

# Capítulo 12

## Cifrado Simétrico en Bloque

### Seguridad Informática y Criptografía



v 4.1



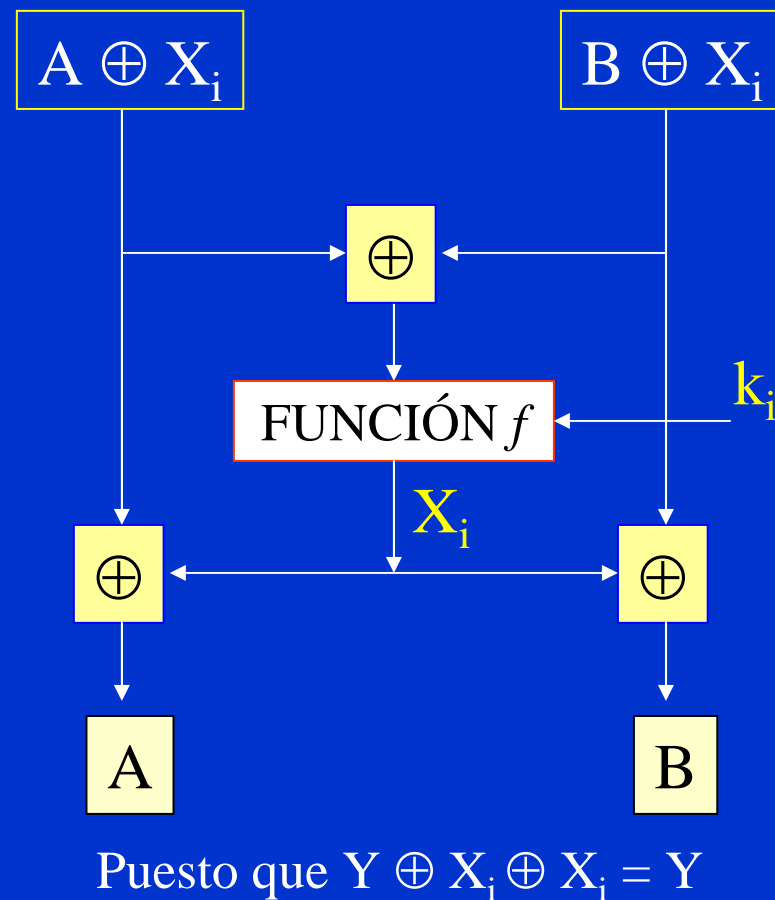
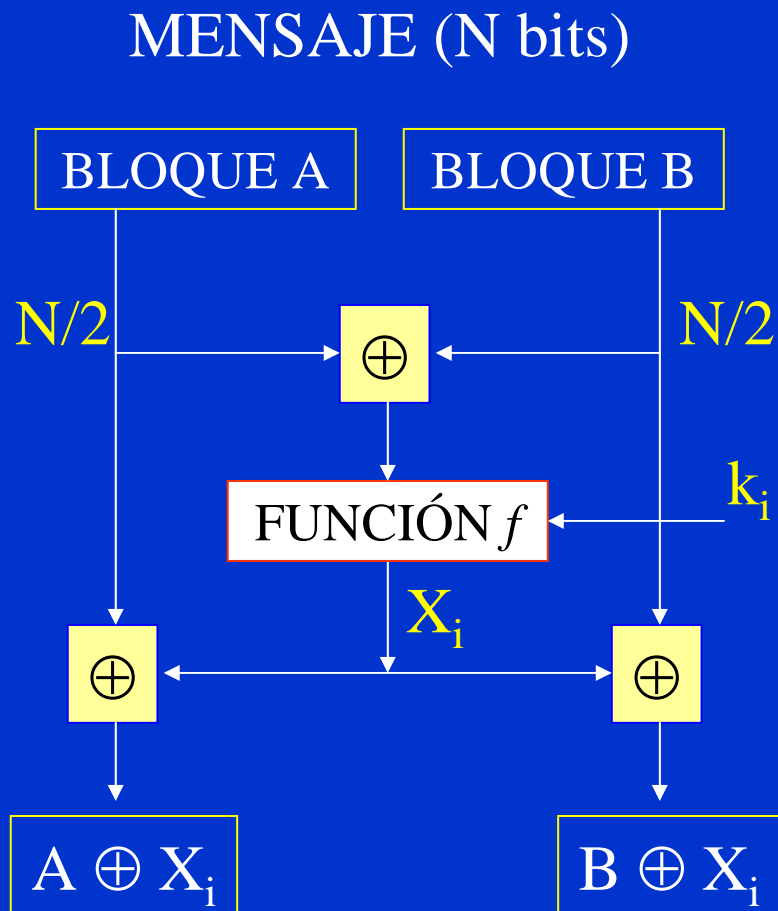
Material Docente de  
Libre Distribución

Ultima actualización del archivo: 01/03/06  
Este archivo tiene: 119 diapositivas

Dr. Jorge Ramió Aguirre  
Universidad Politécnica de Madrid

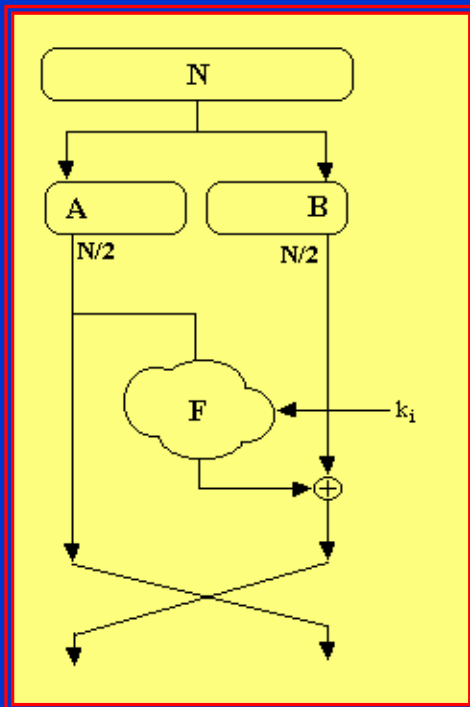
Este archivo forma parte de un curso completo sobre Seguridad Informática y Criptografía. Se autoriza el uso, reproducción en computador y su impresión en papel, sólo con fines docentes y/o personales, respetando los créditos del autor. Queda prohibida su comercialización, excepto la edición en venta en el Departamento de Publicaciones de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, España.

# Cifrado y descifrado genérico en bloque



# Cifrado tipo Feistel

Horst Feistel: inventor (IBM) del algoritmo LUCIFER a comienzos de los años 70. El algoritmo fue utilizado por el Reino Unido. En 1974 se propone a la NSA como estándar y en ese año dará origen al DES.



- Dado un bloque de  $N$  bits (típico 64) éste se dividirá en dos mitades.
- Existirá una función unidireccional  $F$  (muy difícil de invertir).
- Se realizan operaciones con la clave  $k_i$  sólo con una mitad del bloque, y se permutan en cada vuelta las dos mitades, operación que se repite durante  $n$  vueltas.

[http://en.wikipedia.org/wiki/Feistel\\_network](http://en.wikipedia.org/wiki/Feistel_network)



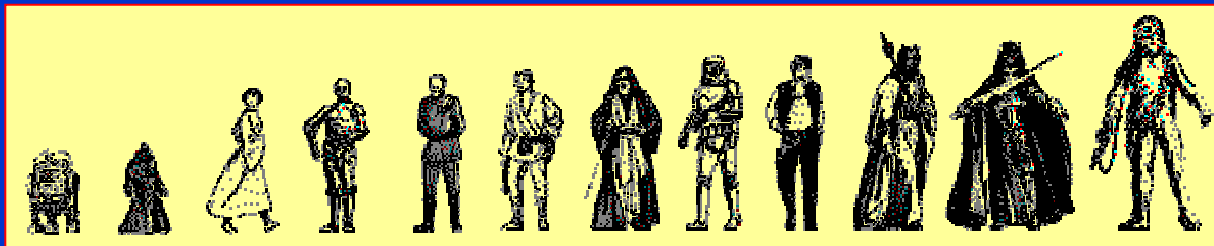
# Un ejemplo básico de cifrado tipo Feistel

El algoritmo usará bloques de tamaño 8 caracteres. Tendrá dos vueltas y en cada vuelta realizará una operación de sustitución  $S$  y una permutación  $P$  sobre la 1ª mitad.

**Sustitución:**  $C_i = (M_i + 1) \bmod 27$

**Permutación:**  $C_i = \Pi_{3241}$  (el carácter 1º pasa a la 4ª posición en el criptograma, el 4º a la 3ª, el 2º a la 2ª y el 3º a la 1ª)

Mensaje:  $M =$  **STAR WARS, LA MISIÓN CONTINÚA**



# Cifrado tipo Feistel en cuerpo $n = 27$

$S_i: +1 \text{ mod } 27$

$P_i: \Pi_{3241}$

Primera  
vuelta

Segunda  
vuelta

|  |   |      |      |      |      |      |      |
|--|---|------|------|------|------|------|------|
| M = STAR WARS, LA MISIÓN CONTINÚA  |   |      |      |      |      |      |      |
| M <sub>1</sub>   | = | STAR | WARS | LAMI | SION | CONT | INUA |
| S <sub>1</sub>   | = | TUBS | WARS | MBNJ | SION | DPÑU | INUA |
| P <sub>1</sub>   | = | BUST | WARS | NBJM | SION | ÑPUD | INUA |
| <div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div></div> |   |      |      |      |      |      |      |
| M <sub>2</sub>   | = | WARS | BUST | SION | NBJM | INUA | ÑPUD |
| S <sub>2</sub>   | = | XBST | BUST | TJPÑ | NBJM | JÑVB | ÑPUD |
| P <sub>2</sub>   | = | SBTX | BUST | PJÑT | NBJM | VÑBJ | ÑPUD |

**C = SBTX BUST PJÑT NBJM VÑBJ ÑPUD**

Aunque le parezca increíble, el DES hará prácticamente lo mismo trabajando con bits y con funciones un poco más “complejas”.

# Cifradores de bloque más conocidos

| Algoritmo | Bloque (bits) | Clave (bits) | Vueltas |
|-----------|---------------|--------------|---------|
| Lucifer   | 128           | 128          | 16      |
| DES       | 64            | 56           | 16      |
| Loki      | 64            | 64           | 16      |
| RC2       | 64            | variable     | --      |
| CAST      | 64            | 64           | 8       |
| Blowfish  | 64            | variable     | 16      |
| IDEA      | 64            | 128          | 8       |

|          |     |           |          |
|----------|-----|-----------|----------|
| Skipjack | 64  | 80        | 32       |
| Rijndael | 128 | 128 o más | flexible |



## Características de estos algoritmos (1)

- **Lucifer**: algoritmo original tipo Feistel usado a comienzos de los años 70 por en el Reino Unido y que posteriormente dará lugar al DES.
- **DES**: algoritmo tipo Feistel que se convirtió en estándar durante casi treinta años. Hoy es vulnerable por su pequeña longitud de clave y ha dejado de ser estándar mundial.
- **Loki**: algoritmo australiano similar al DES, también de tipo Feistel.
- **RC2**: algoritmo propuesto por Ron Rivest y que se incluye en navegadores de Internet desde 1999.
- **CAST**: algoritmo canadiense tipo Feistel que se ofrece como uno de los algoritmos de cifra en últimas versiones de PGP.

## Características de estos algoritmos (2)

- **Blowfish**: algoritmo de tipo Feistel propuesto por Bruce Schneier.
- **IDEA**: algoritmo europeo usado principalmente en el correo electrónico PGP.
- **Skipjack**: propuesta de nuevo estándar en USA 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.

Encontrará las especificaciones de éstos y otros algoritmos de cifra simétrica y asimétrica en la siguiente página web.

<http://www.quadibloc.com/crypto/intro.htm>





## Otros cifradores de bloque

| Algoritmo | Bloque (bits) | Clave (bits) | Vueltas     |
|-----------|---------------|--------------|-------------|
| Twofish   | 128           | variable     | variable    |
| Khufu     | 64            | 512          | 16, 24, 32  |
| Khafre    | 64            | 128          | más vueltas |
| Gost      | 64            | 256          | 32          |
| RC5       | variable      | variable     | variable    |
| SAFER 64  | 64            | 64           | 8           |
| Akelarre  | variable      | variable     | variable    |
| FEAL      | 64            | 64           | 32          |

De éstos, los más conocidos son Twofish -uno de los candidatos a AES- y que lo encontraremos en últimas versiones de PGP y RC5.

## Características de estos algoritmos

- **Twofish**: Propuesto por Bruce Schneier después de Blowfish, de tipo Feistel, diseño simple, sin claves débiles y multiplataforma.
- **Khufu**: algoritmo propuesto por Ralph Merkle con una clave generada con un sistema de “cajas” S.
- **Khafre**: algoritmo propuesto por Ralph Merkle en el que la clave ya no depende de las cajas S.
- **Gost**: algoritmo similar al DES con cajas S secretas propuesto en la Unión Soviética.
- **RC5**: algoritmo propuesto por Ron Rivest; realiza operaciones or exclusivo, suma modular y desplazamiento de bits.
- **SAFER 64**: algoritmo propuesto por James Massey.
- **Akelarre**: algoritmo español propuesto en 1996 por el CSIC, Consejo Superior de Investigaciones Científicas.
- **FEAL**: algoritmo propuesto en Japón.

# Algunas tasas de cifra comparativas


Velocidad de cifra de algoritmos en un PC 486 a 33 MHz

| Algoritmo       | Kbytes/seg | Algoritmo       | Kbytes/seg |
|-----------------|------------|-----------------|------------|
| DES             | 35         | Triple DES      | 12         |
| IDEA            | 53         | FEAL (32 v)     | 91         |
| Khufu (16 v)    | 221        | Khufu (32 v)    | 115        |
| RC5 (8 v)       | 127        | RC5 (16 v)      | 65         |
| SAFER (6 v)     | 81         | SAFER (12 v)    | 41         |
| Blowfish (12 v) | 182        | Blowfish (20 v) | 110        |

**Fuente:** Criptografía Digital. Fundamentos y Aplicaciones. José Pastor y Miguel Angel Sarasa, Pressas Universitarias de Zaragoza (1998).






Dada la baja velocidad del PC ☺, estos valores son sólo indicativos para una comparación.

# Algoritmos DES, IDEA y AES

Profundizaremos  en estas diapositivas en los algoritmos DES, Triple DES, IDEA y AES.

¿Por qué?



-  **DES** es un cifrador de Feistel, ha sido un estándar y en aplicaciones bancarias se seguirá usando durante algún tiempo.
-  DES es de muy fácil comprensión y usa cajas S al igual que varios algoritmos más modernos como el actual estándar AES.
-  **Triple DES** sigue siendo un estándar en e-commerce.
-  **IDEA** es un algoritmo seguro que hace uso de los conceptos de inversos en un cuerpo finito, como todos los algoritmos de cifra modernos, y se usa entre otros en la aplicación PGP.
-  **AES** (Rijndael) es el nuevo estándar de cifra avanzada.

## Modos de cifra

Todos los algoritmos pueden usarse aplicando diversos modos de cifra, entre ellos:

- **ECB**: Electronic **C**ode**B**ook (libro electrónico de códigos)
- **CBC**: Cipher **B**lock **C**haining (encadenamiento de bloques)
- **CFB**: Cipher **F**eed**B**ack (realimentación de bloques)
- **OFB**: Output **F**eed**B**ack (realimentación bloque de salida)

Analizaremos cada uno de ellos para el caso del DES, aunque el estudio es extensible a todos los demás ya que en estos modos el cifrador se considera una caja negra.

<http://www.itl.nist.gov/fipspubs/fip81.htm>



# Data Encryption Standard DES

DES (Data Encryption Standard) ha sido el estándar utilizado mundialmente durante 25 años, generalmente en la banca. Hoy presenta signos de envejecimiento y ha sucumbido a los diversos criptoanálisis que contra él se viene realizando hace ya años.

## FECHAS DE INTERÉS



**1973:** En EEUU la NBS National Bureau of Standards llama a concurso público para buscar un algoritmo criptográfico estándar.

**1974:** La NSA National Security Agency declara desierto el primer concurso, publica unas segundas especificaciones y elige Lucifer, algoritmo original de IBM (años 70) con **variaciones**.

**1976:** El DES se adopta como estándar y se autoriza para ser utilizado en las comunicaciones no clasificadas del gobierno.

# Especificaciones del algoritmo DES

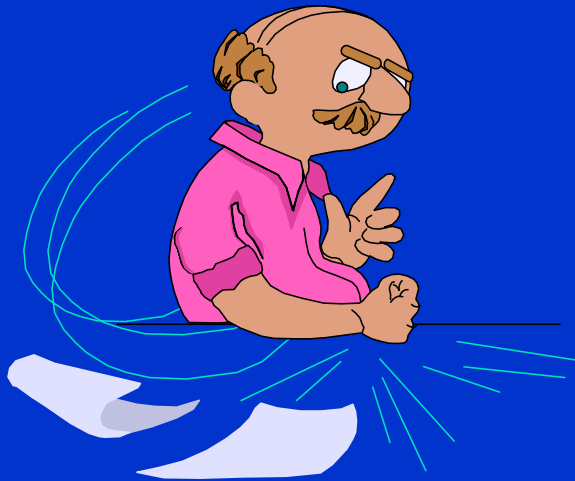
## Especificaciones del concurso

- El nivel de seguridad computacional debe ser alto.
- El algoritmo debe ser fácil de entender y deberá estar especificado en todos sus detalles.
- La seguridad del sistema no debe verse afectada por la publicación y divulgación del algoritmo.
- Debe estar disponible para cualquier usuario.
- Deberá poder usarse en diferentes aplicaciones.
- Fabricación con dispositivos electrónicos de bajo costo.
- Se debe poder usar como validación.
- Debe ser exportable.

No se cumplen en 1973 pero sí en 1974, aunque ...

## El papel de la NSA en el DES

La NSA, National Security Administration, impone una limitación en la longitud de la clave:



$K = 72.057.594.037.927.936$

De los 128 bits de Lucifer, NSA deja la clave en 64 bits. La clave efectiva sólo son 56 bits puesto que al ser datos de 8 bits, no ASCII, se conoce el bit de paridad.

Luego, el espacio de claves será  $2^{56} = 7.2 \cdot 10^{16}$ , tan sólo setenta y dos mil billones de valores, un valor pequeño en criptografía.



## ¿Reducción del tamaño de la clave?

Hay distintas versiones sobre esta reducción del espacio de claves: una habla de la dificultad de diseñar chips capaces de operar de forma eficiente con una clave de 128 bits en esos años 70; la otra sobre una política de seguridad interna para proteger información sensible ante ataques externos y ser capaces, eso sí, de practicar criptoanálisis en un tiempo razonable.



Es muy posible que ambas razones tengan su justificación técnica y política. Ud. puede pensar lo que quiera ☺

# Especificaciones técnicas finales del DES

- Bloque a cifrar: 64 bits
- Clave: 8 bytes (con paridad, no caracteres ASCII)
- Normas ANSI:
  - X3.92: Descripción del algoritmo.
  - X3.108: Descripción de los modos de operación (ECB, CBC, OFB).
- Fácil implementación en un circuito integrado.

Veremos su descripción y modos de operación. En la página que se indica encontrará las especificaciones del DES.

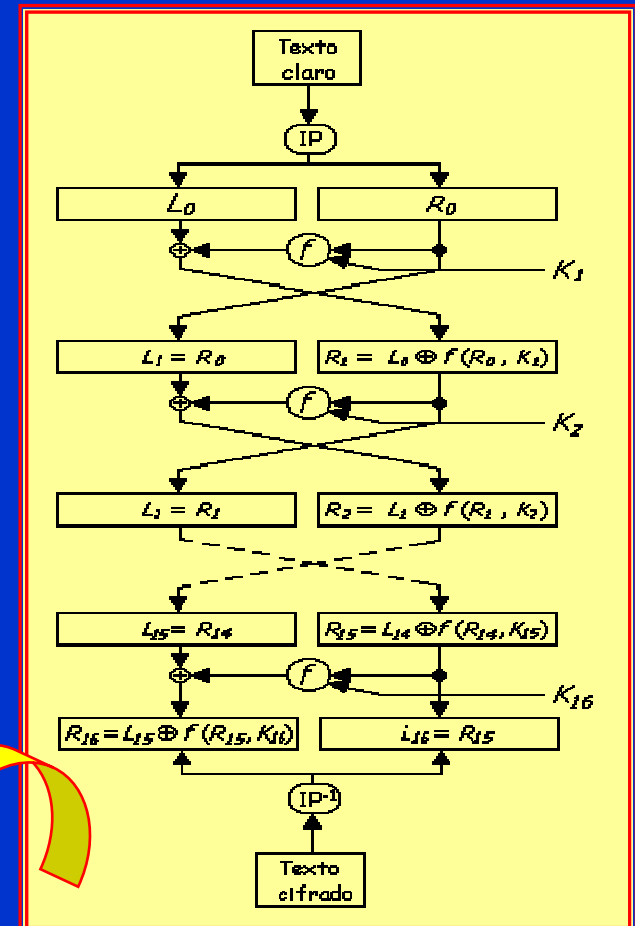
<http://www.itl.nist.gov/fipspubs/fip46-2.htm>



# Visión general del DES

- ❖ Cifrador de bloque
- ❖ Tipo Feistel
- ❖ Longitud de clave de 56 bits
- ❖ Realiza 16 vueltas.
- ❖ La cifra del bloque central usa técnicas de sustituciones y permutaciones.
- ❖ Para poder realizar las sumas or exclusivo, usará permutaciones con expansión y compresión para igualar el número de bits.

En el descifrado se aplican claves y desplazamientos en sentido inverso



# Permutación inicial del DES: tabla IP

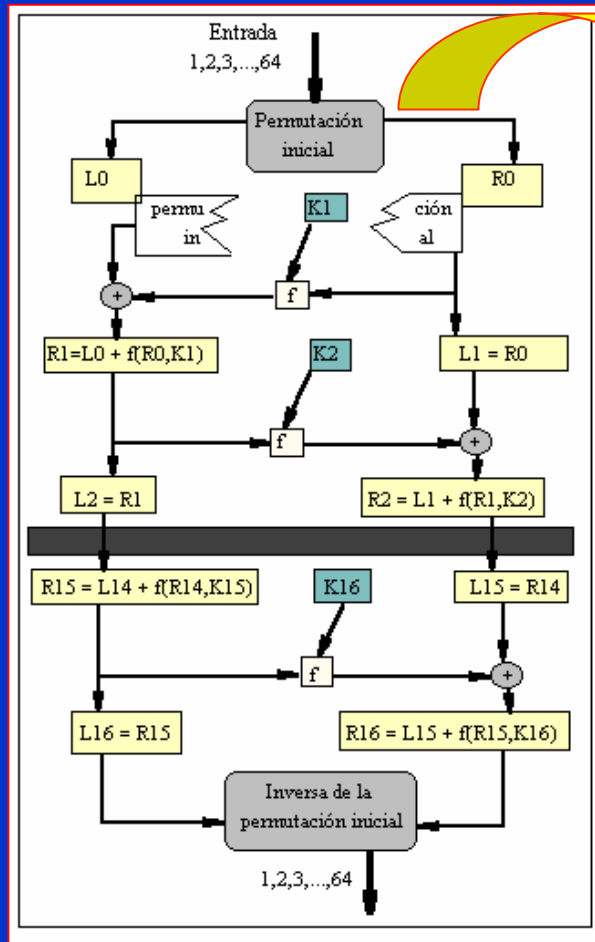


Tabla IP sobre bloque de texto  
(no tiene interés criptográfico)

|    |    |    |    |    |    |    |   |
|----|----|----|----|----|----|----|---|
| 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 lleva a la posición 40

# Bloques izquierdo y derecho de texto

|    |    |    |    |    |    |    |   |
|----|----|----|----|----|----|----|---|
| 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 |

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

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



Observe la distribución correlativa que existe entre los bits del bloque izquierdo  $L_0$  y del bloque derecho  $R_0$  de texto. Este tipo de distribución de los bits en tablas, a simple vista caprichosa ☺, será muy común en el DES.

# Permutación final del DES: tabla IP<sup>-1</sup>

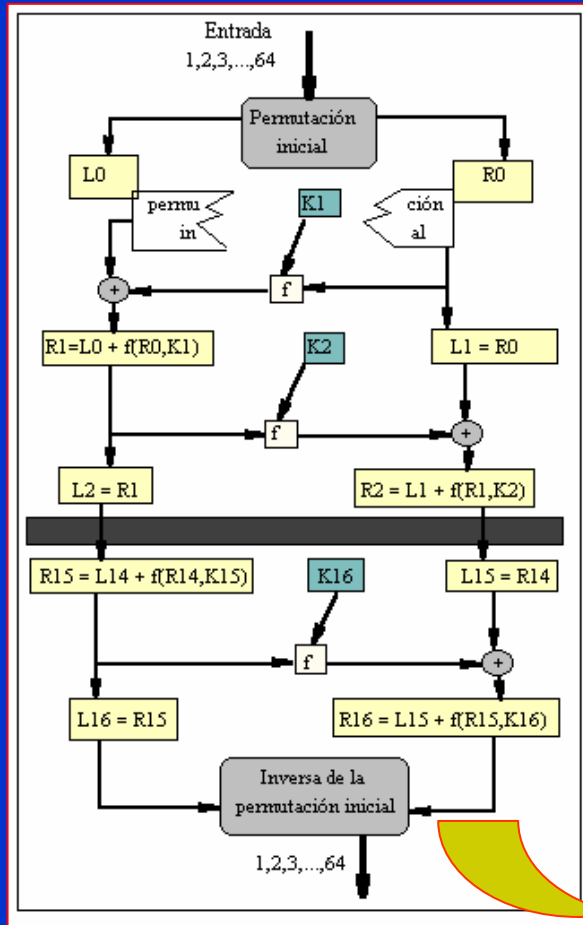


Tabla IP<sup>-1</sup>

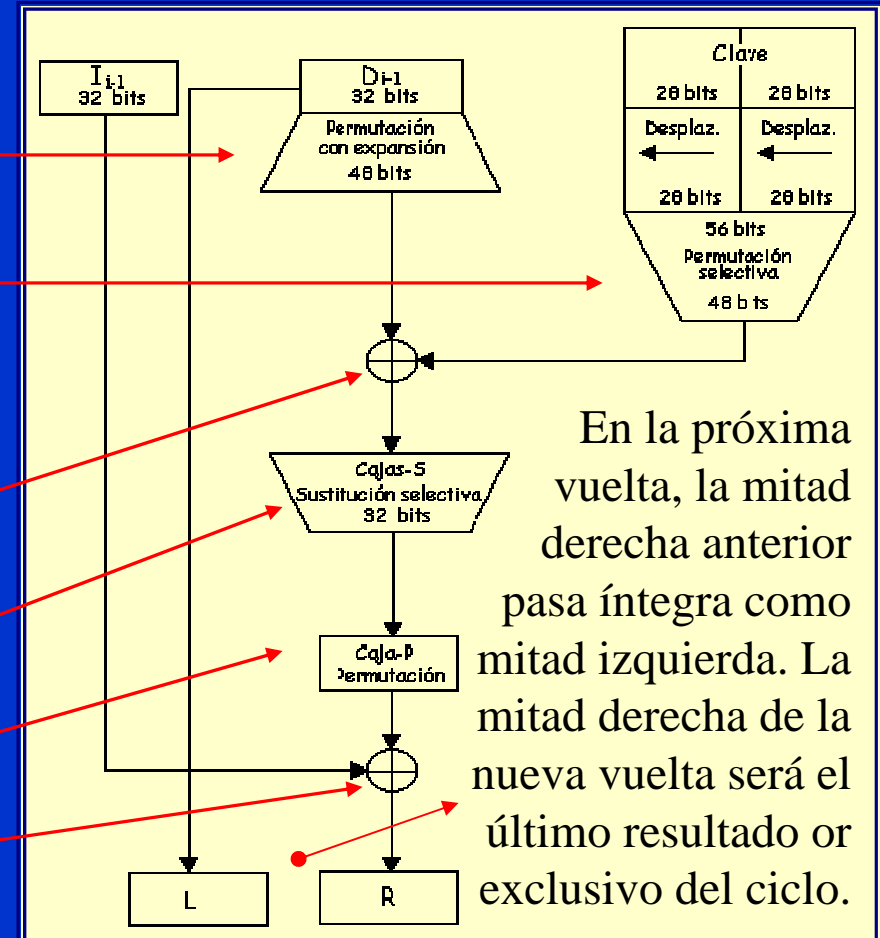
|    |   |    |    |    |    |    |    |
|----|---|----|----|----|----|----|----|
| 40 | 8 | 48 | 16 | 56 | 24 | 64 | 32 |
| 39 | 7 | 47 | 15 | 55 | 23 | 63 | 31 |
| 38 | 6 | 46 | 14 | 54 | 22 | 62 | 30 |
| 37 | 5 | 45 | 13 | 53 | 21 | 61 | 29 |
| 36 | 4 | 44 | 12 | 52 | 20 | 60 | 28 |
| 35 | 3 | 43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

El bit 40 vuelve a la posición 1 y todos los demás bits a su posición inicial antes de IP.

# Operaciones en cada ciclo del DES

## EN CADA CICLO:

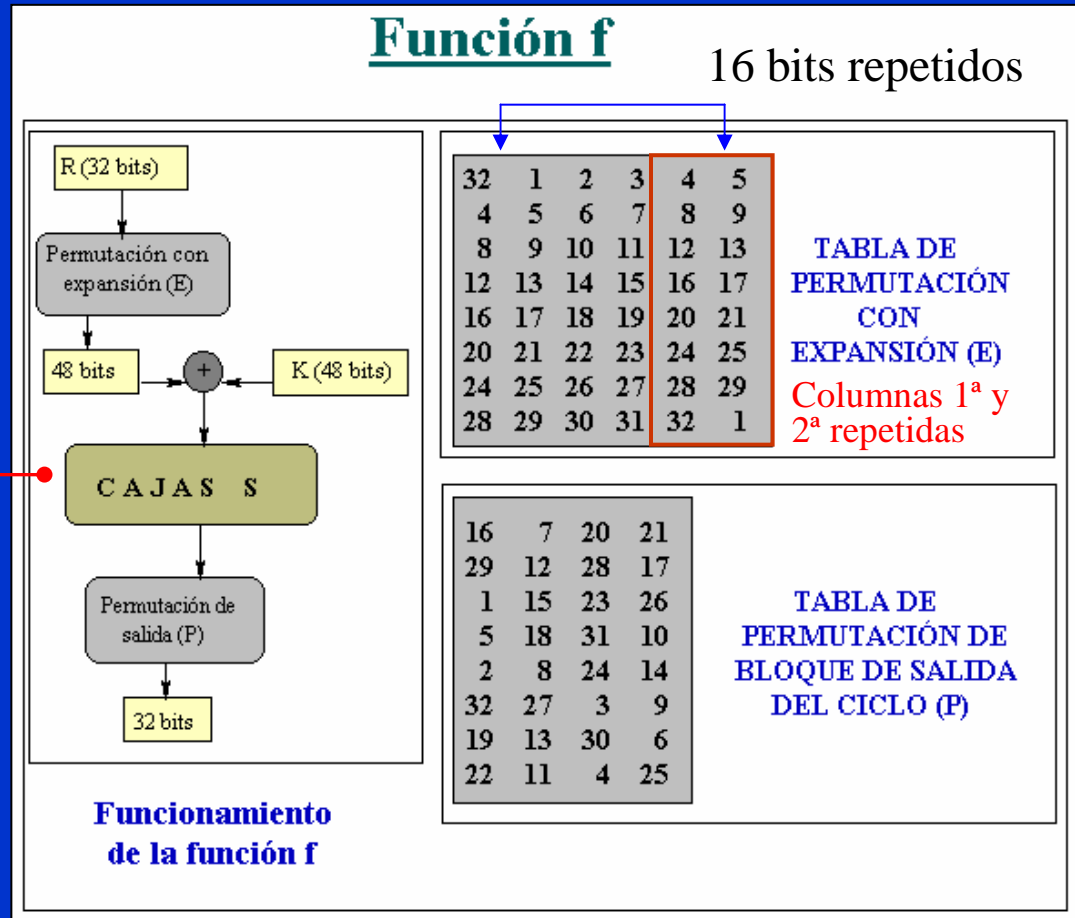
- Se permuta la mitad derecha  $R_i$  aplicando expansión a 48 bits
- La clave de 56 bits se desplaza, permuta y se seleccionan los 48 bits de  $K_i$  de cada vuelta.
- La nueva mitad derecha  $R_i$  y la clave  $K_i$  se suman XOR
- Se reducen los 48 bits de salida a 32 bits mediante las Cajas-S
- Se permuta el resultado
- El resultado se suma XOR con la mitad izquierda  $L_i$



# Módulo de cifra en DES

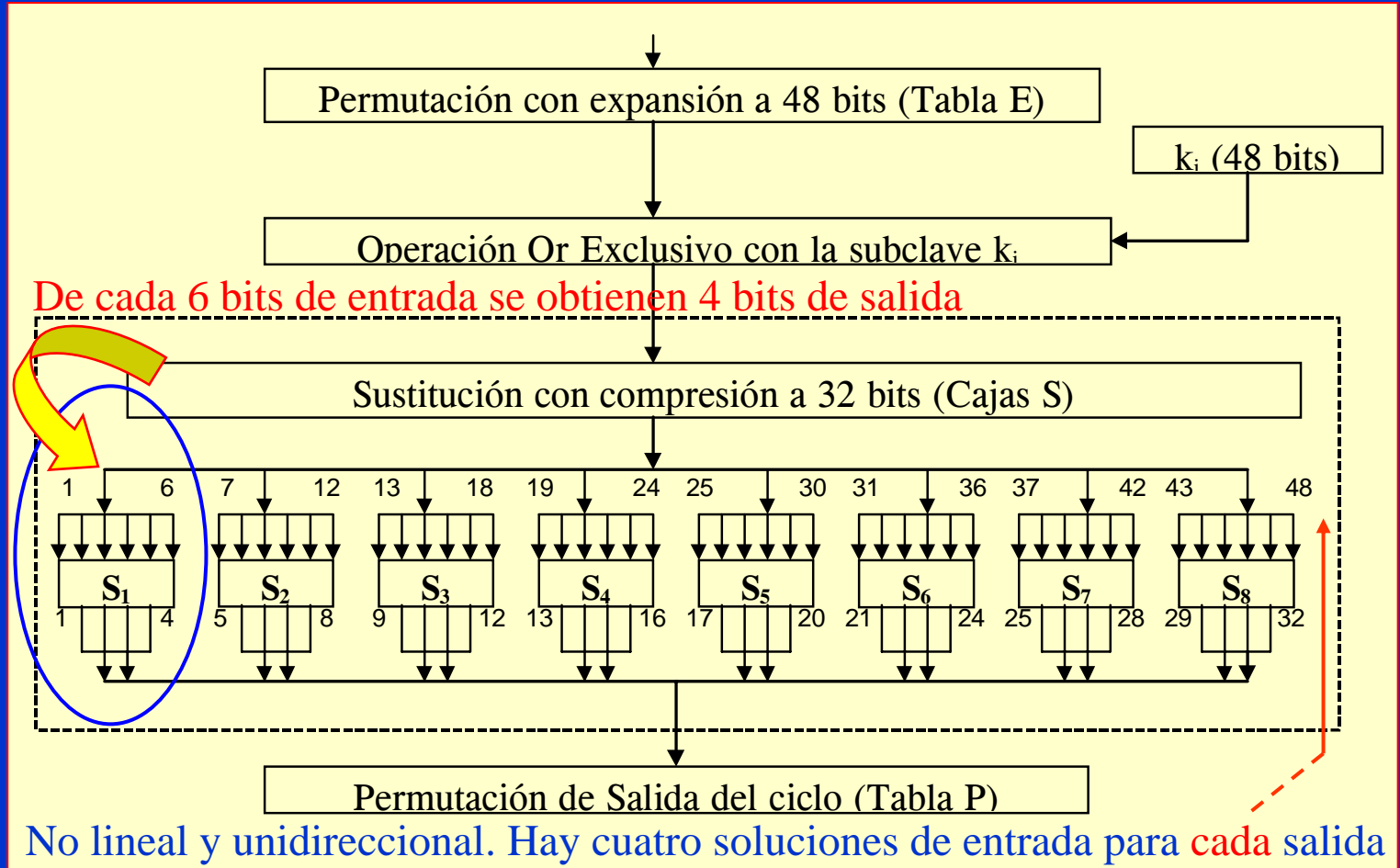
Esquema de la función de cifra  $f$  en cada ciclo

En las cajas  $S$  se consigue la fortaleza del algoritmo. Es una función unidireccional y no lineal.





# Operación de las cajas S en el DES



# Valores de las cajas $S_1$ y $S_2$ del DES

COLUMNAS

|       |   |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|----|----|----|---|----|----|----|----|----|----|----|----|----|----|----|----|
| $S_1$ |   | 0  | 1  | 2  | 3 | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| F     | 0 | 14 | 4  | 13 | 1 | 2  | 15 | 11 | 8  | 3  | 10 | 6  | 12 | 5  | 9  | 0  | 7  |
| I     | 1 | 0  | 15 | 7  | 4 | 14 | 2  | 13 | 1  | 10 | 6  | 12 | 11 | 9  | 5  | 3  | 8  |
| L     | 2 | 4  | 1  | 14 | 8 | 13 | 6  | 2  | 11 | 15 | 12 | 9  | 7  | 3  | 10 | 5  | 0  |
| A     | 3 | 15 | 12 | 8  | 2 | 4  | 9  | 1  | 7  | 5  | 11 | 3  | 14 | 10 | 0  | 6  | 13 |
| S     |   |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |

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     |   |    |    |    |    |    |    |    |    |    |   |    |    |    |    |    |    |

# Valores de las cajas $S_3$ y $S_4$ del DES

COLUMNAS

| $S_3$ |   | 0  | 1  | 2  | 3  | 4 | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|----|----|----|----|---|----|----|----|----|----|----|----|----|----|----|----|
| F     | 0 | 10 | 0  | 9  | 14 | 6 | 3  | 15 | 5  | 1  | 13 | 12 | 7  | 11 | 4  | 2  | 8  |
| I     | 1 | 13 | 7  | 0  | 9  | 3 | 4  | 6  | 10 | 2  | 8  | 5  | 14 | 12 | 11 | 15 | 1  |
| L     | 2 | 13 | 6  | 4  | 9  | 8 | 15 | 3  | 0  | 11 | 1  | 2  | 12 | 5  | 10 | 14 | 7  |
| A     | 3 | 1  | 10 | 13 | 0  | 6 | 9  | 8  | 7  | 4  | 15 | 14 | 3  | 11 | 5  | 2  | 12 |
| S     |   |    |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |

COLUMNAS

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

# Valores de las 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 |
|-------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| F     | 0 | 2  | 12 | 4  | 1  | 7  | 10 | 11 | 6  | 8  | 5  | 3  | 15 | 13 | 0  | 14 | 9  |
| I     | 1 | 14 | 11 | 2  | 12 | 4  | 7  | 13 | 1  | 5  | 0  | 15 | 10 | 3  | 9  | 8  | 6  |
| L     | 2 | 4  | 2  | 1  | 11 | 10 | 13 | 7  | 8  | 15 | 9  | 12 | 5  | 6  | 3  | 0  | 14 |
| A     | 3 | 11 | 8  | 12 | 7  | 1  | 14 | 2  | 13 | 6  | 15 | 0  | 9  | 10 | 4  | 5  | 3  |
| S     |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

COLUMNAS

| $S_6$ |   | 0  | 1  | 2  | 3  | 4 | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|----|----|----|----|---|----|----|----|----|----|----|----|----|----|----|----|
| F     | 0 | 12 | 1  | 10 | 15 | 9 | 2  | 6  | 8  | 0  | 13 | 3  | 4  | 14 | 7  | 5  | 11 |
| I     | 1 | 10 | 15 | 4  | 2  | 7 | 12 | 9  | 5  | 6  | 1  | 13 | 14 | 0  | 11 | 3  | 8  |
| L     | 2 | 9  | 14 | 15 | 5  | 2 | 8  | 12 | 3  | 7  | 0  | 4  | 10 | 1  | 13 | 11 | 6  |
| A     | 3 | 4  | 3  | 2  | 12 | 9 | 5  | 15 | 10 | 11 | 14 | 1  | 7  | 6  | 0  | 8  | 13 |
| S     |   |    |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |

# Valores de las cajas $S_7$ y $S_8$ del DES

COLUMNAS

| $S_7$ |   | 0  | 1  | 2  | 3  | 4  | 5 | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|----|----|----|----|----|---|----|----|----|----|----|----|----|----|----|----|
| F     | 0 | 4  | 11 | 2  | 14 | 15 | 0 | 8  | 13 | 3  | 12 | 9  | 7  | 5  | 10 | 6  | 1  |
| I     | 1 | 13 | 0  | 11 | 7  | 4  | 9 | 1  | 10 | 14 | 3  | 5  | 12 | 2  | 15 | 8  | 6  |
| L     | 2 | 1  | 4  | 11 | 13 | 12 | 3 | 7  | 14 | 10 | 15 | 6  | 8  | 0  | 5  | 9  | 2  |
| A     | 3 | 6  | 11 | 13 | 8  | 1  | 4 | 10 | 7  | 9  | 5  | 0  | 15 | 14 | 2  | 3  | 12 |
| S     |   |    |    |    |    |    |   |    |    |    |    |    |    |    |    |    |    |

COLUMNAS

| $S_8$ |   | 0  | 1  | 2  | 3 | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|----|----|----|---|----|----|----|----|----|----|----|----|----|----|----|----|
| F     | 0 | 13 | 2  | 8  | 4 | 6  | 15 | 11 | 1  | 10 | 9  | 3  | 14 | 5  | 0  | 12 | 7  |
| I     | 1 | 1  | 15 | 13 | 8 | 10 | 3  | 7  | 4  | 12 | 5  | 6  | 11 | 0  | 14 | 9  | 2  |
| L     | 2 | 7  | 11 | 4  | 1 | 9  | 12 | 14 | 2  | 0  | 6  | 10 | 13 | 15 | 3  | 5  | 8  |
| A     | 3 | 2  | 1  | 14 | 7 | 4  | 10 | 8  | 13 | 15 | 12 | 9  | 0  | 3  | 5  | 6  | 11 |
| S     |   |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |

# Ejemplo de operación de cajas S del DES

## Ejemplo:

Sean los bits 7 al 12 los siguientes: 101100

Los bits corresponderán entonces a la entrada de la caja  $S_2$

Para seleccionar la fila tomamos los bits extremos:  $10_2 = 2_{10} = 2$

Para seleccionar la columna tomamos los bits centrales:  $0110_2 = 6_{10} = 6$

La caja  $S_2$  indica una salida igual a  $13_{10} = 1101_2$

explicación

| S2 | 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  |

Entrada: 101100 (6 bits)

Salida: 1101 (4 bits)

# Cálculo de subclaves en el DES (PC-1)

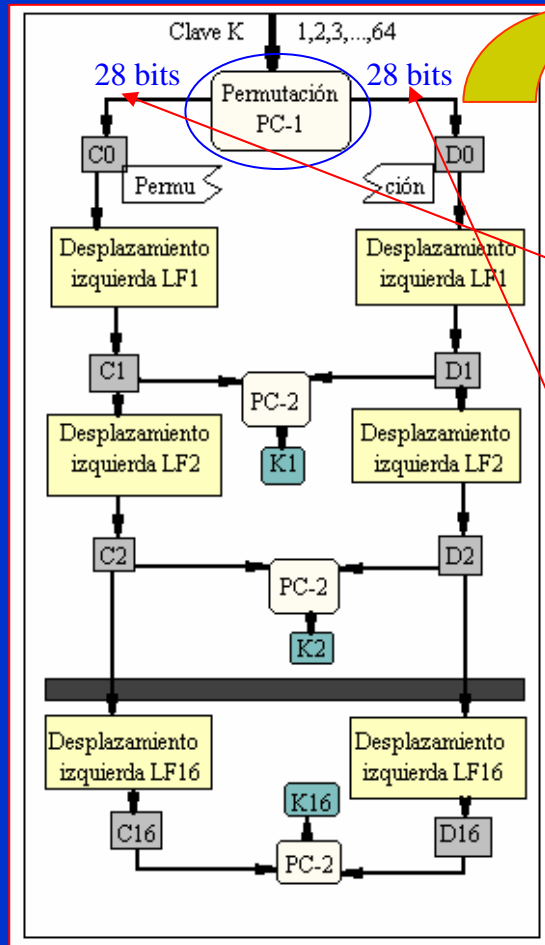


Tabla PC-1 (56 bits)

|    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|
| 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  |

Se han eliminado los bits de paridad:  
8, 16, 24, 32, 40, 48, 56, 64.

# Cálculo de subclaves en el DES (PC-2)

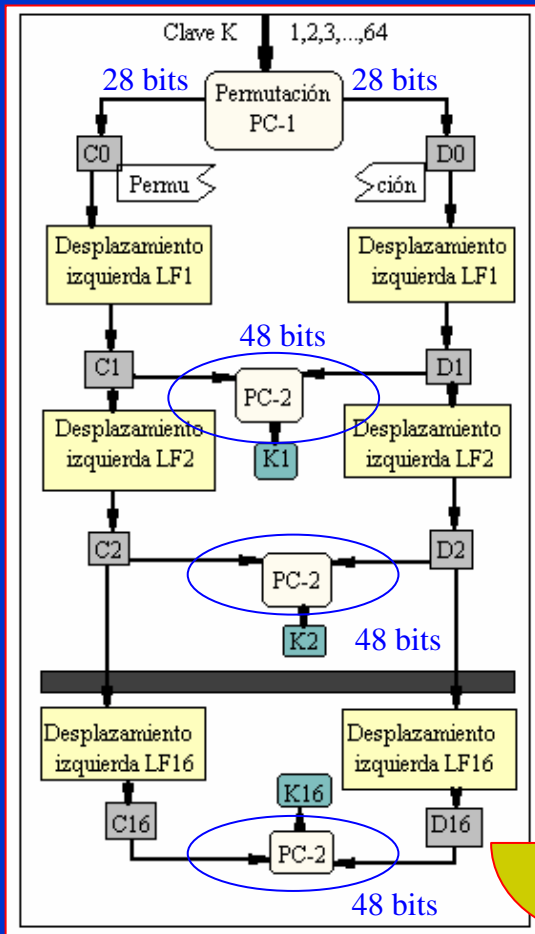


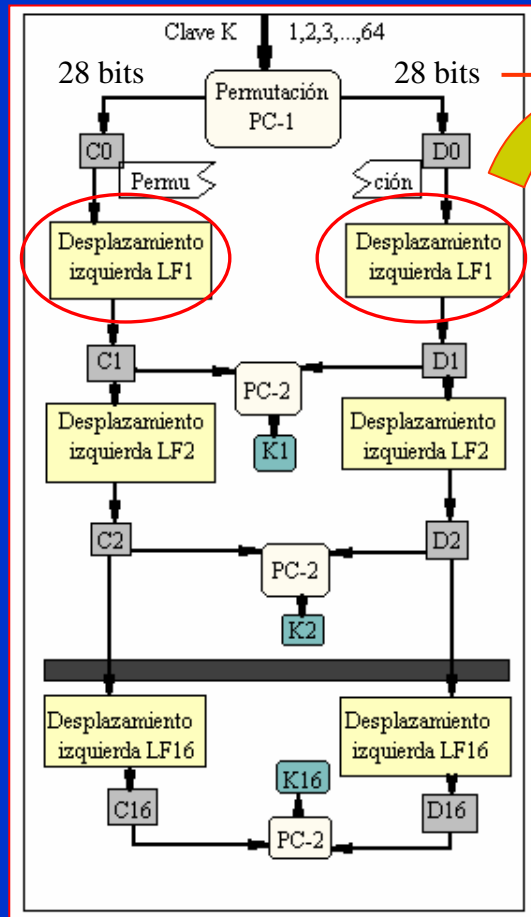
Tabla PC-2 (48 bits)  $\Rightarrow k_1, k_2, \dots, k_{16}$

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 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 los bits:  
9, 18, 22, 25, 35, 38, 43, 54.



# Desplazamiento de subclaves en el DES



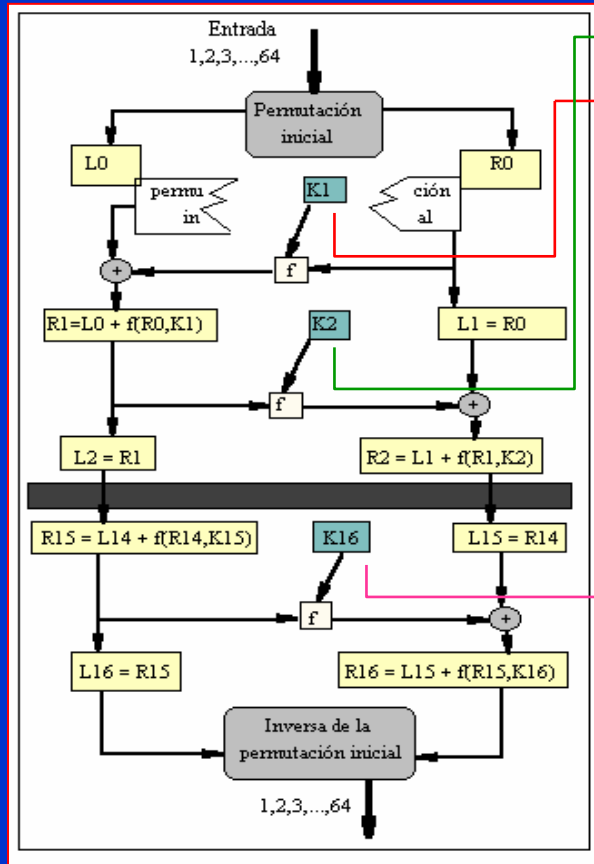
Se produce un desplazamiento total igual a 28, todos los bits de cada bloque  $C_i$  y  $D_i$

$LF_1, LF_2, \dots, LF_{16}$

| Vuelta i | Bits Desp. Izda. | Vuelta i | Bits Desp. Izda. |
|----------|------------------|----------|------------------|
| 1        | 1                | 9        | 1                |
| 2        | 1                | 10       | 2                |
| 3        | 2                | 11       | 2                |
| 4        | 2                | 12       | 2                |
| 5        | 2                | 13       | 2                |
| 6        | 2                | 14       | 2                |
| 7        | 2                | 15       | 2                |
| 8        | 2                | 16       | 1 $\Sigma$       |

# Operación de descifrado en el DES

64 bits de criptograma



Se toman en sentido contrario:

$k_{16}, k_{15}, k_{14}, k_{13}, k_{12}, k_{11}, k_{10},$   
 $k_9, k_8, k_7, k_6, k_5, k_4, k_3, k_2, k_1$

Como se aplica un desplazamiento de 28 bits en cada bloque de clave, entonces  $D_{16} = D_0$  y  $C_{16} = C_0$

Los desplazamientos para el cálculo de las subclaves de descifrado son los mismos de la tabla anterior pero ahora se toman hacia la derecha, puesto que los desplazamientos coinciden.

# Claves débiles y semidébiles

## Claves débiles en hexadecimal:

Una clave es débil si se verifica que:  $E_k[E_k(M)] = M$

Además de 0000000000000000 y FFFFFFFF (que son obvias) serán débiles estas cuatro claves:

|                  |                  |
|------------------|------------------|
| 0101010101010101 | FEFEFEFEFEFEFEFE |
| EOEOEOEO1F1F1F1F | 1F1F1F1F0E0E0E0E |

Los bloques C y D de la clave son todos 0s ó 1s.

## Claves semidébiles en hexadecimal:

Una clave es semidébil si se verifica que:  $E_{k_1}[E_{k_2}(M)] = M$

Son claves  $k_1, k_2$  semidébiles las siguientes seis parejas:

|                    |                   |
|--------------------|-------------------|
| (01FE01FE01FE01FE, | FE01FE01FE01FE01) |
| (1FE01FE00EF10EF1, | E01FE01FF10EF10E) |
| (01E001E001F101F1, | E001E001F101F101) |
| (1FFE1FFE0EFE0EFE, | FE1FFE1FFE0EFE0E) |
| (011F011F010E010E, | 1F011F010E010E01) |
| (E0FEE0FEF1FEF1FE, | FEE0FEE0FEF1FEF1) |

Además de éstas, hay otras ecuaciones que verifican dichas claves.

# Ejemplo de cálculo de claves en DES

**Ejemplo 1:** a partir de las tablas PC-1 y PC-2, encuentre la secuencia de bits de los registros  $C_1$  y  $D_1$  y la subclave  $k_1$ .

**Solución:**

```
i = 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
C1 = 49 41 33 25 17 09 01 58 50 42 34 26 18 10 02 59 51 43 35 27 19 11 03 60 52 44 36 57
i = 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
D1 = 55 47 39 31 23 15 07 62 54 46 38 30 22 14 06 61 53 45 37 29 21 13 05 28 20 12 04 63
k1 = 10 51 34 60 49 17 33 57 02 09 19 42 03 35 26 25 44 58 59 01 36 27 18 41
      22 28 39 54 37 04 47 30 05 53 23 29 61 21 38 63 15 20 45 14 13 62 55 31
```

**Ejemplo 2:** si la clave es PRUEBALO, encuentre  $C_0$  y  $D_0$  y la subclave  $k_1$ . Para encontrar  $C_0$  y  $D_0$  escriba en ASCII la clave y elimine el último bit de cada byte como si fuese el de paridad.

**Solución:**

```
C0 = 0000 0000 1111 1111 0000 0000 0000
C1 = 1001 0010 1100 1100 1100 0000 0111
k1 = 101000 001001 001001 000010 101101 010100 100111 100100
```

## Modo de cifra ECB

Recuerde que estos modos son válidos para todos los cifradores en bloque

**Electronic CodeBook:** cifra cada bloque con la clave  $k$  de forma independiente. Por lo tanto, el resultado es como si se codificase mediante un gran libro electrónico de códigos.

👉 Recuerde: **codificar** no es lo mismo que **cifrar**.

### Debilidades:

- ☹ Se podría reconstruir ese libro electrónico sin necesidad de conocer la clave.
- ☹ Aparece el problema denominado de comienzos y finales fijos que permiten un tipo de ataque sencillo.
- ☹ Se ataca a través de la repetición de bloques similares.

# Características del modo ECB en DES



Cada bloque de 64 bits del texto en claro se pasa por el cifrador, usando la misma clave de 64 bits.



Para bloques de texto en claro iguales, se obtiene siempre el mismo criptograma.

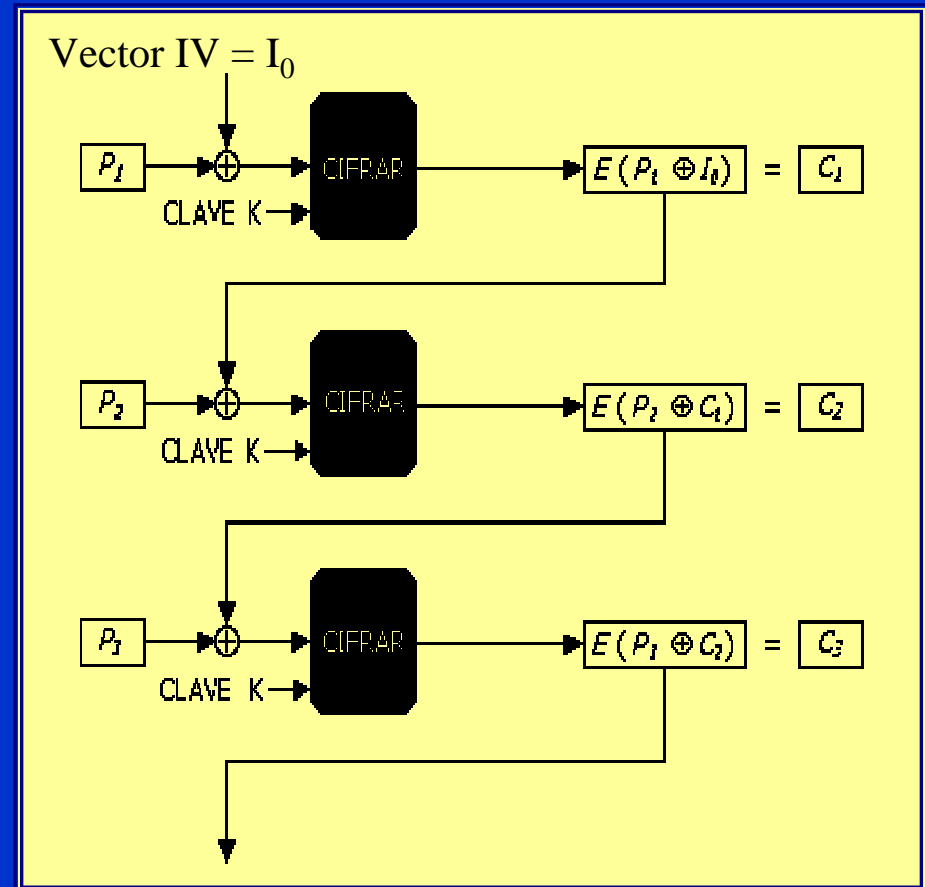


Como a cada bloque de texto en claro le corresponde un único código o texto cifrado de salida y éste es constante, este modo de cifra lleva por nombre Libro Electrónico de Códigos. Es como si tuviésemos un gran libro de código con un código distinto para cada mensaje.

# Modo de cifra CBC en DES

**Cipher Block Chaining:**  
cifra por encadenamiento  
de bloques (el más común)

- Se encadenan los bloques de texto en claro con el bloque del criptograma anterior.
- Usa un vector de inicialización IV de 64 bits que se guarda en secreto.



# Operaciones de cifra modo CBC en DES

## Cifrado

El vector IV se suma XOR a los 64 bits de texto en claro.

Se cifra con la clave K esa suma.

El resultado  $C_i$  se usa como vector IV para el nuevo bloque.

## Descifrado

Se descifra el primer bloque con vector IV:

$$P_1 = D(C_1) \oplus I_0$$

$$P_1 = D[E(P_1 \oplus I_0)] \oplus I_0$$

Se guarda el bloque  $C_{i-1}$  en un registro. Se descifra el bloque  $C_i$  y luego XOR entre esos bloques:

$$M_i = D(C_i) \oplus C_{i-1}$$

## CARACTERÍSTICAS:

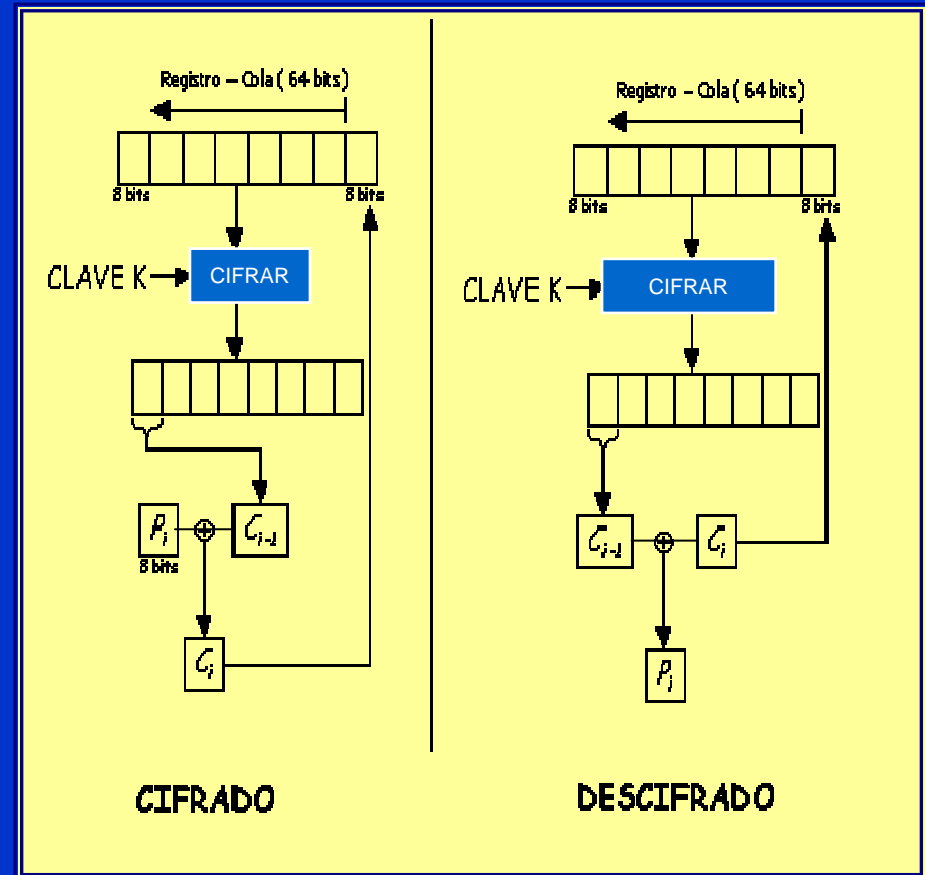
Evita el ataque por repetición de bloque. Enmascara el mensaje lo mismo que la cifra en flujo. El espacio de claves es igual a 64 bits. La propagación de un error afecta a dos bloques contiguos.



# Modo de cifra CFB en DES

**Cipher FeedBack:** cifrado por realimentación de bloques

- Se pueden cifrar unidades de datos más pequeñas que bloques, por lo general un byte.
- Se usa un registro de desplazamiento RD de 64 bits como vector inicial IV.



# Operaciones de cifra modo CFB en DES

## Cifrado

Se suma XOR cada byte del texto claro con bytes resultado de la cifra de RD y la clave K. El byte  $C_i$  se envía al registro; se desplazan a la izquierda 8 bits hasta formar otro RD y se repite el proceso de cifra.

## Descifrado

Se cifra el registro RD. Se obtienen de esta forma los elementos de  $C_{i-d}$ . Se suma XOR los  $C_{i-d}$  con los  $C_i$  del criptograma para obtener  $P_i$ . Se realimenta  $C_i$  al registro RD y se repite el proceso.

## CARACTERÍSTICAS:

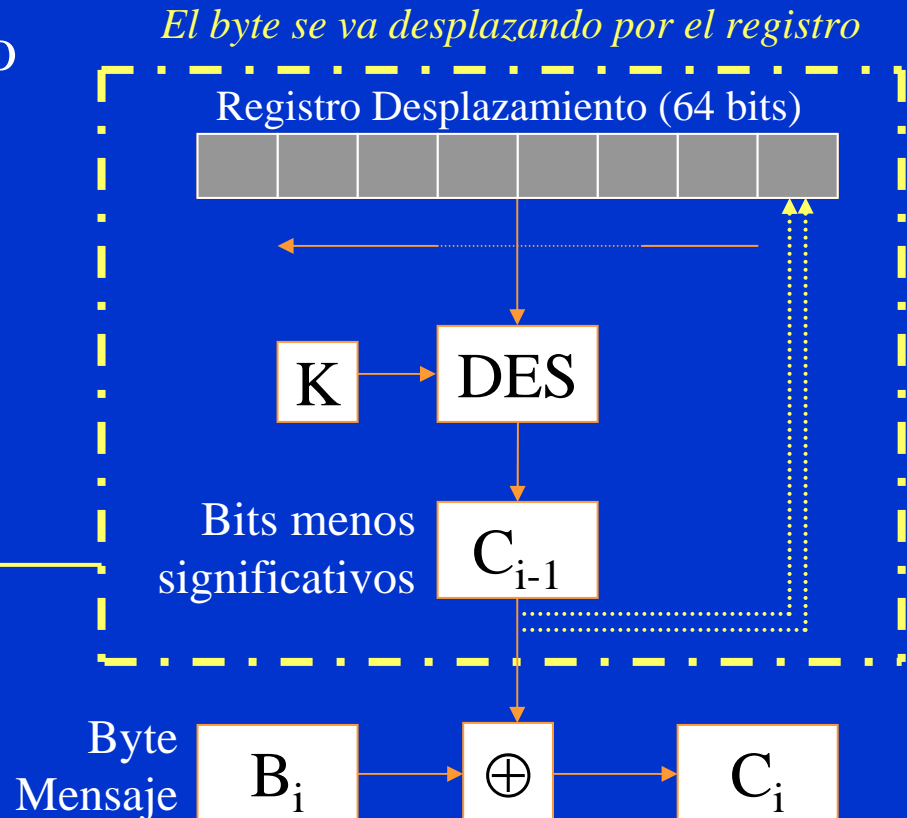
Evita el ataque por repetición de bloque; enmascara el mensaje como en cifra en flujo, el espacio de claves es igual a 64 bits; la propagación de un error se limita a un bloque.

# Modo de cifra OFB en DES

**Output FeedBack:** cifrado por realimentación de bloques de salida

La realimentación de la señal se realiza antes de la operación XOR.

El DES, la clave y el Registro RD actúan como un generador de secuencia cifrante.



Si la cifra se realiza bit a bit, OFB se convierte en cifrador de flujo.

## Características del modo OFB en DES

- ✓ Evita el ataque por repetición de bloque.
- ✓ Produce un enmascaramiento del mensaje similar al de un cifrador de flujo.
- ✓ El espacio de claves es igual a 64 bits.
- ✓ La propagación de un error afecta sólo a un byte, el que se realimenta en el registro de desplazamiento.
- ✓ Las operaciones de cifrado y descifrado son iguales.

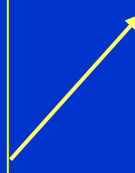
A pesar de las propiedades interesantes de los últimos modos, el más utilizado en los sistemas de cifra de diversos protocolos es el CBC.

## Cifrado múltiple en un grupo

Si un sistema forma un grupo, entonces cifrar un mensaje  $M$  con una clave  $k_1$  y luego el resultado con una clave  $k_2$ , es lo mismo que cifrar el mensaje con una única clave  $k_3$ .

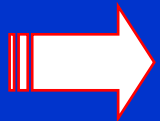
Por ejemplo, el cifrador de Vigenère es un grupo como se demuestra a continuación. Sea  $k_1 = \text{PACO}$  y  $k_2 = \text{CINE}$  y el mensaje a cifrar  $M = \text{ESTO ES UN GRUPO}$ .

|         |      |      |      |   |
|---------|------|------|------|---|
| $M_1 =$ | ESTO | ESUN | GRUP | O |
| $k_1 =$ | PACO | PACO | PACO | P |
| $C_1 =$ | TSVD | TSWB | VRWE | E |



|         |      |      |      |   |
|---------|------|------|------|---|
| $M_2 =$ | TSVD | TSWB | VRWE | E |
| $k_2 =$ | CINE | CINE | CINE | C |
| $C_2 =$ | VAIH | VAJF | XZJI | G |

Obtendremos lo mismo si ciframos el mensaje  $M$  con la clave  $k_3 = k_1 + k_2 = \text{PACO} + \text{CINE} = \text{RIOS}$ .



## El DES no es un grupo

$M_1$  = ESTO ESUN GRUP O

$k_1$  = PACO PACO PACO P

$C_1$  = TSVD TSWB VRWE E

$M_2$  = TSVD TSWB VRWE E

$k_2$  = CINE CINE CINE C

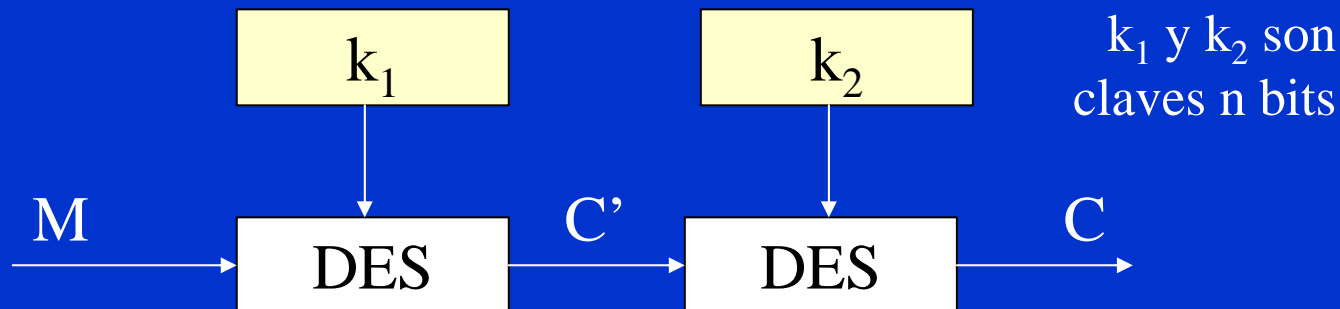
$C_2$  = VAIH VAJF XZJI G

|   |  |   |
|---|--|---|
|  | $M_3$ = ESTO ESUN GRUP O<br>$k_3$ = RIOS RIOS RIOS R<br>$C_3$ = VAIH VAJF XZJI G |  |
|---|--|---|

Como ejercicio compruebe que a resultados similares llega si, por ejemplo, usa ahora los siguientes pares de claves: LAPALA y LANUCA; PASA y NADA; PAÑOS y TERMA. ¿Cuáles son las claves  $k_3$  en cada caso? 😊

El DES no será un grupo y, por lo tanto, permitirá el cifrado múltiple. Esto aumentará el tamaño efectivo de la clave.

## ¿Podríamos usar un doble DES?

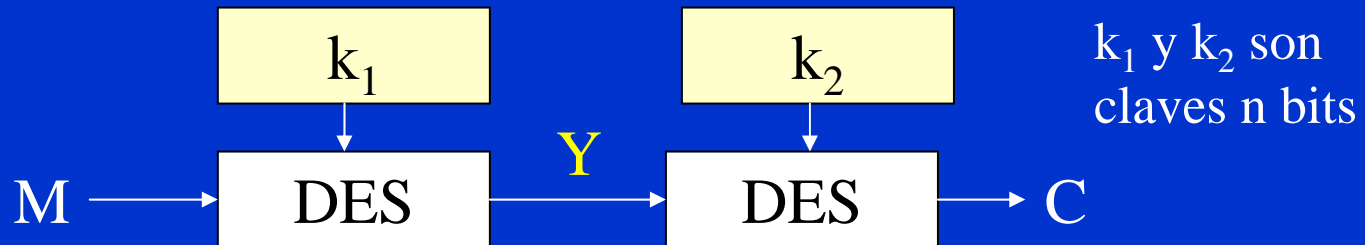


### ¿Se duplica la longitud de la clave?

En este modelo, cabe esperar que la longitud efectiva de la clave sea  $2^{2n}$  donde  $n$  representa la longitud de bits de las claves  $k_1$  y  $k_2$ . No obstante esto no es cierto.

En realidad el tamaño de la clave resultante en este caso es equivalente a  $2^{n+1}$ , un aumento insignificante (un solo bit) para un valor de  $n$  grande (típico) y por esta razón no se usa.

# Ataque por encuentro a medio camino

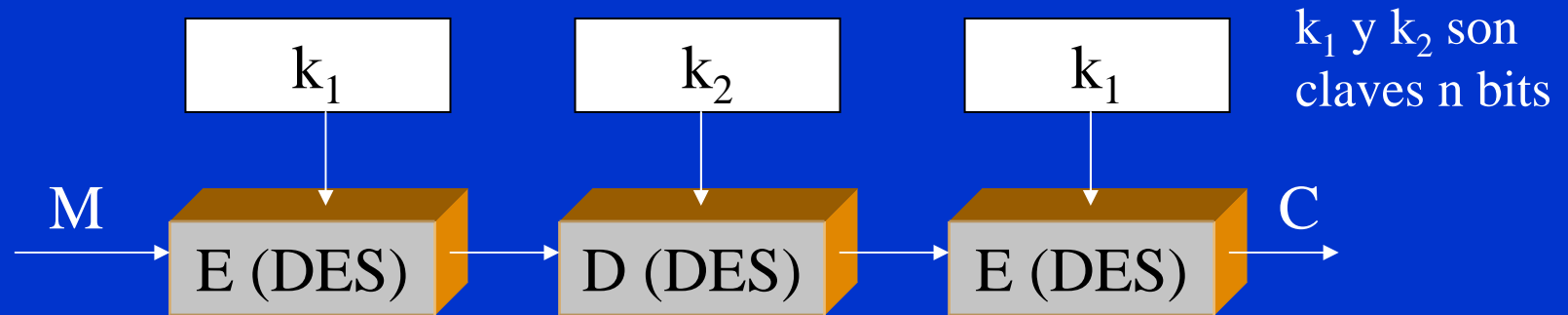


- Se describe el criptograma  $C$  por fuerza bruta usando las  $2^n$  claves posibles y realizando entonces  $2^n$  cálculos. Se obtiene así  $Y$ .
- Con los “textos intermedios”  $Y$  se forma una tabla ordenada de textos cifrados con sus correspondientes valores  $k_2$ .
- Se cifra el texto en claro  $M$  conocido con todas las claves  $k_1$  y se comparan los resultados con  $Y$ , realizando un máximo de  $2^n$  cálculos.
- Una de las claves será la verdadera y se ha realizado un número menor que  $2^n + 2^n = 2^{n+1}$  cálculos. Luego la clave real es igual a  $2^{n+1}$ .

Este ataque se conoce con el nombre de meet-in-the-middle.

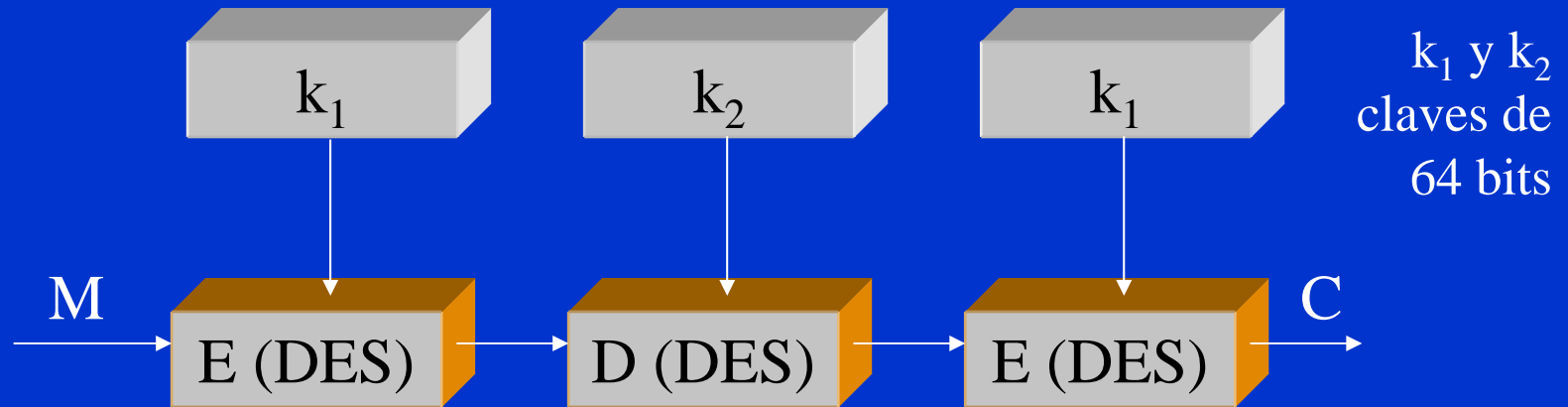


# Triple DES tipo EDE



- En este caso se logra un valor efectivo de longitud de clave igual a  $2^{2n}$  bits, es decir  $2^{2 \cdot 56} = 2^{112}$  bits efectivos.
- El modelo anterior con sólo dos claves es compatible con el DES de clave única cuando  $k_1 = k_2$ . Es más eficiente y equivalente al cifrado triple con claves  $k_1, k_2, k_3$ .
- Este modelo fue propuesto por Matyas y Meyer de IBM, se conoce como EDE (Encrypt-Decrypt-Encrypt) y es inmune a ataques por encuentro a medio camino.

# Usos de Triple DES







Aunque el algoritmo DES haya sufrido diversos ataques y no se haya vuelto a certificar por el NIST como estándar de cifrado, el Triple DES sí tiene una gran seguridad debido al tamaño de su clave de 112 bits efectivos y sigue siendo válido en el año 2005. De hecho, era el algoritmo propuesto en el protocolo SET y se encuentra, entre otras aplicaciones, en el programa PGP.

<http://www.rsasecurity.com/rsalabs/node.asp?id=2231>



# International Data Encryption Algorithm IDEA

## Historia del IDEA

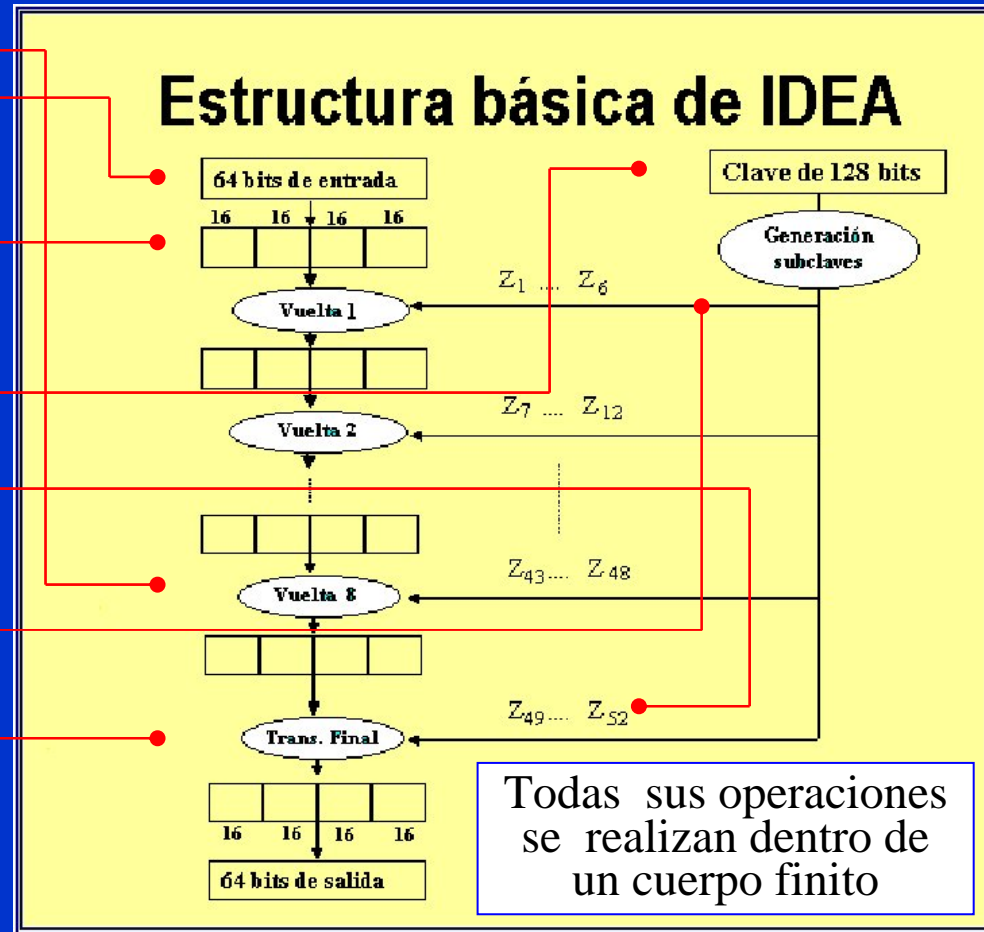
-  En 1990 Xuejia Lai y James Massey proponen el PES, Proposed Encryption Standard.
-  En 1991 -debido a los avances de Biham y Shamir en el criptoanálisis diferencial- los autores proponen el IPES, Improved Proposed Encryption Standard.
-  En 1992 los autores proponen finalmente el algoritmo IDEA, International Data Encryption Algorithm.
-  En 1999 el algoritmo IDEA, mucho más seguro que el DES y sus versiones, se comienza a usar ampliamente en el sistema de correo electrónico seguro PGP.

[http://en.wikipedia.org/wiki/International\\_Data\\_Encryption\\_Algorithm](http://en.wikipedia.org/wiki/International_Data_Encryption_Algorithm)



# Estructura y esquema de IDEA

- Cifra bloques de 64 bits en 8 vueltas
- Divide la entrada  $M$  en cuatro bloques de 16 bits
- Se generan 52 subclaves de 16 bits a partir de la clave maestra de 128 bits
- Usa 6 claves por vuelta
- Hay una transformación final con 4 claves para invertir operación inicial



# Operaciones matemáticas en IDEA

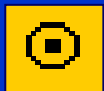
## Operaciones básicas



XOR



Suma módulo  $2^{16}$  (mod 65.536)



Multiplicación módulo  $2^{16}+1$  (65.537)

Es primo y se  
asegura el inverso  
multiplicativo



Todas las operaciones se realizan con bloques de 16 bits y el *truco* está en que los bloques cuyo valor sea 0 (16 bits) se cambiarán por la constante  $2^{16}$  ...de 17 bits ☺. Simplemente representa la salida con n bits, no teniendo en cuenta el bit de desbordamiento. Un ejemplo con números pequeños: ➡

# Operaciones $+$ , $\otimes$ y $\oplus$ en grupo pequeño

Ejemplo dentro de un grupo  $n$  pequeño

Como  $2^n + 1$  debe ser primo, sea  $n = 2$  ya que  $2^2 = 4$  y  $2^2 + 1 = 5$

| X |    | Y |    | X + Y |    | X $\otimes$ Y |    | X $\oplus$ Y |    |
|---|----|---|----|-------|----|---------------|----|--------------|----|
| 0 | 00 | 0 | 00 | 0     | 00 | 1             | 01 | 0            | 00 |
| 0 | 00 | 1 | 01 | 1     | 01 | 0             | 00 | 1            | 01 |
| 0 | 00 | 2 | 10 | 2     | 10 | 3             | 11 | 2            | 10 |
| 0 | 00 | 3 | 11 | 3     | 11 | 2             | 10 | 3            | 11 |
| 1 | 01 | 0 | 00 | 1     | 01 | 0             | 00 | 1            | 01 |
| 1 | 01 | 1 | 01 | 2     | 10 | 1             | 01 | 0            | 00 |
| 1 | 01 | 2 | 10 | 3     | 11 | 2             | 10 | 3            | 11 |
| 1 | 01 | 3 | 11 | 0     | 00 | 3             | 11 | 2            | 10 |
| 2 | 10 | 0 | 00 | 2     | 10 | 3             | 11 | 2            | 10 |
| 2 | 10 | 1 | 01 | 3     | 11 | 2             | 10 | 3            | 11 |
| 2 | 10 | 2 | 10 | 0     | 00 | 0             | 00 | 0            | 00 |
| 2 | 10 | 3 | 11 | 1     | 01 | 1             | 01 | 1            | 01 |
| 3 | 11 | 0 | 00 | 3     | 11 | 2             | 10 | 3            | 11 |
| 3 | 11 | 1 | 01 | 0     | 00 | 3             | 11 | 2            | 10 |
| 3 | 11 | 2 | 10 | 1     | 01 | 1             | 01 | 1            | 01 |
| 3 | 11 | 3 | 11 | 2     | 10 | 0             | 00 | 0            | 00 |

$n = 2$   
dos bits

Veremos cómo se opera con la multiplicación. La suma y el or exclusivo son operaciones similares.

Operaciones:  $+$  mod  $2^n$  (mod 4),  $\otimes$  mod  $2^n+1$  (mod 5), XOR (mod 2)

## Ejemplo de operación $\otimes$ en IDEA

Diagram illustrating the construction of a finite field  $GF(2^4)$  using polynomial multiplication modulo an irreducible polynomial. The diagram shows five columns:  $X$ ,  $Y$ ,  $X+Y$ ,  $X \otimes Y$ , and  $X \oplus Y$ . Each column contains a 4x4 grid of values. Callouts explain specific calculations:

- $0 \otimes 1 = 2^2 \times 1 = 4 = 4 \bmod 5 = 4 = 0$  (por definición)
- $0 \otimes 2 = 2^2 \times 2 = 8 = 8 \bmod 5 = 3$
- $0 \otimes 3 = 2^2 \times 3 = 12 = 12 \bmod 5 = 2$

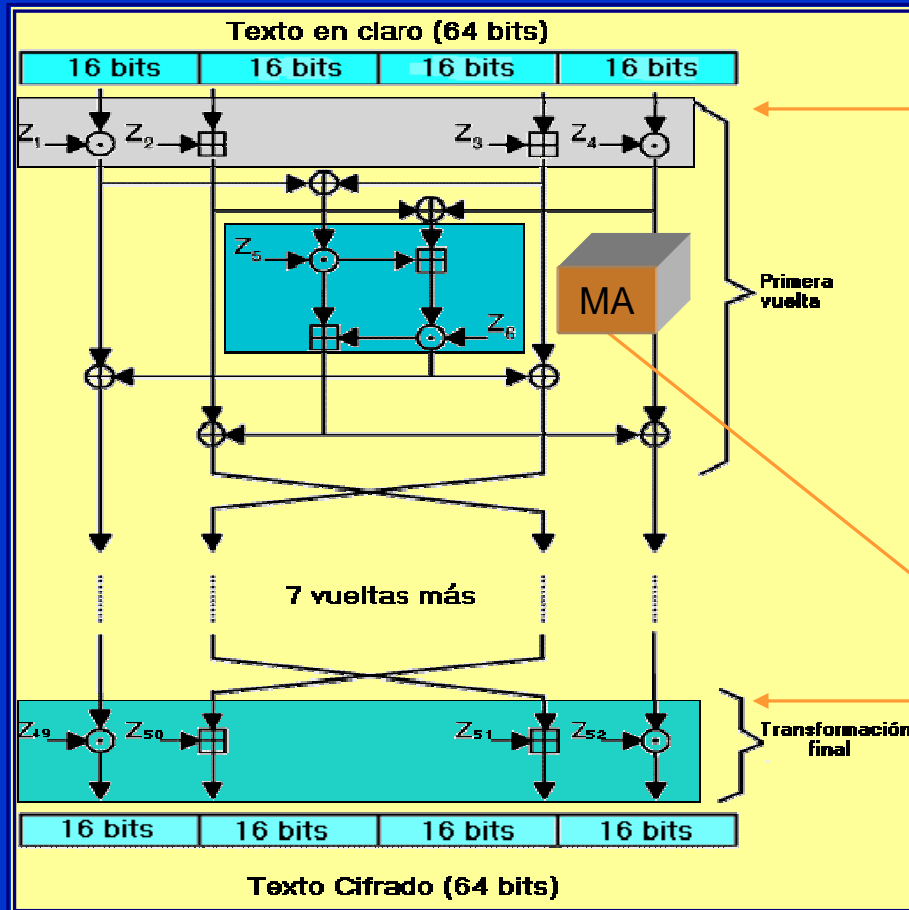
Recuerde que 0 es igual a  $2^n = 4$  por lo que:

$$0 \otimes 0 = 2^2 \times 2^2 = 16 \bmod 5 = 1$$

Operaciones:  $+$  mod  $2^n$  (mod 4),  $\otimes$  mod  $2^{n+1}$  (mod 5), XOR (mod 2)

Los demás cálculos con los diferentes valores de X e Y son todos similares

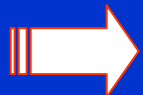
# Detalles del algoritmo IDEA



Operación cifrado

Operaciones inversas al comienzo y al final del algoritmo. Esto permite usar el mismo algoritmo para cifrar que para descifrar.

Bloque principal





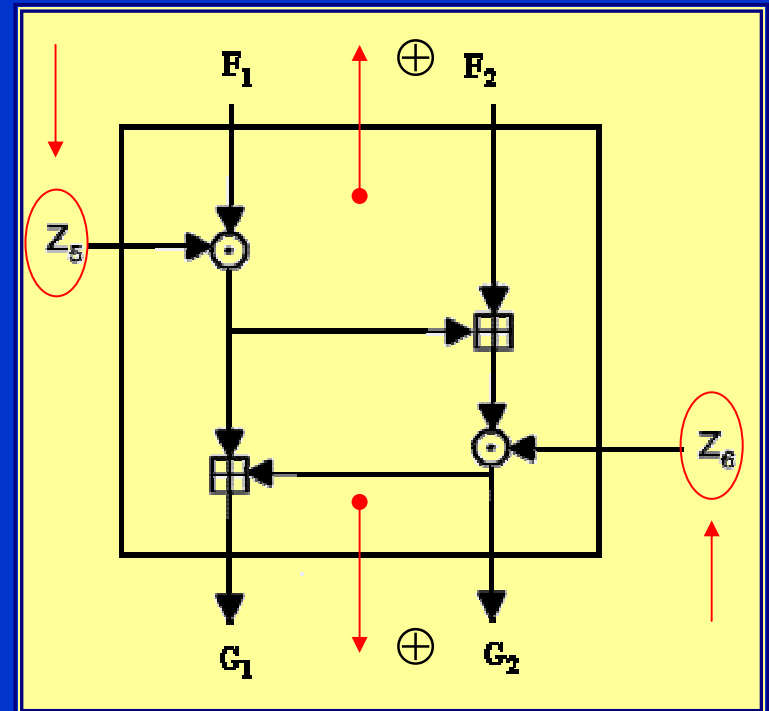
# Bloque principal de IDEA



Estas tres operaciones provocan confusión y no cumplen las leyes distributiva ni asociativa.

La estructura que crea la difusión es un bloque básico denominado Estructura MA Multiplication / Addition.

Usa sólo dos claves por cada vuelta y sus entradas  $F_1$ ,  $F_2$  así como sus salidas  $G_1$ ,  $G_2$  están conectadas por XOR.



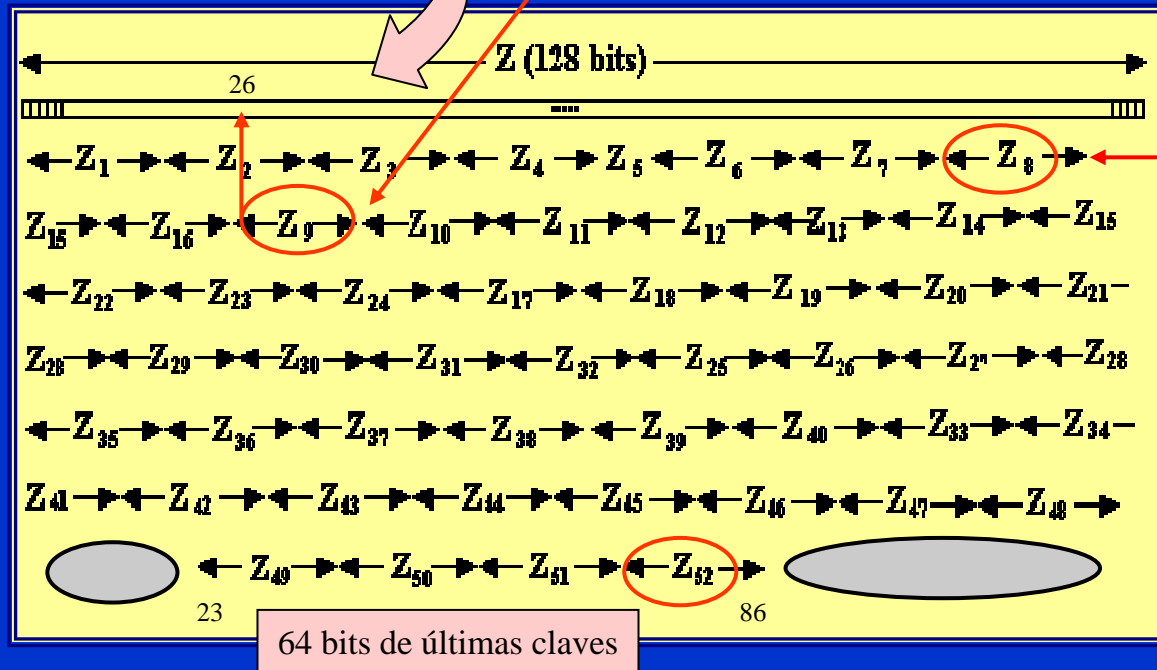
# Generación de claves en IDEA

A partir de una entrada de 128 bits, se generan las 52 subclaves de cifrado.

Se produce un desplazamiento de 25 bits a la izquierda en cada una de las 7 fases de generación de claves.

Los 64 últimos bits de la fase 7 no se usan.

Con los primeros 128 bits se generan 8 subclaves de 16 bits cada una.



# Desplazamientos de la clave en IDEA

En cada operación sobre la clave de 128 bits, se obtienen 8 claves de 16 bits de las que sólo se usan 6 en cada vuelta. Las claves restantes se guardan para la siguiente vuelta.

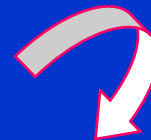
## Clave Principal k = 128 bits

001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032  
033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064  
065 066 067 068 069 070 071 072 073 074 075 076 077 078 079 080 081 082 083 084 085 086 087 088 089 090 091 092 093 094 095 096  
097 098 099 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128

|    | Primeros 16 bits de clave                                       | .... | Ultimos 16 bits de clave  |
|----|---|------|---|
| 1ª | 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 | .... | 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 |
| 2ª | 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 | .... | 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 |
| 3ª | 051 052 053 054 055 056 057 058 059 060 061 062 063 064 065 066 | .... | 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 |
| 4ª | 076 077 078 079 080 081 082 083 084 085 086 087 088 089 090 091 | .... | 060 061 062 063 064 065 066 067 068 069 070 071 072 073 074 075 |
| 5ª | 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 | .... | 085 086 087 088 089 090 091 092 093 094 095 096 097 098 099 100 |
| 6ª | 126 127 128 001 002 003 004 005 006 007 008 009 010 011 012 103 | .... | 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 |
| 7ª | 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 | .... | 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 |

# Claves usadas por IDEA en cada en vuelta

La distribución de bits de subclaves en cada vuelta sigue una lógica



|                 |   |  |
|-----------------|---|--|
| Primera vuelta: | $k_1 k_2 k_3 k_4 k_5 k_6$                   | $B[1 \dots 96]$                              |
| Segunda vuelta: | $k_7 k_8 k_9 k_{10} k_{11} k_{12}$          | $B[97 \dots 128; 26 \dots 89]$               |
| Tercera vuelta: | $k_{13} k_{14} k_{15} k_{16} k_{17} k_{18}$ | $B[90 \dots 128; 1 \dots 25; 51 \dots 82]$   |
| Cuarta vuelta:  | $k_{19} k_{20} k_{21} k_{22} k_{23} k_{24}$ | $B[83 \dots 128; 1 \dots 50]$                |
| Quinta vuelta:  | $k_{25} k_{26} k_{27} k_{28} k_{29} k_{30}$ | $B[76 \dots 128; 1 \dots 43]$                |
| Sexta vuelta:   | $k_{31} k_{32} k_{33} k_{34} k_{35} k_{36}$ | $B[44 \dots 75; 101 \dots 128; 1 \dots 36]$  |
| Séptima vuelta: | $k_{37} k_{38} k_{39} k_{40} k_{41} k_{42}$ | $B[37 \dots 100; 126 \dots 128; 1 \dots 29]$ |
| Octava vuelta:  | $k_{43} k_{44} k_{45} k_{46} k_{47} k_{48}$ | $B[30 \dots 125]$                            |
| Transformación: | $k_{49} k_{50} k_{51} k_{52}$               | $B[23 \dots 86]$                             |

# Primeras claves en cada vuelta en IDEA

Las primeras claves de cada vuelta  $k_1, k_7, k_{13}, k_{19}, k_{25}, k_{31}, k_{37}$  y  $k_{43}$  usan un conjunto diferente de bits. Excepto en las vueltas primera y octava, los 96 bits de subclave usados en cada vuelta, no son contiguos. Debido al desplazamiento en cada fase de 25 bits a la izquierda, se hace muy difícil el ataque a la clave.

|           |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| $K_1:$    | 001 | 002 | 003 | 004 | 005 | 006 | 007 | 008 | 009 | 010 | 011 | 012 | 013 | 014 | 015 | 016 |
| $K_7:$    | 097 | 098 | 099 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 |
| $K_{13}:$ | 090 | 091 | 092 | 093 | 094 | 095 | 096 | 097 | 098 | 099 | 100 | 101 | 102 | 103 | 104 | 105 |
| $K_{19}:$ | 083 | 084 | 085 | 086 | 087 | 088 | 089 | 090 | 091 | 092 | 093 | 094 | 095 | 096 | 097 | 098 |
| $K_{25}:$ | 076 | 077 | 078 | 079 | 080 | 081 | 082 | 083 | 084 | 085 | 086 | 087 | 088 | 089 | 090 | 091 |
| $K_{31}:$ | 044 | 045 | 046 | 047 | 048 | 049 | 050 | 051 | 052 | 053 | 054 | 055 | 056 | 057 | 058 | 059 |
| $k_{37}:$ | 037 | 038 | 039 | 040 | 041 | 042 | 043 | 044 | 045 | 046 | 047 | 048 | 049 | 050 | 051 | 052 |
| $k_{43}:$ | 030 | 031 | 032 | 033 | 034 | 035 | 036 | 037 | 038 | 039 | 040 | 041 | 042 | 043 | 044 | 045 |

# Ejemplo de cálculo de claves en IDEA

Si la clave es “**IDEA es la clave**” de 16 caracteres (128 bits), encuentre los 16 bits de la segunda clave de la cuarta vuelta.  
Solución:

|     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |    |    |    |    |    |    |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|
| 01  | 02  | 03  | 04  | 05  | 06  | 07  | 08  | 09  | 10  | 11  | 12  | 13  | 14  | 15  | 16  | 17  | 18  | 19  | 20  | 21  | 22  | 23  | 24  | 25 | 26 | 27 | 28 | 29 | 30 |
| 0   | 1   | 0   | 0   | 1   | 0   | 0   | 1   | 0   | 1   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 1   | 0   | 1   | 0  | 1  | 0  | 0  | 0  | 0  |
| 31  | 32  | 33  | 34  | 35  | 36  | 37  | 38  | 39  | 40  | 41  | 42  | 43  | 44  | 45  | 46  | 47  | 48  | 49  | 50  | 51  | 52  | 53  | 54  | 55 | 56 | 57 | 58 | 59 | 60 |
| 0   | 1   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 1   | 0   | 0   | 1   | 0   | 1   | 0   | 1   | 1   | 1   | 0   | 0   | 1  | 1  | 0  | 0  | 1  | 0  |
| 61  | 62  | 63  | 64  | 65  | 66  | 67  | 68  | 69  | 70  | 71  | 72  | 73  | 74  | 75  | 76  | 77  | 78  | 79  | 80  | 81  | 82  | 83  | 84  | 85 | 86 | 87 | 88 | 89 | 90 |
| 0   | 0   | 0   | 0   | 0   | 1   | 1   | 0   | 1   | 1   | 0   | 0   | 0   | 1   | 1   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 1   | 0   | 0  | 0  | 0  | 0  | 0  | 1  |
| 91  | 92  | 93  | 94  | 95  | 96  | 97  | 98  | 99  | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 |    |    |    |    |    |    |
| 1   | 0   | 0   | 0   | 1   | 1   | 0   | 1   | 1   | 0   | 1   | 1   | 0   | 0   | 0   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 1  |    |    |    |    |    |
| 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 |     |     |     |     |     |     |     |     |     |     |    |    |    |    |    |    |
| 1   | 1   | 0   | 1   | 1   | 0   | 0   | 1   | 1   | 0   | 0   | 1   | 0   | 1   |     |     |     |     |     |     |     |     |     |     |    |    |    |    |    |    |

Como en cada vuelta se usan 6 subclaves, la segunda clave de la cuarta vuelta será la número  $3*6+2 = 20$ . Como la clave 19 termina en el bit 98, la clave 20 serán los 16 bits siguientes, es decir del 99 al 114:  $k_{20} = 10110001\ 10000101$ .

# Descifrado con IDEA

El algoritmo IDEA, al igual que el DES, permite cifrar y descifrar con la misma estructura. Como las operaciones se hacen dentro de un cuerpo finito, en este caso las claves se toman como los inversos de las operaciones XOR, suma mod  $2^{16}$  y producto mod  $2^{16}+1$ , dependiendo de las operaciones realizadas en la fase de cifrado.

## INVERSOS

Inverso XOR: se aplica la misma función



Inverso aditivo: suma mod  $2^{16}$

$$Z_j \oplus -Z_j = 0$$

Inverso multiplicativo: producto mod  $2^{16}+1$

$$Z_j \odot Z_j^{-1} = 1$$

# Claves de descifrado en IDEA

|                        |                    |                    |                        |                   |                   |
|------------------------|--------------------|--------------------|------------------------|-------------------|-------------------|
| $d_1 = k_{49}^{-1}$    | $d_2 = -k_{50}$    | $d_3 = -k_{51}$    | $d_4 = k_{52}^{-1}$    | $d_5 = k_{47}$    | $d_6 = k_{48}$    |
| $d_7 = k_{43}^{-1}$    | $d_8 = -k_{45}$    | $d_9 = -k_{44}$    | $d_{10} = k_{46}^{-1}$ | $d_{11} = k_{41}$ | $d_{12} = k_{42}$ |
| $d_{13} = k_{37}^{-1}$ | $d_{14} = -k_{39}$ | $d_{15} = -k_{38}$ | $d_{16} = k_{40}^{-1}$ | $d_{17} = k_{35}$ | $d_{18} = k_{36}$ |
| $d_{19} = k_{31}^{-1}$ | $d_{20} = -k_{33}$ | $d_{21} = -k_{32}$ | $d_{22} = k_{34}^{-1}$ | $d_{23} = k_{29}$ | $d_{24} = k_{30}$ |
| $d_{25} = k_{25}^{-1}$ | $d_{26} = -k_{27}$ | $d_{27} = -k_{26}$ | $d_{28} = k_{28}^{-1}$ | $d_{29} = k_{23}$ | $d_{30} = k_{24}$ |
| $d_{31} = k_{19}^{-1}$ | $d_{32} = -k_{21}$ | $d_{33} = -k_{20}$ | $d_{34} = k_{22}^{-1}$ | $d_{35} = k_{17}$ | $d_{36} = k_{18}$ |
| $d_{37} = k_{13}^{-1}$ | $d_{38} = -k_{15}$ | $d_{39} = -k_{14}$ | $d_{40} = k_{16}^{-1}$ | $d_{41} = k_{11}$ | $d_{42} = k_{12}$ |
| $d_{43} = k_7^{-1}$    | $d_{44} = -k_9$    | $d_{45} = -k_8$    | $d_{46} = k_{10}^{-1}$ | $d_{47} = k_5$    | $d_{48} = k_6$    |
| $d_{49} = k_1^{-1}$    | $d_{50} = -k_2$    | $d_{51} = -k_3$    | $d_{52} = k_4^{-1}$    |                   |                   |

Inversos de la suma

Inversos del XOR

Inversos del producto

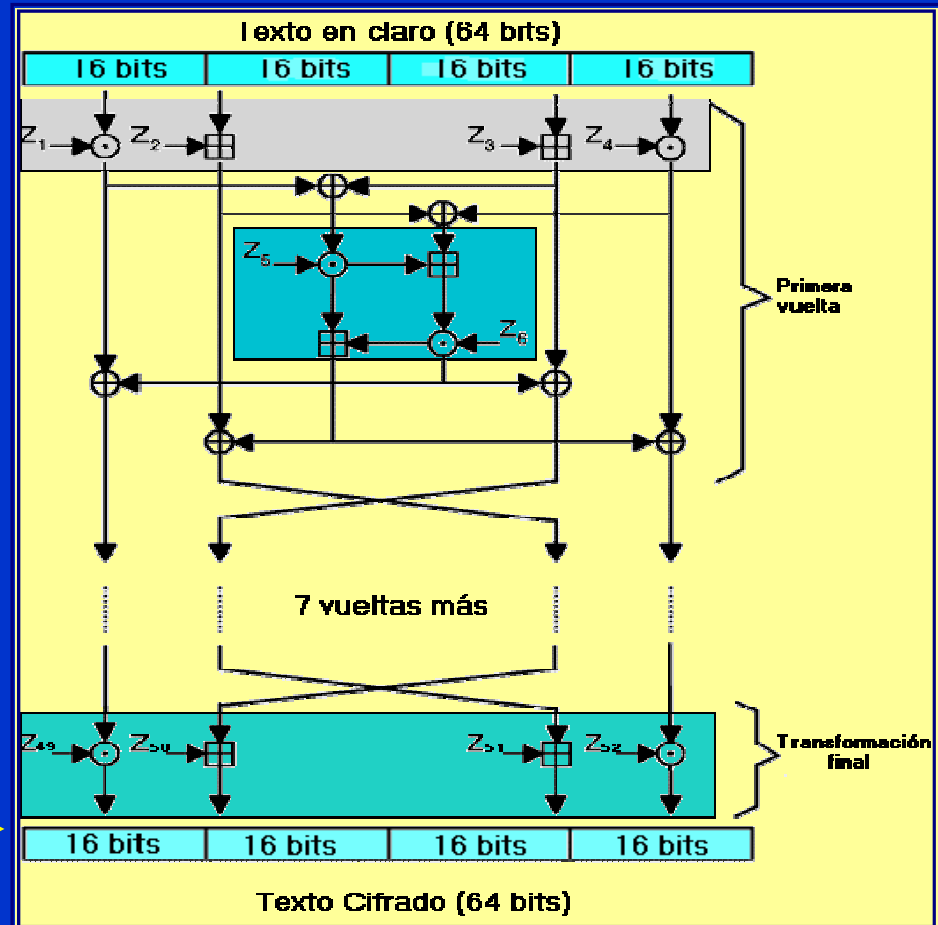


# Operación de descifrado con IDEA

## Módulo IDEA

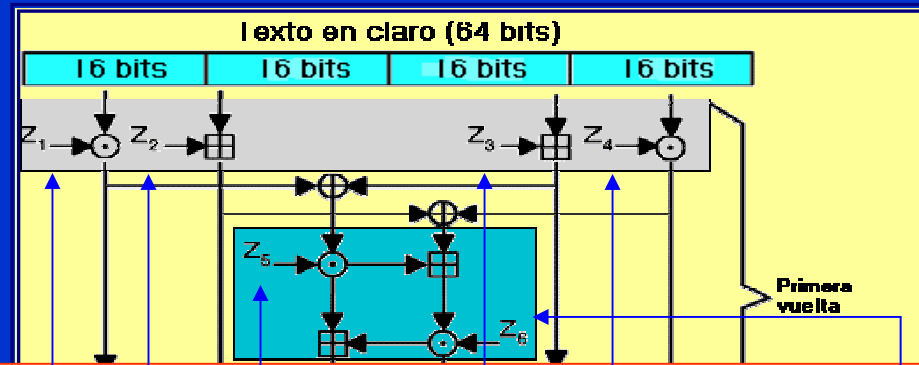
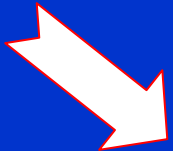
Para descifrar, cada bloque de criptograma se dividirá en cuatro subbloques de 16 bits

Las operaciones se hacen ahora hacia arriba



# Uso de claves inversas en descifrado IDEA

Ultimas 6 claves de descifrado



$$\begin{aligned} d_1 &= k_{49}^{-1} \\ d_7 &= k_{43}^{-1} \\ d_{13} &= k_{37}^{-1} \\ d_{19} &= k_{31}^{-1} \\ d_{25} &= k_{25}^{-1} \\ d_{31} &= k_{19}^{-1} \\ d_{37} &= k_{13}^{-1} \\ d_{43} &= k_7^{-1} \\ d_{49} &= k_1^{-1} \end{aligned}$$

$$\begin{aligned} d_2 &= -k_{50} \\ d_8 &= -k_{45} \\ d_{14} &= -k_{39} \\ d_{20} &= -k_{33} \\ d_{26} &= -k_{27} \\ d_{32} &= -k_{21} \\ d_{38} &= -k_{15} \\ d_{44} &= -k_9 \\ d_{50} &= -k_2 \end{aligned}$$

$$\begin{aligned} d_3 &= -k_{51} \\ d_9 &= -k_{44} \\ d_{15} &= -k_{38} \\ d_{21} &= -k_{32} \\ d_{27} &= -k_{26} \\ d_{33} &= -k_{20} \\ d_{39} &= -k_{14} \\ d_{45} &= -k_8 \\ d_{51} &= -k_3 \end{aligned}$$

$$\begin{aligned} d_4 &= k_{52}^{-1} \\ d_{10} &= k_{46}^{-1} \\ d_{16} &= k_{40}^{-1} \\ d_{22} &= k_{34}^{-1} \\ d_{28} &= k_{28}^{-1} \\ d_{34} &= k_{22}^{-1} \\ d_{40} &= k_{16}^{-1} \\ d_{46} &= k_{10}^{-1} \\ d_{52} &= k_4^{-1} \end{aligned}$$

$$\begin{aligned} d_5 &= k_{47} \\ d_{11} &= k_{41} \\ d_{17} &= k_{35} \\ d_{23} &= k_{29} \\ d_{29} &= k_{23} \\ d_{35} &= k_{17} \\ d_{41} &= k_{11} \\ d_{47} &= k_5 \end{aligned}$$

$$\begin{aligned} d_6 &= k_{48} \\ d_{12} &= k_{42} \\ d_{18} &= k_{36} \\ d_{24} &= k_{30} \\ d_{30} &= k_{24} \\ d_{36} &= k_{18} \\ d_{42} &= k_{12} \\ d_{48} &= k_6 \end{aligned}$$

# Fortaleza del algoritmo IDEA

- IDEA se muestra inmune ante un criptoanálisis diferencial. Sus autores conocían esta debilidad del DES y lo hicieron resistente.
- Joan Daemen descubre en 1992 una clase de claves débiles. La siguiente clave  $k = 0000,0000,0x00,0000,0000,000x,xxxx,x000$  en hexadecimal es débil, en el sentido de que un criptoanalista podría identificarla en un ataque con texto en claro elegido. Las posiciones  $x$  pueden ser cualquier número en hexadecimal.
- La probabilidad de que se use este tipo de claves es sólo de uno en  $2^{96}$  y se puede, además, eliminar por diseño.
- No se conoce a la fecha ningún sistema o algoritmo de ataque que haya criptoanalizado el IDEA.
- Joan Daemen y Vincent Rijmen crearán en 1997 el RIJNDAEL, nuevo estándar mundial del NIST desde finales de 2001.

<http://www.cosic.esat.kuleuven.ac.be/publications/article-140.pdf>



## Otros algoritmos: RC2

- Cifrador en bloque de clave variable propuesto por Ron Rivest.
- El código es secreto industrial de RSA Data Security Inc.
- Tamaño del bloque de texto: 64 bits.
- Con una clave con tamaño variable (de 8 a 1.024 bits) forma una tabla de 128 bytes (1.024 bits) que depende de la clave inicial.
- No usa cajas S y es casi tres veces más rápido que DES.
- Se usa en SMIME con longitudes de clave de 40, 64 y 128 bits.
- Los algoritmos RC2 y RC4 (este último cifrador de flujo) se incluyen en productos para la exportación, como navegadores.
- Operaciones primitivas de cifra: suma en módulo  $2^{32}$ , operación or exclusivo, complemento de bits, operación AND y rotación circular a la izquierda.
- Realiza 18 vueltas conocidas como mixing y mashing.

## Otros algoritmos: RC5

- RC5 es un cifrador en bloque de tamaño variable de Ron Rivest.
- Cifra bloques de texto de 32, 64 ó 128 bits.
- Tamaño de clave hasta 2.048 bits, en función número de vueltas.
- Número de vueltas de 0 a 255.
- Versiones específicas: RC5 –w/r/b donde w es el tamaño de la palabra (16, 32 ó 64 bits) -RC5 cifra bloques de dos palabras-, r es el número de vueltas y b es el tamaño en octetos de la clave K. El valor propuesto por Rivest como mínimo es RC5 –32/12/16.
- Rutina expansión de clave: se expande K para llenar una tabla.
- Rutinas de cifrado y descifrado: usa primitivas de suma módulo  $2^w$ , or exclusivo y rotación circular a la izquierda.
- Características: muy rápido, arquitectura simple, bajos requisitos de memoria y alta seguridad. Las rotaciones dependientes de los datos le fortalecen ante el criptoanálisis diferencial.

## Otros algoritmos: SAFER 64 y 128

- SAFER: Secure and Fast Encryption Routine (James Massey).
- Cifra bloques de texto de 64 bits. Cada bloque de texto a cifrar se divide en 8 bytes.
- Tamaño de clave: 64 ó 128 bits.
- Número de vueltas de 0 a 10; mínimo recomendable 6.
- Operaciones de cifrado y descifrado distintas basadas en bytes, que orientan su uso en aplicaciones de tarjetas inteligentes.
- En cada vuelta hay operaciones or y sumas normales, potencias y logaritmos discretos en  $p = 257$ , usando 45 como raíz primitiva.
- Al final del algoritmo hay tres niveles de operaciones lineales conocidas como Pseudo Transformaciones de Hadamard, PTH, cuyo objetivo es aumentar la difusión de los bits.
- Existen versiones SAFER SK-64 y SK-128 más seguras ante claves débiles que sus antecesoras.

## Otros algoritmos: Blowfish

- Cifrador tipo Feistel de clave variable (Bruce Schneier).
- Cifra bloques de texto de 64 bits.
- Tamaño de clave: de 32 hasta 448 bits. Se generan 18 subclaves de 32 bits y cuatro cajas S de 8x32 bits, en total 4.168 bytes.
- Número de vueltas: 16, en cada una de ellas se realiza una permutación función de la clave y una sustitución que es función de la clave y los datos.
- Operaciones básicas: or exclusivo y suma módulo  $2^{32}$ .
- Cajas S: en cada vuelta hay cuatro con 256 entradas cada una.
- Características: compacto porque necesita sólo 5 K de memoria, es muy rápido (5 veces más veloz que DES), es conceptualmente simple y su fortaleza puede variarse según longitud de la clave. Usa una función F con las cuatro cajas S y operaciones básicas de suma y or exclusivo que provocan un efecto de avalancha.

## Otros algoritmos: CAST 128

- Cifrador Feistel propuesto por C. Adams y S. Tavares (Canadá).
- Cifra bloques de texto de 64 bits con claves de 40 hasta 128 bits en incrementos de octetos.
- Cifra en 16 vueltas.
- Usa ocho cajas S de 8 bits de entrada y 32 bits de salida con unas funciones no lineales óptimas (funciones bent), cuatro cajas en procesos de cifra y las otras cuatro para la generación de claves. Cada caja es un array de 32 columnas y 256 filas. Los 8 bits de entrada seleccionan una fila y los 32 bits de ésta es la salida.
- Operaciones básicas: suma y resta módulo  $2^{32}$ , or exclusivo y rotaciones circulares hacia la izquierda.
- Características: inmune a ataques por criptoanálisis diferencial y lineal; algoritmo estándar de cifra en últimas versiones de PGP.



## Otros algoritmos: Skipjack

- Ha sido desarrollado por la NSA, National Security Agency, está contenido en los chip Clipper y Capstone y su implementación sólo está permitida en hardware.
- Cifra bloques de 64 bits con una clave de 80 bits.
- Los usuarios depositan sus claves secretas en diversas agencias de gobierno.
- Usa 32 vueltas en cada bloque de cifra.
- Los detalles del algoritmo no son públicos.
- Características: imposición de los EEUU para comunicaciones con la administración, tiene una puerta trasera que puede dejar en claro la cifra, nadie puede asegurar que el algoritmo tenga la suficiente fortaleza pero los Estados Unidos piensa usarlo en su DMS, Defense Messaging System. Ha sido duramente criticado.

## El DES deja de ser un estándar

- ⌚ El DES se adopta como estándar en 1976.
- ⌚ El NIST certifica al DES en 1987 y luego en 1993.
- ⌚ Durante esos años se estandariza como algoritmo de cifra en todo el mundo. Su uso principal lo encontramos en el cifrado de la información intercambiada en transacciones de dinero entre un cajero automático y el banco respectivo.
- ⌚ En 1997 NIST no certifica al DES y llama a un concurso internacional para buscar un nuevo estándar mundial de cifra denominado *AES Advanced Encryption Standard*.
- ⌚ Precisamente entre 1997 y 1999 el DES se enfrenta a tres ataques o desafíos conocidos como DES Challenge que impulsa y promueve la compañía RSA.

<http://www.nist.gov/>



## 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 😊.
- 💣 13 enero 1998: DES Challenge II-1. Se rompe la clave en **39 días** con un ataque tipo distribuido por distributed.net que llega a evaluar 34.000 millones de claves por segundo y debe recorrer el 88% del espacio de claves 😞.
- 💣 13 julio de 1998: DES Challenge II-2. Electronic Frontier Foundation EFF 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.

## DES Challenge III



18 enero 1999: DES Challenge III. Se unen 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 tras recorrer el 22% del espacio de claves. Se trata del último desafío propuesto por RSA que pone en evidencia la capacidad de ataques distribuidos a través de los tiempos muertos de procesador de máquinas conectadas a Internet que, con un programa cliente, van resolviendo un pequeño trozo del espacio de claves, comunicándose para ello con un servidor. Recuerde, el DES no ha sido criptoanalizado, se ha roto la cifra sólo por el pequeño tamaño de su clave.

<http://www.rsasecurity.com/rsalabs/node.asp?id=2108>



# Magnitudes de tiempo y criptoanálisis

| Longitud de la clave | Tiempo necesario para romper la clave        |
|----------------------|--|
| 40 bits              | 2 segundos                                   |
| 48 bits              | 9 minutos                                    |
| 56 bits              | 40 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 |

La tabla muestra el tiempo medio de criptoanálisis necesario para romper una clave de cifra simétrica mediante fuerza bruta, en este caso usando la potencia de cálculo alcanzada en el DES Challenge III en 1999, unos 250.000 millones de claves por segundo con la máquina DES Cracker y unos 100.000 computadores a través de Internet. Según la ley de Moore, en el 2006 esta potencia de cálculo se multiplicaría por 20 o más.

La seguridad de 128 bits de una cifra simétrica es equivalente a la de 1.024 bits de cifra asimétrica.

# El nuevo estándar en cifra AES

## AES: Advanced Encryption Standard

- El DES, estándar desde 1976, pasa la certificación de la NBS National Bureau of Standards en 1987 y en 1993.
- En 1997 el NIST National Institute of Standards and Technology (antigua NBS) no certifica al DES y llama a concurso público para un nuevo algoritmo estándar, el AES.

<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>



<http://www.iaik.tu-graz.ac.at/research/krypto/AES/>



- En octubre del año 2000 el NIST elige el algoritmo belga **Rijndael** como nuevo estándar para cifrado del siglo XXI.

[http://www.criptored.upm.es/guiateoria/gt\\_m480a.htm](http://www.criptored.upm.es/guiateoria/gt_m480a.htm)



# Características del algoritmo AES

**Rijndael**: autores Vincent **Rijmen** & Joan **Daemen**

- No es de 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: 128, 192 y 256 bits (estándar) o bien múltiplo de 4 bytes.
- Tamaño del bloque de texto: 128 bits o múltiplo de 4 bytes.
- Operaciones modulares a nivel de byte (representación en forma de polinomios) y de palabra de 4 bytes: 32 bits.
- Número de etapas flexible según necesidades del usuario.
- Usa un conjunto de Cajas S similares a las del DES.

<http://www.iaik.tu-graz.ac.at/research/krypto/AES/old/%7Erijmen/rijndael/>



# Operaciones con bytes en AES

Unidad básica de tratamiento: el byte

- **Suma y multiplicación.** Son cálculos en Campos de Galois  $GF(2^8)$  con 8 bits. Para la reducción de exponente se usará un polinomio primitivo  $p(x) = x^8 + x^4 + x^3 + x + 1$ .
- **Producto por x.** Esta operación conocida como  $xtime(a)$  al igual que en el caso anterior usa la reducción de exponente. Puede implementarse fácilmente con desplazamientos y operaciones or exclusivo.

*Ejemplos*



## Ejemplo de suma en $GF(2^8)$

Vamos a sumar los valores hexadecimales 57 y 83:

$$A = 57_{16} = 0101\ 0111_2 \quad B = 83_{16} = 1000\ 0011_2$$

que expresados en polinomios dentro de  $GF(2^8)$  serán:

$$A = 0101\ 0111_2 = x^6 + x^4 + x^2 + x + 1$$

$$B = 1000\ 0011_2 = x^7 + x + 1$$

Sumando:  $A+B = (x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) \bmod 2$

$$A+B = (x^7 + x^6 + x^4 + x^2 + 2x + 2) \bmod 2$$

$$A+B = x^7 + x^6 + x^4 + x^2 = 1101\ 0100_2 = d4_{16}$$

Y lo mismo se obtiene con la suma Or exclusivo:

$$0101\ 0111 \oplus 1000\ 0011 = 1101\ 0100_2 = d4_{16}$$

## Ejemplo de producto en $GF(2^8)$ (1)

Vamos a multiplicar los valores hexadecimales 57 y 83:

$$A = 57_{16} = 0101\ 0111_2 \quad B = 83_{16} = 1000\ 0011_2$$

que expresados en polinomios dentro de  $GF(2^8)$  serán:

$$A = 0101\ 0111_2 = x^6 + x^4 + x^2 + x + 1$$

$$B = 1000\ 0011_2 = x^7 + x + 1$$

$$A*B = (x^6 + x^4 + x^2 + x + 1)*(x^7 + x + 1) \bmod 2$$

$$A*B = x^{13} + x^{11} + x^9 + x^8 + 2x^7 + x^6 + x^5 + x^4 + x^3 + 2x^2 + 2x + 1$$

Reduciendo mod 2

$$A*B = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$$

Este resultado hay que reducirlo por  $p(x) = x^8 + x^4 + x^3 + x + 1$

## Ejemplo de producto en $GF(2^8)$ (2)

Como el polinomio irreducible es  $p(x) = x^8 + x^4 + x^3 + x + 1$

$(x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1) \bmod (x^8 + x^4 + x^3 + x + 1)$

Para dejar los valores dentro de  $GF(2^8)$  vemos que un divisor será  $x^5 + x^3$  puesto que  $x^5 x^8 = x^{13}$ ;  $x^5 x^4 = x^9$ ;  $x^5 x^3 = x^8$  y  $x^3 x^8 = x^{11}$ .

$$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \quad \left| \begin{array}{l} x^8 + x^4 + x^3 + x + 1 \\ x^5 + x^3 \end{array} \right.$$

$$\begin{array}{r} x^{13} \phantom{+ x^{11}} + x^9 + x^8 + x^6 + x^5 \\ \hline x^{11} \phantom{+ x^9} \phantom{+ x^8} + x^4 + x^3 + 1 \\ \hline x^{11} + x^7 \phantom{+ x^9} + x^6 \phantom{+ x^8} + x^4 + x^3 \\ \hline x^7 \phantom{+ x^{11}} + x^6 \phantom{+ x^9} \phantom{+ x^8} + 1 \end{array}$$

*multiplicando  $p(x)$  por  $x^5$*

*reduciendo mod 2*

*multiplicando  $p(x)$  por  $x^3$*

*reduciendo mod 2*

Resultado:  $A * B = x^7 + x^6 + 1 = 1100\ 0001_2 = c1_{16}$

# El mismo resultado pero de otra forma

Están fuera del cuerpo de 8 bits

$$x^8 = x^4 + x^3 + x + 1$$

$$A*B = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$$

$$x^{13} = x^5 * x^8 = x^5 * (x^4 + x^3 + x + 1) = x^9 + x^8 + x^6 + x^5$$

$$x^{13} = x * (x^4 + x^3 + x + 1) + (x^4 + x^3 + x + 1) + x^6 + x^5$$

$$x^{13} = (x^5 + x^4 + x^2 + x) + (x^4 + x^3 + x + 1) + x^6 + x^5$$

$$x^{13} = x^6 + x^3 + x^2 + 1$$

Es mucho más complejo, pero repitiendo el mismo desarrollo para  $x^{11}$ ,  $x^9$  y  $x^8$ , reduciendo en cada caso mod 2, obtenemos:

$$A*B = x^7 + x^6 + 1 = 1100\ 0001 = c1_{16}$$

# Transformaciones o capas del AES

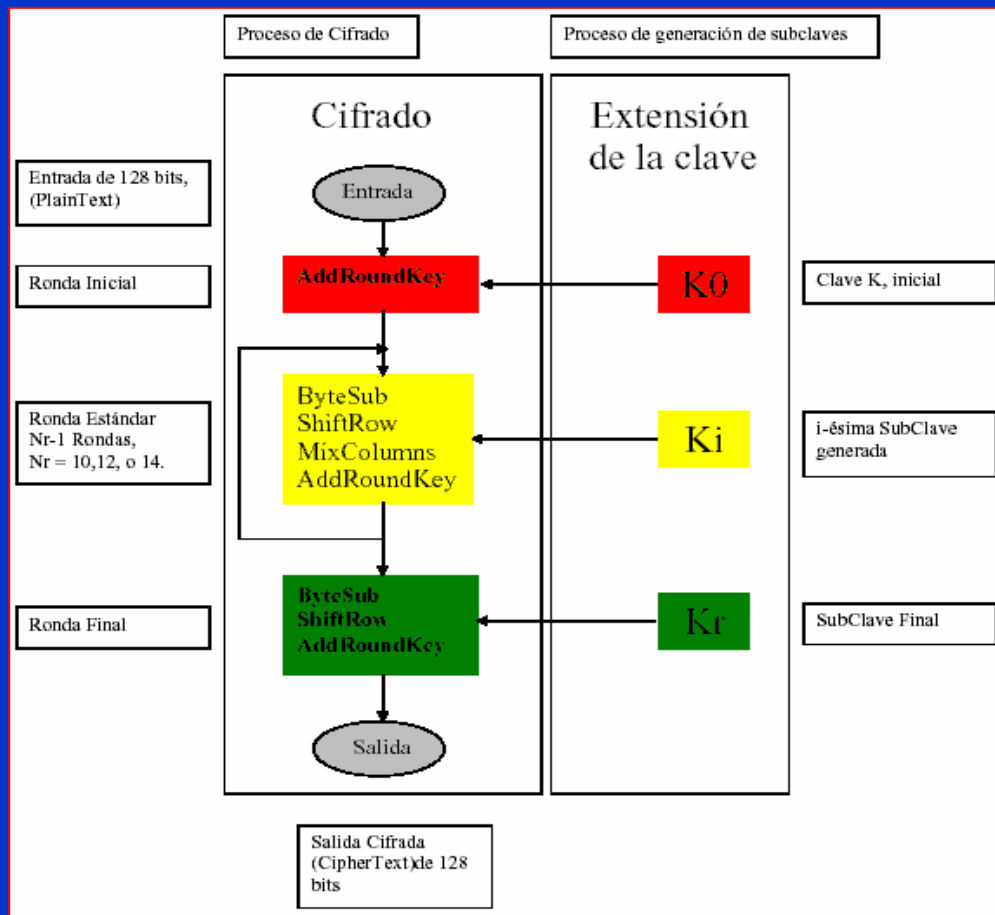
- Hay tres transformaciones distintas llamadas capas en las que se tratan los bits. Estas constan de:
  - Capa de Mezcla Lineal: en ella se busca la difusión de los bits.
  - Capa No Lineal: se trata de una zona similar a las cajas S del DES.
  - Capa Clave: operaciones con una función or exclusivo de la subclave y la información de esta etapa intermedia.
- Las transformaciones realizadas en cada paso del algoritmo se denominan estados. Estos estados se representa 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.

En la siguiente página web encontrará una extensa explicación de las operaciones en el algoritmo Rijndael con interesantes ilustraciones.

<http://www.quadibloc.com/crypto/co040401.htm>



# Esquema general del AES



Funciones en cifrado:

- *AddRoundKey*
- *ByteSub*
- *ShiftRow*
- *MixColumns*

Funciones en descifrado:

- *InvAddRoundKey*
- *InvByteSub*
- *InvShiftRow*
- *InvMixColumns*

Se realizará además una expansión de la clave  $K$  para generar desde  $K_0$  hasta  $K_r$ .

Figura y tablas tomadas de:

[http://www.criptored.upm.es/guiateoria/gt\\_m117i.htm](http://www.criptored.upm.es/guiateoria/gt_m117i.htm)



# Estados de entrada y claves del AES

|       |       |          |          |
|-------|-------|----------|----------|
| $a_0$ | $a_4$ | $a_8$    | $a_{12}$ |
| $a_1$ | $a_5$ | $a_9$    | $a_{13}$ |
| $a_2$ | $a_6$ | $a_{10}$ | $a_{14}$ |
| $a_3$ | $a_7$ | $a_{11}$ | $a_{15}$ |

Bloque de texto 16 bytes (128 bits)  
 $Nb = 128/32 = 4$

Estados

|       |       |          |          |
|-------|-------|----------|----------|
| $k_0$ | $k_4$ | $k_8$    | $k_{12}$ |
| $k_1$ | $k_5$ | $k_9$    | $k_{13}$ |
| $k_2$ | $k_6$ | $k_{10}$ | $k_{14}$ |
| $k_3$ | $k_7$ | $k_{11}$ | $k_{15}$ |

Clave de 16 bytes (128 bits)  
 $Nk = 128/32 = 4$  (10 rondas)

|       |       |          |          |          |          |
|-------|-------|----------|----------|----------|----------|
| $k_0$ | $k_4$ | $k_8$    | $k_{12}$ | $k_{16}$ | $k_{20}$ |
| $k_1$ | $k_5$ | $k_9$    | $k_{13}$ | $k_{17}$ | $k_{21}$ |
| $k_2$ | $k_6$ | $k_{10}$ | $k_{14}$ | $k_{18}$ | $k_{22}$ |
| $k_3$ | $k_7$ | $k_{11}$ | $k_{15}$ | $k_{19}$ | $k_{23}$ |

Clave de 24 bytes (192 bits)  
 $Nk = 192/32 = 6$  (12 rondas)

|       |       |          |          |          |          |          |          |
|-------|-------|----------|----------|----------|----------|----------|----------|
| $k_0$ | $k_4$ | $k_8$    | $k_{12}$ | $k_{16}$ | $k_{20}$ | $k_{24}$ | $k_{28}$ |
| $k_1$ | $k_5$ | $k_9$    | $k_{13}$ | $k_{17}$ | $k_{21}$ | $k_{25}$ | $k_{29}$ |
| $k_2$ | $k_6$ | $k_{10}$ | $k_{14}$ | $k_{18}$ | $k_{22}$ | $k_{26}$ | $k_{30}$ |
| $k_3$ | $k_7$ | $k_{11}$ | $k_{15}$ | $k_{19}$ | $k_{23}$ | $k_{27}$ | $k_{31}$ |

Clave de 32 bytes (256 bits)  
 $Nk = 256/32 = 8$  (16 rondas)

# Combinaciones de estados

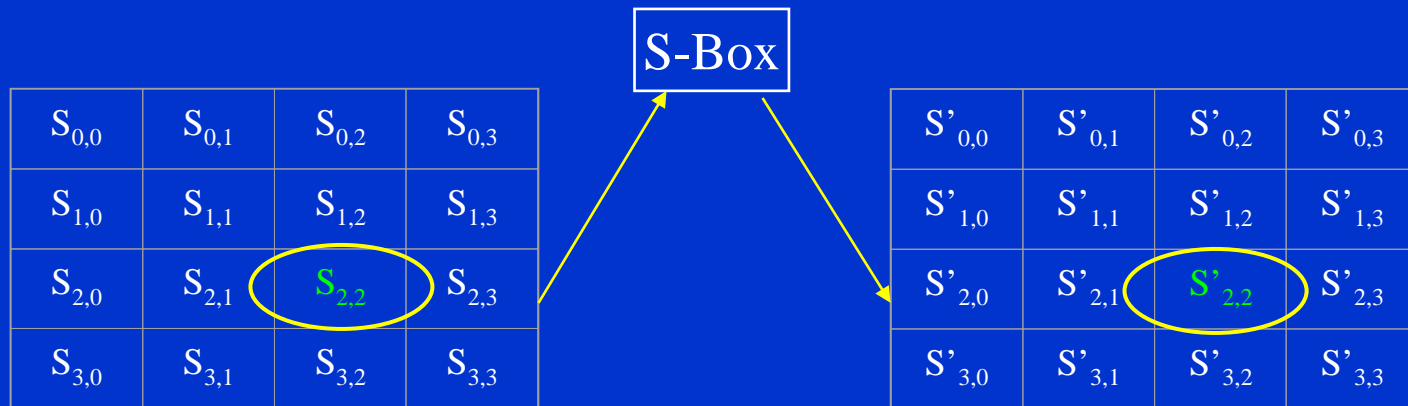
| Combinaciones posibles de estados en AES | Longitud del bloque<br>(Nb palabras) | Longitud de la clave<br>(Nk palabras) | Número de Rondas<br>(Nr) |
|--|--------------------------------------|---------------------------------------|--------------------------|
| AES – 128                           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

Para las funciones de cifrado y descifrado se usarán 4 transformaciones orientadas a bytes:

1. Sustitución de un byte mediante una tabla S-box.
2. Desplazamiento de columnas de un estado.
3. Mezcla de datos dentro de cada columna de estado.
4. Añade una clave de vuelta al estado.



# Función ByteSub



Se trata de una función no lineal que se realiza a través de una S-box.  
La S-box se construye:

a) calculando el inverso en  $GF(2^8)$ , y

b) 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 y  $c_i$  es el  $i$ ésimo bit del byte  $c$  cuyo valor es  $\{63\}_{16}$  o  $\{011000011\}_2$ .



# Tabla de inversos en mod $2^8$

|   | 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 |

Lógicamente se cumple que:

Si:

$$\text{inv } x = y$$

Entonces:

$$\text{inv } y = x$$

Por ejemplo:

$$\text{inv } c4 = da$$

$$\text{inv } da = c4$$

# Representación de la función ByteSub

La transformación afín anterior queda:

$$\begin{aligned} b'_0 &= b_0 \oplus b_4 \oplus b_5 \oplus b_6 \oplus b_7 \oplus c_0 & b'_4 &= b_4 \oplus b_0 \oplus b_1 \oplus b_2 \oplus b_3 \oplus c_4 \\ b'_1 &= b_1 \oplus b_5 \oplus b_6 \oplus b_7 \oplus b_0 \oplus c_1 & b'_5 &= b_1 \oplus b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus c_5 \\ b'_2 &= b_2 \oplus b_5 \oplus b_7 \oplus b_0 \oplus b_1 \oplus c_2 & b'_6 &= b_6 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5 \oplus c_6 \\ b'_3 &= b_3 \oplus b_7 \oplus b_0 \oplus b_1 \oplus b_2 \oplus c_3 & b'_7 &= b_7 \oplus b_3 \oplus b_4 \oplus b_5 \oplus b_6 \oplus c_7 \end{aligned}$$

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

**Representación matricial**

Valor  $\{63\}_{16}$  o  $\{011000011\}_2$

Es el inverso del valor de entrada

# Tabla ByteSub

Usando la siguiente tabla, se llega a igual resultado que calculando el inverso y luego aplicando la transformación matricial mostrada en la diapositiva anterior.

➞ En la siguiente diapositiva hay un ejemplo para el valor **5a** mostrado.

|   | 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 | <b>be</b> | 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 |

# Ejemplo de operación ByteSub

Se pide calcular el ByteSub de **5a**

$$\mathbf{5a} = 01011010 = x^6 + x^4 + x^3 + x + 1$$

$$\text{inv}(5A) = 22 = \mathbf{00100010} \text{ (según la tabla dada)}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{1} \end{pmatrix}$$

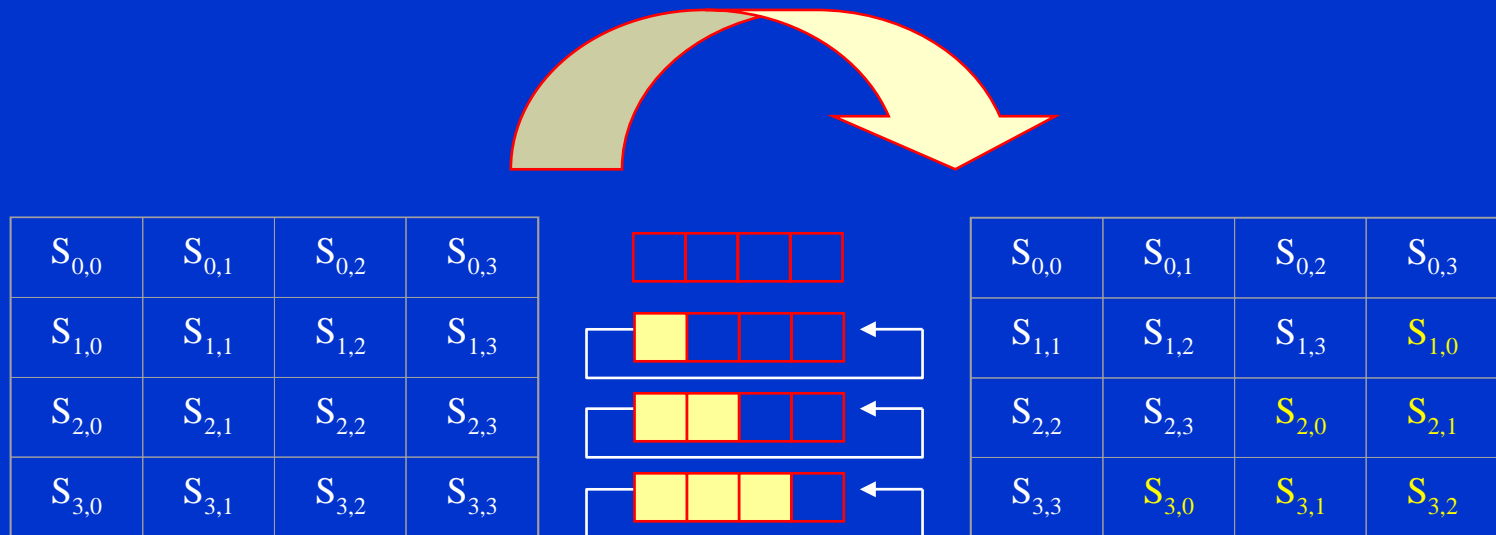
Al mismo valor se llega si en la tabla buscamos la intersección entre la fila **5** y la columna **a**: el resultado es el valor **be**.

Operando filas por columnas y sumando al resultado el valor  $\{011000011\}_2$  se obtiene: **1011 1110 = be**.

# Función ShiftRow

La función consiste en desplazar bloques de un byte hacia la izquierda módulo columna (en este caso 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 como se muestra.



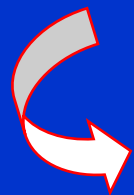
# Función MixColumns

Opera sobre columnas que son consideradas como un polinomio sobre  $GF(2^8)$  multiplicando las columnas módulo  $x^4 + 1$  por este polinomio fijo, en donde los valores  $\{ \}$  están en hexadecimal, que es primo relativo con  $x^4 + 1$  y por tanto asegura el inverso.

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

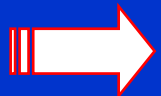
Por tanto, recuerde que  $\{03\} = x + 1$ ,  $\{02\} = x$ ,  $\{01\} = 1$ .

Representación  
matricial de la  
función  
MixColumns



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

Luego, las operaciones sobre columnas se expresan como:



# Ejemplo de operación MixColumns

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

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 |

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

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

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

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

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

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

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

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

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

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

$$\text{Luego: } S'_{0,0} = 25$$

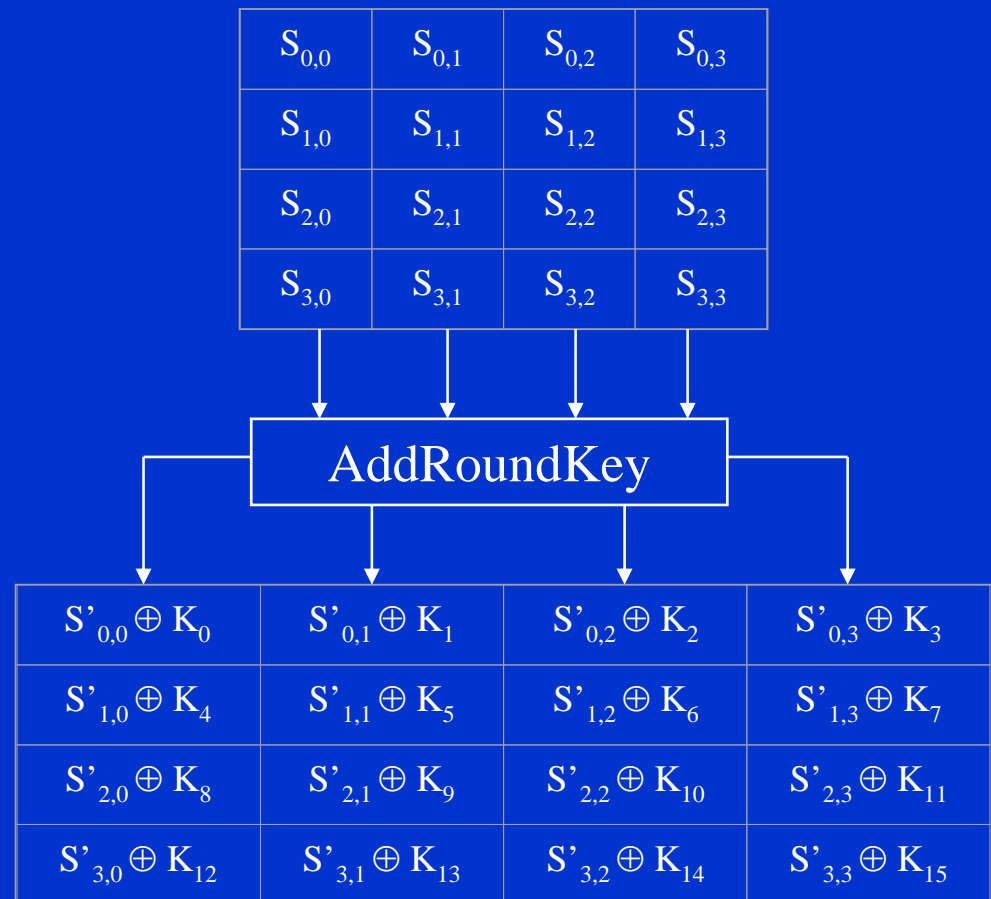
Los bytes hasta  $S'_{4,4}$  se calculan de forma similar.



# Función AddRoundKey

Se sumarán or exclusivo el estado intermedio con la clave de cada ronda.

En la ronda 0 será el or exclusivo entre el texto de entrada y la clave inicial; en las rondas siguientes (p.e. 1 a 9) será el or exclusivo de las subclave de cada ronda con la salida de la función MixColumns y en la última ronda (10) el or exclusivo de la subclave de estado 10 y la salida de ShiftRows.



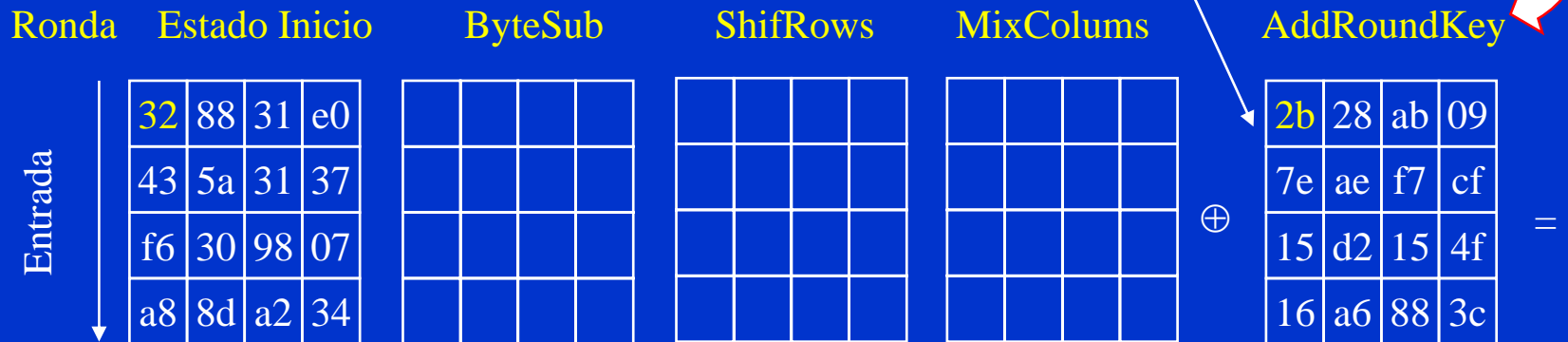
# AddRoundKey en la vuelta 0

Si el bloque de entrada y la clave son de 128 bits, (Nb=4 y Nk =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



|    |    |    |    |
|----|----|----|----|
| 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

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

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

# AddRoundKey en la vuelta 10

| Ronda  | Estado Inicio  | ByteSub | ShifRows | MixColumns | AddRoundKey |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|--------|--|---------|----------|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| 9      | ...  |         |          |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 10     | <table> <tr><td>eb</td><td>59</td><td>8b</td><td>1b</td></tr> <tr><td>40</td><td>2e</td><td>a1</td><td>c3</td></tr> <tr><td>f2</td><td>38</td><td>13</td><td>42</td></tr> <tr><td>1e</td><td>84</td><td>e7</td><td>d2</td></tr> </table> | eb      | 59       | 8b         | 1b          | 40 | 2e | a1 | c3 | f2 | 38 | 13 | 42 | 1e | 84 | e7 | d2 | <table> <tr><td>e9</td><td>cb</td><td>3d</td><td>af</td></tr> <tr><td>09</td><td>31</td><td>32</td><td>2e</td></tr> <tr><td>89</td><td>07</td><td>7d</td><td>2c</td></tr> <tr><td>72</td><td>5f</td><td>94</td><td>b5</td></tr> </table> | e9 | cb | 3d | af | 09 | 31 | 32 | 2e | 89 | 07 | 7d | 2c | 72 | 5f | 94 | b5 | <table> <tr><td>e9</td><td>cb</td><td>3d</td><td>af</td></tr> <tr><td>31</td><td>32</td><td>2e</td><td>09</td></tr> <tr><td>7d</td><td>2c</td><td>89</td><td>07</td></tr> <tr><td>b5</td><td>72</td><td>5f</td><td>94</td></tr> </table> | e9 | cb | 3d | af | 31 | 32 | 2e | 09 | 7d | 2c | 89 | 07 | b5 | 72 | 5f | 94 | <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> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | $\oplus$ <table> <tr><td>d0</td><td>c9</td><td>e1</td><td>b6</td></tr> <tr><td>14</td><td>ee</td><td>3f</td><td>63</td></tr> <tr><td>f9</td><td>25</td><td>0c</td><td>0c</td></tr> <tr><td>a8</td><td>89</td><td>c8</td><td>a6</td></tr> </table> | d0 | c9 | e1 | b6 | 14 | ee | 3f | 63 | f9 | 25 | 0c | 0c | a8 | 89 | c8 | a6 | = |
| eb     | 59   | 8b      | 1b       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 40     | 2e   | a1      | c3       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| f2     | 38   | 13      | 42       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 1e     | 84   | e7      | d2       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| e9     | cb   | 3d      | af       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 09     | 31   | 32      | 2e       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 89     | 07   | 7d      | 2c       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 72     | 5f   | 94      | b5       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| e9     | cb   | 3d      | af       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 31     | 32   | 2e      | 09       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 7d     | 2c   | 89      | 07       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| b5     | 72   | 5f      | 94       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|        |  |         |          |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|        |  |         |          |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|        |  |         |          |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|        |  |         |          |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| d0     | c9   | e1      | b6       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 14     | ee   | 3f      | 63       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| f9     | 25   | 0c      | 0c       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| a8     | 89   | c8      | a6       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| Salida | <table> <tr><td>39</td><td>02</td><td>dc</td><td>19</td></tr> <tr><td>25</td><td>dc</td><td>11</td><td>6a</td></tr> <tr><td>84</td><td>09</td><td>85</td><td>0b</td></tr> <tr><td>1d</td><td>fb</td><td>97</td><td>32</td></tr> </table> | 39      | 02       | dc         | 19          | 25 | dc | 11 | 6a | 84 | 09 | 85 | 0b | 1d | fb | 97 | 32 |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 39     | 02   | dc      | 19       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 25     | dc   | 11      | 6a       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 84     | 09   | 85      | 0b       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
| 1d     | fb   | 97      | 32       |            |             |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |

➤ Como se observa, en esta décima y última vuelta (para Nb = 4 y Nk = 4) sólo se aplican las funciones ByteSub, ShifRows y AddRoundKey.

➤ Ejemplo tomado del documento oficial del NIST:

- Como se observa, en esta décima y última vuelta (para  $N_b = 4$  y  $N_k = 4$ ) sólo se aplican las funciones ByteSub, ShiftRows y AddRoundKey.
- Ejemplo tomado del documento oficial del NIST:

<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>



# Expansión de la clave en AES

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

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

- ✓ La expansión generará 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>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> | ... |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----------------|-----------------|-----------------|-----------------|-----|

# Expansión de la clave en función de $N_k$

- ✓ Las primeras  $N_k$  palabras se copiarán de la clave principal.
- ✓ Las restantes  $N_b \cdot (N_r + 1) - N_k$  palabras se generarán mediante un algoritmo que será diferente si  $N_k \leq 6$  o bien  $N_k > 6$ .

## ➤ Si $N_k \leq 6$

- Si la posición  $i$  dentro del array  $W(i)$  es múltiplo del valor  $N_k$ :

☞  $W(i) = W(i - N_k) \text{ xor } [\text{ByteSub}(\text{RotWord}[W(i - 1)]) \text{ xor } \text{Rcon}(i/N_k)]$

- 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 $N_k > 6$

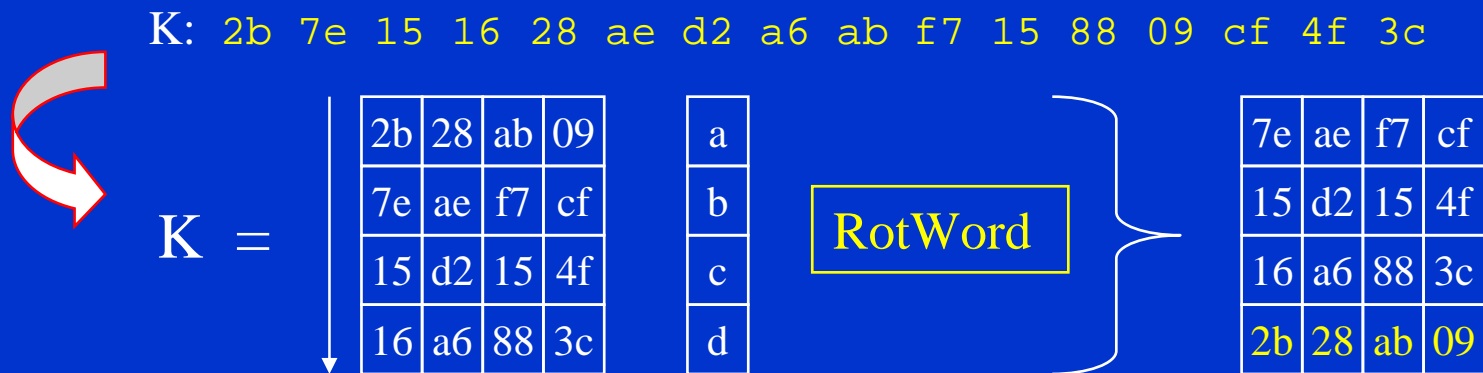
- El valor de la variable  $i$  debe satisfacer la expresión  $i \bmod N_k = 4$ .

- Las palabras de subclaves se calcularán:

☞  $W(i) = W(i - N_k) \text{ xor } \text{ByteSub}[W(i - 1)]$

# Funciones RotWord y Rcon

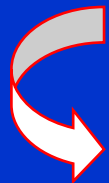
- ❖ **RotWord** rota una posición a la izquierda los bytes de la palabra. Si la palabra de 4 bytes es (a, b, c, d) Rotword devolverá (b, c, d, a).



- ❖ **Rcon** genera la constante  $Rcon(j) = [R(j), \{00\}, \{00\}, \{00\}]$  de 32 bits y donde  $R(j)$  es el elemento  $GF(2^8)$  correspondiente al valor  $x^{j-1}$ . Su cálculo se verá en el siguiente ejemplo.

# Expansión $W(4)$ para una clave de 128 bits

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 |

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



- $W(0) = 2b\ 7e\ 15\ 16$
- $W(1) = 28\ ae\ d2\ a6$
- $W(2) = ab\ f7\ 15\ 88$
- $W(3) = 09\ cf\ 4f\ 3c$

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

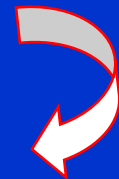
- $\text{temp} = W(3) = 09\ cf\ 4f\ 3c$
- $\text{RotWord}(\text{temp}) = cf\ 4c\ 3c\ 09 \rightarrow \text{temp}$
- $\text{ByteSub}(\text{temp}) = 8a\ 84\ eb\ 01 \rightarrow \text{temp}$
- $\text{Rcon}(4/4) = \text{Rcon}(1)$ ;  $j = 1 \Rightarrow x^{j-1} = x^0 = 01$
- $\text{Rcon}(1) = [01, 00, 00, 00]$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



2b 7e 15 16  
8b 84 eb 01

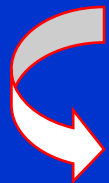
a0 fa fe 17

|    |    |    |    |
|----|----|----|----|
| a0 | 88 | 23 | 2a |
| fa | 54 | a3 | 6c |
| fe | 2c | 39 | 76 |
| 17 | b1 | 39 | 05 |



# Expansión $W(5)$ para una clave de 128 bits

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 |

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



- $W(0) = 2b\ 7e\ 15\ 16$
- $W(1) = 28\ ae\ d2\ a6$
- $W(2) = ab\ f7\ 15\ 88$
- $W(3) = 09\ cf\ 4f\ 3c$

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

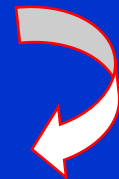
- $W(i) = W(i - N_k) \text{ xor } W(i - 1)$
- $W(5) = W(5 - 4) \text{ xor } W(5 - 1)$
- $W(5) = W(1) \text{ xor } W(4)$
- Como  $W(4) = a0\ fa\ fe\ 17$  (valor ya calculado en el paso anterior)



$\oplus$   
28 ae d2 a6  
a0 fa fe 17

88 54 2c b1

|    |    |    |    |
|----|----|----|----|
| a0 | 88 | 23 | 2a |
| fa | 54 | a3 | 6c |
| fe | 2c | 39 | 76 |
| 17 | b1 | 39 | 05 |





# Expansión genérica para clave de 128 bits

- Si  $K$  es de 128 bits,  $N_k = 4$ . Supondremos un bloque de texto de 128 bits ( $N_b = 4$ ).
- La longitud del array  $W$  será  $(4*[10+1]) = 44$  palabras de 4 bytes.
- En las cuatro primeras posiciones (0 a 3) se copia la clave principal  $K$ .
- Las restantes 40 palabras de las posiciones 4 a 43 ( $4 \leq i \leq 43$ ) se calcularán según las expresiones de la diapositiva anterior.
- Para  $W(i)$  con valores de  $i$  múltiplo de  $N_k$  (4, 8, 12, 16, 20, 24, 28, 32, 36, 40), la palabra de 4 bytes se calculará tomando una palabra que está cuatro posiciones antes que ella, y realizando luego una operación xor con una función de transformación (ByteSub - RotWord - Rcon) de la palabra que se encuentra una posición antes que la actual.
- Para el resto de valores de  $i$  (5, 6, 7, 9, ..., 41, 42, 43) la palabra de 4 bytes se calculará realizando una operación xor entre la palabra que se encuentra cuatro posiciones antes en el array y la palabra que se encuentra una posición antes que la actual.

# Descifrado en AES: InvAddRoundKey

Se invierte el orden de las transformaciones y se toman las funciones inversas.

Según el esquema general ya visto, se tiene el cuadro siguiente.

## InvAddRoundKey

### Vuelta Final

- InvAddRoundKey
- InvShiftRow
- InvByteSub

### Vuelta estándar

- InvAddRoundKey
- InvMixColumn
- InvShiftRow
- InvByteSub

### Salida

- InvAddRoundKey

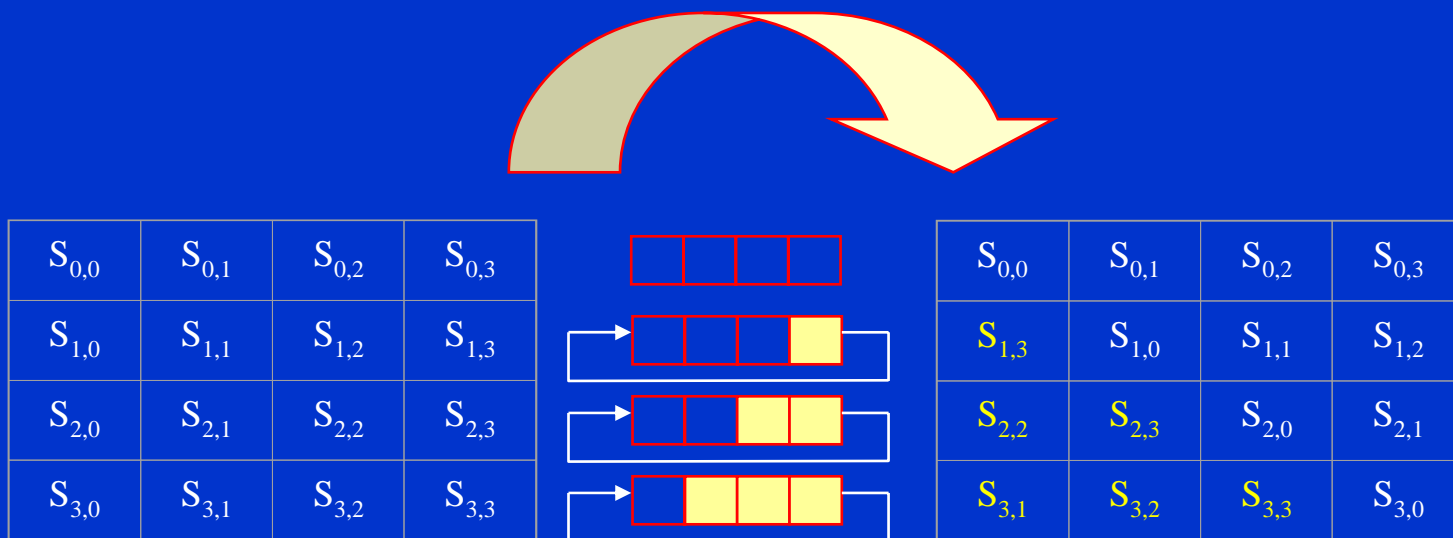


Como se ha usado una operación xor en esta transformación, el inverso es el mismo valor. Por lo tanto aplicaremos las mismas claves de cifrado, pero de forma inversa, desde  $K_r$  a  $K_0$ .

# InvShiftRows

En este caso se desplazan bloques de un byte hacia la derecha módulo columna (en este caso 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 como se muestra.



# Tabla InvByteSub

El inverso consistirá en devolver los valores a su posición original en la tabla ByteSub.

Recordando el valor mostrado en la tabla ByteSub, se muestra que el InvByteSub de **be** es igual a **5a**.

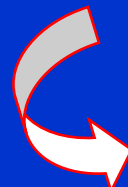
|   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | a  | b  | c  | d  | e         | f  |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|
| 0 | 52 | 09 | 6a | d5 | 30 | 36 | a5 | 38 | bf | 40 | a3 | 9e | 81 | f3 | d7        | fb |
| 1 | 7c | e3 | 39 | 82 | 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 | <b>5a</b> | 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 |

# 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.

$$b(x) = \{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\}$$

Representación  
matricial de la  
función  
InvMixColumns



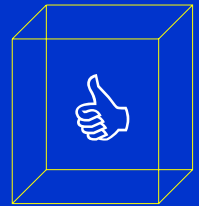
$$\begin{pmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{pmatrix} = \begin{pmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{pmatrix} \begin{pmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{pmatrix} \quad \text{Para } 0 \leq C < Nb$$

Como ejercicio, recupere el valor encontrado en el ejercicio de la función MixColumns mostrado en una diapositiva anterior.

# Resumen de los sistemas de clave secreta

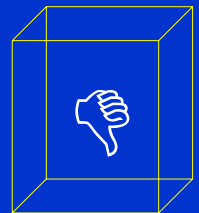
## Pros y contras de los Sistemas de Clave Secreta

- El emisor y el receptor comparten una misma clave.
- La seguridad depende sólo del secreto de la clave.
- La velocidad de cifra es muy alta y los sistemas con un espacio de clave con cientos de bits son muy seguros.
- Permitirán autenticar los mensajes con MACs.



... pero

- 
- Es imposible establecer un sistema de distribución y gestión de claves eficiente entre emisor y receptor.
  - Carecen de una firma digital, al menos en un sentido amplio y sencillo.



Fin del capítulo

## Cuestiones y ejercicios (1 de 3)

1. ¿Qué particularidad tiene el cifrado tipo Feistel?
2. ¿Qué importante diferencia tiene el algoritmo Skipjack con respecto a todos los demás? Razone si eso es bueno o malo en criptografía.
3. ¿Cuál es la razón principal de la debilidad del algoritmo DES?
4. ¿Con cuál de las dos versiones respecto a la reducción de clave aplicada al DES por la NSA se queda Ud.? Razone las dos respuestas posibles.
5. ¿Qué tamaño de bloque de mensaje cifra el DES, con qué longitud de clave y con cuántas vueltas?
6. ¿Tiene algún interés criptográfico la tabla de permutación inicial IP que se repite en sentido contrario al final en el DES? ¿Por qué?
7. ¿Qué distribución especial observa en los dos bloques de texto a cifrar  $L_0$  y  $R_0$  en DES? ¿Qué separación en bits hay entre ellos?

## Cuestiones y ejercicios (2 de 3)

8. ¿Cómo se las arregla DES para realizar operaciones suma módulo dos con sub-bloques de texto de 32 bits y sub-claves de 56 bits?
9. ¿Qué dos importantes funciones cumplen las cajas S en el DES?
10. En la caja  $S_3$  del DES entra la secuencia de bits 101101, ¿qué sale?
11. Si la clave DES en ASCII (no números) es HOLAPACO, ¿cuáles serán la primera y segunda sub-claves de cifrado?
12. ¿Por qué no debe usarse nunca el modo de cifra ECB?
13. ¿Podemos usar el DES como un generador de secuencia cifrante?
14. ¿Por qué decimos que el DES no es un grupo? ¿Qué significa eso?
15. ¿En qué consiste un ataque por encuentro a medio camino?
16. ¿Por qué se usa en el triple DES un cifrado con sólo dos claves tipo EDE y no con tres como su nombre indica?



## Cuestiones y ejercicios (3 de 3)

17. ¿Por qué en IDEA se usa una palabra de 16 bits y no de 32 bits? ¿Se podría usar una palabra de 8 bits ó 24 bits? Justifique su respuesta.
18. Encuentre los resultados de las tres operaciones básicas para un sistema simulado IDEA que trabaja con 4 bits.
19. ¿Qué tamaño de bloque cifra IDEA, con qué longitud de clave y con cuántas vueltas?
20. ¿Cómo se generan las sub-claves en IDEA?
21. ¿Cuáles son las claves  $Z_9$  y  $Z_{10}$  en un sistema IDEA en el que la clave maestra en ASCII es  $K = \text{UnaClaveDePrueba}$ .
22. Encuentre las claves de descifrado de las siguientes claves de cifra en IDEA:  $k_{12} = 3.256$ ;  $k_{13} = 34.517$ ;  $k_{14} = 45.592$ .
23. Sume y multiplique 31 y 18 en  $GF(2^8)$  según algoritmo Rijndael.
24. Invente Ud. mismo diversos ejercicios sobre el algoritmo AES.

Use el portapapeles

## Prácticas del tema 12 (1/6)

Software safeDES:

[http://www.criptored.upm.es/software/sw\\_m001j.htm](http://www.criptored.upm.es/software/sw_m001j.htm)



1. Con la clave ASCII K = **123Clave** cifre el mensaje 8 bytes M = **Hola Ana**. Usando el portapapeles, descifre el criptograma y observe si hay relleno. Repita la cifra del mensaje usando ahora K = **A9A83CFA8B16CF0D** una clave hexadecimal y compruebe nuevamente si hay relleno.
2. Con esas dos claves cifre el mensaje de 15 bytes M = **No me respondes**, descifrelo y compruebe que ahora hay un relleno de un byte para formar un segundo bloque de texto en hexadecimal de 64 bits.
3. Repita el apartado 2 para el mensaje M = **Ya no te saludo más Lucía**.
4. Cifre el mensaje M = **No sale lo mismo**, con K = **1111111111111111** en hexadecimal y compruebe que al descifrar ya no puede usar el portapapeles tomando el criptograma como texto ASCII.
5. Repita el descifrado copiando la entrada del criptograma en hexadecimal y pegándolo en ese formato como entrada a descifrar. Saque conclusiones.

Use el portapapeles

## Prácticas del tema 12 (2/6)

6. Con la clave ASCII  $K = \text{BCBCBCBC}$  cifre el mensaje  $M = \text{Sale lo mismo}$ . Descifrelo con esa clave y con la clave  $K = \text{BCCCCCCB}$ . Explique y justifique lo que ha sucedido.
7. Repita la cifra ahora con las claves hexadecimal  $K = \text{1111112222111111}$  y  $K = \text{2222221111222222}$ . Explique y justifique lo que ha sucedido.
8. Cifre el mensaje  $M = \text{Cifra con clave débil}$ , con la clave  $K = \text{ààààññññ}$ . Vuelva a cifrar el criptograma con la misma clave y compruebe que se cumple la relación  $E_k[E_k(M)] = M$ . No olvide copiar el texto cifrado en hexadecimal para volver a cifrar. Justifique lo que ha pasado.
9. Repita la cifra del apartado 8 con las seis claves débiles del DES, en este caso representadas en hexadecimal.
10. Cifre el mensaje  $M = \text{Ahora son claves semidébiles}$ , con las seis parejas de claves semidébiles  $E_{k1}|E_{k2}$  del DES en hexadecimal y compruebe que se cumple la relación  $E_{k1}[E_{k2}(M)] = M$ .

Use el portapapeles

## Prácticas del tema 12 (3/6)

11. Cifre el mensaje  $M = \text{Probaremos un ataque por fuerza bruta}$ , con la clave  $K = \text{AAABBBAAA}$ . Abra una nueva ventana y descifrelo usando el portapapeles y tomando como entrada el criptograma en hexadecimal. Con ambos textos (claro y criptograma) en hexadecimal proceda a un ataque monousuario con clave inicial  $\text{AAABA000}$  y clave final  $\text{AAABBBBB}$ .
12. ¿Qué puede decir con respecto al tiempo de criptoanálisis real y el tiempo que se requeriría para recorrer todo el espacio de claves dado?
13. Comente y justifique el número de claves válidas encontradas.
14. Repita el ataque con clave inicial  $\text{AAABB777}$  y final  $\text{AAABFFFF}$ . Comente lo que ha sucedido con respecto al espacio de claves elegido.
15. Repita el ejemplo 11 usando un ataque de simulación multiusuario con la clave inicial  $\text{AAABB000}$  y la clave final  $\text{AAABBBBB}$ . Elija el número de procesos desde 1 hasta 10 y observe lo que sucede con el tiempo de ataque a la clave. Justifique lo que ha visto.

Use el portapapeles

## Prácticas del tema 12 (4/6)

16. Repita el ejemplo anterior usando un ataque de simulación multiusuario con la misma clave inicial **AAAB000** pero una clave final **AAABFFFF**.
17. Cifre el mensaje  $M = \text{Ahora atacaremos claves en hexadecimal}$ , con la clave  $K = 1111222233334444$ . La clave inicial es **111122223327BF6F** y la clave final **11112222333B72EE**. Proceda al ataque monousuario.
18. Con la calculadora de Windows en hexadecimal reste la clave inicial de la clave final y encuentre luego su valor en decimal. ¿Por qué no coincide este valor con el número de claves distintas que indica el programa?
19. Si tiene entorno de red, realice este ataque multiusuario con un ordenador trabajando como servidor (Inicio Ejecutar command Enter ipconfig) y los demás como clientes. Elija un rango de claves 10 ó 20 veces mayor.
20. ¿Cuánto tiempo tardaría su computador en romper una clave real de DES?  
¿Cuántos computadores trabajando en paralelo necesitaríamos para romper una clave DES en 24 horas usando este programa safeDES?

Use el portapapeles

## Prácticas del tema 12 (5/6)

Software CryptoIDEA:

[http://www.criptored.upm.es/software/sw\\_m001f.htm](http://www.criptored.upm.es/software/sw_m001f.htm)



1. Cree un archivo txt con el siguiente texto “Una clave con letras C” Cree una clave  $K = \text{CCCCCCCCCCCCCCCC}$  de nombre ClaveTodasC. Cifre el archivo y haga un seguimiento de las subclaves de cifra generadas. ¿Por qué aparecen bloques de 8 subclaves con valores iguales y cuatro al final?
2. Compruebe con la calculadora de Windows las subclaves  $Z_1$  y  $Z_9$ .
3. Cifre nuevamente ese archivo con una clave de nombre UnaClaveMejor y valor  $K = \text{EstaClaveEsBuena}$ . Vuelva a hacer el seguimiento de subclaves, observe lo que sucede y justifíquelo.
4. Compruebe con la calculadora de Windows las subclaves  $Z_1$  y  $Z_2$ .
5. Con esta nueva clave cifre el archivo con texto “De64bits” y vea el archivo cifrado con cuatro bloques de 16 bits. Cifre ahora el archivo con texto mayor “Más de 64 bits” y vuelva a ver la cifra. Saque conclusiones.
6. Descifre esta última cifra y observe las subclaves de descifrado.

Use el portapapeles

## Prácticas del tema 12 (6/6)

7. Con las herramientas del programa compruebe que las claves de descifrado son las inversas multiplicativas, aditivas o del or exclusivo en función de la zona del algoritmo donde se han usado las claves de cifrado, por ejemplo:

$$d_{47} = k_5$$

$$d_{48} = k_6$$

$$d_{49} = \text{inv}(k_1, 65537) \text{ multiplicativo}$$

$$d_{50} = \text{inv}(k_2, 65536) \text{ aditivo (el complemento a } n)$$

$$d_{51} = \text{inv}(k_3, 65536) \text{ aditivo (el complemento a } n)$$

$$d_{52} = \text{inv}(k_4, 65537) \text{ multiplicativo}$$

8. ¿Qué sucede con las claves de cifrado y descifrado si la clave K tiene sólo cuatro caracteres **ABCD**?
9. Observe los bloques de cifrado de 16 bits, en función del tamaño del archivo de entrada. Si  $M \leq 8$  bytes hay 4 bloques, si  $8 < M \leq 16$  bytes hay 8 bloques, etc.

**Software de laboratorio del AES:** próximamente en página web de la asignatura.