI32

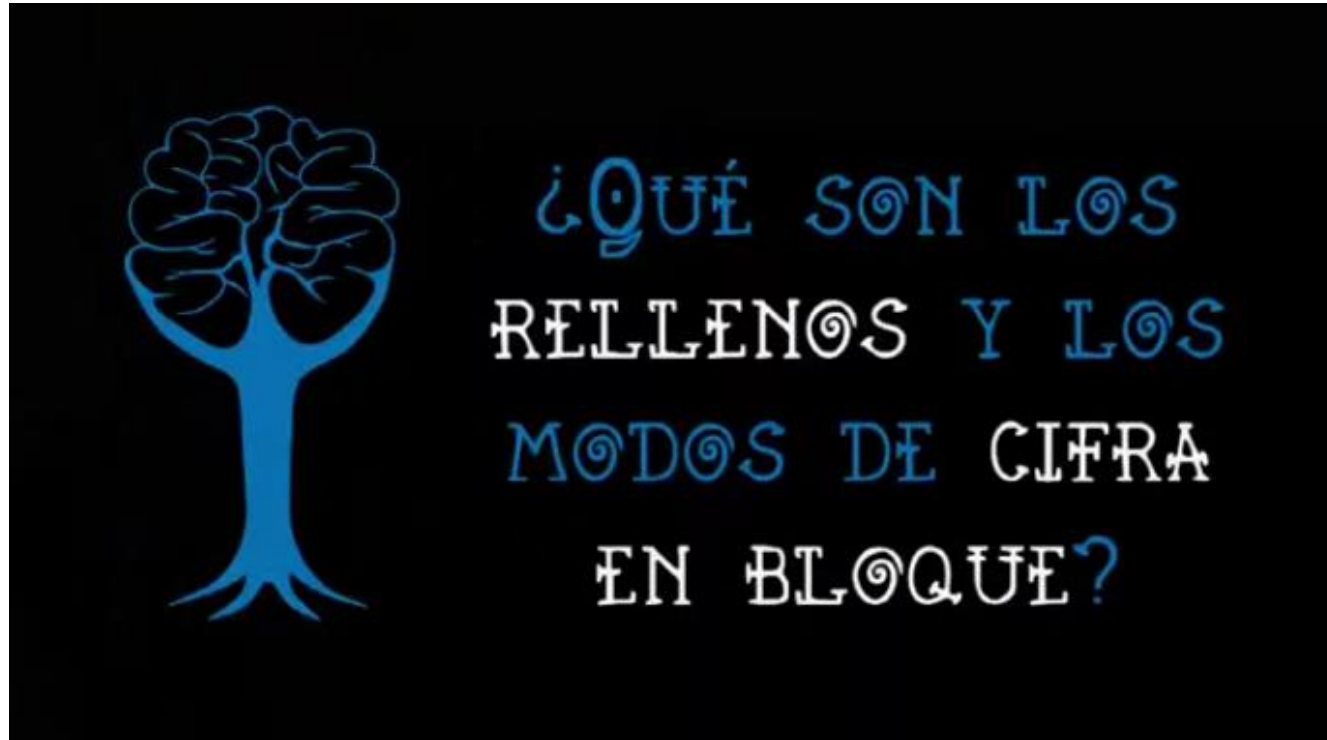


# Archivo colores.bmp editado con Paint 3D



Un píxel gris  
0x CCCCCC  
Que está entre  
un píxel verde  
0x 00FF00  
y un píxel azul  
0x 0000FF

# Más información en píldoras Thoth



[https://www.youtube.com/watch?v=AFYhB\\_MjZLw](https://www.youtube.com/watch?v=AFYhB_MjZLw)

# Conclusiones de la lección 8.4a

- Entre los modos de cifra en bloque más importantes, hay un conjunto de modos con confidencialidad, entre ellos ECB, CBC, CFB, OFB y CTR
- Los modos con confidencialidad e integridad se verán en una clase próxima
- En esta clase hemos analizado el modo ECB Electronic codebook y CBC Cipher block chaining
- Está prohibido el uso de ECB ya que permite ataques por inicios y finales iguales de texto en claro y ataques por reenvío de bloques seleccionados
- CBC, que fue muy popular hasta 2015, evita esos ataques porque al utilizar un vector de inicialización y ser éste aleatorio y diferente en cada cifra, cada operación de cifra aunque sea con la misma clave  $K$ , entrega un criptograma diferente, pero sin embargo no permitirá un cifrado en paralelo

# Lectura recomendada (1/2)

- Block Cipher Modes, Computer Security Resource Center, NIST
  - <https://csrc.nist.gov/Projects/block-cipher-techniques/BCM>
- NIST Special Publication 800-38A, 2001 Edition, Recommendation for Block Cipher Modes of Operation
  - <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf>
- Block cipher mode of operation, Wikipedia
  - [https://en.wikipedia.org/wiki/Block\\_cipher\\_mode\\_of\\_operation](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation)
- Block Cipher modes of Operation
  - <https://www.geeksforgeeks.org/block-cipher-modes-of-operation/>
- Block Ciphers Modes of Operation
  - <http://www.crypto-it.net/eng/theory/modes-of-block-ciphers.html>

# Lectura recomendada (2/2)

- Which encryption method supports random reads?, Cryptography
  - <https://crypto.stackexchange.com/questions/10879/which-encryption-method-supports-random-reads>
- Freeware Hex Editor XVI32, Version 2.55
  - <http://www.chmaas.handshake.de/delphi/freeware/xvi32/xvi32.htm>
- HexEd.it (online)
  - <https://hexed.it/>
- Guion píldora formativa Thoth nº 31, ¿Qué son los rellenos y los modos de cifra en bloque?, proyecto Thoth, Jorge Ramió, 2015
  - <https://www.criptored.es/thoth/material/texto/pildora031.pdf>

# Class4crypt c4c8.4b

## Módulo 8. Criptografía simétrica en bloque

### Lección 8.4b. CFB, OFB y CTR, modos de cifra con confidencialidad

8.4b.1 Modos de cifra aprobados por el NIST

8.4b.2. Modo de cifra CFB Cipher feedback

8.4b.3. Modo de cifra OFB Output feedback

8.4b.4. Modo de cifra CTR Counter

8.4b.5 Comparativa entre modos de cifra con confidencialidad

Class4crypt c4c8.4b CFB, OFB y CTR, modos de cifra con confidencialidad  
[https://www.youtube.com/watch?v=Elm62\\_ec4MU](https://www.youtube.com/watch?v=Elm62_ec4MU)



# Modos de cifra aprobados por el NIST

- *Currently, NIST has approved fourteen modes of the approved block ciphers in a series of special publications.*
- *As summarized on the Current Modes page, there are:*
  - *Eight confidentiality modes (ECB, CBC, OFB, CFB, CTR, XTS-AES, FF1, and FF3)*
  - *One authentication mode (CMAC)*
  - *Five combined modes for confidentiality and authentication (CCM, GCM, KW, KWP, and TKW)*

*Referencia: Block Cipher Modes, Computer Security Resource Center, NIST*

<https://csrc.nist.gov/Projects/block-cipher-techniques/BCM>

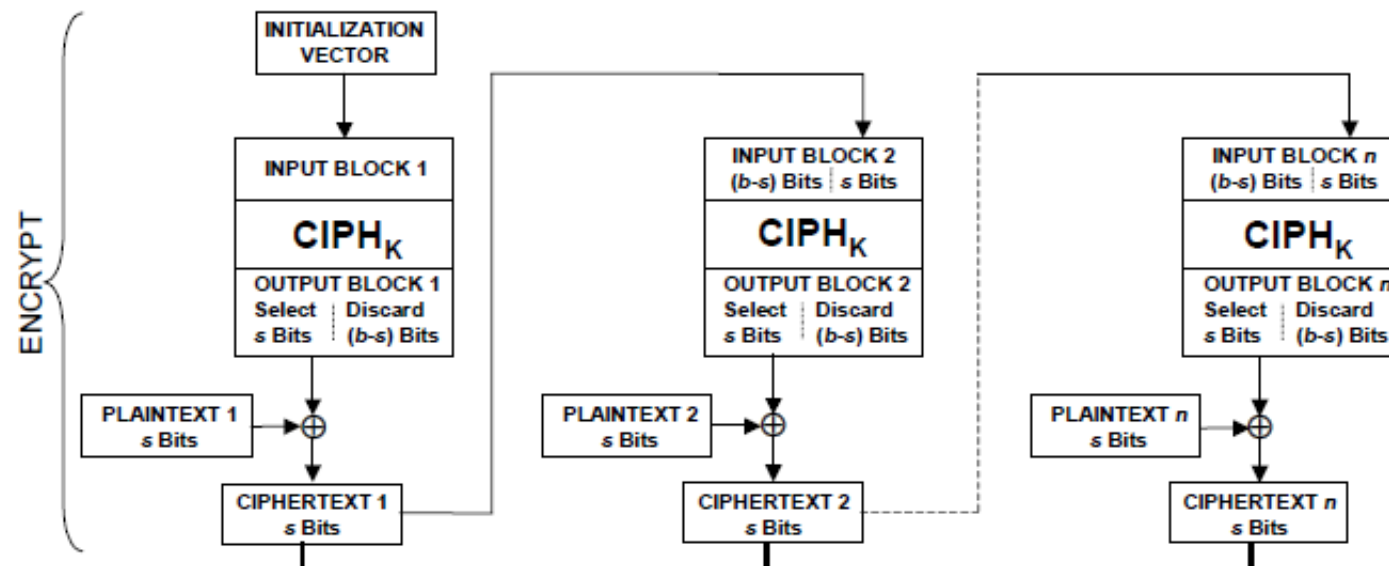


# Modos de cifra bloque con confidencialidad

- Modos de cifra en bloque con confidencialidad a estudiar
  - ECB Electronic codebook
  - CBC Cipher block chaining
  - CFB Cipher feedback ✓
  - OFB Output feedback ✓
  - CTR Counter ✓
- No veremos aquí estos otros modos
  - XTS-AES, FF1, FF3
  - Tampoco veremos PCBC Propagating cipher block chaining (que se usa entre otros en Kerberos v4 y WASTE) no incluido por el NIST

# 1) Modo CFB en cifrado

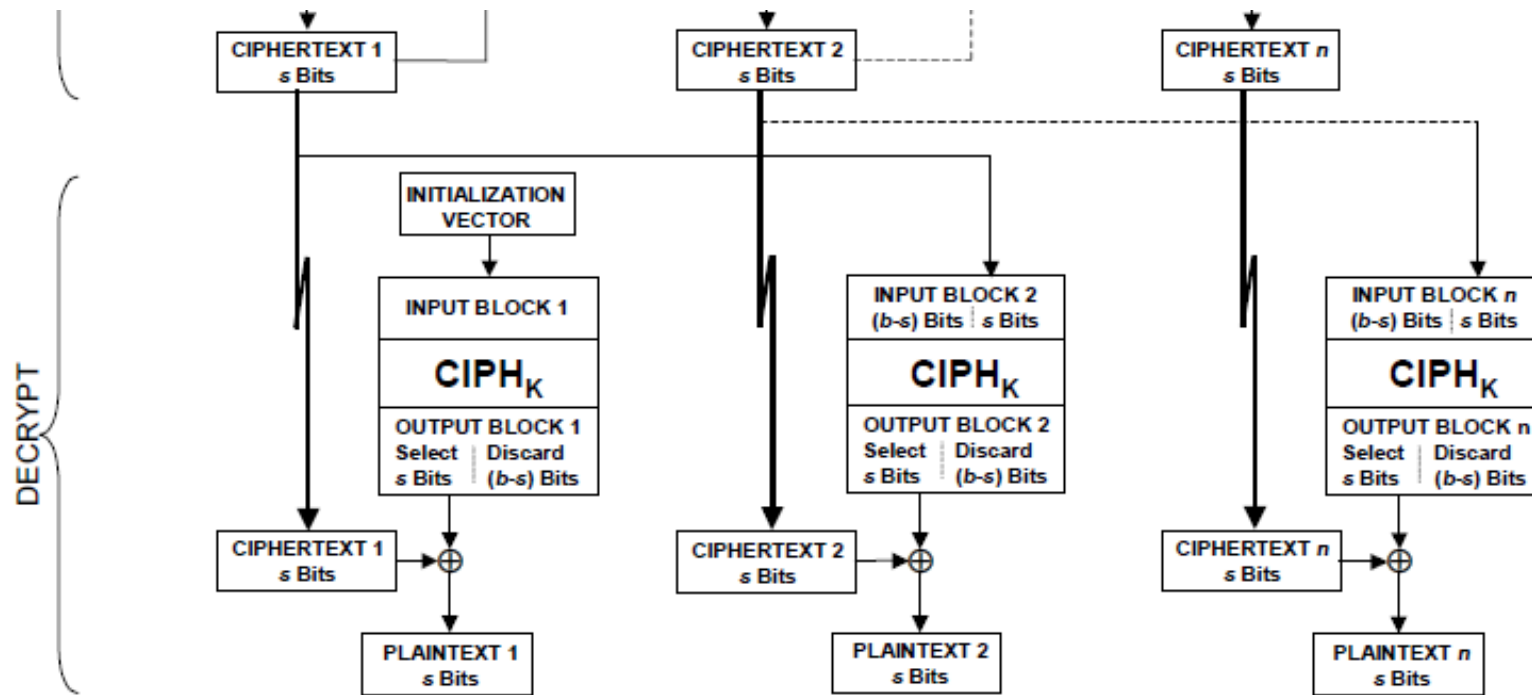
- El modo CFB Cipher feedback o Realimentación de bloques, permite cifrar unidades de datos más pequeñas, por ejemplo CFB de 1 bit, CFB de 8 bits, CFB de 64 bits y CFB de 128 bits. Aplicará relleno a  $s$  bits en el último bloque



- $CIPH_K$  significa cifrar con el algoritmo y la clave  $K$
- Se eligen los  $s$  bits más significativos de la salida, que se cifran xor con el texto
- La entrada del bloque 2 será  $(b-s) | s$  ( $s$  van a la derecha)
- Es un registro cuyos bits se desplazan hacia la izquierda

- Figura: NIST Special Publication 800-38A, 2001 Edition "Recommendation for Block Cipher Modes of Operation", <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf>

# Modo CFB en descifrado



- Usa la misma operación  $CIPH_K$  para obtener el registro, con la misma elección de los  $s$  bits más significativos y el descarte de  $(b-s)$  bits
- El descifrado se hace mediante el xor de  $s$  bits del registro y  $s$  bits de criptograma, por el carácter involutivo de la operación xor

- Figura: NIST Special Publication 800-38A, 2001 Edition "Recommendation for Block Cipher Modes of Operation", <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf>

# Ventajas y desventajas del modo CFB

- Ventajas

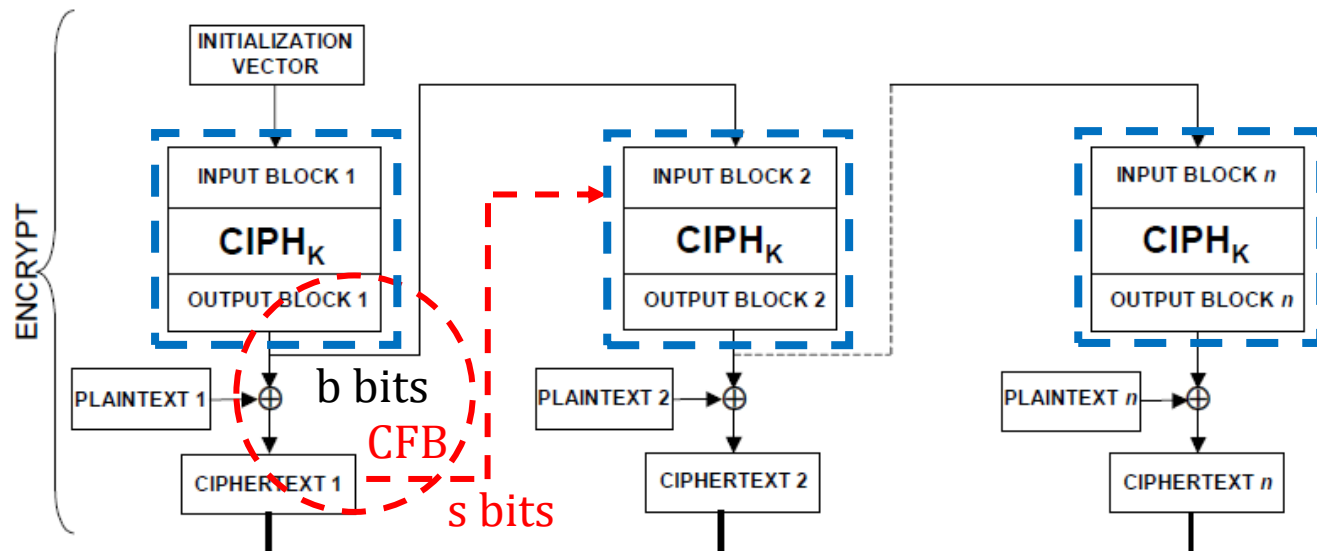
- Puede cifrar unidades de datos más pequeñas que un bloque
- Puede hacer un descifrado en paralelo conocidos K, IV y los criptogramas
- Si IV es aleatorio, al cifrar dos veces el mismo documento con la misma clave, obtenemos criptogramas distintos y la cifra de archivos con mucho formato no mostrará su perfil
- Evita ataques por comienzos y finales iguales y por el reenvío de bloques

- Desventajas

- No se puede hacer un cifrado en paralelo porque excepto en el primer bloque, el bloque de entrada depende del resultado del cifrado anterior
- Un error se propaga en todo ese bloque y en los siguientes

## 2) Modo OFB en cifrado

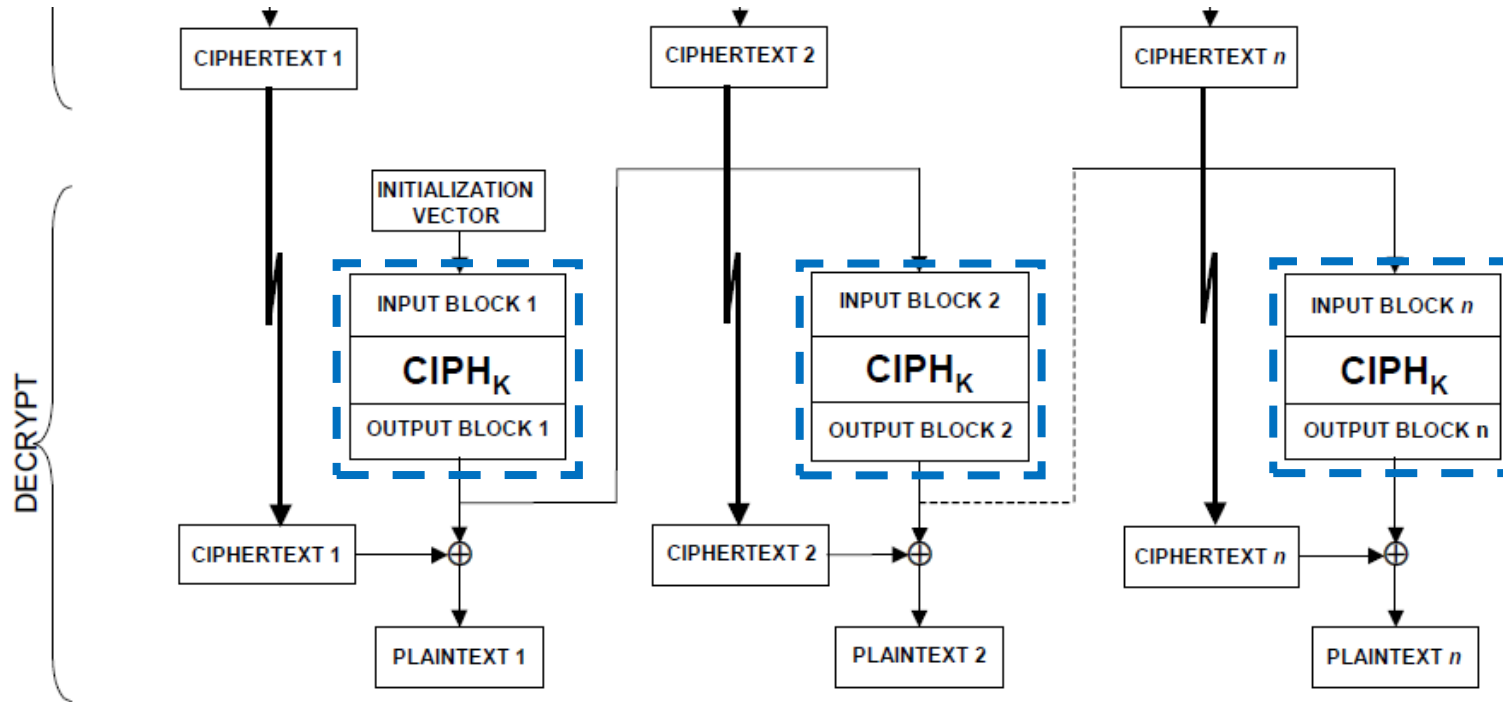
- El modo OFB Output feedback o Realimentación de salida es similar a CFB, con la diferencia de que ahora los  $b$  bits del cifrador sirven de entrada a los nuevos bloques, antes del xor, y no interviene el texto en claro como en CFB



- $CIPH_K$  significa cifrar con el algoritmo y la clave  $K$
- El vector IV debe ser nonce, es decir un número ( $n$ ) aleatorio usado una única vez (once) en cada cifra para la clave  $K$
- Si el último bloque tiene solo  $u$  bits, se usarán los  $u$  bits más significativos de la salida de  $n$

- Figura: NIST Special Publication 800-38A, 2001 Edition "Recommendation for Block Cipher Modes of Operation", <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf>

# Modo OFB en descifrado



- Usa la misma operación  $CIPH_K$  para obtener la misma secuencia de  $b$  bits
- Puesto que el xor es involutivo, permite descifrar el bloque de texto cifrado o criptograma

- Figura: NIST Special Publication 800-38A, 2001 Edition "Recommendation for Block Cipher Modes of Operation", <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf>

# Ventajas y desventajas del modo OFB

- Ventajas

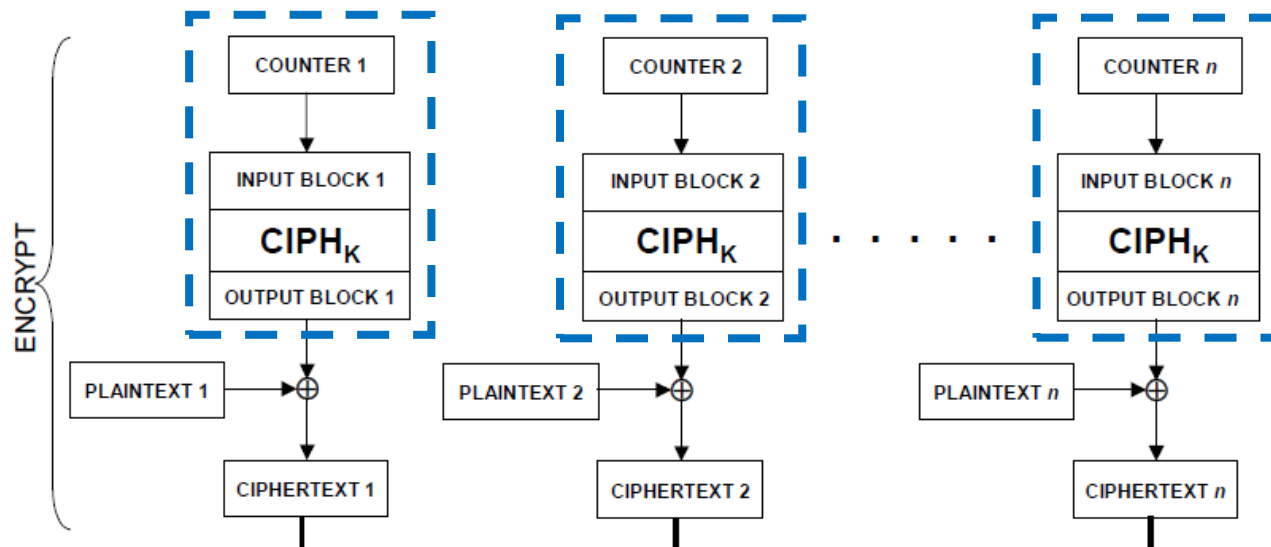
- Si el vector IV es aleatorio y cambia en cada cifra, al cifrar dos veces el mismo documento con la misma clave, obtenemos criptogramas distintos
- La cifra de archivos muy formateados no mostrará el perfil de los mismos
- Evita ataques por comienzos y finales iguales y por el reenvío de bloques
- Se trata de un cifrador de flujo y no aplica relleno

- Desventajas

- No se puede hacer un cifrado en paralelo porque, excepto en el primer bloque, el bloque de entrada depende del resultado del cifrado anterior
- Tampoco permite hacer un descifrado en paralelo
- Un error se propaga en todo ese bloque y en los siguientes

### 3) Modo CTR en cifrado

- El modo CTR Contador es muy similar a OFB, con la diferencia de que en este caso el vector de inicialización Counter es un número aleatorio tipo contador, es decir, su valor se va incrementando en una unidad de bloque en bloque



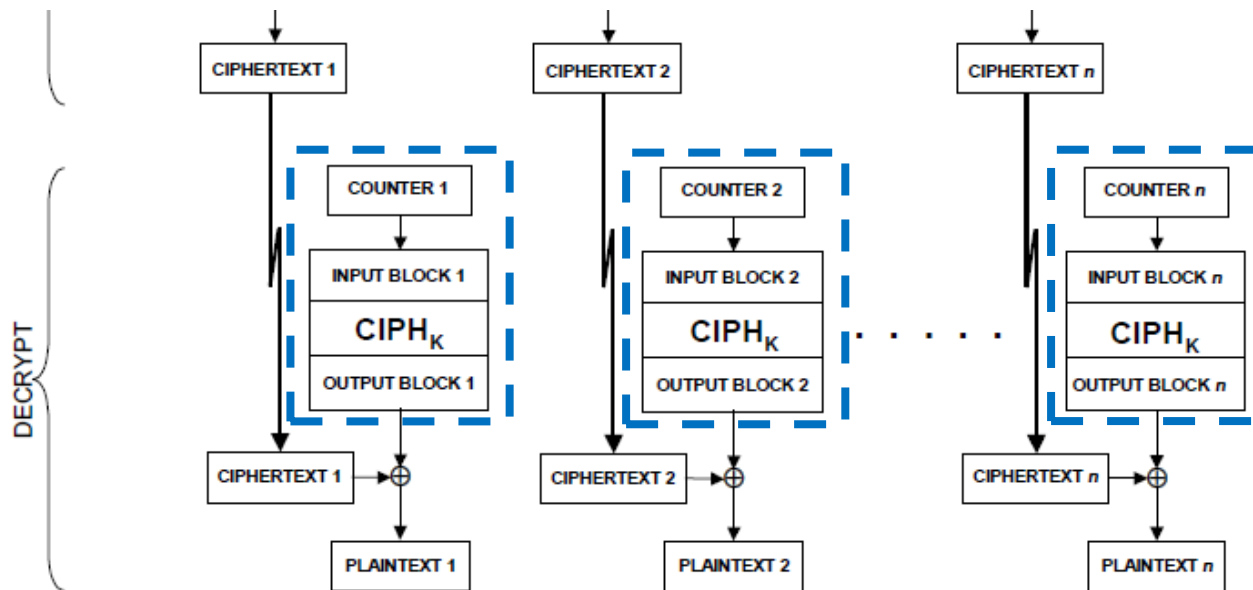
- $CIPH_K$  significa cifrar con el algoritmo y la clave  $K$
- Counter debe ser nonce, es decir un número ( $n$ ) aleatorio usado una única vez (once) en cada cifra para la clave  $K$
- Si el último bloque tiene solo  $u$  bits, se usarán los  $u$  bits más significativos de la salida de  $n$

- Figura: NIST Special Publication 800-38A, 2001 Edition "Recommendation for Block Cipher Modes of Operation", <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf>



# Modo CTR en descifrado

- El descifrado en modo CTR es igual que el cifrado, se hace un xor entre el bloque de criptograma con la salida del generador de secuencia cifrante, al igual que en un cifrador de flujo



- $\text{CIPH}_K$  significa cifrar con el algoritmo y la clave  $K$
- Se descifra haciendo xor entre el bloque de criptograma de  $b$  bits y el bloque de secuencia de clave de  $b$  bits
- Si el último bloque tiene solo  $u$  bits, se usarán los  $u$  bits más significativos de la salida de  $n$

- Figura: NIST Special Publication 800-38A, 2001 Edition "Recommendation for Block Cipher Modes of Operation", <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf>

# Ventajas y desventajas del modo CTR

- Ventajas

- Si el vector IV es aleatorio y cambia en cada cifra, al cifrar dos veces el mismo documento con la misma clave, obtenemos criptogramas distintos
- La cifra de archivos muy formateados no mostrará el perfil de los mismos
- Evita ataques por comienzos y finales iguales y por el reenvío de bloques
- Permite hacer un cifrado y un descifrado en paralelo porque la entrada de cada bloque al algoritmo de cifra solamente depende del contador, cuyo valor se conoce a priori
- Es un buen cifrador de flujo y no aplica relleno
- Un error se propaga solamente en el bloque en que se produjo

# Comparativa final de los modos de cifra

<table><tr><th colspan="2">ECB</th></tr><tr><td colspan="2">Electronic codebook</td></tr><tr><td>Encryption parallelizable:</td><td>Yes</td></tr><tr><td>Decryption parallelizable:</td><td>Yes</td></tr><tr><td>Random read access:</td><td>Yes</td></tr></table>	ECB		Electronic codebook		Encryption parallelizable:	Yes	Decryption parallelizable:	Yes	Random read access:	Yes	<table><tr><th colspan="2">CBC</th></tr><tr><td colspan="2">Cipher block chaining</td></tr><tr><td>Encryption parallelizable:</td><td>No</td></tr><tr><td>Decryption parallelizable:</td><td>Yes</td></tr><tr><td>Random read access:</td><td>Yes</td></tr></table>	CBC		Cipher block chaining		Encryption parallelizable:	No	Decryption parallelizable:	Yes	Random read access:	Yes	<table><tr><th colspan="2">PCBC (*)</th></tr><tr><td colspan="2">Propagating cipher block chaining</td></tr><tr><td>Encryption parallelizable:</td><td>No</td></tr><tr><td>Decryption parallelizable:</td><td>No</td></tr><tr><td>Random read access:</td><td>No</td></tr></table>	PCBC (*)		Propagating cipher block chaining		Encryption parallelizable:	No	Decryption parallelizable:	No	Random read access:	No
ECB																																
Electronic codebook																																
Encryption parallelizable:	Yes																															
Decryption parallelizable:	Yes																															
Random read access:	Yes																															
CBC																																
Cipher block chaining																																
Encryption parallelizable:	No																															
Decryption parallelizable:	Yes																															
Random read access:	Yes																															
PCBC (*)																																
Propagating cipher block chaining																																
Encryption parallelizable:	No																															
Decryption parallelizable:	No																															
Random read access:	No																															
<table><tr><th colspan="2">CFB</th></tr><tr><td colspan="2">Cipher feedback</td></tr><tr><td>Encryption parallelizable:</td><td>No</td></tr><tr><td>Decryption parallelizable:</td><td>Yes</td></tr><tr><td>Random read access:</td><td>Yes</td></tr></table>	CFB		Cipher feedback		Encryption parallelizable:	No	Decryption parallelizable:	Yes	Random read access:	Yes	<table><tr><th colspan="2">OFB</th></tr><tr><td colspan="2">Output feedback</td></tr><tr><td>Encryption parallelizable:</td><td>No</td></tr><tr><td>Decryption parallelizable:</td><td>No</td></tr><tr><td>Random read access:</td><td>No</td></tr></table>	OFB		Output feedback		Encryption parallelizable:	No	Decryption parallelizable:	No	Random read access:	No	<table><tr><th colspan="2">CTR</th></tr><tr><td colspan="2">Counter</td></tr><tr><td>Encryption parallelizable:</td><td>Yes</td></tr><tr><td>Decryption parallelizable:</td><td>Yes</td></tr><tr><td>Random read access:</td><td>Yes</td></tr></table>	CTR		Counter		Encryption parallelizable:	Yes	Decryption parallelizable:	Yes	Random read access:	Yes
CFB																																
Cipher feedback																																
Encryption parallelizable:	No																															
Decryption parallelizable:	Yes																															
Random read access:	Yes																															
OFB																																
Output feedback																																
Encryption parallelizable:	No																															
Decryption parallelizable:	No																															
Random read access:	No																															
CTR																																
Counter																																
Encryption parallelizable:	Yes																															
Decryption parallelizable:	Yes																															
Random read access:	Yes																															

- Fuente Wikipedia: [https://en.wikipedia.org/wiki/Block\\_cipher\\_mode\\_of\\_operation](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation)
- (\*) El modo PCBC Propagating Cipher Block Chaining no incluido por el NIST, es muy similar al CBC. La diferencia es que el “vector inicial” en los bloques segundo y siguientes no será el criptograma anterior sino la suma xor entre el criptograma y el texto en claro anteriores

# Conclusiones de la lección 8.4b

- Entre los modos de cifra en bloque más importantes, hay un conjunto de modos con confidencialidad, entre ellos ECB, CBC, CFB, OFB y CTR. Los modos con confidencialidad e integridad se verán en una clase próxima
- Los modos ECB, CBC y CFB requieren que al último bloque se le añada relleno hasta llegar a  $b$  bits (o bien los  $s$  bits en CFB), si fuese necesario
- No obstante, no será necesario añadir relleno en los modos OFB y CTR
- Los únicos modos de los 5 analizados que permite un cifrado y descifrado en paralelo y, además, el acceso de lectura aleatorio a un criptograma, son el ECB que está prohibido su uso por fuertes debilidades ante ataques, y el CTR
- El modo CTR es un cifrado en flujo real, en el que la secuencia de clave es aleatoria y va cambiando de bloque en bloque

# Lectura recomendada

- Block Cipher Modes, Computer Security Resource Center, NIST
  - <https://csrc.nist.gov/Projects/block-cipher-techniques/BCM>
- NIST Special Publication 800-38A, 2001 Edition, Rec. for Block Cipher Modes of Operation
  - <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf>
- Block cipher mode of operation, Wikipedia
  - [https://en.wikipedia.org/wiki/Block\\_cipher\\_mode\\_of\\_operation](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation)
- Block cipher modes of operation
  - <https://www.geeksforgeeks.org/block-cipher-modes-of-operation/>
  - <http://www.crypto-it.net/eng/theory/modes-of-block-ciphers.html>
- Which encryption method supports random reads?, Cryptography
  - <https://crypto.stackexchange.com/questions/10879/which-encryption-method-supports-random-reads>

# Class4crypt c4c8.5

## Módulo 8. Criptografía simétrica en bloque

### Lección 8.5. Ataques al DES, DES Challenge y 3DES

8.5.1. Debilidades del DES

8.5.2. Ataques en red a la cifra simétrica en bloque usando divide y vencerás

8.5.3. El desafío DES Challenge

8.5.4. Necesidad del cifrado múltiple

8.5.5. Fortaleza real del doble DES por ataque meet in the middle

8.5.6. Características y usos del 3DES

8.5.7. Formato de cifra EDE Encrypt Decrypt Encrypt

Class4crypt c4c8.5 Ataques al DES, DES Challenge y 3DES  
<https://www.youtube.com/watch?v=kU3FP9HbxFs>

# ¿Era débil el DES ya en los años 90?

- En 1977 Diffie y Hellman indican que podría romperse por fuerza bruta
- En 1991 Biham y Shamir presentan el criptoanálisis diferencial
- En 1992 Biham y Shamir demuestran que su fortaleza es de  $2^{47}$  en vez de  $2^{56}$
- En 1994 Matsui presenta un criptoanálisis lineal experimental
- Entre 1997 y 1999 se demuestra mediante el proyecto DES Challenge, desafío al DES liderado por RSA, que DES no soporta un ataque distribuido en red
- En 1999 este desafío en red logra romper al DES en menos de 24 horas
- En 2006 la máquina COPACABANA (Alemania) rompe el DES en 9 días
- En 2008 la máquina RIVYERA (su sucesora) lo rompe en menos de un día. Y siguen los ataques en 2016 y 2017, más rápidos y con menos recursos

# Las certificaciones del DES

- Por ello el DES recibe las siguientes certificaciones por la NBS (National Bureau of Standards) y posteriormente por el NIST (National Institute of Standards and Technology)
  - 1976: el DES se adopta como estándar de cifra simétrica en bloque
  - 1983: se confirma al DES como estándar por primera vez
  - 1988: se confirma al DES como estándar por segunda vez  
En 1988 la NBS pasa a llamarse NIST
  - 1993: se confirma al DES como estándar por tercera vez
  - 1999: se confirma al DES como estándar por cuarta vez, pero se indica que se use preferentemente la variante denominada 3DES o Triple DES y se use DES sólo en sistemas heredados



# Ataque en red a la cifra simétrica en bloque



- La clave es un número único. En el DES del 0 hasta 72.057.594.037.927.935 ( $2^{56} - 1$ )
- Sea nuestra clave un caramelo
- Tenemos 300 departamentos vacíos y dejamos el caramelo en uno de ellos
- Si buscamos ese caramelo solos, habrá que buscar en los 300 departamentos, tardando de media un tiempo  $x$  en encontrarlo
- Pero si tenemos 300 amigos, uno en cada casa, y hacemos una búsqueda de forma simultánea y coordinada, avisando quien lo encuentre a los demás con el teléfono móvil, tardaremos en encontrarlo de media unas 300 veces menos
- Es un ejemplo del principio divide y vencerás aplicando paralelismo

# Divide y vencerás en el DES

- Las 72.057.594.037.927.936 claves del DES se dividen en  $n$  espacios o ventanas
- La ventana de búsqueda  $v$  tendrá un valor inicial  $v_i$  y un valor final  $v_f$  ( $v = v_f - v_i$ )
- Por ejemplo  $v_i = 1.000.001$  y  $v_f = 2.000.000$  para una ventana de tamaño un millón
- Cada máquina buscará la clave en esa ventana  $v$  a una tasa de  $x$  bits/segundo
- Si tenemos  $n$  máquinas trabajando, cada una con su ventana  $v$  definida, haremos el ataque a una tasa de  $n \cdot x$  bits/segundo



# Ataque delimitado al DES con safeDES

- Se conoce el texto en claro M y el cifrado C
- M = 0x 41 54 41 51 55 45 20 41 4C 20 44 45 53 20 4D 4F 44 4F 20 4D 4F 4E 4F 55 53 55 41 52 49 4F

## ATAQUE AL DES MODO MONOUSUARIO

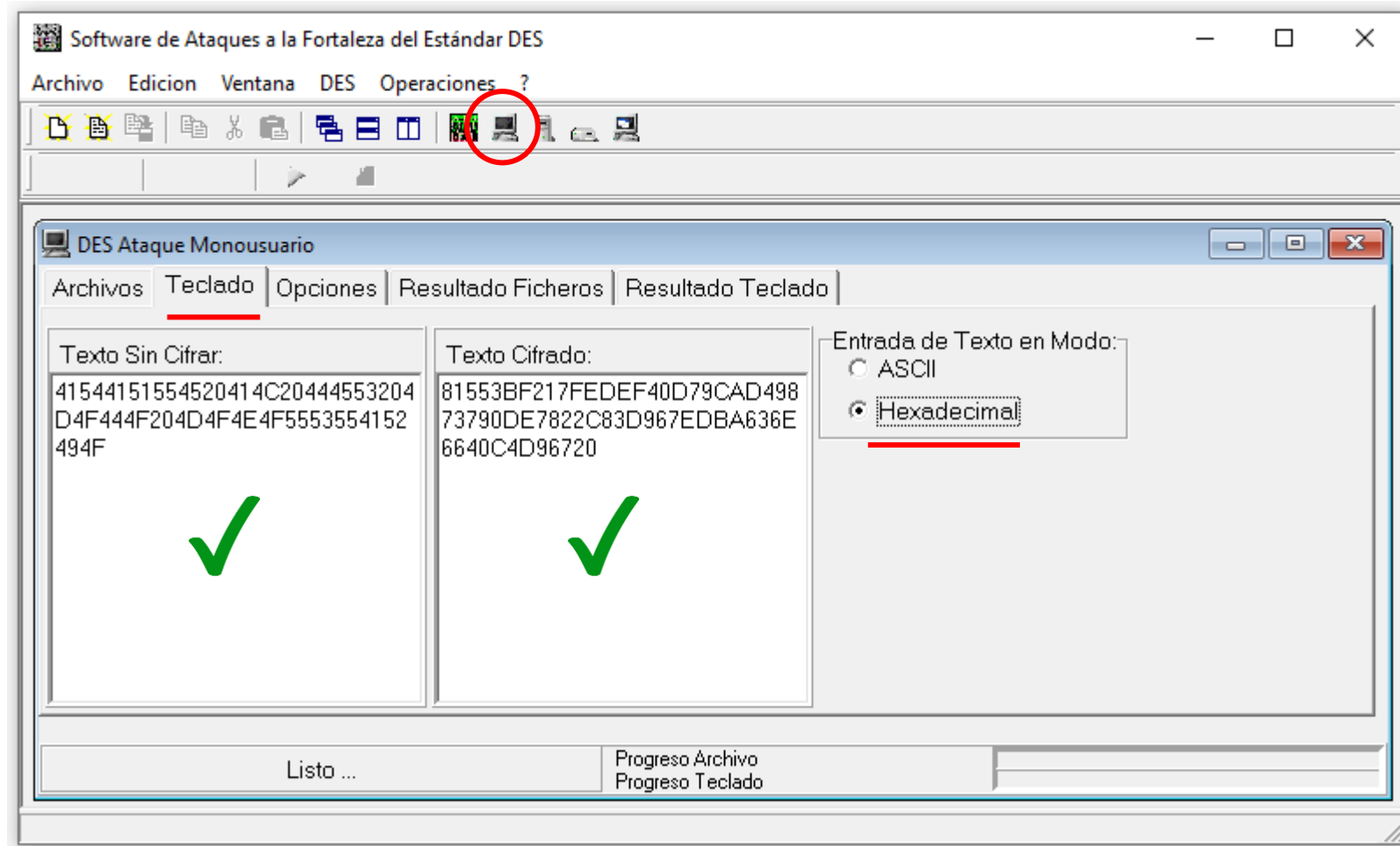
- C = 0x 81 55 3B F2 17 FE DE F4 0D 79 CA D4 98 73 79 0D E7 82 2C 83 D9 67 ED BA 63 6E 66 40 C4 D9 67 20
- $K_i = 0x\ 0123456780000000$
- $K_f = 0x\ 012345678FFFFFFF$



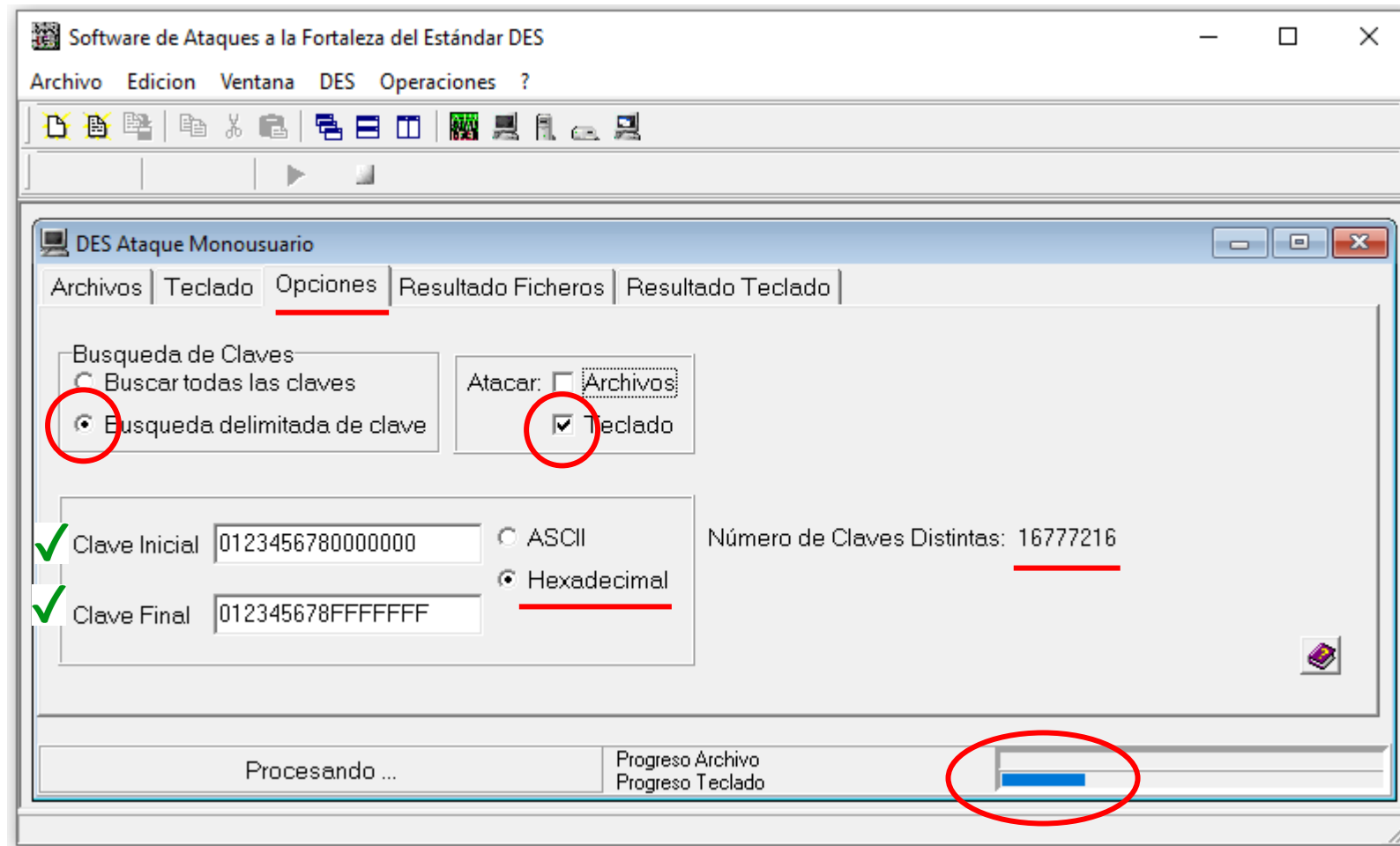
Capturas de pantalla de la resolución de la práctica



# Capturas de pantalla de la práctica (1/3)

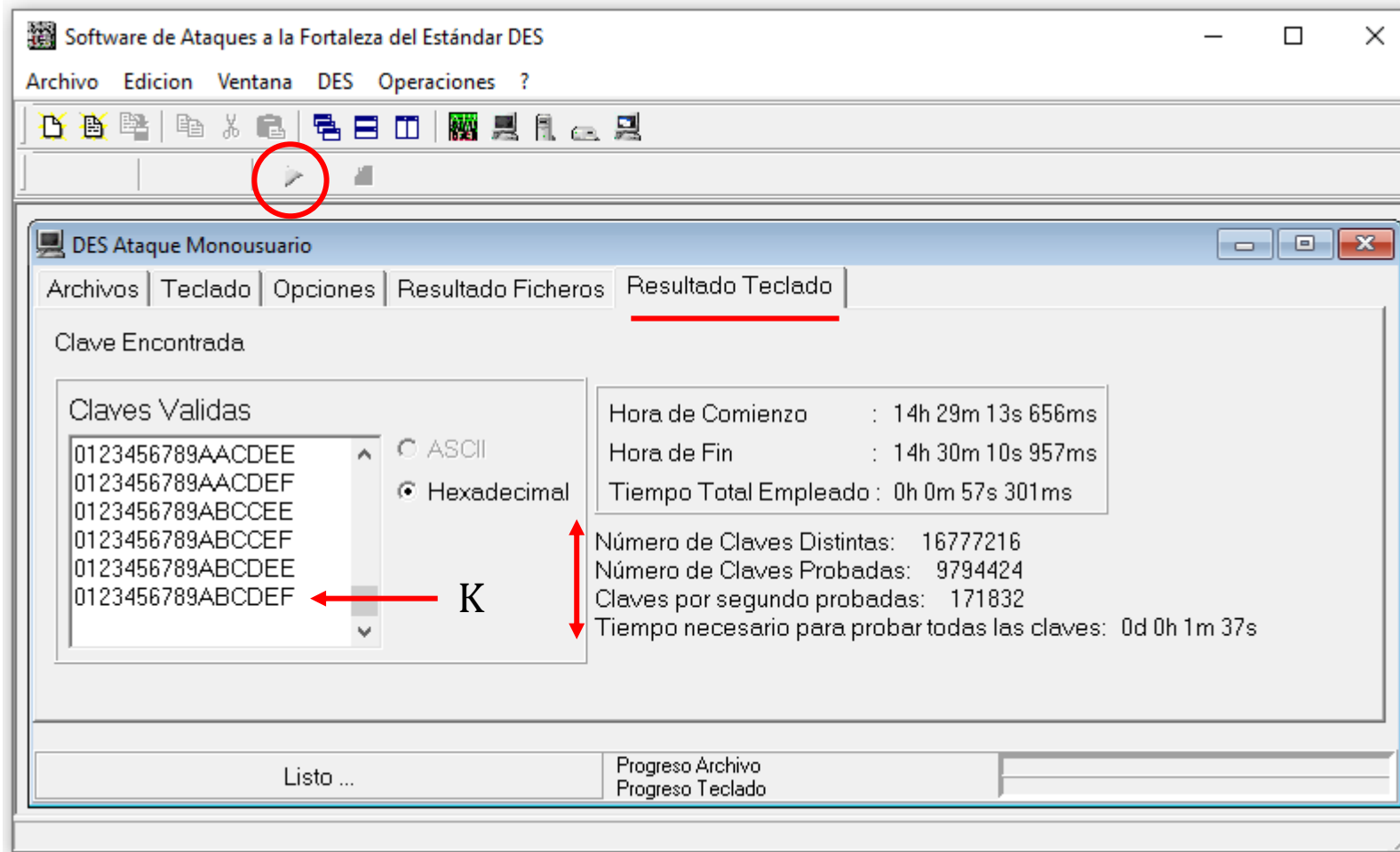


# Capturas de pantalla de la práctica (2/3)





# Capturas de pantalla de la práctica (3/3)



Hay  $2^8 = 256$  claves válidas, en las que el octavo bit de cada uno de los 8 bytes de la clave se elimina, por ser el bit paridad. Son las posiciones pares de los 4 bits (nibble) del cada valor byte en hexadecimal

$K_1 = 0x\ 0022446688AACCEE$

$K_2 = 0x\ 0022446688AACCE\textcolor{blue}{F}$

$K_3 = 0x\ 0022446688AAC\textcolor{blue}{D}EE$

$K_4 = 0x\ 0022446688AACDE\textcolor{blue}{F}$

...

$K_{255} = 0x\ 0123456789ABCDE\textcolor{blue}{E}$

$K_{256} = 0x\ 0123456789ABCDE\textcolor{blue}{F}$

Sin el bit de paridad, todos los valores son el mismo número

# Ataque en red con safeDES



- La misma vulnerabilidad mostrará cualquier algoritmo de cifra simétrica en bloque, como por ejemplo AES
- Por ello, entre otras razones, es necesario hoy en día usar una clave de al menos 128 bits

Ref.: CLCript 12: Ataques por fuerza bruta al DES y DES Challenge III

[https://www.criptored.es/descarga/CLCript entrega 12 Ataques por fuerza bruta a DES y DES Challenge III.pdf](https://www.criptored.es/descarga/CLCript%20entrega%2012%20Ataques%20por%20fuerza%20bruta%20a%20DES%20y%20DES%20Challenge%20III.pdf)

# El desafío de RSA: DES Challenge I y II



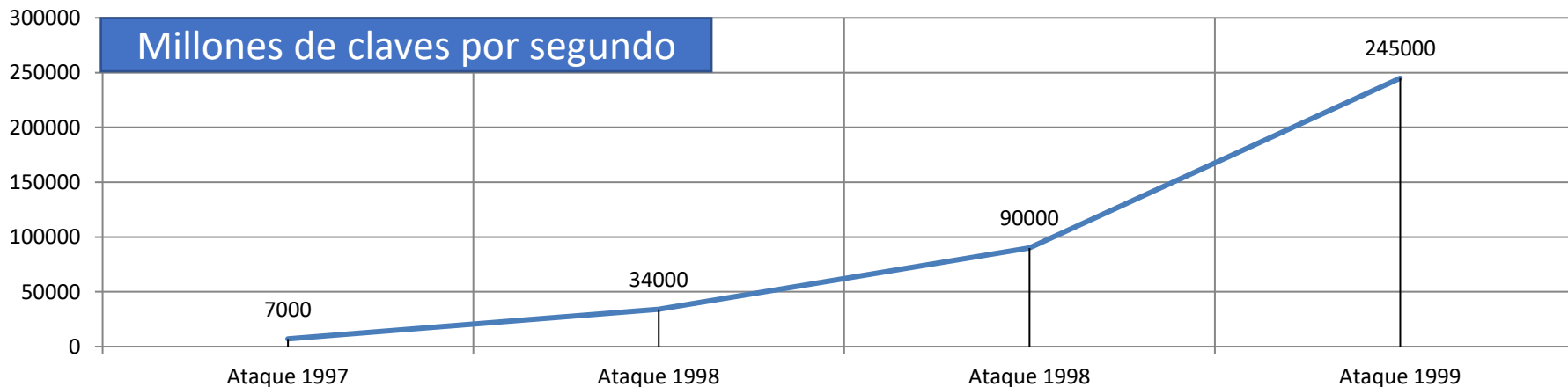
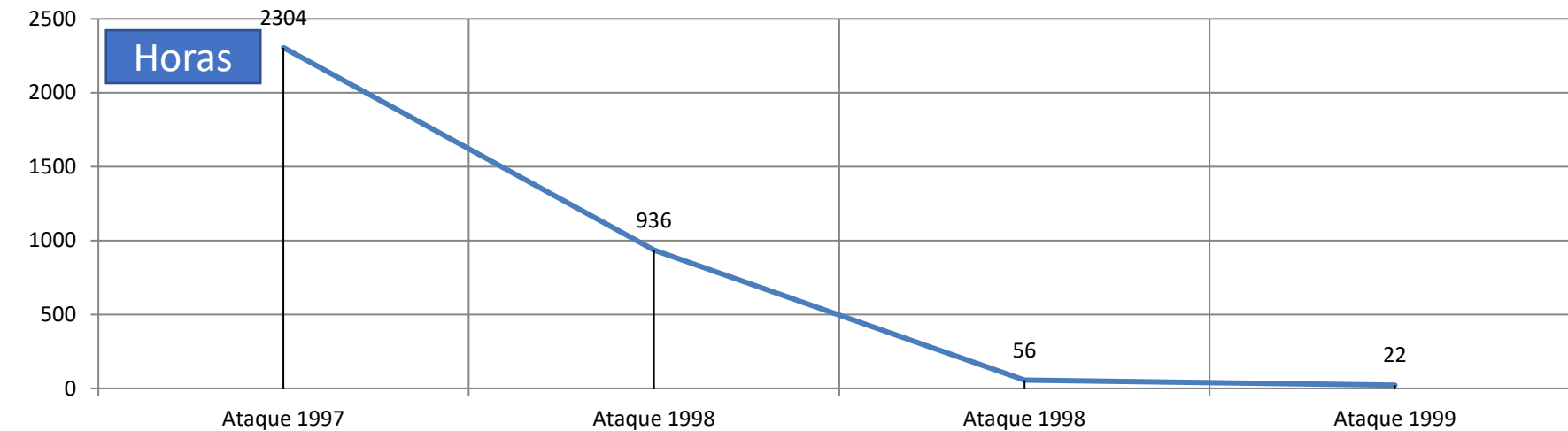
- 29 enero 1997: DES Challenge I. Se rompe la clave en 96 días con 80.000 de ordenadores en Internet que evalúan **7.000 millones** de clave por segundo
  - Para encontrar la clave se debe recorrer el 25% del espacio de claves (buena suerte)
- 13 enero 1998: DES Challenge II-1. Se rompe la clave en 39 días con un ataque de tipo distribuido, desarrollado por distributed.net y que llega a evaluar **34.000 millones** de claves por segundo
  - Ahora se debe recorrer el 88% del espacio de claves (mala suerte)
- 13 julio de 1998: DES Challenge II-2. La EFF Electronic Frontier Foundation, crea el DES Cracker con una inversión de US \$ 200.000 y en 56 horas (2½ días) rompe la clave evaluando **90.000 millones** de claves por segundo



# El desafío de RSA: DES Challenge III

- 18 enero 1999: DES Challenge III. Se utiliza la máquina DES Cracker y distributed.net con 100.000 ordenadores conectados en Internet para romper la clave en 22 horas, menos de 1 día, evaluando **245.000 millones** de claves por segundo
  - Aquí se debe recorrer el 22% del espacio de claves (buena suerte)
- Se trata del último desafío propuesto por RSA, que pone en evidencia la capacidad de ataques distribuidos a la cifra simétrica en bloque, a través de los tiempos muertos del procesador de máquinas conectadas a Internet y que, con un programa cliente, van resolviendo un pequeño trozo del espacio de claves de ataque, comunicándose para ello con un servidor
  - Recuerda que el DES no ha sido criptoanalizado, sólo se ha roto la cifra por fuerza bruta debido al pequeño tamaño de su clave

# Estadísticas del DES Challenge



¿Qué significa una tasa de ataque de 250 mil millones de claves por segundo para un algoritmo actual como el AES?



# ¿250.000 millones c/s y 50% del espacio?

Longitud de la clave	Tiempo necesario para romper la clave
40 bits	2 segundos
48 bits	9 minutos
56 bits	50 horas
64 bits	14 meses
72 bits	305 años
80 bits	78.250 ( $2^{16}$ ) años
96 bits	5.127.160.311 ( $2^{32}$ ) años
112 bits	336.013.578.167.538 ( $2^{48}$ ) años
128 bits	22.020.985.858.787.784.059 ( $2^{64}$ ) años

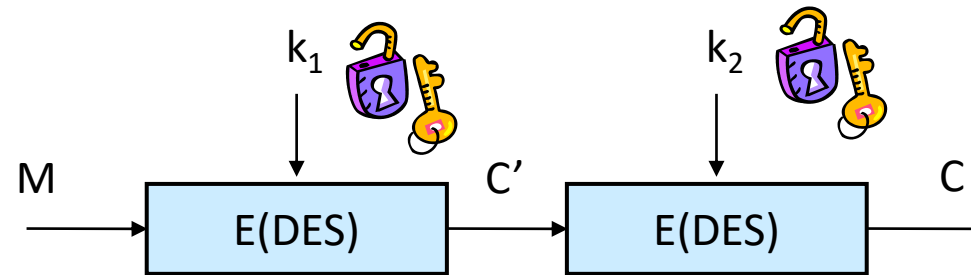
Referencia de tiempo con números grandes	
Edad planeta	10.000.000.000 ( $10^{10} = 2^{34}$ ) años
Edad universo	100.000.000.000 ( $10^{11} = 2^{37}$ ) años

- Tiempo medio de criptoanálisis necesario para romper una clave de cifra simétrica mediante fuerza bruta, usando la potencia de cálculo alcanzada en el DES Challenge III en 1999, 250.000 millones de claves por segundo
- Según la ley de Moore, en el año 2020 esta potencia de cálculo se habría multiplicado por  $2^{12}$ , pero aquí son números tan grandes que  $2^{12}$  no es nada significativo...

# Necesidad del cifrado múltiple en DES

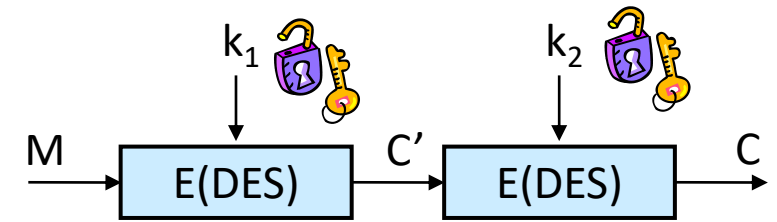
- Un cifrado múltiple es una cifra en cascada, con claves diferentes
- Por ejemplo,  $C = E_{k_3}[E_{k_2}[E_{k_1}(M)]]$  sería un cifrado triple
- Un cifrado tiene estructura de grupo si cumple con esta propiedad:
  - Hacer dos (o más) operaciones de cifrado sucesivas con claves diferentes, es equivalente a hacer un solo cifrado con una clave nueva
  - Vigenère sí tiene estructura de grupo: cifrar con  $k_1 = \text{PACO}$  y después con  $k_2 = \text{CINE}$  es lo mismo que cifrar con  $k = \text{RIOS}$ , pues  $\text{PACO} + \text{CINE} = \text{RIOS}$
- Como el DES no forma grupo, permite un cifrado múltiple
- Y la fortaleza del sistema de cifra aumentará porque el ataque deberá encontrar los valores de  $k_1, k_2, k_3$ , etc., no una sola clave

# ¿Por qué no existe el doble DES?



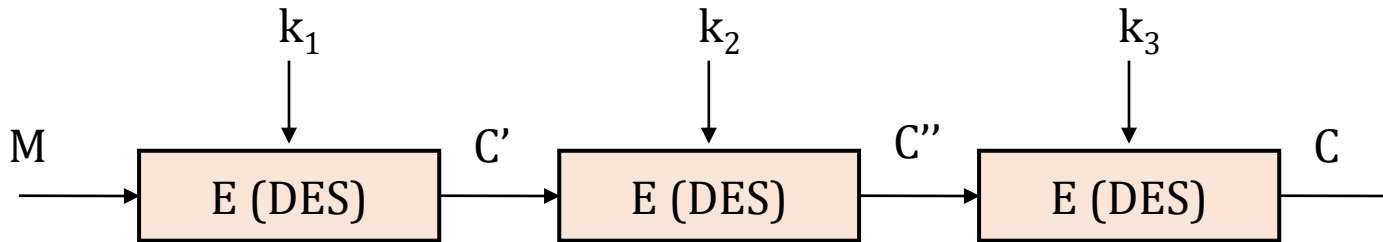
- Como el DES no forma grupo, un ataque por ejemplo por fuerza bruta deberá encontrar las claves  $k_1$  y  $k_2$
- Si  $k$  tiene 56 bits, la fortaleza de este nuevo sistema de cifra doble con DES debería tener una fortaleza de  $56 \times 2 = 112$  bits, pero...
- Todo algoritmo debe ser resistente a un ataque con texto en claro conocido. Y en este esquema doble se puede realizar un ataque meet in the middle, que reduce mucho la fortaleza real del sistema

# Ataque meet in the middle



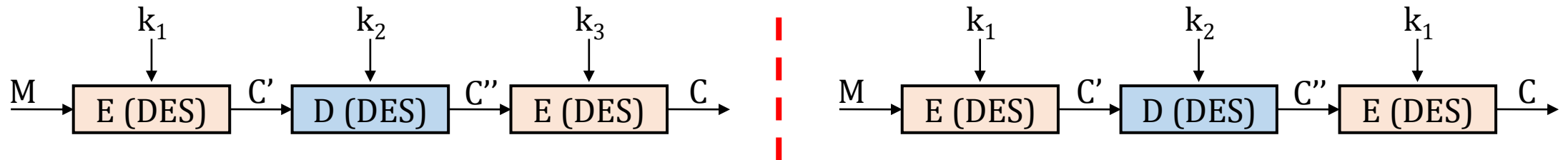
1. Se descifra el criptograma  $C$  por fuerza bruta usando las  $2^{k_2}$  claves posibles y realizando  $2^{k_2}$  cálculos. Se obtienen así todos los resultados posibles de  $C'$
2. Con los textos descifrados intermedios  $C'$  se forma una tabla ordenada de textos descifrados con los correspondientes valores de  $k_2$
3. Se cifra ahora el texto en claro  $M$  conocido con todas las claves  $k_1$  posibles, realizando un máximo de  $2^{k_1}$  cálculos
4. Se comparan los resultados con el criptograma  $C'$ . Un valor de estos cifrados coincidirá con otro de los descifrados y se conocerán los valores de  $k_1$  y  $k_2$
5. Hemos realizado  $2^{k_2} + 2^{k_1}$  operaciones, aunque normalmente serán menos
6. Si  $k_1$  y  $k_2$  son 56 bits del DES, realizaremos máximo  $2^{56} + 2^{56} = 2^{57}$  cálculos, y la fortaleza real del sistema será de 57 bits, tan solo aumenta en un bit...

# Características y usos de 3DES



- En 3DES se cifra de forma encadenada con tres claves diferentes:  $k_1$ ,  $k_2$  y  $k_3$
- 3DES es inhume a un ataque por meet in the middle (encuentro a medio camino) porque el punto medio no es una conexión sino el propio algoritmo
- Como DES tiene 56 bits reales de clave, la clave del 3DES será de  $56 \times 3 = 168$  bits, aceptable pero muy lento para cifrar grandes volúmenes de información
- Se usaba, por ejemplo, en SSL/TLS hasta 2012 y en la aplicación PGP
- Se recomienda junto con AES en el Esquema Nacional de Seguridad (España)

# 3DES modo EDE: compatibilidad con DES



- Modelo EDE (Encrypt Decrypt Encrypt) propuesto por Matyas y Meyer de IBM en 1998, de 168 bits (con  $k_1, k_2, k_3$ ) y de 112 bits (con  $k_1, k_2$ )
- Si hacemos  $k_3 = k_1$  obtenemos el esquema de la segunda figura y el espacio real de claves se reduce a  $2^{2n}$  bits, es decir  $2^{2 \cdot 56} = 2^{112}$  bits efectivos
- Actualmente es un valor muy bajo, pero era interesante en aquellos años:
  - El modelo con dos claves era compatible con el DES de clave única cuando  $k_1 = k_2$
  - Por ejemplo, compatibilidad entre un cliente con DES y un servidor con 3DES
  - Era más eficiente que 3DES normal al usar solamente 2 claves, pero menos seguro



# 3DES EDE en TLS (2010 ECI, 2012 BBVA)

The screenshot displays a web browser window showing the El Corte Inglés website. The URL is [https://www.elcorteingles.es/multienda\\_ssl/comun/eci/TarjetaECI/firmlogin.asp](https://www.elcorteingles.es/multienda_ssl/comun/eci/TarjetaECI/firmlogin.asp). The page features a login form for 'Servicios ON-LINE (acceso mediante PIN)' and 'Servicios ON-LINE (libre acceso)'. Overlaid on the browser are two Windows XP dialog boxes. The 'Propiedades' dialog box shows the connection details for the website, indicating the use of TLS 1.0 with Triple DES (3DES) encryption in EDE mode with a 168-bit key. The 'Propiedades de Fecha y hora' dialog box shows the date and time settings.

**Propiedades**

General

BBVA - Bancos, entidades financieras, credito, productos y servicios BBVA. Adelante

Protocolo: HyperText Transfer Protocol with Privacy

Tipo: No disponible

Conexión: TLS 1.0, Triple DES con cifrado de 168 bits (alta); RSA con intercambio de 2048 bits

Dirección: <https://www.bbva.es/TLBS/tlbs/esp/segmento/particulares/index.jsp>

Tamaño: No disponible

Creado: No disponible

Modificado: No disponible

**Propiedades de Fecha y hora**

Fecha y hora Zona horaria Hora de Internet

Fecha

marzo 2012

17:10:09

Zona horaria actual: Hora estándar romance

**Propiedades de Fecha y hora**

Fecha y hora Zona horaria Hora de Internet

Fecha

enero 2010

18:27:56

Zona horaria actual: Romance Standard Time

**Ver** Ver el certificado de seguridad que verifica la identidad de este sitio web.

**Conexión cifrada: el nivel de cifrado es alto (3DES-EDE-CBC 168 bit)**

La página que está viendo fue cifrada antes de ser transmitida por Internet. El cifrado hace muy difícil que gente no autorizada pueda ver la información que viaja entre dos ordenadores. Por tanto, es muy improbable que alguien haya leído esta página mientras viajaba por la red.

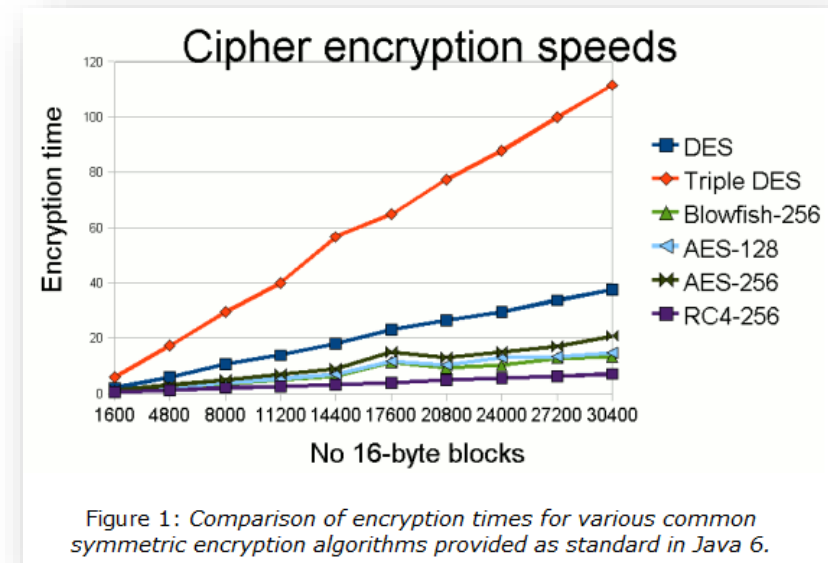
Cifrado 3DES EDE con 168 bits de fortaleza ( $k_1, k_2, k_3$ ) en modo CBC

# Velocidad del DES, 3DES y otros algoritmos

Algoritmo	Texto claro	Tiempo	MBytes/seg.
Blowfish	256 Mbytes	3,98	64,39
AES 128	256 Mbytes	4,20	61,01
AES 192	256 Mbytes	4,82	53,15
AES 256	256 Mbytes	5,31	48,23
AES 128 CTR	256 Mbytes	4,44	57,71
AES 128 OFB	256 Mbytes	4,84	52,93
AES 128 CFB	256 Mbytes	5,38	47,60
AES 128 CBC	256 Mbytes	4,62	55,45
DES	128 Mbytes	6,00	21,34
3DES-XEX3	128 Mbytes	6,16	20,78
3DES-EDE3	64 Mbytes	6,50	9,85

Microsoft Visual C++ .NET 2003 , Windows XP SP Pentium 4 2.1 GHz

[http://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption\\_perf/index.html](http://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption_perf/index.html)



Javamex: Comparison of ciphers

<https://www.javamex.com/tutorials/cryptography/ciphers.shtml>

# Más información en píldoras Thoth



[https://www.youtube.com/watch?v=Sh\\_upEyBW84](https://www.youtube.com/watch?v=Sh_upEyBW84)

# Conclusiones de la Lección 8.5

- DES nace débil por la limitación del tamaño de clave impuesta por la NSA
- Desde los años 90 proliferan los ataques, siendo DES Challenge III el más famoso de ellos al romper por fuerza bruta la clave en menos de un día
- El ataque divide y vencerás a la cifra simétrica afecta especialmente al DES
- Con DES se debe usar un cifrado múltiple para aumentar el valor de la clave
- Doble DES es vulnerable al ataque meet in the middle (no confundir con man) o ataque por encuentro a medio camino, con texto en claro conocido
- En este caso la fortaleza del Doble DES aumenta sólo en un bit, de 56 a 57 bits
- Se usa 3DES EDE en SSL/TLS desde finales de los 90 hasta aprox. 2012
- DES es lento y 3DES mucho más; sólo se usa hoy en cifra local o convencional

# Lectura recomendada (1/2)

- Data Encryption Standard, Wikipedia
  - [https://en.wikipedia.org/wiki/Data\\_Encryption\\_Standard#Chronology](https://en.wikipedia.org/wiki/Data_Encryption_Standard#Chronology)
- Divide-and-conquer algorithm, Wikipedia
  - [https://en.wikipedia.org/wiki/Divide-and-conquer\\_algorithm#Parallelism](https://en.wikipedia.org/wiki/Divide-and-conquer_algorithm#Parallelism)
- CLCRIPT Cuadernos de Laboratorio de Criptografía, CLCRIPT 12, Jorge Ramió, 2019
  - [https://www.criptored.es/software/sw\\_m001s.htm](https://www.criptored.es/software/sw_m001s.htm)
- ASCII Text to Hex Code Converter, RapidTables
  - <https://www.rapidtables.com/convert/number/ascii-to-hex.html>
- RSA's DES Challenge III, Wayback Machine
  - <https://web.archive.org/web/20160317201151/https://www.emc.com/emc-plus/rsa-labs/historical/des-challenge-iii.htm>

# Lectura recomendada (2/2)

- Guion píldora formativa Thoth nº 29, ¿Por qué sucumbe el DES ante un ataque en red?, Jorge Ramió, 2015
  - <https://www.criptored.es/thoth/material/texto/pildora029.pdf>
- Documento Histórico\_DES\_Challenge, Miguel Ángel Jiménez, Jorge Ramió, 2019
  - [https://www.criptored.es/descarga/Historico\\_DES\\_Challenge\\_safeDES.pdf](https://www.criptored.es/descarga/Historico_DES_Challenge_safeDES.pdf)
- Guía de Seguridad de las TIC CCN-STIC 807, Criptología de empleo en el Esquema Nacional de Seguridad, Centro Criptológico Nacional, España, 2017
  - <https://www.ccn-cert.cni.es/series-ccn-stic/800-guia-esquema-nacional-de-seguridad/513-ccn-stic-807-criptologia-de-empleo-en-el-ens/file.html>
- Performance Analysis of Data Encryption Algorithms, Comparison results using Crypto++, Abdel-Karim Al Tamimi
  - [http://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption\\_perf/index.html](http://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption_perf/index.html)
- Javamex: Comparison of ciphers
  - <https://www.javamex.com/tutorials/cryptography/ciphers.shtml>

# Class4crypt c4c8.6a

## Módulo 8. Criptografía simétrica en bloque

### Lección 8.6a. Algoritmo AES parte 1: visión general y fortaleza

8.6a.1. El concurso del NIST para el Advanced Encryption Standard

8.6a.2. Características del algoritmo

8.6a.3. Esquemas de cifrado y de descifrado

8.6a.4. Relleno PKCS#7

8.6a.5. Operaciones de cifrado y de descifrado

8.6a.6. Consideraciones sobre la fortaleza del algoritmo

Class4crypt c4c8.6a Algoritmo AES parte 1: visión general y fortaleza  
<https://www.youtube.com/watch?v=pbmMo3wwp9A>

# El nuevo estándar AES y DES Challenge

- El DES, estándar de cifra simétrica desde 1976 por la National Bureau of Standards NBS, pasa la certificación en 1987 y en 1993
- En 1997 el National Institute of Standards and Technology NIST, antigua NBS, no certifica al DES y llama a concurso público para proveer un nuevo algoritmo estándar de cifra que se llamará AES, acrónimo de Advanced Encryption Standard
- El concurso dura 2 años y participan 15 algoritmos candidatos
- Esto está relacionado con los cuatro ataques distribuidos en red por fuerza bruta que sufre el DES, que comienzan en enero de 1997 y terminan en enero de 1999, cuando se logra romper la clave de 56 bits en menos de 24 horas



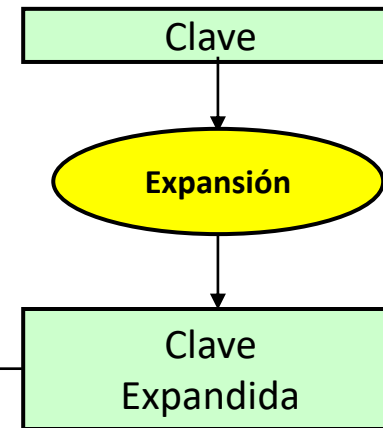
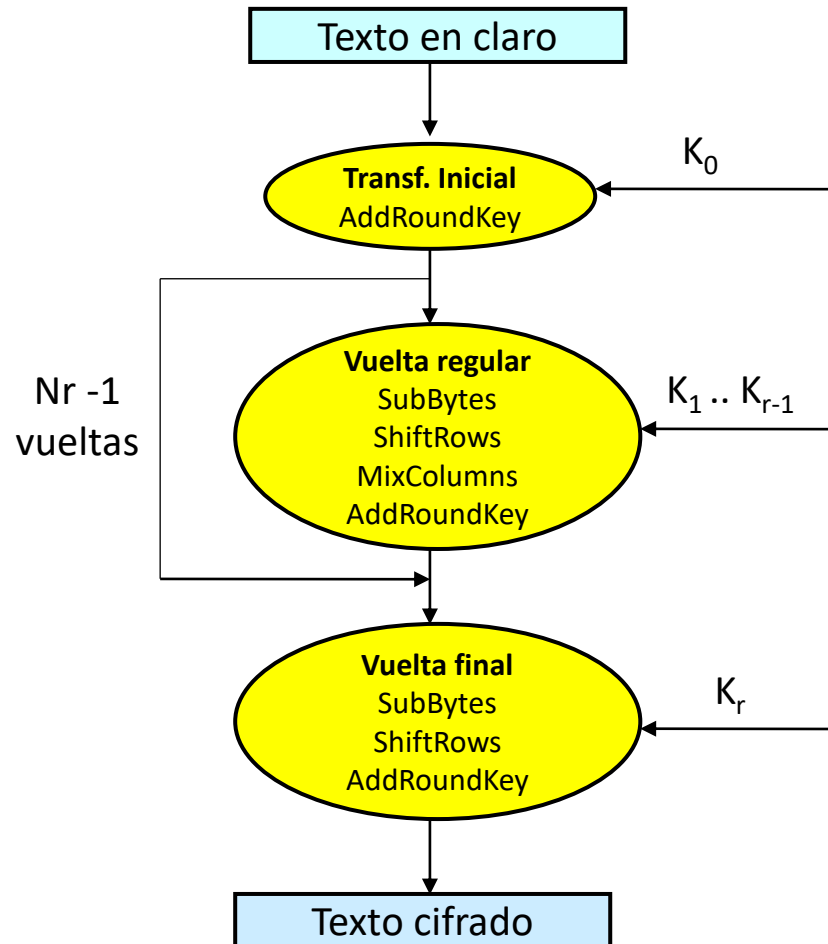
# El NIST selecciona al nuevo estándar

- Los 15 algoritmos que participan en el concurso son CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+, Serpent y Twofish
- Entre los finalistas MARS, RC6, Rijndael, Serpent y Twofish, en octubre de 2000 se anuncia que el ganador es el algoritmo de los investigadores belgas Vincent Rijmen y Joan Daemen, Rijndael
- En noviembre de 2001 el NIST anuncia el nuevo estándar AES
- AES se hace popular por su seguridad y velocidad, compitiendo en aquella década con el algoritmo de flujo RC4 hoy en desuso. Pero no es hasta comienzos de la década de 2010 en que AES comienza a usarse masivamente en protocolos seguros como TLS

# Características del AES

- Cifrador de producto (permutación + sustitución) no tipo Feistel
- Implementado para trabajar en los procesadores de 8 bits usados en tarjetas inteligentes y en CPUs de 32 bits
- Tamaño de clave variable de 128, 192 y 256 bits, valores estándar, o bien múltiplo de 4 bytes
- Tamaño del bloque de texto de 128 bits o múltiplo de 4 bytes
- Operaciones modulares a nivel de byte (representación en forma de polinomios) y con palabras de 4 bytes, es decir 32 bits
- Número de vueltas flexible según las necesidades del usuario
- Usa 4 funciones invertibles para provocar difusión y confusión

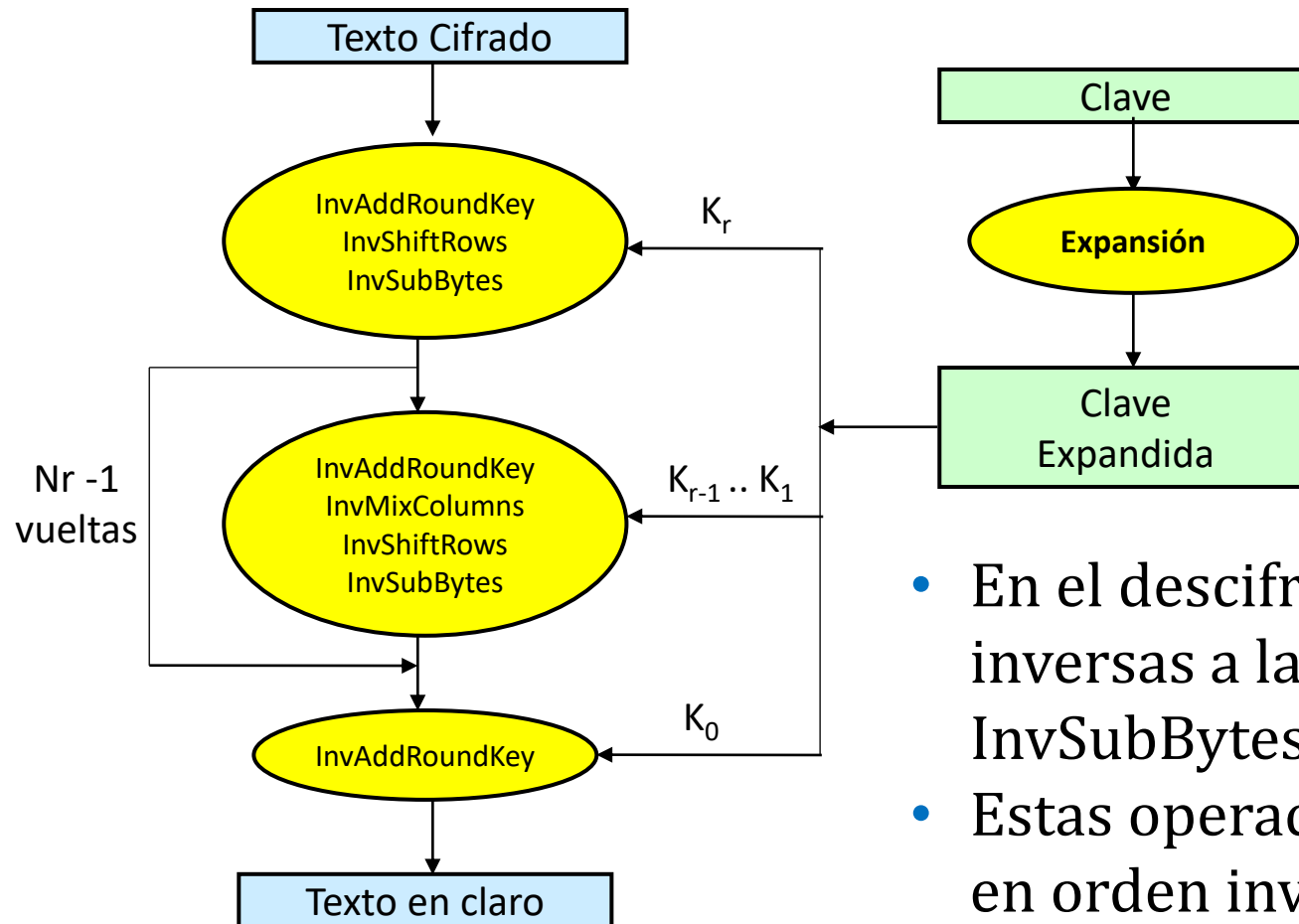
# Esquema del AES en el cifrado



- Los 128 bits del texto en claro se mezclan con los bits de una clave de vuelta siempre de 128 bits, sea  $K$  igual a 128, 192 o 256 bits

- Las claves  $K_r$  de cada vuelta se obtienen con un algoritmo de expansión de claves
- Se aplican 4 funciones: **AddRoundKey**, **SubBytes**, **ShiftRows** y **MixColumns**
- Operaciones de sustitución y permutación con polinomios (campos de Galois)

# Esquema del AES en el descifrado



- Las cuatro funciones empleadas AddRoundKey, SubBytes, ShiftRows y MixColumns son fácilmente invertibles


- En el descifrado se usarán las operaciones inversas a las del cifrado: InvAddRoundKey, InvSubBytes, InvShiftRows, InvMixColumns
- Estas operaciones de descifrado se realizan en orden inverso al usado en el cifrado

# Transformaciones o capas en AES

- Hay tres transformaciones distintas llamadas capas, en las que se tratan los bits
  - Capa de Mezcla Lineal: en ella se busca la difusión de los bits
  - Capa No Lineal: se trata de una zona similar a las cajas S del DES
  - Capa Clave: operaciones con una función or exclusivo de la subclave y la información de esta etapa intermedia
- Las transformaciones realizadas en cada paso del algoritmo se denominan estados
- Estos estados se representan por una matriz de 4 filas y  $N_b = 4$  columnas para el texto en claro, y 4 filas y  $N_k = 4, 6$  u  $8$  columnas para las claves

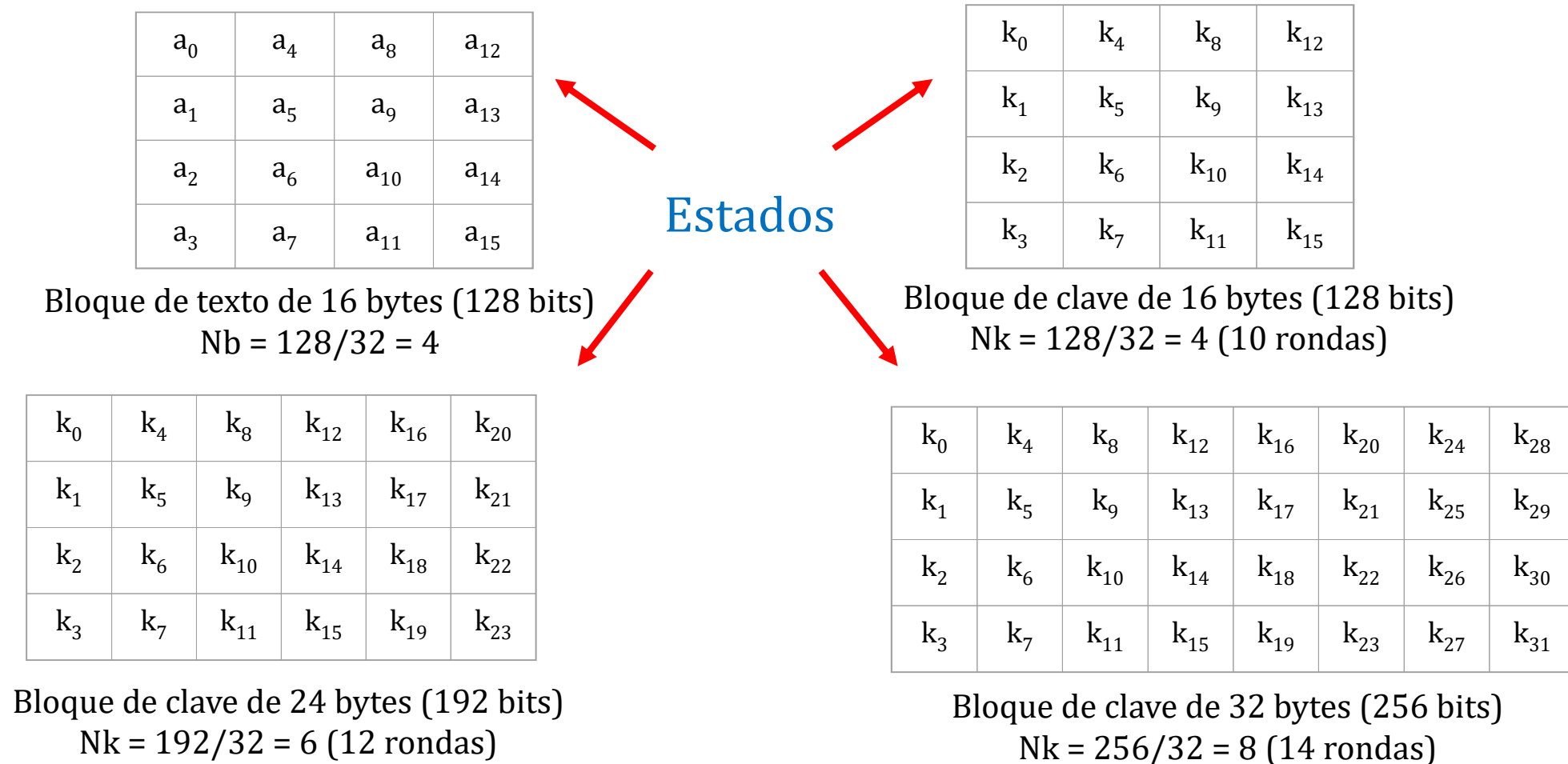
# Matriz de estado para bloques de 16 bytes

- Se trabajará con una matriz de 4 filas por 4 columnas y en cada celda habrá 1 byte, 8 bits
- Se conoce como estado de la matriz al resultado obtenido en cada uno de los pasos u operaciones realizadas
- Se almacenan y leen en la matriz los bytes del bloque de texto en claro por columnas, es decir, según el orden  $S_{0,0}, S_{1,0}, S_{2,0}, S_{3,0}, S_{0,1}, S_{1,1}, S_{2,1}, S_{3,1}, S_{0,2}, S_{1,2}, S_{2,2}, S_{3,2}, S_{0,3}, S_{1,3}, S_{2,3}, S_{3,3}$
- El resultado de la cifra del bloque de texto en claro será el último estado de la matriz



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

# Estados de entrada y claves típicos



# Funciones y estados en bloque de 16 bytes

- Para las funciones de cifrado y descifrado se usarán 4 transformaciones orientadas a bytes
  - Añadido de una clave de vuelta al estado: AddRoundKey
  - Sustitución de un byte mediante una tabla S-box: SubBytes
  - Desplazamiento de filas de un estado: ShiftRows
  - Mezcla de datos dentro de cada columna de estado: MixColumns

Combinaciones posibles de estados en AES	Longitud del bloque (Nb palabras)	Longitud de la clave (Nk palabras)	Número de Rondas (Nr)
AES – 128	4	4	10
AES – 192	4	6	12
AES – 256	4	8	14



# Relleno PKCS#7

- Al cifrar bloques, habrá que considerar un posible relleno en el último bloque
- A diferencia del DES, cuyo relleno Zero padding con ceros se usaba solamente si era necesario, es decir, si en el último bloque de texto en claro había menos de 8 bytes, en AES se usa el relleno PKCS#7 que es obligatorio incluir, incluso si el tamaño del documento es congruente con el bloque a cifrar
- PKCS#7 añade un relleno indicando en hexadecimal los bytes que faltan para completar el último bloque, repitiendo ese número la misma cantidad de veces que bytes nos faltan para completar el bloque
- Un byte de relleno se indica como 01, dos bytes 0202, tres bytes 030303, cuatro bytes 04040404... y 15 bytes 0F0F0F0F0F0F0F0F0F0F0F0F0F0F
- Si el texto en claro tiene 128 bits (16 bytes) o un múltiplo de ese valor, se añadirá un último bloque sólo de relleno con 16 valores 0x 10

# Ejercicios prácticos con AESphere

- $M_1$  = La chica de Ipanema;  $M_2$  = Chica de Ipanema
- $K = 0x\ 0123456789ABCDEF0123456789ABCDEF$  (128 bits)
- $IV = 0x\ AAAABBBBCCCCDDDDDEEEFFFF12345678$  (128 bits)



- Cifrar y descifrar en modo ECB y CBC, de forma Directa y de forma Paso a Paso
- Comprobar los rellenos en cada caso
- Comprobar la operación AddRoundKey en  $K_0$  del ejemplo del documento oficial del NIST y el criptograma final, que aparecen en las diapositivas siguientes

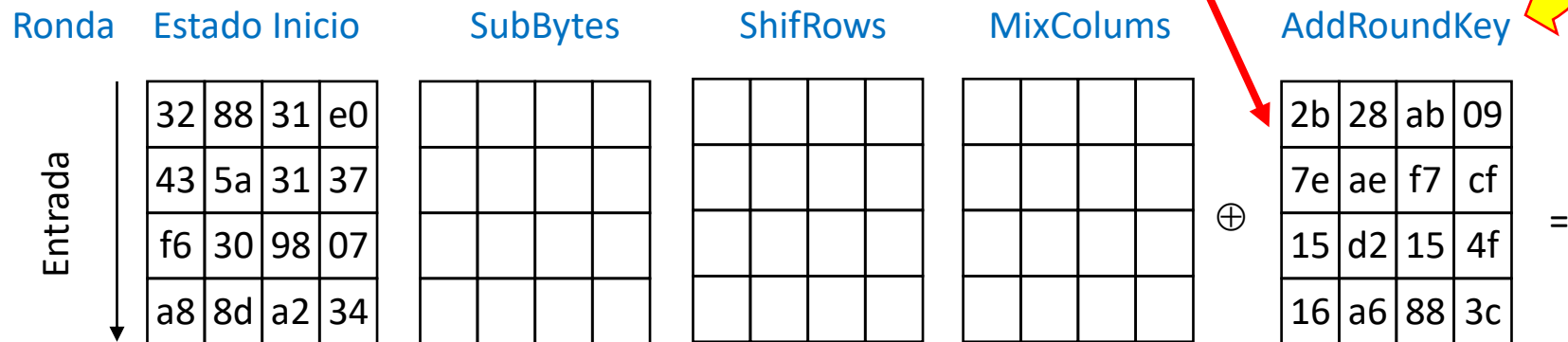
# Ejemplo documento oficial del NIST (1/2)

Si el bloque de entrada y la clave son de 128 bits, ( $N_b = 4$  y  $N_k = 4$ ) con valores

Entrada: 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34

Clave: 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

entonces



1

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

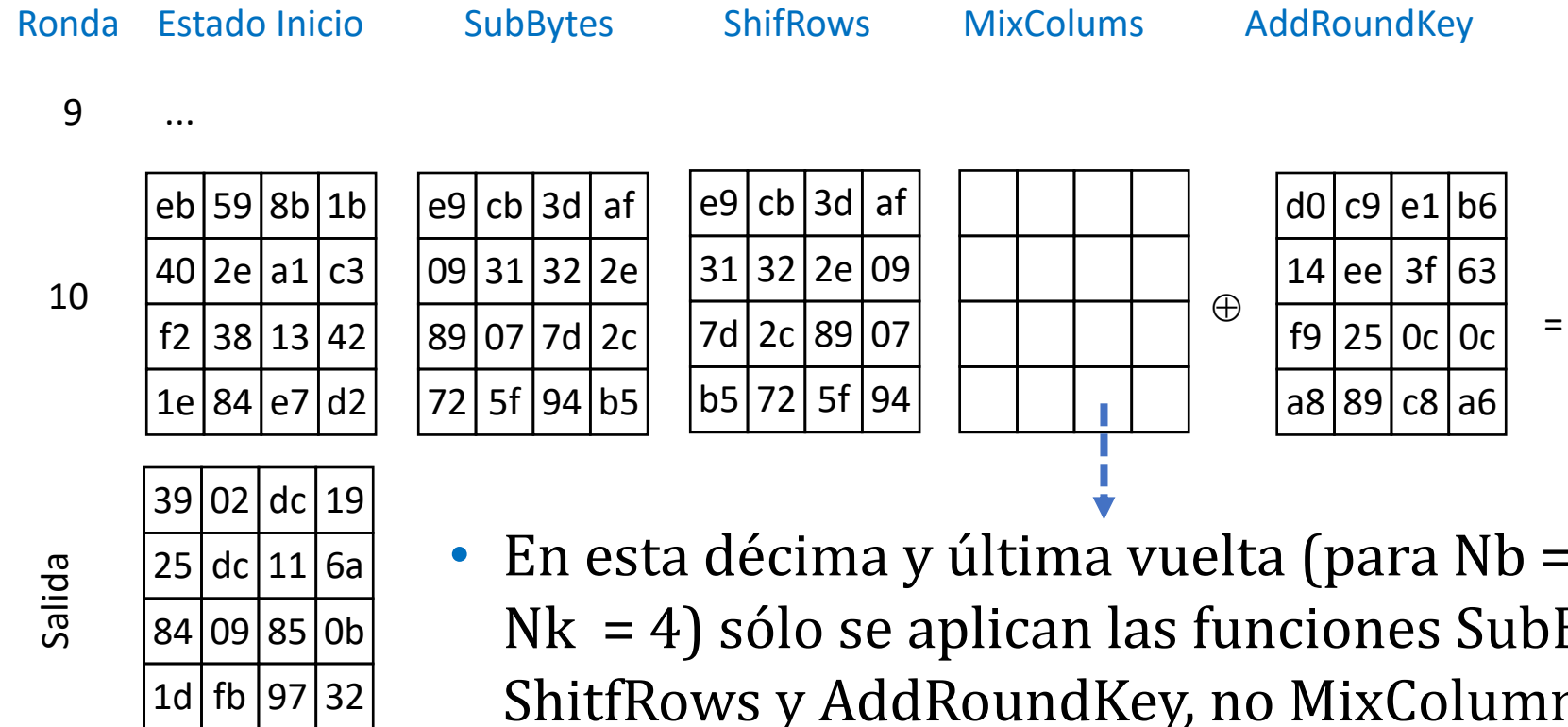
El primer valor del estado siguiente

$S'_{0,0}$  será 32 XOR 2b

$$\begin{array}{r} 0011\ 0010 \\ \oplus\ 0010\ 1011 \\ \hline = 0001\ 1001 = 19 \end{array}$$

La vuelta 10 y el criptograma final se muestran en la próxima diapositiva

# Ejemplo documento oficial del NIST (2/2)

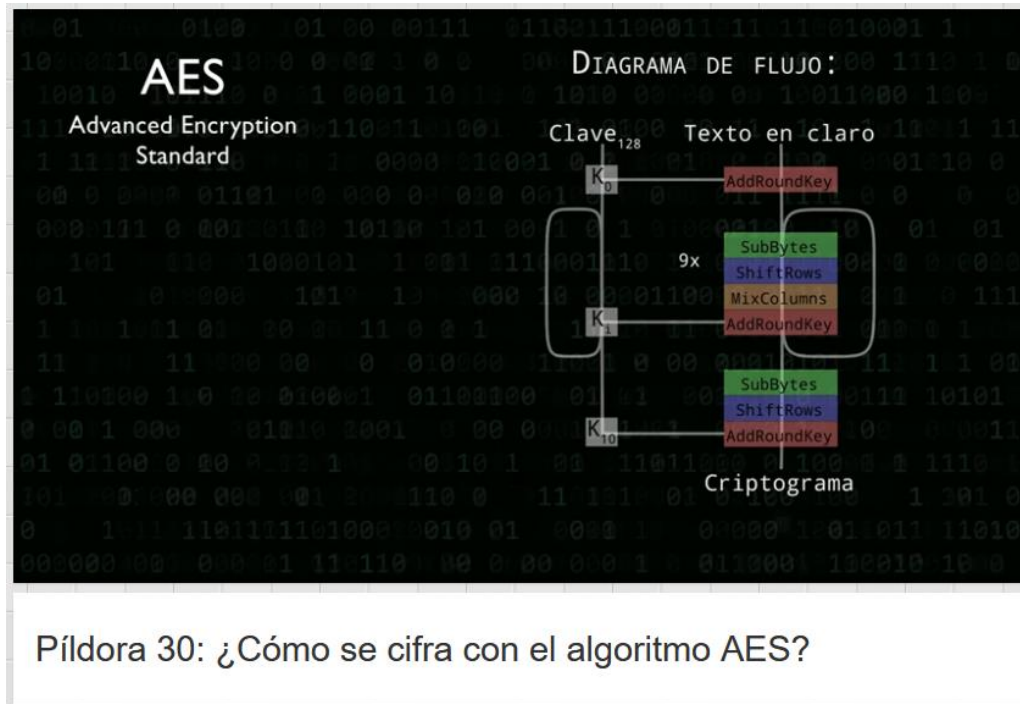


- En esta décima y última vuelta (para  $N_b = 4$  y  $N_k = 4$ ) sólo se aplican las funciones SubBytes, ShiftRows y AddRoundKey, no MixColumns

# ¿Qué tan seguro es AES en 2021?

- Seguridad asociada a la longitud de la clave ante ataques por fuerza bruta
  - Para  $K = 128$  bits, fortaleza de  $2^{127}$  operaciones en media
  - Para  $K = 192$  bits, fortaleza de  $2^{191}$  operaciones en media
  - Para  $K = 256$  bits, fortaleza de  $2^{255}$  operaciones en media
- Existen ataques conocidos pero siempre con menos rondas que las estándar 10, 12 y 14 según la clave, aunque están muy cerca
- Existen otros ataques por canal lateral que podrían considerarse
- Controversia por la pocas vueltas del AES, comentarios de Bruce Schneier y AES256 en archivo seguro de vida de Julen Assange

# Más información en píldoras Thoth



<https://www.youtube.com/watch?v=tzj1RoqRnv0>



[https://www.youtube.com/watch?v=AFYhB\\_MjZLw](https://www.youtube.com/watch?v=AFYhB_MjZLw)

# Conclusiones de la Lección 8.6a

- El algoritmo Rijndael se proclama en 2000 como el nuevo estándar de cifra simétrica en bloque AES, Advanced Encryption Standard, después de los 4 ataques sufridos por DES en el DES Challenge, desde 1997 hasta 1999
- Actualmente AES trabaja con bloques de texto de 128 bits y claves estándar de 128, 192 y 256 bits, que pueden aumentar de tamaño en el futuro
- Cifra con las funciones AddRoundKey, SubBytes, ShiftRows y MixColumns y descifra con InvAddRoundKey, InvSubBytes, InvShiftRows e InvMixColumns
- Para bloques de 128 bits, opera sobre una matriz de estado de 4 filas por 4 columnas, es decir 16 bytes, con 4 palabras de 32 bits leídas en columnas
- Para claves de 128, 192 y 256 bits se usan 10, 12 y 14 vueltas, empleando en cada vuelta una clave diferente mediante un algoritmo de expansión de clave



# Lectura recomendada (1/3)

- Guion píldora formativa Thoth nº 30, ¿Cómo se cifra con el algoritmo AES?, Jorge Ramió, 2015
  - <https://www.criptored.es/thoth/material/texto/pildora030.pdf>
- Guion píldora formativa Thoth nº 31, ¿Qué son los rellenos y los modos de cifra en bloque?, Jorge Ramió, 2015
  - <https://www.criptored.es/thoth/material/texto/pildora031.pdf>
- Cryptographic Standards and Guidelines, AES Development, Computer Security Resource Center CSRC, NIST
  - <https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development>



# Lectura recomendada (2/3)

- Federal Information, Processing Standards Publication 197, Announcing the Advanced Encryption Standard AES, 2001
  - <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- Advanced Encryption Standard, Wikipedia
  - [https://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)
- AESphere, A. Gómez, R. Sierra, dirección J. Ramió, 2014
  - [https://www.criptored.es/software/sw\\_m001p.htm](https://www.criptored.es/software/sw_m001p.htm)
- The Rijndael Animation, Rijndael Inspector, E. Zabala, 2008
  - <http://www.formaestudio.com/rijndaelinspector/>
- ¿Cómo de seguro es la seguridad de 256 bits?, 3Blue1Brown, 2017
  - [https://www.youtube.com/watch?v=S9JGmA5\\_unY](https://www.youtube.com/watch?v=S9JGmA5_unY)

# Lectura recomendada (3/3)

- Another New AES Attack, Blog Schneier on Security, Bruce Schneier, July 2009
  - [https://www.schneier.com/blog/archives/2009/07/another\\_new\\_aes.html](https://www.schneier.com/blog/archives/2009/07/another_new_aes.html)
- ¿Fue una buena idea usar AES256 con el archivo INSURANCE de Wikileaks?, Fernando Acero, 2010
  - <https://fernando-acero.livejournal.com/78069.html>
- Wikileaks, Assange y la tecnología y criptografía que rodean al fichero secreto Insurance.AES256, microsiervos, 2011
  - <https://www.microsiervos.com/archivo/seguridad/wikileaks-assange-criptografia.html>

# Class4crypt c4c8.6b

## Módulo 8. Criptografía simétrica en bloque

### Lección 8.6b. Algoritmo AES parte 2: Campos de Galois y expansión de clave

8.6b.1. Resumen y esquema del AES

8.6b.2. Representación de bytes en GF

8.6b.3. Operaciones de suma y multiplicación en campos de Galois

8.6b.4. Inversos en GF ( $2^8$ )

8.6b.5. Funciones RotWord y Rcon

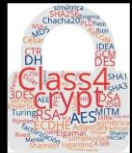
8.6b.6. Algoritmo de expansión de clave

Class4crypt c4c8.6b Algoritmo AES parte 2: Campos de Galois y expansión de clave

[https://www.youtube.com/watch?v=Elm62\\_ec4MU](https://www.youtube.com/watch?v=Elm62_ec4MU)

# Resumen y esquema del AES

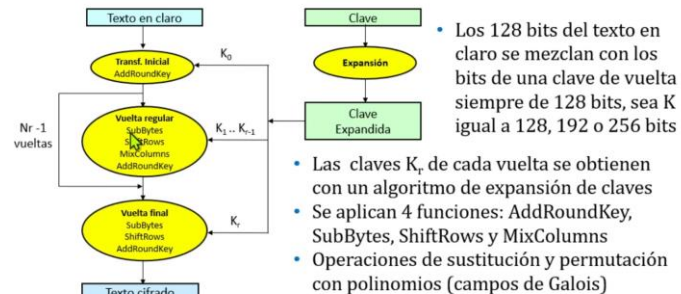
Módulo 8. Criptografía simétrica en bloque  
Lección 8.6a Algoritmo AES parte 1: visión general y fortaleza



Clase c4c8.6a  
26/05/2021

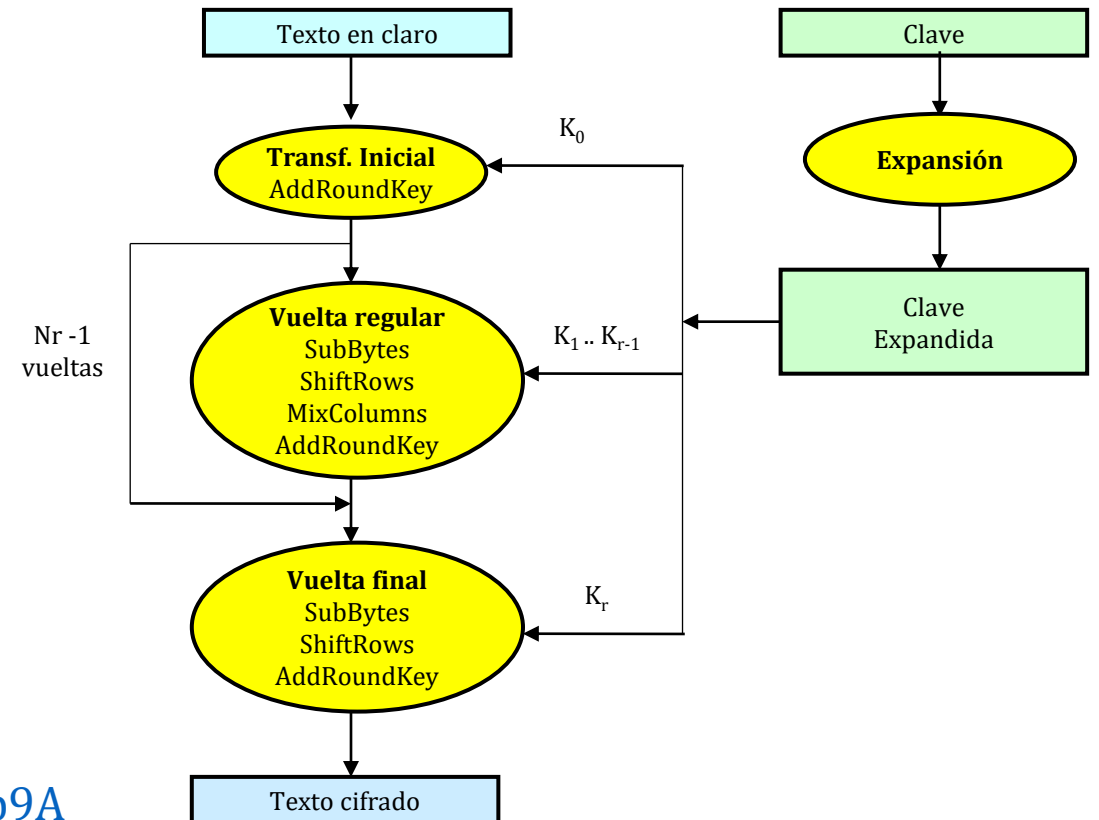


## Esquema del AES en el cifrado



Class4crypt c4c8.6a - © jorgeramio 2021

Lección 8.6a - página 13



- Para repasar conceptos y características del AES:
- <https://www.youtube.com/watch?v=pbmMo3wwp9A>

# Campos de Galois GF ( $2^8$ ) en AES

- En AES la unidad básica de tratamiento será el byte
- Los campos de Galois son interesantes en criptografía porque sus elementos tendrán inverso aditivo y multiplicativo
- En AES interesará usar GF ( $2^8$ ) para representar los 8 bits de un byte (en hexadecimal) puesto que la base 2 significa restos 0 y 1
- Dado que los bits de un byte son  $b_7b_6b_5b_4b_3b_2b_1b_0$ , si éstos son los coeficientes de un polinomio  $\{b_i = 0, 1\}$ , entonces un byte será:
  - $p(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0$
- Así, el byte  $b9 = 1011\ 1001$  tendrá este polinomio asociado:
  - $p(x) = x^7 + x^5 + x^4 + x^3 + 1$

# Suma en campos Galois $GF(2^8)$

- La suma se realizará módulo 2 y será equivalente a un xor
  - Calcular la suma  $GF(2^8)$  de e7 + 8a con polinomios
    - $e7 = 1110\ 0111 = x^7 + x^6 + x^5 + x^2 + x + 1$
    - $8a = 1000\ 1010 = x^7 + x^3 + x$
    - $(x^7 + x^6 + x^5 + x^2 + x + 1) + (x^7 + x^3 + x) \bmod 2$
    - $2x^7 + x^6 + x^5 + x^3 + x^2 + 2x + 1 \bmod 2$
    - $e7 + 8a = x^6 + x^5 + x^3 + x^2 + 1 = 0110\ 1101 = 6d$
  - Calcular la suma  $e7 \oplus 8a$ 
    - $e7 = 1110\ 0111$
    - $8a = 1000\ 1010$   
 $= 0110\ 1101 = x^6 + x^5 + x^3 + x^2 + 1$
- Al ser una suma XOR, el valor máximo del polinomio será siempre  $x^7$ , luego nunca habrá problemas con la potencia

# Multiplicación en campos Galois $GF(2^8)$

- Al realizar una multiplicación con dos polinomios en  $GF(2^8)$  podemos tener elementos que estén fuera del cuerpo
- Por ejemplo  $\{57\} * \{83\} = 0101\ 0111 \times 1000\ 0011$ 
  - $\{57\} * \{83\} = (x^6 + x^4 + x^2 + x + 1) * (x^7 + x + 1)$
  - $(x^6x^7 + x^6x + x^6) + (x^4x^7 + x^4x + x^4) + (x^2x^7 + x^2x + x^2) + (xx^7 + xx + x) + (x^7 + x + 1)$
  - $x^{13} + x^{11} + x^9 + x^8 + 2x^7 + x^6 + x^5 + x^4 + x^3 + 2x^2 + 2x + 1 \text{ mod } 2$
  - $\{57\} * \{83\} = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$
- Como  $x^{13}$ ,  $x^{11}$ ,  $x^9$  y  $x^8$  están fuera del cuerpo, se reducirán por el polinomio primitivo  $m(x) = x^8 + x^4 + x^3 + x + 1$
- Es decir, reemplazamos  $x^8$  por  $(x^4 + x^3 + x + 1)$

# Reduciendo $x^{13}$ , $x^{11}$ y $x^9$ con $m(x)$ (1/2)

- $x^{13} = x^5 * x^8 = x^5 * (x^4 + x^3 + x + 1) = x^9 + x^8 + x^6 + x^5$
- $x^{13} = x^9 + x^8 + x^6 + x^5 = x * x^8 + x^8 + x^6 + x^5$
- $x^{13} = x * (x^4 + x^3 + x + 1) + (x^4 + x^3 + x + 1) + x^6 + x^5$
- $x^{13} = (x^5 + x^4 + x^2 + x) + (x^4 + x^3 + x + 1) + x^6 + x^5$
- $x^{13} = x^6 + 2x^5 + x^3 + x^2 + 2x + 1 \text{ mod } 2$
- $x^{13} = x^6 + x^3 + x^2 + 1$
- $x^{11} = x^3 * x^8 = x^3 * (x^4 + x^3 + x + 1) = x^7 + x^6 + x^4 + x^3 \text{ mod } 2$
- $x^{11} = x^7 + x^6 + x^4 + x^3$
- $x^9 = x * x^8 = x * (x^4 + x^3 + x + 1) = x^5 + x^4 + x^2 + x \text{ mod } 2$
- $x^9 = x^5 + x^4 + x^2 + x$



# Reduciendo $x^{13}$ , $x^{11}$ y $x^9$ con $m(x)$ (2/2)

- Reemplazando  $x^{13}$ ,  $x^{11}$ ,  $x^9$  y  $x^8$  en  $\{57\}*\{83\}$  tenemos:
- $\{57\}*\{83\} = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$
- $\{57\}*\{83\} = (x^6 + x^3 + x^2 + 1) + (x^7 + x^6 + x^4 + x^3) + (x^5 + x^4 + x^2 + x) + (x^4 + x^3 + x + 1) + x^6 + x^5 + x^4 + x^3 + 1$
- $\{57\}*\{83\} = x^7 + 3x^6 + 2x^5 + 4x^4 + 4x^3 + 2x^2 + 2x + 3 \text{ mod } 2$
- $\{57\}*\{83\} = x^7 + x^6 + 1$
- $\{57\}*\{83\} = 1100\ 0001 = c1$
- Esta multiplicación con reducción polinómica nos va a permitir encontrar los inversos de cada byte, desde 00 hasta FF en  $m(x)$

# Comprobación inv (44, (m(x)) = 2d (1/2)

- Vamos multiplicar  $\{44\} * \{2d\} \bmod m(x)$
- $\{44\} * \{2d\} = \{0100\ 0100\} \{0010\ 1101\}$
- $\{44\} * \{2d\} = \{x^6 + x^2\} * \{x^5 + x^3 + x^2 + 1\}$
- $\{44\} * \{2d\} = (x^6x^5 + x^6x^3 + x^6x^2 + x^6) + (x^2x^5 + x^2x^3 + x^2x^2 + x^2)$
- $\{44\} * \{2d\} = x^{11} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^2 \bmod 2$
- $\{44\} * \{2d\} = x^{11} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^2$
- Reduciendo mod  $m(x)$ ,  $x^8 = (x^4 + x^3 + x + 1)$
- $x^{11} = x^3 * x^8 = x^3 * (x^4 + x^3 + x + 1) = x^7 + x^6 + x^4 + x^3$
- $x^9 = x * x^8 = x * (x^4 + x^3 + x + 1) = x^5 + x^4 + x^2 + x$

# Comprobación inv (44, (m(x)) = 2d (2/2)

- Reemplazando en  $\{44\} * \{2d\} = x^{11} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^2$
- $\{44\} * \{2d\} = (x^7 + x^6 + x^4 + x^3) + (x^5 + x^4 + x^2 + x) + (x^4 + x^3 + x + 1) + x^7 + x^6 + x^5 + x^4 + x^2$
- $\{44\} * \{2d\} = 2x^7 + 2x^6 + 2x^5 + 4x^4 + 2x^3 + 2x^2 + 2x + 1 \text{ mod } 2$
- $\{44\} * \{2d\} = 1$
- Por lo tanto, se comprueba que  $\text{inv}(44, m(x)) = 2d$
- Repitiendo esta operación con todos los bytes desde {00} hasta {ff}, obtenemos la tabla de inversos en  $\text{GF}(2^8)$  que se muestra en la siguiente diapositiva, necesaria para la función SubBytes

# Tabla de inversos en GF ( $2^8$ )

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	00	01	8d	f6	cb	52	7b	d1	e8	4f	29	c0	b0	e1	e5	c7
1	74	b4	aa	4b	99	2b	60	5f	58	3f	fd	cc	ff	40	ee	b2
2	3a	6e	5a	f1	55	4d	a8	c9	c1	0a	98	15	30	44	a2	c2
3	2c	45	92	6c	f3	39	66	42	f2	35	20	6f	77	bb	59	19
4	1d	fe	37	67	2d	31	f5	69	a7	64	ab	13	54	25	e9	09
5	ed	5c	05	ca	4c	24	87	bf	18	3e	22	f0	51	ec	61	17
6	16	5e	af	d3	49	a6	36	43	f4	47	91	df	33	93	21	3b
7	79	b7	97	85	10	b5	ba	3c	b6	70	d0	06	a1	fa	81	82
8	83	7e	7f	80	96	73	be	56	9b	9e	95	d9	f7	02	b9	a4
9	de	6a	32	6d	d8	8a	84	72	2a	14	9f	88	f9	dc	89	9a
a	fb	7c	2e	c3	8f	b8	65	48	26	c8	12	4a	ce	e7	d2	62
b	0c	e0	1f	ef	11	75	78	71	a5	8e	76	3d	bd	bc	86	57
c	0b	28	2f	a3	da	d4	e4	0f	a9	27	53	04	1b	fc	ac	e6
d	7a	07	ae	63	c5	db	e2	ea	94	8b	c4	d5	9d	f8	90	6b
e	b1	0d	d6	eb	c6	0e	cf	ad	08	4e	d7	e3	5d	50	1e	b3
f	5b	23	38	34	68	46	03	8c	dd	9c	7d	a0	cd	1a	41	1c

- Se cumple que:
- Si  $\text{inv}(A, m(x)) = B$
- Entonces  $\text{inv}(B, m(x)) = A$
- En la figura:
  - $\text{inv}(c4, m(x)) = da$
  - $\text{inv}(da, m(x)) = c4$
  - $da * c4 \bmod m(x) = 1$
- Además, es fácil comprobar:
  - $\text{inv}(00, m(x)) = 00$
  - $\text{inv}(01, m(x)) = 01$

# Expansión de claves en AES

- Número de bits de las subclaves para valores estándar de  $N_b$  y  $N_k$

Bloque / Clave	$N_k = 4$ (128 bits)	$N_k = 6$ (192 bits)	$N_k = 8$ (256 bits)
$N_b = 4$ 128 bits	$N_r = 10$ 1.408 bits	$N_r = 12$ 1.664 bits	$N_r = 14$ 1.920 bits
$N_b = 6$ 192 bits	$N_r = 12$ 2.304 bits	$N_r = 12$ 2.496 bits	$N_r = 14$ 2.880 bits
$N_b = 8$ 256 bits	$N_r = 14$ 3.840 bits	$N_r = 14$ 3.328 bits	$N_r = 14$ 3.840 bits

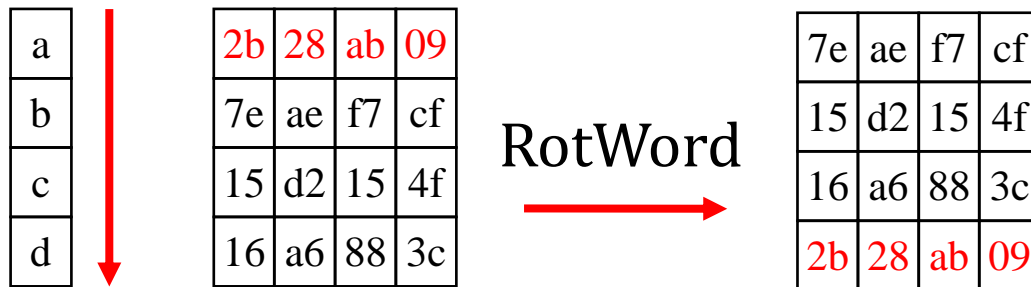
- La expansión genera los bytes de las subclaves a partir de la clave  $K$  principal
- Será un array lineal  $W$  de palabras de 4 bytes y con longitud  $N_b \cdot (N_r + 1)$

$W_0$	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$	$W_7$	$W_8$	$W_9$	$W_{10}$	$W_{11}$	$W_{12}$	$W_{13}$	$W_{14}$	...
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	----------	----------	----------	----------	----------	-----

# RotWord y Rcon para expansión de claves

- RotWord rota una posición a la izquierda los bytes de la palabra
- Si la palabra es (a, b, c, d) RotWord devolverá (b, c, d, a)

Si K: 2b7e1516 28aed2a6  
abf71588 09cf4f3c



- Rcon genera la constante  $Rcon(j) = [R(j), \{00\}, \{00\}, \{00\}]$  de 32 bits
- $R(j)$  es el elemento  $GF(2^8)$  correspondiente al valor  $x^{j-1}$
- Puede ver un ejemplo de  $Rcon(j)$  en una próxima diapositiva, cuando se calcule  $W(4)$  y  $W(5)$

# Algoritmo de expansión de claves

- Las primeras  $N_k$  palabras se copiarán de la clave principal. Y las restantes  $N_b \cdot (N_r + 1) - N_k$  palabras, se generarán con este algoritmo
- Si  $N_k \leq 6$ 
  - Si la posición  $i$  dentro del array  $W(i)$  es múltiplo del valor  $N_k$ :
  - $W(i) = W(i - N_k) \text{ xor } [\text{SubByte}(\text{RotWord}[W(i - 1)]) \text{ xor } \text{Rcon}(i/N_k)]$
  - Si la posición  $i$  dentro del array  $W(i)$  no es múltiplo del valor  $N_k$ :
  - $W(i) = W(i - N_k) \text{ xor } W(i - 1)$
- Si  $N_k > 6$ 
  - El valor de la variable  $i$  debe satisfacer la expresión  $i \bmod N_k = 4$
  - Las palabras de subclaves se calcularán:
  - $W(i) = W(i - N_k) \text{ xor } \text{SubByte}[W(i - 1)]$

# Expansión $W(4)$ para $K = 128$ bits - NIST\*

(\*) Documento oficial Announcing the AES, Appendix A - Key Expansion Examples, NIST, Nov 26, 2001

Sea  $K$ : 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c ( $N_k = 4$ )

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

$W(0)$	$W(1)$	$W(2)$	$W(3)$
--------	--------	--------	--------

$\Rightarrow$

$W(0) = 2b\ 7e\ 15\ 16$

$W(1) = 28\ ae\ d2\ a6$

$W(2) = ab\ f7\ 15\ 88$

$W(3) = 09\ cf\ 4f\ 3c$

Cálculo de  $W(4)$  con  $i = 4$ , múltiplo de  $N_k$

- Registro temp =  $W(3) = 09\ cf\ 4f\ 3c$
- RotWord (temp) =  $cf\ 4c\ 3c\ 09 \rightarrow$  temp
- SubByte (temp) =  $8a\ 84\ eb\ 01 \rightarrow$  temp
- $Rcon(4/4) = Rcon(1); j = 1 \Rightarrow x^{j-1} = x^0 = 01$
- $Rcon(1) = [01, 00, 00, 00]$
- $Rcon(1) \text{ xor temp} = 8b\ 84\ eb\ 01 \rightarrow$  temp
- $W(4) = W(0) \text{ xor temp}$

$$\begin{array}{r} 2b\ 7e\ 15\ 16 \\ \oplus\ 8b\ 84\ eb\ 01 \\ \hline a0\ fa\ fe\ 17 \end{array}$$

a0			
fa			
fe			
17			

$W(4) = a0\ fa\ fe\ 17$



# Expansión $W(5)$ para $K = 128$ bits - NIST\*

(\*) Documento oficial Announcing the AES, Appendix A - Key Expansion Examples, NIST, Nov 26, 2001

Sea  $K$ : 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c ( $N_k = 4$ )

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

$W(0)$	$W(1)$	$W(2)$	$W(3)$
--------	--------	--------	--------

$\Rightarrow$

$W(0) = 2b\ 7e\ 15\ 16$

$W(1) = 28\ ae\ d2\ a6$

$W(2) = ab\ f7\ 15\ 88$

$W(3) = 09\ cf\ 4f\ 3c$

Cálculo de  $W(5)$  con  $i = 5$ , no múltiplo de  $N_k$

- $W(i) = W(i - N_k) \text{ xor } W(i - 1)$
- $W(5) = W(5 - 4) \text{ xor } W(5 - 1)$
- $W(5) = W(1) \text{ xor } W(4)$
- Como  $W(4) = \text{a0 fa fe 17}$
- Observa que para las tres columnas (palabras) que faltan de la clave de esta vuelta, columnas 1, 2 y 3, solo se hace un xor de la clave actual con la palabra tres posiciones a la izquierda

$$\begin{array}{r}
 28\ ae\ d2\ a6 \\
 \oplus\ a0\ fa\ fe\ 17 \\
 \hline
 88\ 54\ 2c\ b1
 \end{array}$$

a0	88		
fa	54		
fe	2c		
17	b1		

$W(4) = a0\ fa\ fe\ 17$

$W(5) = 88\ 54\ 2c\ b1$

# Más información en píldoras Thoth



<https://www.youtube.com/watch?v=tzj1RoqRnv0>

# Conclusiones de la Lección 8.6b

- AES cifra con polinomios y usa 4 funciones fácilmente invertibles para usarlas en el descifrado: SubBytes, ShiftRows, MixColumns y AddRoundKey
- AddRoundKey es una operación xor, suma módulo 2, y ShiftRows se trata de un desplazamiento módulo 4, ambas funciones con inversa directas
- En SubBytes habrá que reducir módulo  $x^8 + x^4 + x^3 + x + 1$  y en MixColumns reducir módulo  $x^4 + 1$ , sus inversas serán nuevas funciones
- El algoritmo de expansión de clave generará nuevas palabras clave para cada vuelta, que operarán con la matriz de estado en la función AddRoundKey
- Dependiendo del tamaño de clave, esta expansión tendrá un funcionamiento diferente, pero siempre se harán operaciones con palabras clave anteriores
- En la expansión de claves se usarán dos nuevas funciones, RotWord y Rcon

# Lectura recomendada

- Guion píldora formativa Thoth nº 30, ¿Cómo se cifra con el algoritmo AES?, Jorge Ramió, 2015
  - <https://www.criptored.es/thoth/material/texto/pildora030.pdf>
- Federal Information, Processing Standards Publication 197, Announcing the Advanced Encryption Standard AES, 2001
  - <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- Criptosistema Rijndael. A Fondo, Alfonso Muñoz, 2004
  - [https://www.criptored.es/guiateoria/gt\\_m480a.htm](https://www.criptored.es/guiateoria/gt_m480a.htm)
- Libro Electrónico de Seguridad Informática y Criptografía, Versión 4.1, Módulo 12 Cifrado Simétrico en Bloque, Jorge Ramió, marzo 2006
  - [https://www.criptored.es/guiateoria/gt\\_m001a.htm](https://www.criptored.es/guiateoria/gt_m001a.htm)

# Class4crypt c4c8.6c

## Módulo 8. Criptografía simétrica en bloque

### Lección 8.6c. AES parte 3: SubBytes, ShiftRows, MixColumns, AddRoundKey

8.6c.1. Esquema del AES y recuerdo de operaciones en GF ( $2^8$ )

8.6c.2. Función SubBytes

8.6c.3. Función ShiftRows

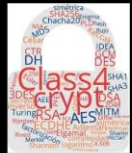
8.6c.4. Función MixColumns

8.6c.5. Función AddRoundKey

Class4crypt c4c8.6c Algoritmo AES parte 3: SubBytes, ShiftRows, MixColumns, AddRoundKey  
<https://www.youtube.com/watch?v=ZCcN4sr4IMw>

# Resumen y esquema del AES

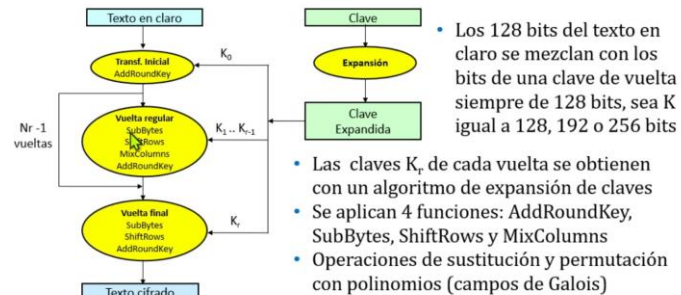
Módulo 8. Criptografía simétrica en bloque  
Lección 8.6a Algoritmo AES parte 1: visión general y fortaleza



Clase c4c8.6a  
26/05/2021

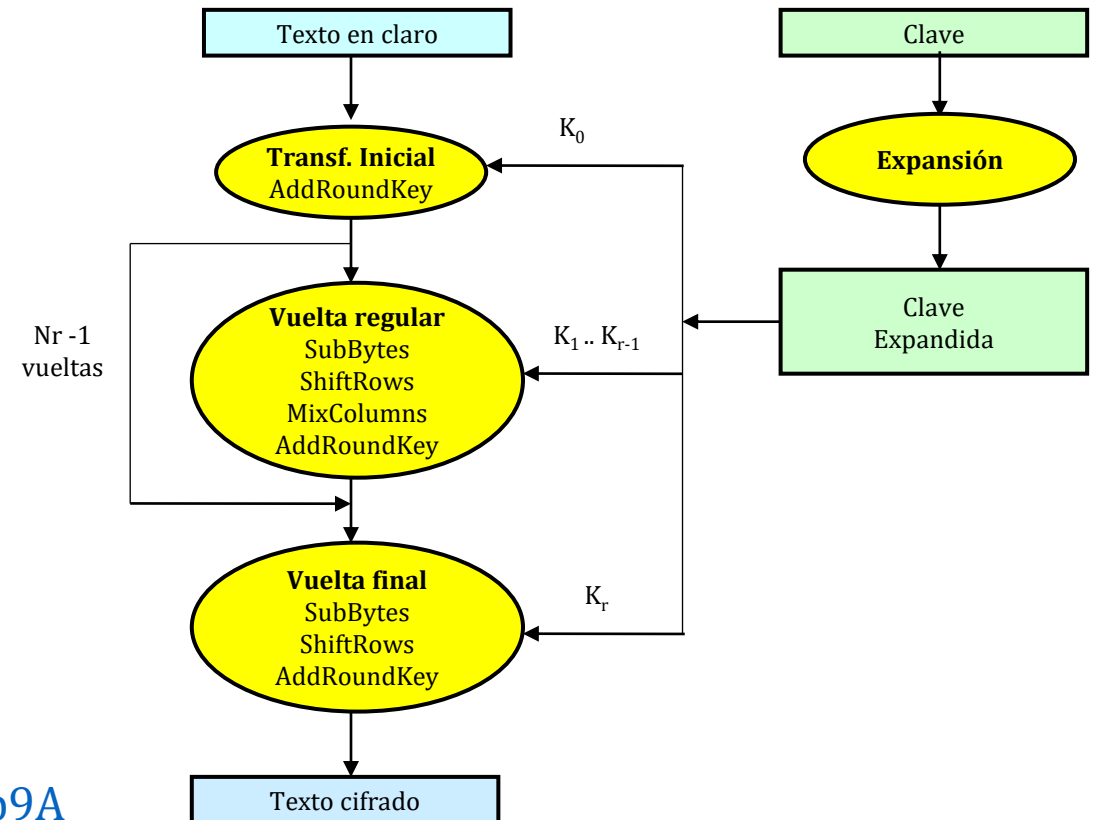


## Esquema del AES en el cifrado



Class4crypt c4c8.6a - © jorgeramio 2021

Lección 8.6a - página 13



- Para repasar conceptos y características del AES:
- <https://www.youtube.com/watch?v=pbmMo3wwp9A>

# Resumen de operaciones en GF ( $2^8$ )

## Módulo 8. Criptografía simétrica en bloque Lección 8.6b AES parte 2: Campos de Galois y extensión de clave



Clase c4c8.6b  
03/06/2021



### Expansión de claves en AES

- Número de bits de las subclaves para valores estándar de Nb y Nk

Bloque / Clave	Nk = 4 (128 bits)	Nk = 6 (192 bits)	Nk = 8 (256 bits)
Nb = 4 128 bits	Nr = 10 1.408 bits	Nr = 12 1.664 bits	Nr = 14 1.920 bits
Nb = 6 192 bits	Nr = 12 2.304 bits	Nr = 12 2.496 bits	Nr = 14 2.880 bits
Nb = 8 256 bits	Nr = 14 3.840 bits	Nr = 14 3.328 bits	Nr = 14 3.840 bits

- La expansión genera los bytes de las subclaves a partir de la clave K principal
- Será un array lineal W de palabras de 4 bytes y con longitud Nb\*(Nr + 1)

W <sub>0</sub>	W <sub>1</sub>	W <sub>2</sub>	W <sub>3</sub>	W <sub>4</sub>	W <sub>5</sub>	W <sub>6</sub>	W <sub>7</sub>	W <sub>8</sub>	W <sub>9</sub>	W <sub>10</sub>	W <sub>11</sub>	W <sub>12</sub>	W <sub>13</sub>	W <sub>14</sub>	...
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----

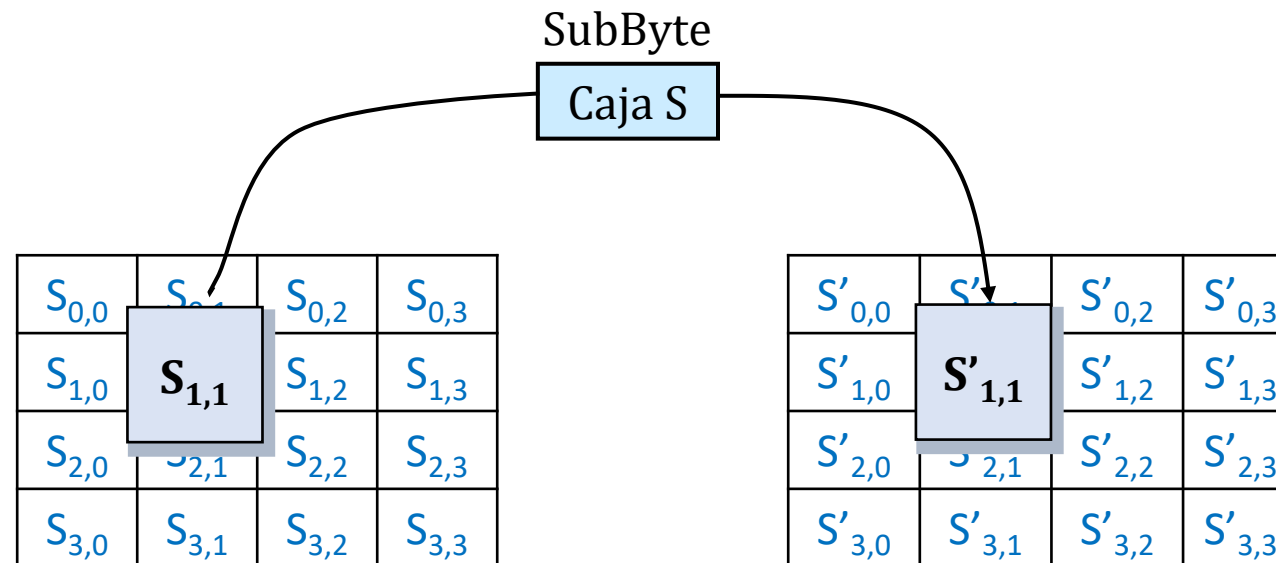
Class4crypt c4c8.6b - © jorgeramio 2021

Lección 8.6b - página 19

- Para repasar conceptos suma y multiplicación en GF ( $2^8$ ):
- [https://www.youtube.com/watch?v=Elm62\\_ec4MU](https://www.youtube.com/watch?v=Elm62_ec4MU)

# Función SubBytes

- Cada uno de los bytes de la matriz de estado es sustituido a través de una caja S de 8x8, es decir, con 8 bits de entrada y 8 bits de salida y cuya finalidad es introducir no linealidad en la cifra





# Cálculo de la función SubBytes

- La S-box se construye de la siguiente manera:
  1. Calculando el inverso del byte en GF ( $2^8$ )
  2. Calculando la siguiente transformación afín sobre GF(2)
    - $b'_i = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$
    - Donde
      - $0 \leq i < 8$
      - $b_i$  es el  $i$ ésimo bit del byte
      - $c_i$  es el  $i$ ésimo bit del byte  $c$  cuyo valor es  $\{63\}_{16}$  o  $\{01100011\}_2$
    - La transformación afín queda como se indica en la diapositiva siguiente

# Transformación afín en SubBytes

$$b'_0 = b_0 \oplus b_4 \oplus b_5 \oplus b_6 \oplus b_7 \oplus c_0$$

$$b'_1 = b_1 \oplus b_5 \oplus b_6 \oplus b_7 \oplus b_0 \oplus c_1$$

$$b'_2 = b_2 \oplus b_5 \oplus b_7 \oplus b_0 \oplus b_1 \oplus c_2$$

$$b'_3 = b_3 \oplus b_7 \oplus b_0 \oplus b_1 \oplus b_2 \oplus c_3$$

$$b'_4 = b_4 \oplus b_0 \oplus b_1 \oplus b_2 \oplus b_3 \oplus c_4$$

$$b'_5 = b_1 \oplus b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus c_5$$

$$b'_6 = b_6 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5 \oplus c_6$$

$$b'_7 = b_7 \oplus b_3 \oplus b_4 \oplus b_5 \oplus b_6 \oplus c_7$$

↑  
Lectura del byte

$$\begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Representación matricial

Valor  $\{63\}_{16}$  o  $\{01100011\}_2$

Es el inverso de la entrada

# Ejemplo operación SubByte de 5a - NIST\*

(\*) Documento oficial Announcing the AES, Appendix B – Cipher Example, NIST, November 26, 2001

$$5a = 0101\ 1010 = x^6 + x^4 + x^3 + x$$

inv (5a) = 22 = 0010 0010 (tabla de inversos entregada en clase anterior)

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{matrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{matrix}$$

- El resultado es el valor 1011 1110 = be
- Al mismo valor se llega si en la tabla en la siguiente diapositiva buscamos la intersección entre la fila 5 y la columna a

- Multiplicamos filas por columnas y sumamos {01100011}
- $1*0 + 0*1 + 0*0 + 0*0 + 1*0 + 1*1 + 1*0 + 1*0 = 1 \bmod 2 = 1 \oplus 1 = 0$

# Tabla de la función SubBytes

- La tabla contiene todos los valores entre 0 (0x 00) y 255 (0x FF)
- La formulación matemática anterior pretende minimizar la relación entre la entrada y la salida, una relación no lineal
- Para aplicar la caja S sobre un byte XY en hexadecimal
  - X representa la fila
  - Y representa la columna
  - En la figura SubByte 5a = be

	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

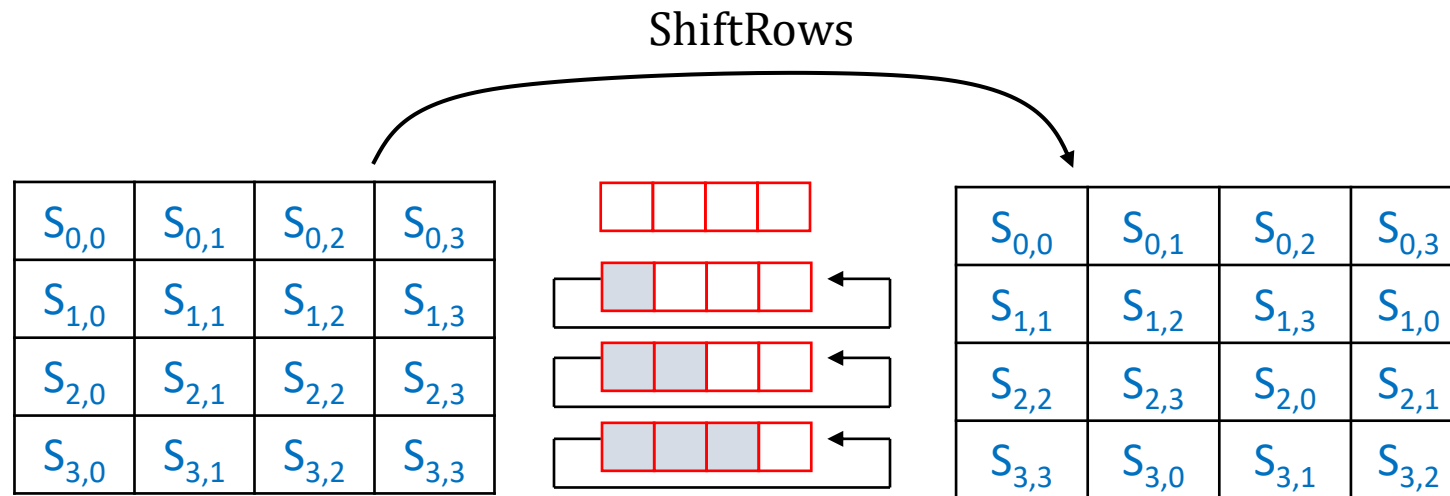
# Tabla de la función InvSubBytes

- El inverso SubBytes consiste en realizar la función inversa de la transformación afín anterior y el inverso del byte GF ( $2^8$ ) que devuelve los valores originales de la tabla SubBytes
- Como teníamos que SubBytes de 5a = be, nos ponemos en la fila del nibble b y en la columna del nibble e, como se ve en la tabla, obteniendo InvSubByte be = 5a

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	a0	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	d0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

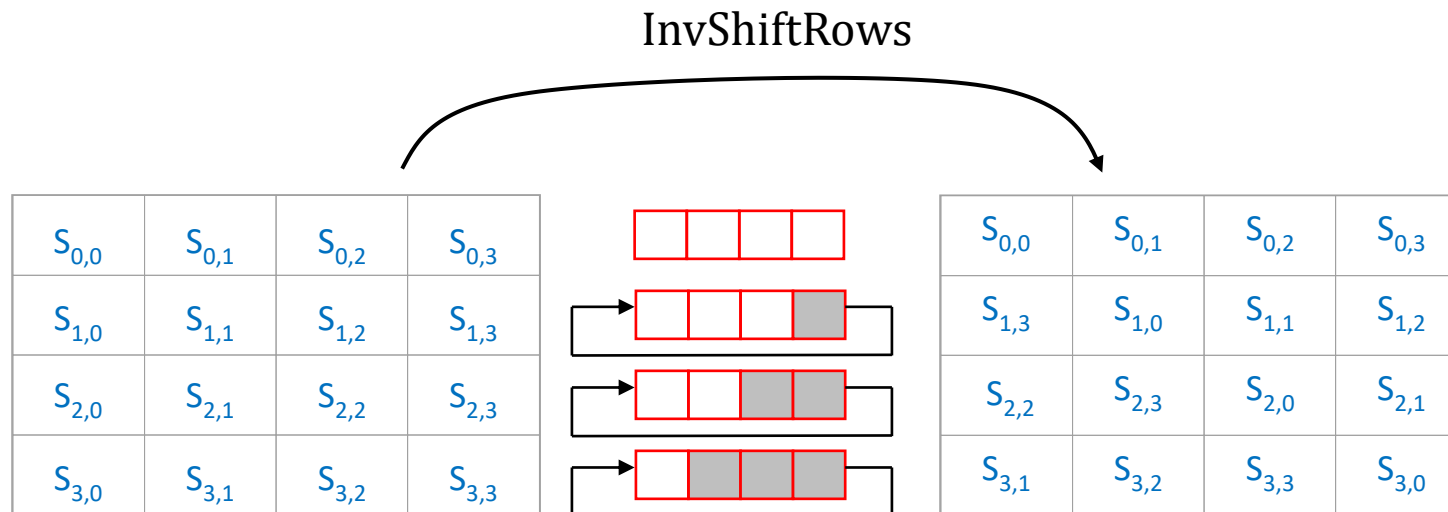
# Función ShiftRows

- En esta transformación se permutan cíclicamente los contenidos de las filas del estado. Tiene por objeto aumentar la difusión. La fila 0 no desplaza, la fila 1 desplaza un byte, la fila 2 desplaza dos bytes y la fila 3 desplaza tres bytes, hacia la izquierda



# Función InvShifRows

- En este caso se desplazan bloques de un byte hacia la derecha módulo 4 dentro de una fila. Así la fila 0 no desplaza, la fila 1 desplaza un byte, la fila 2 desplaza dos bytes y la fila 3 desplaza tres bytes a la derecha



# Función MixColumns

- Esta transformación opera sobre el estado, columna por columna, para maximizar la difusión
- Son operaciones con polinomios  $GF(2^8)$  en que cada byte es considerado como un polinomio
- Cada columna se multiplicará módulo  $x^4 + 1$ , polinomio no irreducible y no necesariamente invertible, pero sí para el polinomio  $a(x)$  con inverso  $a^{-1}(x)$
- $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$

- Donde:

- $0x\ 03 = 0000\ 0011 = x + 1$

- $0x\ 02 = 0000\ 0010 = x$

- $0x\ 01 = 0000\ 0001 = 1$

$$\begin{pmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{pmatrix} \quad \begin{matrix} \text{Para} \\ 0 \leq C < Nb \end{matrix}$$



# Ejemplo operación MixColumns - NIST\*

(\*) Documento oficial Announcing the AES, Appendix B – Cipher Example, NIST, November 26, 2001

$$\begin{aligned} S'_{0,C} &= (\{02\} S_{0,C}) \oplus (\{03\} S_{1,C}) \oplus S_{2,C} \oplus S_{3,C} \\ S'_{1,C} &= S_{0,C} \oplus (\{02\} S_{1,C}) \oplus (\{03\} S_{2,C}) \oplus S_{3,C} \\ S'_{2,C} &= S_{0,C} \oplus S_{1,C} \oplus (\{02\} S_{2,C}) \oplus (\{03\} S_{3,C}) \\ S'_{3,C} &= (\{03\} S_{0,C}) \oplus S_{1,C} \oplus S_{2,C} \oplus (\{02\} S_{3,C}) \end{aligned}$$

El primer byte de estado  $S'_{0,0}$  quedará:

$$S'_{0,0} = \{02\}S_{0,0} \oplus \{03\}S_{1,0} \oplus S_{2,0} \oplus S_{3,0}$$

$$S'_{0,0} = \{02\}e1 \oplus \{03\}fb \oplus 96 \oplus 7c$$

$$\{02\}e1 = x(x^7 + x^6 + x^5 + 1)$$

$$\{02\}e1 = x^8 + x^7 + x^6 + x$$

$$\{02\}e1 = (x^8 + x^7 + x^6 + x) \bmod x^4 + 1 = d2$$

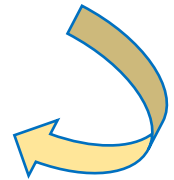
$$\{03\}fb = (x + 1)(x^7 + x^6 + x^5 + x^4 + x^3 + x + 1)$$

$$\{03\}fb = x^8 + x^3 + x^2 + 1$$

$$\{03\}fb = (x^8 + x^3 + x^2 + 1) \bmod x^4 + 1 = 1d$$

Si suponemos que el estado intermedio es el indicado

e1	a8	63	0d
fb	18	f4	c8
96	5b	73	11
7c	a0	e6	fd



Reducción mod  $x^4 + 1$  →

$$S'_{0,0} = d2 \oplus 1d \oplus 96 \oplus 7c = 25$$

Los bytes  $S'_{1,0}$  hasta  $S'_{3,3}$  se calculan de forma similar

# Reducción $(x^8 + x^7 + x^6 + x) \bmod x^4 + 1$

Diapositiva añadida después de haber grabado la clase

- Teníamos:
  - $(x^8 + x^7 + x^6 + x) \bmod x^4 + 1$
  - Un divisor para eliminar  $x^8$  es  $x^4$  porque  $x^4 x^4 = x^8$
  - Multiplicamos el módulo  $(x^4 + 1)$  por  $x^4$  y obtenemos  $(x^8 + x^4)$
- Tenemos dos polinomios  $(x^8 + x^7 + x^6 + x)$  y  $(x^8 + x^4)$  y sumamos mod 2

Byte		$x^7$	$x^6$	$x^5$	$x^4$	$x^3$	$x^2$	$x$	1
$p_1(x)$	$x^8$	$x^7$	$x^6$					$x$	
$p_2(x)$	$x^8$				$x^4$				
$\oplus$		$x^7$	$x^6$		$x^4$			$x$	
$p_3(x) = x^7 + x^6 + x^4 + x = 1101\ 0010 = 0x\ d2$									

# Reducción $(x^8 + x^3 + x^2 + 1) \bmod x^4 + 1$

Diapositiva añadida después de haber grabado la clase

- Teníamos:
  - $(x^8 + x^3 + x^2 + 1) \bmod x^4 + 1$
  - Un divisor para eliminar  $x^8$  es  $x^4$  porque  $x^4 x^4 = x^8$
  - Multiplicamos el módulo  $(x^4 + 1)$  por  $x^4$  y obtenemos  $(x^8 + x^4)$
- Tenemos dos polinomios  $(x^8 + x^3 + x^2 + 1)$  y  $(x^8 + x^4)$  y sumamos mod 2

Byte		$x^7$	$x^6$	$x^5$	$x^4$	$x^3$	$x^2$	$x$	1
$p_1(x)$	$x^8$					$x^3$	$x^2$		1
$p_2(x)$	$x^8$				$x^4$				
$\oplus$					$x^4$	$x^3$	$x^2$		1
$p_3(x) = x^4 + x^3 + x^2 + 1 = 0001\ 1101 = 0x\ 1d$									

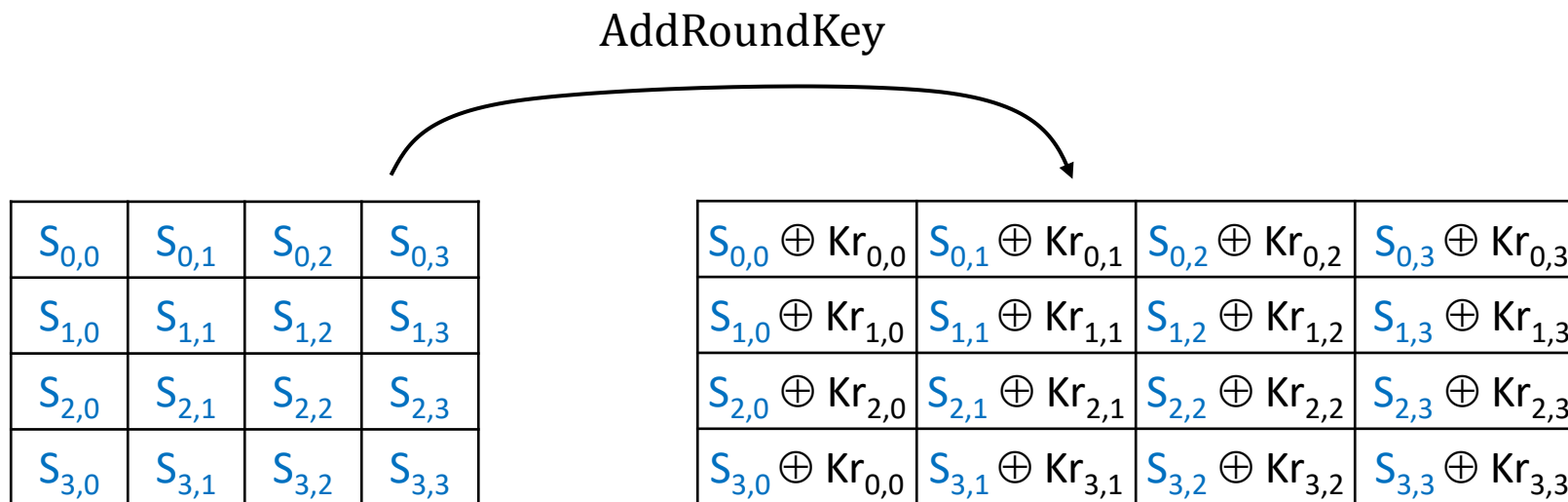
# Función InvMixColumns

- Como la transformación MixColumns multiplicaba por un polinomio fijo  $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$ , primo relativo con  $x^4 + 1$ , su inverso será  $a^{-1}(x)$
- Cada columna se multiplicará módulo  $x^4 + 1$  con el polinomio  $a^{-1}(x)$
- $a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$ 
  - Donde:
    - $0x \ 0b = 0000 \ 1011 = x^3 + x + 1$
    - $0x \ 0d = 0000 \ 1101 = x^3 + x^2 + 1$
    - $0x \ 09 = 0000 \ 1001 = x^3 + 1$
    - $0x \ 0e = 0000 \ 1110 = x^3 + x^2 + x$

$$\begin{pmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{pmatrix} = \begin{pmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{pmatrix} \begin{pmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{pmatrix} \quad \begin{matrix} \text{Para} \\ 0 \leq C < Nb \end{matrix}$$

# Función AddRoundKey e InvAddRoundKey

- Se realiza la suma módulo 2 (XOR) de la matriz de estado con la matriz de la clave  $K_r$  que será diferente en cada vuelta según el algoritmo de expansión de clave
- Su inversa es la misma al ser involutivo el xor



# Ejemplo AddRoundKey - NIST\*

(\*) Documento oficial Announcing the AES, Appendix B – Cipher Example, NIST, November 26, 2001

Round Number	Start of Round	After SubBytes	After ShiftRows	After MixColumns	Round Key Value																																																																																
input	<table><tr><td>32</td><td>88</td><td>31</td><td>e0</td></tr><tr><td>43</td><td>5a</td><td>31</td><td>37</td></tr><tr><td>f6</td><td>30</td><td>98</td><td>07</td></tr><tr><td>a8</td><td>8d</td><td>a2</td><td>34</td></tr></table>	32	88	31	e0	43	5a	31	37	f6	30	98	07	a8	8d	a2	34	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td>2b</td><td>28</td><td>ab</td><td>09</td></tr><tr><td>7e</td><td>ae</td><td>f7</td><td>cf</td></tr><tr><td>15</td><td>d2</td><td>15</td><td>4f</td></tr><tr><td>16</td><td>a6</td><td>88</td><td>3c</td></tr></table>	2b	28	ab	09	7e	ae	f7	cf	15	d2	15	4f	16	a6	88	3c
32	88	31	e0																																																																																		
43	5a	31	37																																																																																		
f6	30	98	07																																																																																		
a8	8d	a2	34																																																																																		
2b	28	ab	09																																																																																		
7e	ae	f7	cf																																																																																		
15	d2	15	4f																																																																																		
16	a6	88	3c																																																																																		
1	<table><tr><td>19</td><td>a0</td><td>9a</td><td>e9</td></tr><tr><td>3d</td><td>f4</td><td>c6</td><td>f8</td></tr><tr><td>e3</td><td>e2</td><td>8d</td><td>48</td></tr><tr><td>be</td><td>2b</td><td>2a</td><td>08</td></tr></table>	19	a0	9a	e9	3d	f4	c6	f8	e3	e2	8d	48	be	2b	2a	08	<table><tr><td>d4</td><td>e0</td><td>b8</td><td>1e</td></tr><tr><td>27</td><td>bf</td><td>b4</td><td>41</td></tr><tr><td>11</td><td>98</td><td>5d</td><td>52</td></tr><tr><td>ae</td><td>f1</td><td>e5</td><td>30</td></tr></table>	d4	e0	b8	1e	27	bf	b4	41	11	98	5d	52	ae	f1	e5	30	<table><tr><td>d4</td><td>e0</td><td>b8</td><td>1e</td></tr><tr><td>bf</td><td>b4</td><td>41</td><td>27</td></tr><tr><td>5d</td><td>52</td><td>11</td><td>98</td></tr><tr><td>30</td><td>ae</td><td>f1</td><td>e5</td></tr></table>	d4	e0	b8	1e	bf	b4	41	27	5d	52	11	98	30	ae	f1	e5	<table><tr><td>04</td><td>e0</td><td>48</td><td>28</td></tr><tr><td>66</td><td>cb</td><td>f8</td><td>06</td></tr><tr><td>81</td><td>19</td><td>d3</td><td>26</td></tr><tr><td>e5</td><td>9a</td><td>7a</td><td>4c</td></tr></table>	04	e0	48	28	66	cb	f8	06	81	19	d3	26	e5	9a	7a	4c	<table><tr><td>a0</td><td>88</td><td>23</td><td>2a</td></tr><tr><td>fa</td><td>54</td><td>a3</td><td>6c</td></tr><tr><td>fe</td><td>2c</td><td>39</td><td>76</td></tr><tr><td>17</td><td>b1</td><td>39</td><td>05</td></tr></table>	a0	88	23	2a	fa	54	a3	6c	fe	2c	39	76	17	b1	39	05
19	a0	9a	e9																																																																																		
3d	f4	c6	f8																																																																																		
e3	e2	8d	48																																																																																		
be	2b	2a	08																																																																																		
d4	e0	b8	1e																																																																																		
27	bf	b4	41																																																																																		
11	98	5d	52																																																																																		
ae	f1	e5	30																																																																																		
d4	e0	b8	1e																																																																																		
bf	b4	41	27																																																																																		
5d	52	11	98																																																																																		
30	ae	f1	e5																																																																																		
04	e0	48	28																																																																																		
66	cb	f8	06																																																																																		
81	19	d3	26																																																																																		
e5	9a	7a	4c																																																																																		
a0	88	23	2a																																																																																		
fa	54	a3	6c																																																																																		
fe	2c	39	76																																																																																		
17	b1	39	05																																																																																		

# Ejemplo SubBytes - NIST\*

(\*) Documento oficial Announcing the AES, Appendix B – Cipher Example, NIST, November 26, 2001

Round Number	Start of Round	After SubBytes	After ShiftRows	After MixColumns	Round Key Value																																																																																	
input	<table><tr><td>32</td><td>88</td><td>31</td><td>e0</td></tr><tr><td>43</td><td>5a</td><td>31</td><td>37</td></tr><tr><td>f6</td><td>30</td><td>98</td><td>07</td></tr><tr><td>a8</td><td>8d</td><td>a2</td><td>34</td></tr></table>	32	88	31	e0	43	5a	31	37	f6	30	98	07	a8	8d	a2	34	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td>2b</td><td>28</td><td>ab</td><td>09</td></tr><tr><td>7e</td><td>ae</td><td>f7</td><td>cf</td></tr><tr><td>15</td><td>d2</td><td>15</td><td>4f</td></tr><tr><td>16</td><td>a6</td><td>88</td><td>3c</td></tr></table> $\oplus$	2b	28	ab	09	7e	ae	f7	cf	15	d2	15	4f	16	a6	88	3c	=
32	88	31	e0																																																																																			
43	5a	31	37																																																																																			
f6	30	98	07																																																																																			
a8	8d	a2	34																																																																																			
2b	28	ab	09																																																																																			
7e	ae	f7	cf																																																																																			
15	d2	15	4f																																																																																			
16	a6	88	3c																																																																																			
1	<table><tr><td>19</td><td>a0</td><td>9a</td><td>e9</td></tr><tr><td>3d</td><td>f4</td><td>c6</td><td>f8</td></tr><tr><td>e3</td><td>e2</td><td>8d</td><td>48</td></tr><tr><td>be</td><td>2b</td><td>2a</td><td>08</td></tr></table>	19	a0	9a	e9	3d	f4	c6	f8	e3	e2	8d	48	be	2b	2a	08	<table><tr><td>d4</td><td>e0</td><td>b8</td><td>1e</td></tr><tr><td>27</td><td>bf</td><td>b4</td><td>41</td></tr><tr><td>11</td><td>98</td><td>5d</td><td>52</td></tr><tr><td>ae</td><td>f1</td><td>e5</td><td>30</td></tr></table>	d4	e0	b8	1e	27	bf	b4	41	11	98	5d	52	ae	f1	e5	30	<table><tr><td>d4</td><td>e0</td><td>b8</td><td>1e</td></tr><tr><td>bf</td><td>b4</td><td>41</td><td>27</td></tr><tr><td>5d</td><td>52</td><td>11</td><td>98</td></tr><tr><td>30</td><td>ae</td><td>f1</td><td>e5</td></tr></table>	d4	e0	b8	1e	bf	b4	41	27	5d	52	11	98	30	ae	f1	e5	<table><tr><td>04</td><td>e0</td><td>48</td><td>28</td></tr><tr><td>66</td><td>cb</td><td>f8</td><td>06</td></tr><tr><td>81</td><td>19</td><td>d3</td><td>26</td></tr><tr><td>e5</td><td>9a</td><td>7a</td><td>4c</td></tr></table> $\oplus$	04	e0	48	28	66	cb	f8	06	81	19	d3	26	e5	9a	7a	4c	= <table><tr><td>a0</td><td>88</td><td>23</td><td>2a</td></tr><tr><td>fa</td><td>54</td><td>a3</td><td>6c</td></tr><tr><td>fe</td><td>2c</td><td>39</td><td>76</td></tr><tr><td>17</td><td>b1</td><td>39</td><td>05</td></tr></table>	a0	88	23	2a	fa	54	a3	6c	fe	2c	39	76	17	b1	39	05	
19	a0	9a	e9																																																																																			
3d	f4	c6	f8																																																																																			
e3	e2	8d	48																																																																																			
be	2b	2a	08																																																																																			
d4	e0	b8	1e																																																																																			
27	bf	b4	41																																																																																			
11	98	5d	52																																																																																			
ae	f1	e5	30																																																																																			
d4	e0	b8	1e																																																																																			
bf	b4	41	27																																																																																			
5d	52	11	98																																																																																			
30	ae	f1	e5																																																																																			
04	e0	48	28																																																																																			
66	cb	f8	06																																																																																			
81	19	d3	26																																																																																			
e5	9a	7a	4c																																																																																			
a0	88	23	2a																																																																																			
fa	54	a3	6c																																																																																			
fe	2c	39	76																																																																																			
17	b1	39	05																																																																																			

# Ejemplo ShiftRows - NIST\*

(\*) Documento oficial Announcing the AES, Appendix B – Cipher Example, NIST, November 26, 2001

Round Number	Start of Round	After SubBytes	After ShiftRows	After MixColumns	Round Key Value																																																																																	
input	<table><tr><td>32</td><td>88</td><td>31</td><td>e0</td></tr><tr><td>43</td><td>5a</td><td>31</td><td>37</td></tr><tr><td>f6</td><td>30</td><td>98</td><td>07</td></tr><tr><td>a8</td><td>8d</td><td>a2</td><td>34</td></tr></table>	32	88	31	e0	43	5a	31	37	f6	30	98	07	a8	8d	a2	34	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td>2b</td><td>28</td><td>ab</td><td>09</td></tr><tr><td>7e</td><td>ae</td><td>f7</td><td>cf</td></tr><tr><td>15</td><td>d2</td><td>15</td><td>4f</td></tr><tr><td>16</td><td>a6</td><td>88</td><td>3c</td></tr></table> $\oplus$	2b	28	ab	09	7e	ae	f7	cf	15	d2	15	4f	16	a6	88	3c	=
	32	88	31	e0																																																																																		
	43	5a	31	37																																																																																		
	f6	30	98	07																																																																																		
	a8	8d	a2	34																																																																																		
2b	28	ab	09																																																																																			
7e	ae	f7	cf																																																																																			
15	d2	15	4f																																																																																			
16	a6	88	3c																																																																																			
1	<table><tr><td>19</td><td>a0</td><td>9a</td><td>e9</td></tr><tr><td>3d</td><td>f4</td><td>c6</td><td>f8</td></tr><tr><td>e3</td><td>e2</td><td>8d</td><td>48</td></tr><tr><td>be</td><td>2b</td><td>2a</td><td>08</td></tr></table>	19	a0	9a	e9	3d	f4	c6	f8	e3	e2	8d	48	be	2b	2a	08	<table><tr><td>d4</td><td>e0</td><td>b8</td><td>1e</td></tr><tr><td>27</td><td>bf</td><td>b4</td><td>41</td></tr><tr><td>11</td><td>98</td><td>5d</td><td>52</td></tr><tr><td>ae</td><td>f1</td><td>e5</td><td>30</td></tr></table>	d4	e0	b8	1e	27	bf	b4	41	11	98	5d	52	ae	f1	e5	30	<table><tr><td>d4</td><td>e0</td><td>b8</td><td>1e</td></tr><tr><td>bf</td><td>b4</td><td>41</td><td>27</td></tr><tr><td>5d</td><td>52</td><td>11</td><td>98</td></tr><tr><td>30</td><td>ae</td><td>f1</td><td>e5</td></tr></table>	d4	e0	b8	1e	bf	b4	41	27	5d	52	11	98	30	ae	f1	e5	<table><tr><td>04</td><td>e0</td><td>48</td><td>28</td></tr><tr><td>66</td><td>cb</td><td>f8</td><td>06</td></tr><tr><td>81</td><td>19</td><td>d3</td><td>26</td></tr><tr><td>e5</td><td>9a</td><td>7a</td><td>4c</td></tr></table> $\oplus$	04	e0	48	28	66	cb	f8	06	81	19	d3	26	e5	9a	7a	4c	= <table><tr><td>a0</td><td>88</td><td>23</td><td>2a</td></tr><tr><td>fa</td><td>54</td><td>a3</td><td>6c</td></tr><tr><td>fe</td><td>2c</td><td>39</td><td>76</td></tr><tr><td>17</td><td>b1</td><td>39</td><td>05</td></tr></table>	a0	88	23	2a	fa	54	a3	6c	fe	2c	39	76	17	b1	39	05	
	19	a0	9a	e9																																																																																		
	3d	f4	c6	f8																																																																																		
	e3	e2	8d	48																																																																																		
	be	2b	2a	08																																																																																		
d4	e0	b8	1e																																																																																			
27	bf	b4	41																																																																																			
11	98	5d	52																																																																																			
ae	f1	e5	30																																																																																			
d4	e0	b8	1e																																																																																			
bf	b4	41	27																																																																																			
5d	52	11	98																																																																																			
30	ae	f1	e5																																																																																			
04	e0	48	28																																																																																			
66	cb	f8	06																																																																																			
81	19	d3	26																																																																																			
e5	9a	7a	4c																																																																																			
a0	88	23	2a																																																																																			
fa	54	a3	6c																																																																																			
fe	2c	39	76																																																																																			
17	b1	39	05																																																																																			



# Ejemplo MixColumns - NIST\*

(\*) Documento oficial Announcing the AES, Appendix B – Cipher Example, NIST, November 26, 2001

Round Number	Start of Round	After SubBytes	After ShiftRows	After MixColumns	Round Key Value																																																																																	
input	<table><tr><td>32</td><td>88</td><td>31</td><td>e0</td></tr><tr><td>43</td><td>5a</td><td>31</td><td>37</td></tr><tr><td>f6</td><td>30</td><td>98</td><td>07</td></tr><tr><td>a8</td><td>8d</td><td>a2</td><td>34</td></tr></table>	32	88	31	e0	43	5a	31	37	f6	30	98	07	a8	8d	a2	34	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td>2b</td><td>28</td><td>ab</td><td>09</td></tr><tr><td>7e</td><td>ae</td><td>f7</td><td>cf</td></tr><tr><td>15</td><td>d2</td><td>15</td><td>4f</td></tr><tr><td>16</td><td>a6</td><td>88</td><td>3c</td></tr></table> $\oplus$	2b	28	ab	09	7e	ae	f7	cf	15	d2	15	4f	16	a6	88	3c	=
	32	88	31	e0																																																																																		
	43	5a	31	37																																																																																		
	f6	30	98	07																																																																																		
	a8	8d	a2	34																																																																																		
2b	28	ab	09																																																																																			
7e	ae	f7	cf																																																																																			
15	d2	15	4f																																																																																			
16	a6	88	3c																																																																																			
1	<table><tr><td>19</td><td>a0</td><td>9a</td><td>e9</td></tr><tr><td>3d</td><td>f4</td><td>c6</td><td>f8</td></tr><tr><td>e3</td><td>e2</td><td>8d</td><td>48</td></tr><tr><td>be</td><td>2b</td><td>2a</td><td>08</td></tr></table>	19	a0	9a	e9	3d	f4	c6	f8	e3	e2	8d	48	be	2b	2a	08	<table><tr><td>d4</td><td>e0</td><td>b8</td><td>1e</td></tr><tr><td>27</td><td>bf</td><td>b4</td><td>41</td></tr><tr><td>11</td><td>98</td><td>5d</td><td>52</td></tr><tr><td>ae</td><td>f1</td><td>e5</td><td>30</td></tr></table>	d4	e0	b8	1e	27	bf	b4	41	11	98	5d	52	ae	f1	e5	30	<table><tr><td>d4</td><td>e0</td><td>b8</td><td>1e</td></tr><tr><td>bf</td><td>b4</td><td>41</td><td>27</td></tr><tr><td>5d</td><td>52</td><td>11</td><td>98</td></tr><tr><td>30</td><td>ae</td><td>f1</td><td>e5</td></tr></table>	d4	e0	b8	1e	bf	b4	41	27	5d	52	11	98	30	ae	f1	e5	<table><tr><td>04</td><td>e0</td><td>48</td><td>28</td></tr><tr><td>66</td><td>cb</td><td>f8</td><td>06</td></tr><tr><td>81</td><td>19</td><td>d3</td><td>26</td></tr><tr><td>e5</td><td>9a</td><td>7a</td><td>4c</td></tr></table> $\oplus$	04	e0	48	28	66	cb	f8	06	81	19	d3	26	e5	9a	7a	4c	<table><tr><td>a0</td><td>88</td><td>23</td><td>2a</td></tr><tr><td>fa</td><td>54</td><td>a3</td><td>6c</td></tr><tr><td>fe</td><td>2c</td><td>39</td><td>76</td></tr><tr><td>17</td><td>b1</td><td>39</td><td>05</td></tr></table> $\oplus$	a0	88	23	2a	fa	54	a3	6c	fe	2c	39	76	17	b1	39	05	=
	19	a0	9a	e9																																																																																		
	3d	f4	c6	f8																																																																																		
	e3	e2	8d	48																																																																																		
	be	2b	2a	08																																																																																		
d4	e0	b8	1e																																																																																			
27	bf	b4	41																																																																																			
11	98	5d	52																																																																																			
ae	f1	e5	30																																																																																			
d4	e0	b8	1e																																																																																			
bf	b4	41	27																																																																																			
5d	52	11	98																																																																																			
30	ae	f1	e5																																																																																			
04	e0	48	28																																																																																			
66	cb	f8	06																																																																																			
81	19	d3	26																																																																																			
e5	9a	7a	4c																																																																																			
a0	88	23	2a																																																																																			
fa	54	a3	6c																																																																																			
fe	2c	39	76																																																																																			
17	b1	39	05																																																																																			

# Comprobación con software AESphere

The image displays the AESphere software interface, which is used for testing AES encryption and decryption. The main window, titled 'AESphere - Ventana Principal', features a menu bar with 'Archivo', 'Herramientas', and 'Ayuda'. Below the menu, there are four large circular buttons: a green padlock for 'CIFRAR' (Encrypt), a red padlock for 'DESCIFRAR' (Decrypt), an orange gear for 'ATAQUES' (Attacks), and a blue wrench for 'OPERACIONES' (Operations). A 'Comprobar vectores' button is located in the top right corner of the main window.

The 'Operaciones' window, titled 'AESphere - Operaciones', shows four diagrams illustrating the AES operations:

- SubBytes:** A diagram showing a 4x4 state matrix  $a$  being transformed into a 4x4 state matrix  $b$  using the SubBytes operation. The element  $a_{2,2}$  is highlighted in orange and mapped to  $b_{2,2}$ .
- ShiftRows:** A diagram showing a 4x4 state matrix  $a$  being transformed into a 4x4 state matrix  $b$  using the ShiftRows operation. The element  $a_{2,2}$  is highlighted in orange and mapped to  $b_{2,2}$ .
- MixColumns:** A diagram showing a 4x4 state matrix  $a$  being transformed into a 4x4 state matrix  $b$  using the MixColumns operation. The element  $a_{2,2}$  is highlighted in orange and mapped to  $b_{2,2}$ .
- AddRoundKey:** A diagram showing a 4x4 state matrix  $a$  being transformed into a 4x4 state matrix  $b$  using the AddRoundKey operation. The element  $a_{2,2}$  is highlighted in orange and mapped to  $b_{2,2}$ .

The 'Operaciones' window also includes a menu bar with 'Archivo', 'Operaciones', and 'Ayuda', and an 'Atrás' button in the bottom right corner.

[https://www.criptored.es/software/sw\\_m001p.htm](https://www.criptored.es/software/sw_m001p.htm)

# Más información en píldoras Thoth



<https://www.youtube.com/watch?v=tzj1RoqRnv0>

# Conclusiones de la Lección 8.6c

- AES opera con polinomios y usa 4 funciones fácilmente invertibles, usando la función directa para el cifrado y la función inversa para el descifrado, con las mismas claves en ambos casos
  - SubBytes para el cifrado e InvSubBytes para el descifrado
  - ShiftRows para el cifrado e InvShiftRows para el descifrado
  - MixColumns para el cifrado e InvMixColumns para el descifrado
  - AddRoundKey para el cifrado e InvAddRoundKey para el descifrado
- SubBytes y MixColumns trabajan a nivel de byte y palabra de 32 bits, y si fuese necesario se reducirán a un módulo que será un polinomio primitivo
- ShiftRows es un simple desplazamiento de bytes y AddRoundKey es una función xor

# Lectura recomendada (1/2)

- Guion píldora formativa Thoth nº 30, ¿Cómo se cifra con el algoritmo AES?, Jorge Ramió, 2015
  - <https://www.criptored.es/thoth/material/texto/pildora030.pdf>
- Federal Information, Processing Standards Publication 197, Announcing the Advanced Encryption Standard AES, 2001
  - <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- Criptosistema Rijndael. A Fondo, Alfonso Muñoz, 2004
  - [https://www.criptored.es/guiateoria/gt\\_m480a.htm](https://www.criptored.es/guiateoria/gt_m480a.htm)
- AESphere, Antonio Gómez, Roberto Sierra, dirección Jorge Ramió, 2014
  - [https://www.criptored.es/software/sw\\_m001p.htm](https://www.criptored.es/software/sw_m001p.htm)

# Lectura recomendada (2/2)

- Libro Electrónico de Seguridad Informática y Criptografía, Versión 4.1, Módulo 12 Cifrado Simétrico en Bloque, Jorge Ramió, marzo 2006
  - [https://www.criptored.es/guiateoria/gt\\_m001a.htm](https://www.criptored.es/guiateoria/gt_m001a.htm)