Cifrado en bloque

Principios de cifradores en bloque

Cifradores en bloque populares

- DES (y Triple DES) [1977]
  (antiguo estándar EEUU, TDES se sigue utilizando en banca y otras aplicaciones por compatibilidad)

- **AES [2001]**
  (es el estándar actual y más utilizado en la práctica)

- IDEA [1991]
  (estándar internacional que no ha disfrutado del éxito d[redacted])

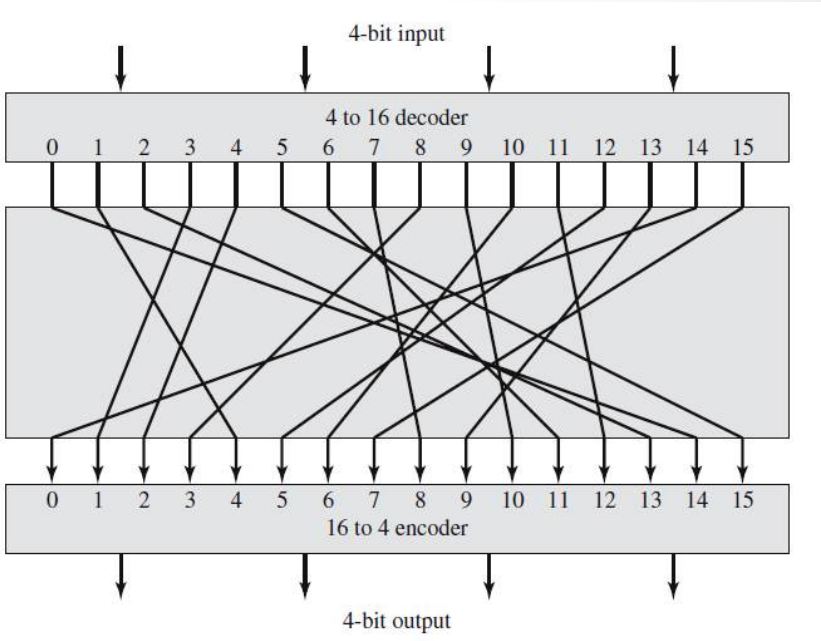
- RC5 [1994] / RC6 [1998]
  (algoritmos d[redacted] son cifradores en bloque al contrario qu[redacted] no han tenido uso en la práctica)

- Blowfish [1993] / Twofish [1998]
  (algoritmos d[redacted] se han empleado en el mundo de[redacted] forman la base d[redacted])

Motivación para la estructur Feistel

- Un cifrador en bloque se puede modelar como una función de transformación de **n** a **n** bits
- Ha de ser una transformación reversible:
  - Cada bloque de entrada produce un único bloque de salida
- El número de transformaciones reversibles posibles es $2^n!$ *(inmanejable en la práctica)*
- Todo cifrador se puede modelar como una sustitución generalizada de bloques:

cifrador en bloque ideal



Motivación para la estructur Feistel

- Problemas:
  - Si se utiliza u bloque pequeño, es susceptible d ataque estadístico
  - Un sustitución generalizada para un bloque grande e impráctica desde el punto de vista de la implementación y e rendimiento
  - En dicha implementación, l clave consiste en l sustitución concreta
- Feistel indica que es necesaria un *aproximación* al cifrador en bloque ideal

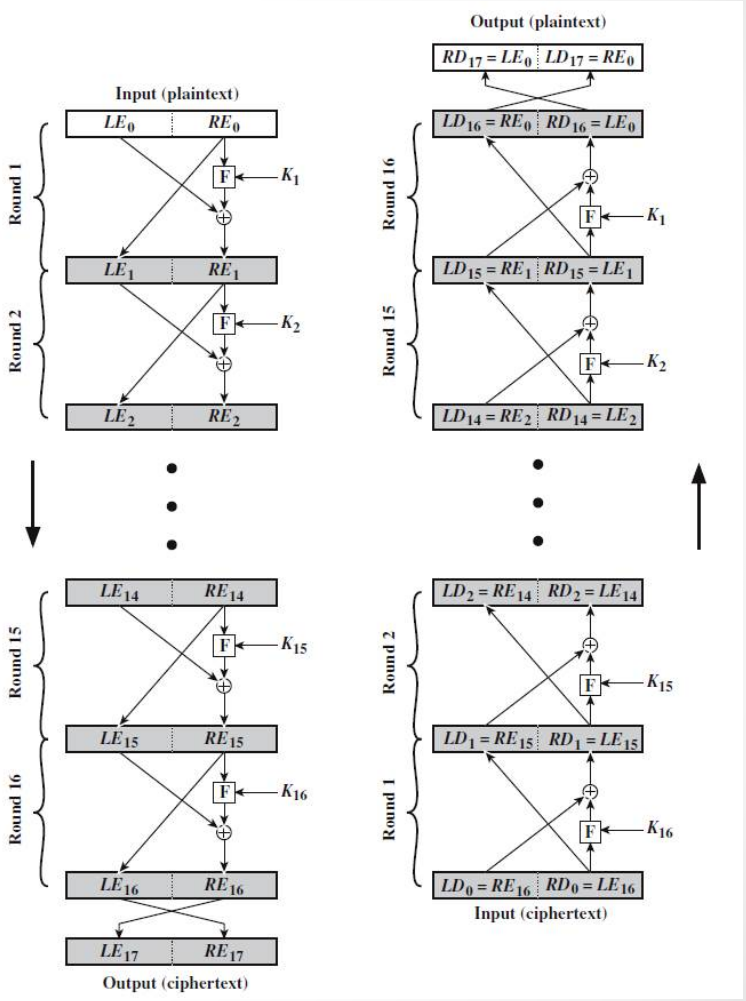
El cifrado Feistel

- Feistel propone aproximar el cifrador en bloque ideal mediante un *cifrador producto*
- Cifrador producto:
  - La ejecución de dos o más cifradores sencillos en secuencia para lograr mayor seguridad
- Desarrollar un cifrador con **k** bits de clave y **n** bits de bloque
  - Permite un total de 2^k transformaciones y no $2^n!$

- Propone un cifrador que alterna entre:
  - Sustitución
    - Cada elemento se sustituye de forma unívoca por otro elemento
  - Permutación
    - Cada secuencia de elementos se sustituye por una permutación de dicha secuencia (reordenación)
- Conceptos similares a los de *difusión* y *confusión* de Shannon

El cifrad Feistel

- El bloque de texto en claro se divide en dos mitades
- Estas mitades pasan por un serie de rondas y se combinan al final para producir el texto cifrado
- En cada ronda se realiza:
  - Sustitución: función F
  - Permutación: intercambio de las dos mitades
- Cad ronda es estructuralmente idéntica pero parametrizada por una subclav única
- Esta estructura es un forma particular de la red de sustitución y permutación (SPN) de Shannon



El cifrad Feistel

- La implementación de la red Feistel depende de
  - Tamaño d bloque
    - Tradicionalmente 64bits, AES es 128 bits
  - Tamaño d clave
    - El mínimo actual es 128 bits, idealmente 256 bits
  - Número d rondas
    - Un número común es 16, pero depende del diseño concreto
  - Algoritmo de generación d subclaves
    - Su complejidad dificulta el criptoanálisis
  - Función de rond F
    - Su complejidad dificulta el criptoanálisis

- Otras consideraciones
  - Rendimiento en software
    - Necesario un rendimiento elevado en software y no únicamente en hardware
  - Facilidad de análisis
    - Cuanto más sencillo sea el algoritmo mayor será la certeza de que no posee vulnerabilidades ocultas u otros problemas de diseño

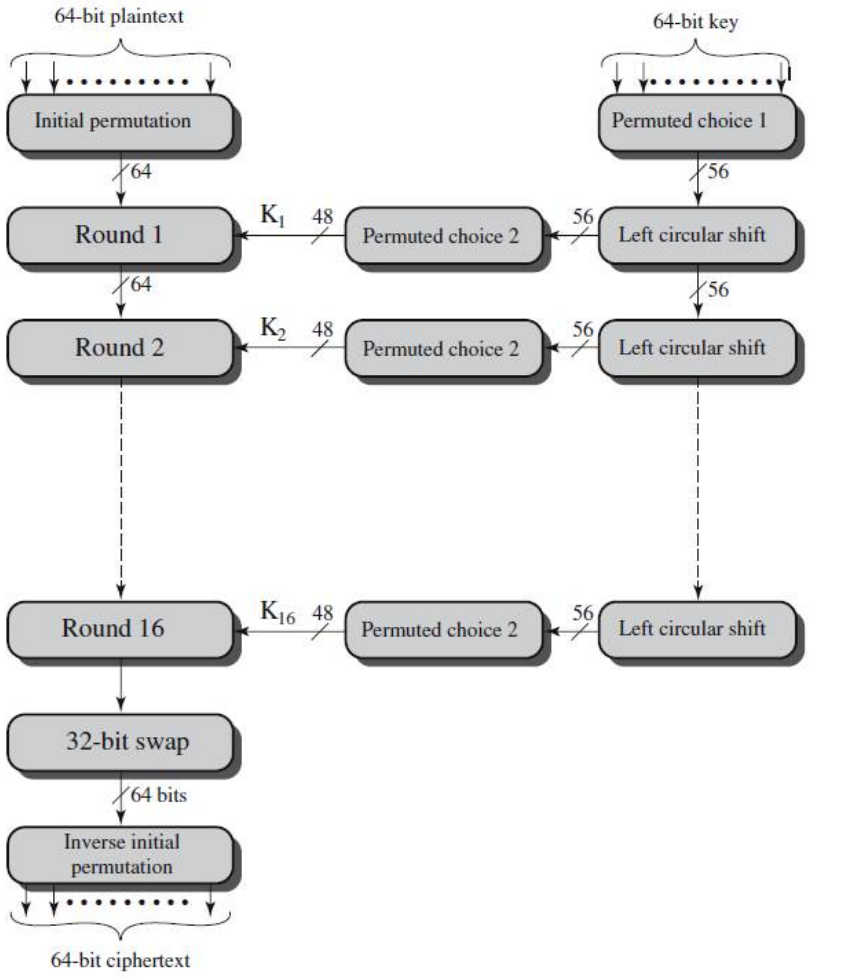
El Estándar de Cifrado de Datos (DES)

El estándar de cifrado de datos (DES)

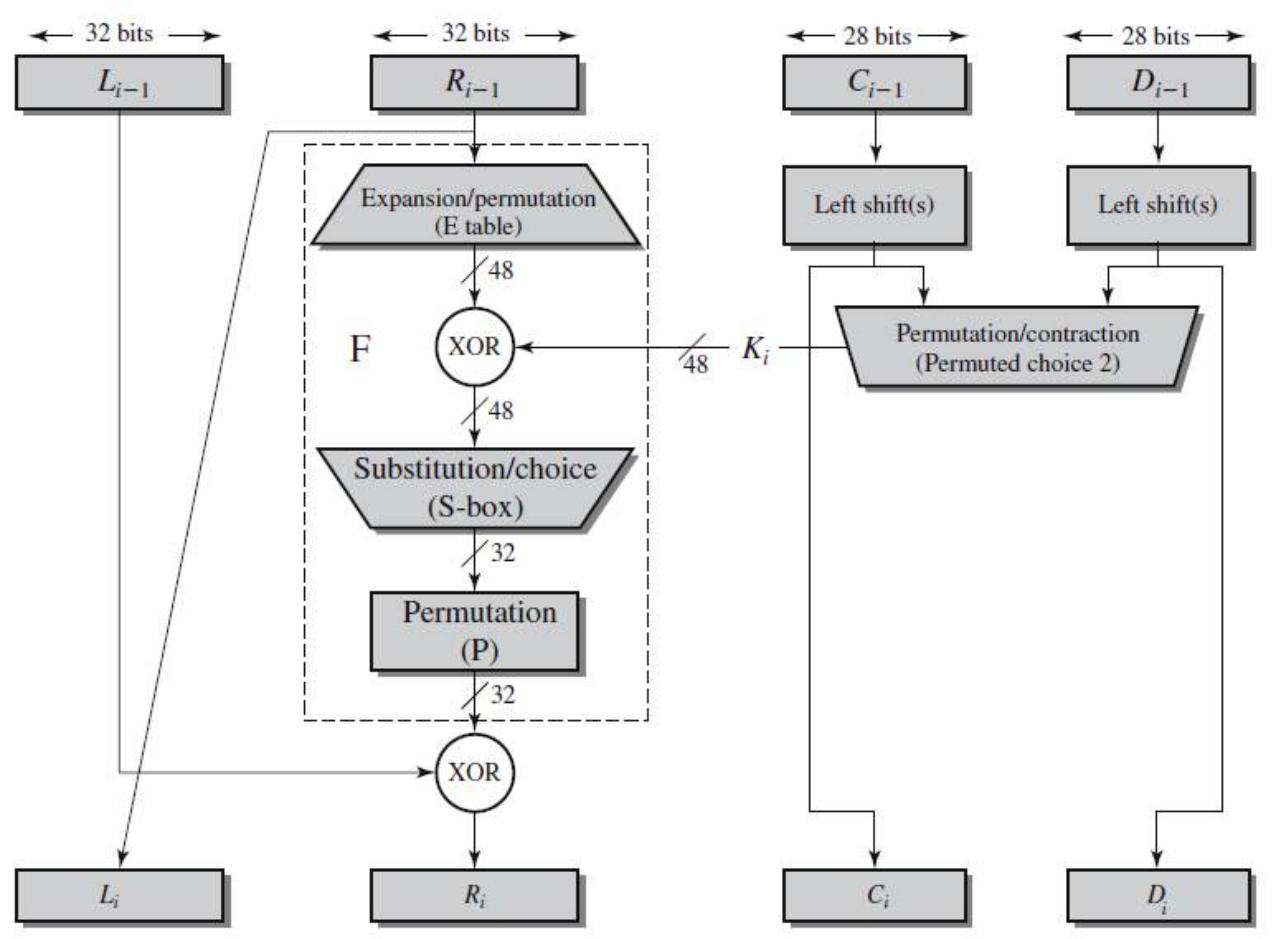
- Lucifer
  - Proyecto de IBM, de finales de los 60, dirigido por Feistel
  - Bloque de 64bits, clave de 128bits
- Versión refinada de Lucifer
  - Interés comercial de IBM
  - Dirigido por Tuchman y Meyer y consultores de NSA
  - Clave de 56bits para caber en un único chip
- 1973, NBS solicita algoritmos para un estándar nacional de cifrado
  - IBM somete a Lucifer
  - Se elige como DES en 1977
- Múltiples críticas
  - Reducción de clave significativa (de 128 a 56)
  - Diseño de cajas “clasificado”
  - La historia ha demostrado que son infundadas

Cifrado en DES (esquema)

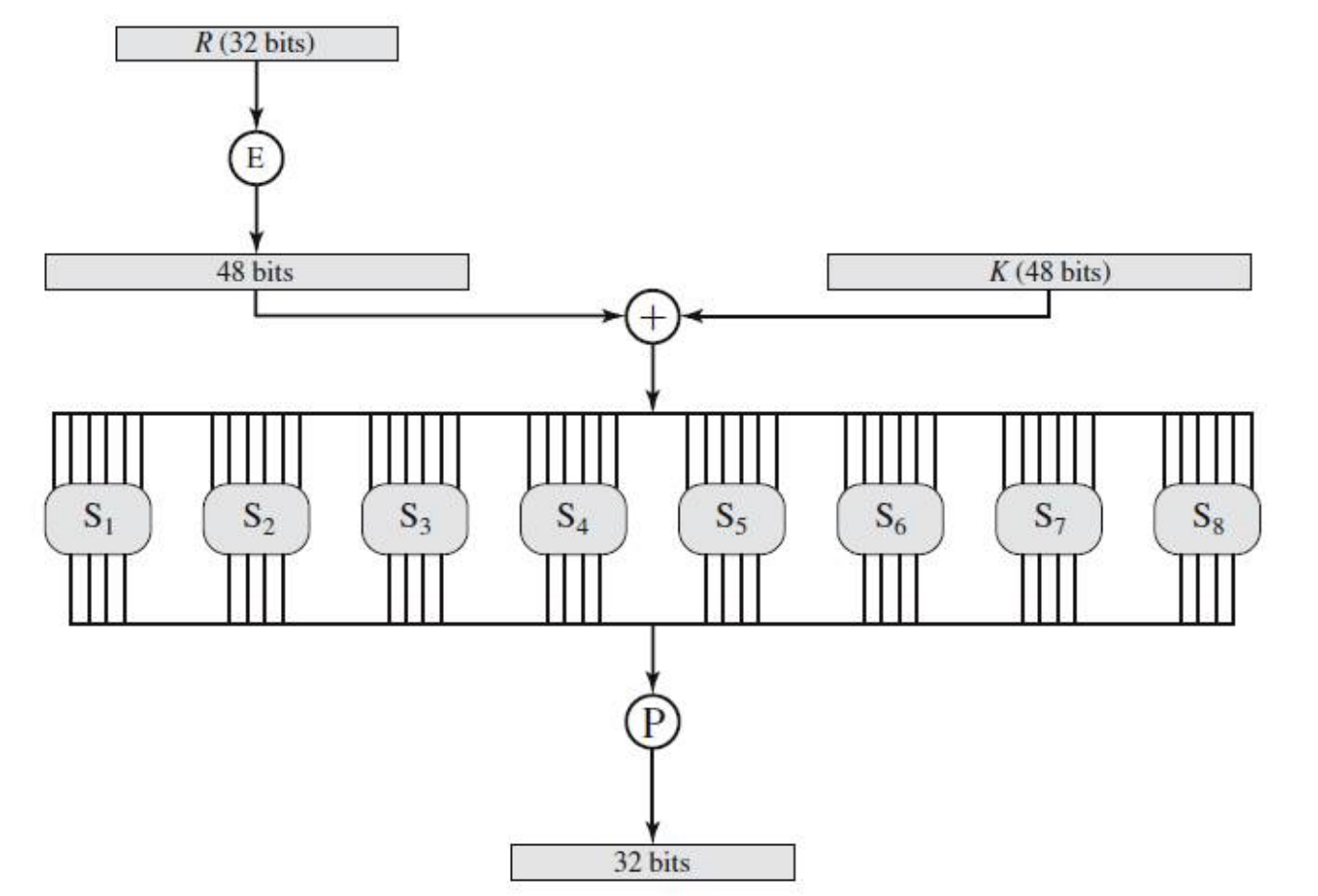
- Dos entradas
  - Texto en claro (64bits)
  - Clave (64bits, sólo 56 útiles)
- Tres fases
  - Permutación inicial
  - 16 rondas de SPN, intercambio de mitades
  - Permutación inicial inversa
- Generación de subclaves
  - Desplazamiento circular (56bits)
  - Selección permutada (48bits)
- Descifrado
  - Mismo algoritmo con subclaves en orden inverso



Cifrado en DES (ronda)



Cifrado en DES (box)



Cifrado en DES (box)

S_1

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

S_2

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

S_3

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

S_4

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

S_5

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

S_6

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

S_7

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

S_8

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

La seguridad de DES

- Claves de 56bits
  - ¿Ataque por fuerza bruta impráctico?
  - 1000 años con 1M pruebas/s
  - En 1977, 20M$ 1M chips con 1M pruebas/s, 10 horas
  - En 1998, EFF 250K$, 3 días
  - Es más complicado que simplemente recorrer las claves *(*hay que identificar correctamente el texto en claro)*
- Cajas de sustitución (S-boxes)
  - Su diseño nunca ha sido público
  - Sospechas de debilidades ocultas
  - Nunca se ha encontrado nada significativo
- Ataques por temporización *(timing attacks)*
  - Se basan en medir las diferencias de tiempo de ejecución en función de la entrada permitiendo obtener información de la clave o estado interno
  - En principio no son aplicables a DES puesto que siempre tarda lo mismo, aunque todos los algoritmos basados en S-box permiten ciertos ataques de temporización basados en accesos a caché.

Criptoanálisis

Criptoanálisis Diferencial

- Esta técnica no se hace pública hasta 1990
- Capaz de romper DES en 2^{47} pruebas con 2^{47} textos en claro elegidos
- No es muy eficaz con DES, IBM admite que conocían la técnica desde 1974
  (no publicar los ataques que encuentran es una estrategia frecuente de la NSA)
- Los cambios realizados a Lucifer (S-boxes y permutación P) fueron para mejorar la resistencia al criptoanálisis diferencial
  - Lucifer de 8 rondas -> 256 textos elegidos
  - DES de 8 rondas -> 2^{14} textos elegidos

Criptoanálisis Lineal

- Consiste en encontrar aproximaciones lineales (sistemas de ecuaciones) a las operaciones realizadas en DES
- Capaz de romper DES con 2^{43} textos conocidos
- A pesar de ser una mejora significativa, sigue sin ser un ataque práctico

Estándar de Cifrado Avanzado (AES)

El estándar de cifrado avanzado (AES)

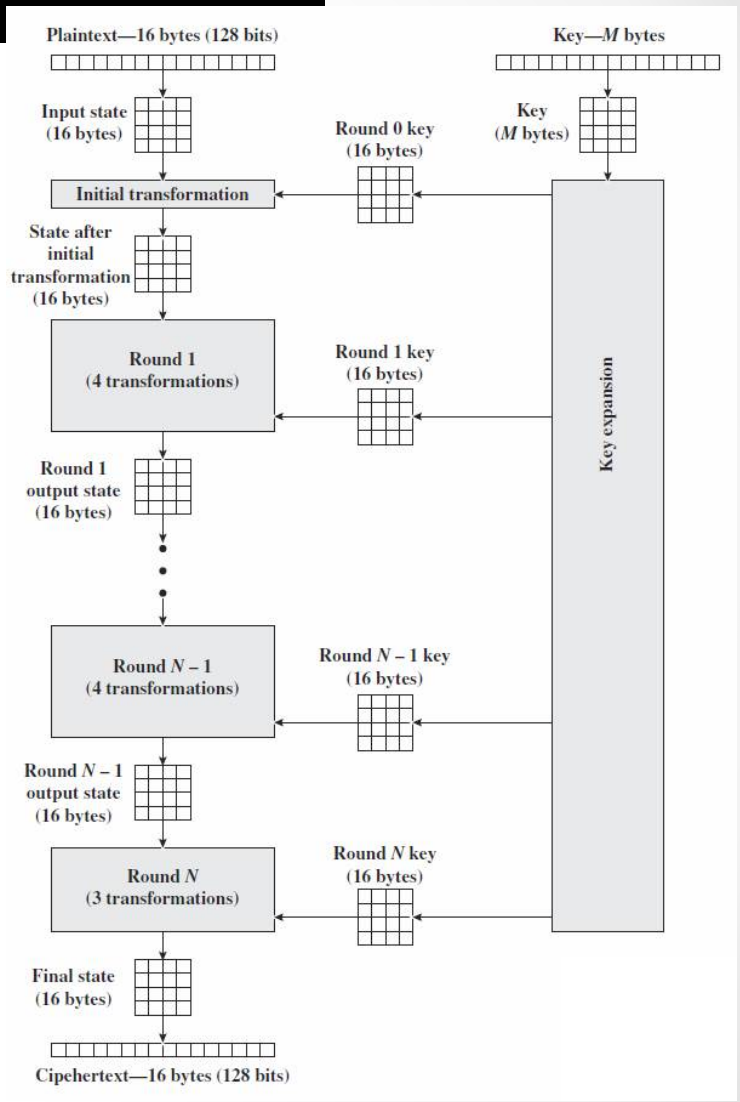
- Fue publicado por el NIST en 2001
- Diseñado para remplazar a DES
- 5 años de competición, 15 algoritmos propuestos
- No utiliza una red Feistel
- Originalmente llamado Rijndael
- Diseñado por Joan Daemen y Vincent Rijmen (Bélgica)
- Los procesadores más modernos incluyen instrucciones para su aceleración por hardware *(AES-NI)*

Aritmética de cuerpos finitos

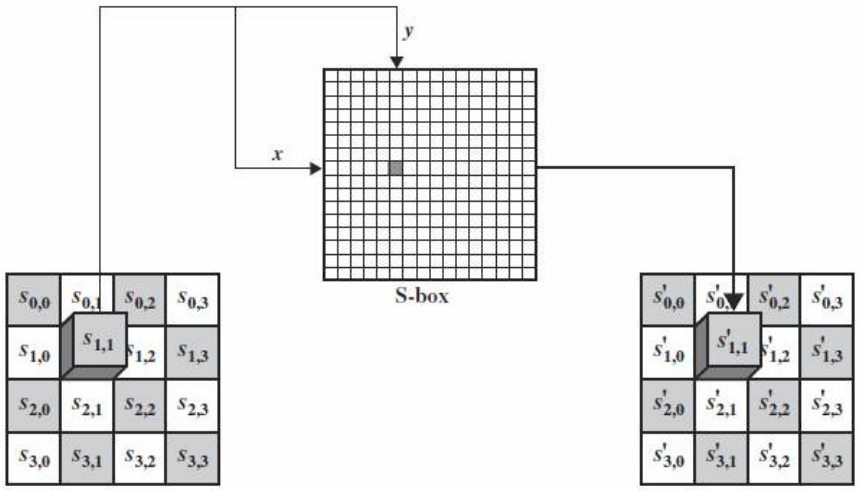
- En AES se realizan todas las operaciones sobre bytes
  - Suma, producto y división sobre GF(2^8)
  - Permite una implementación más eficiente en software
- En álgebra, un cuerpo es un conjunto que permite realizar suma, resta, multiplicación y división obteniendo resultados dentro de ese mismo conjunto
- Los enteros usando aritmética modular (mod 2^n) no son un cuerpo
- Los cuerpos de Galois (GF) se basan en aritmética polinomial y sí son cuerpo
- GF(2^n) contiene 2^n elementos formando un cuerpo

Estructura de AES

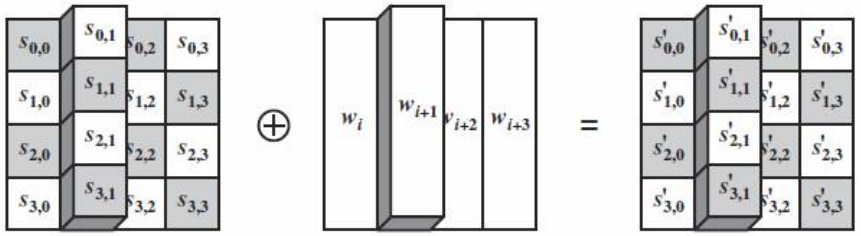
- Bloque de 128bits (16 bytes)
- Clave de 128, 192 o 256 (AES-128,…,AES-256)
- El bloque se representa como una matriz de 4x4 bytes.
- Consiste en 10, 12 o 14 rondas (128, 192 o 256bits de clave, respectivamente)
- Cada ronda consiste en 4 funciones de transformación
  - SubBytes, ShiftRows, MixColumns, AddRoundKey



Funciones de transformación de AES

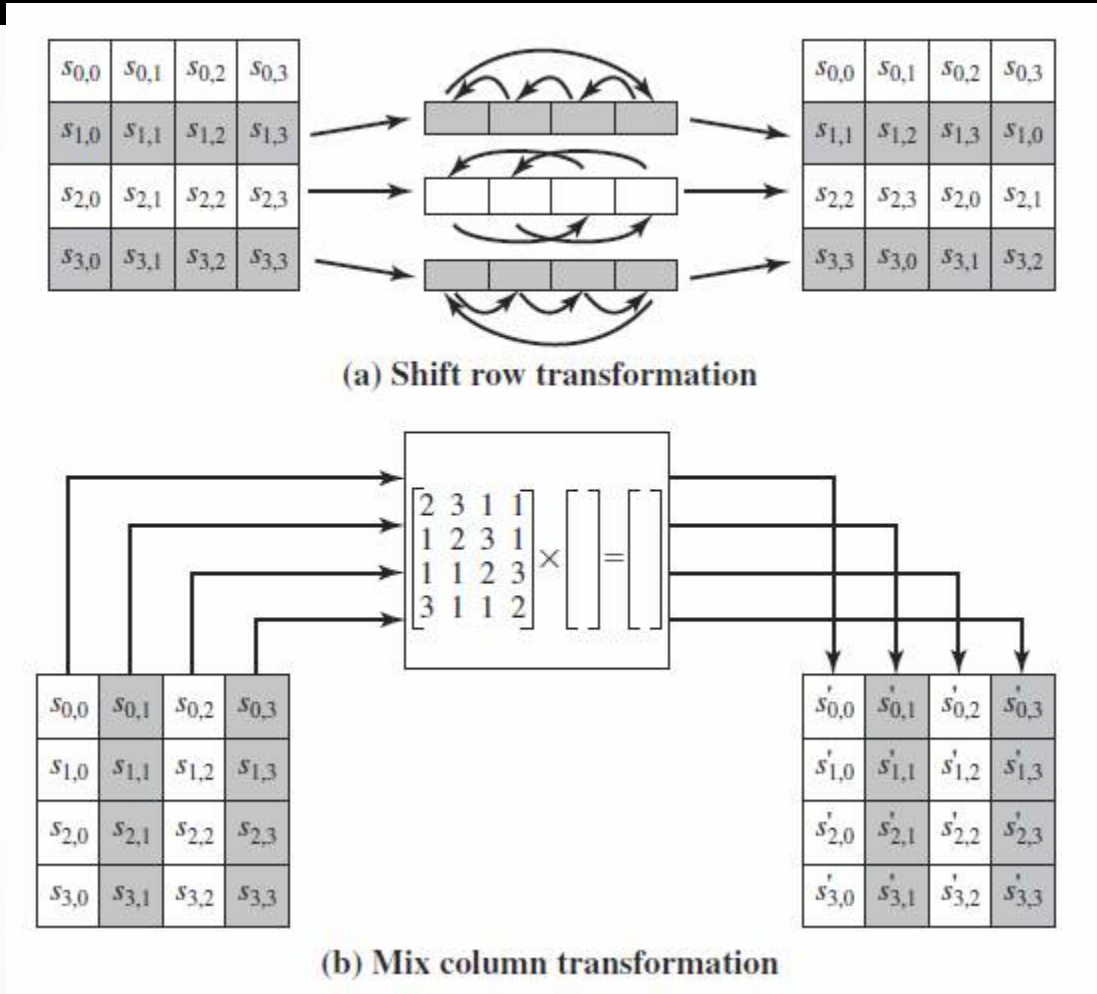


(a) Substitute byte transformation

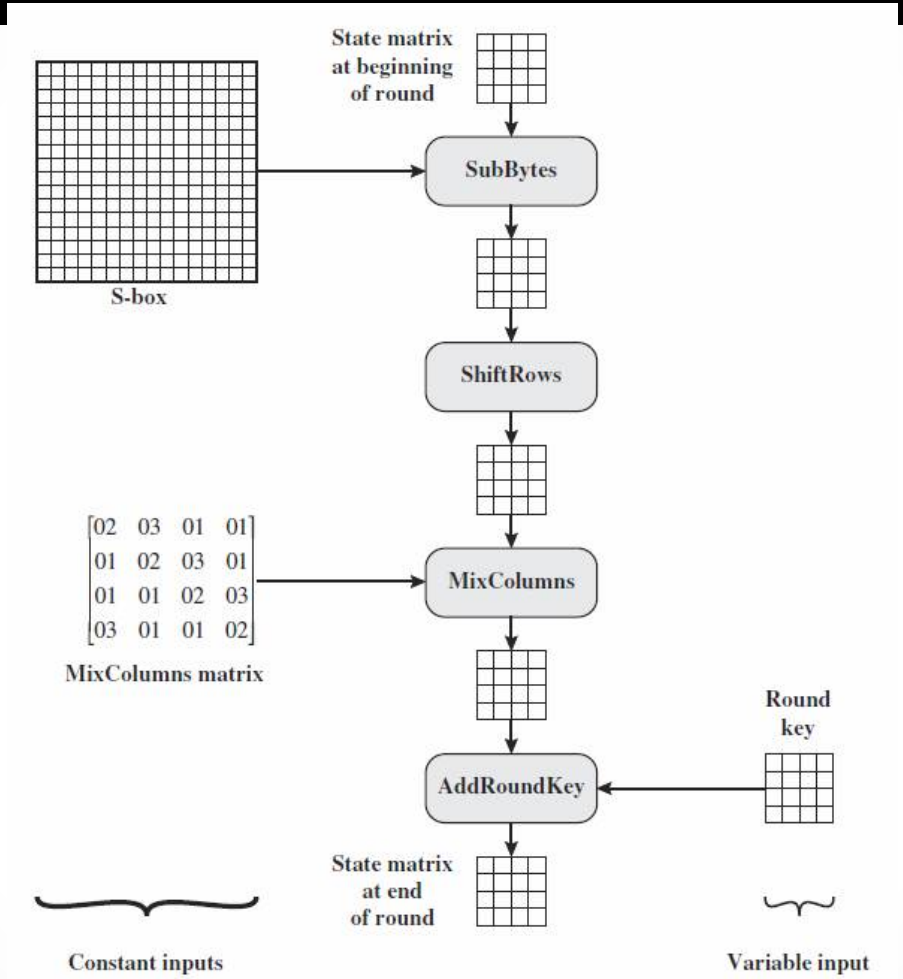


(b) Add round key transformation

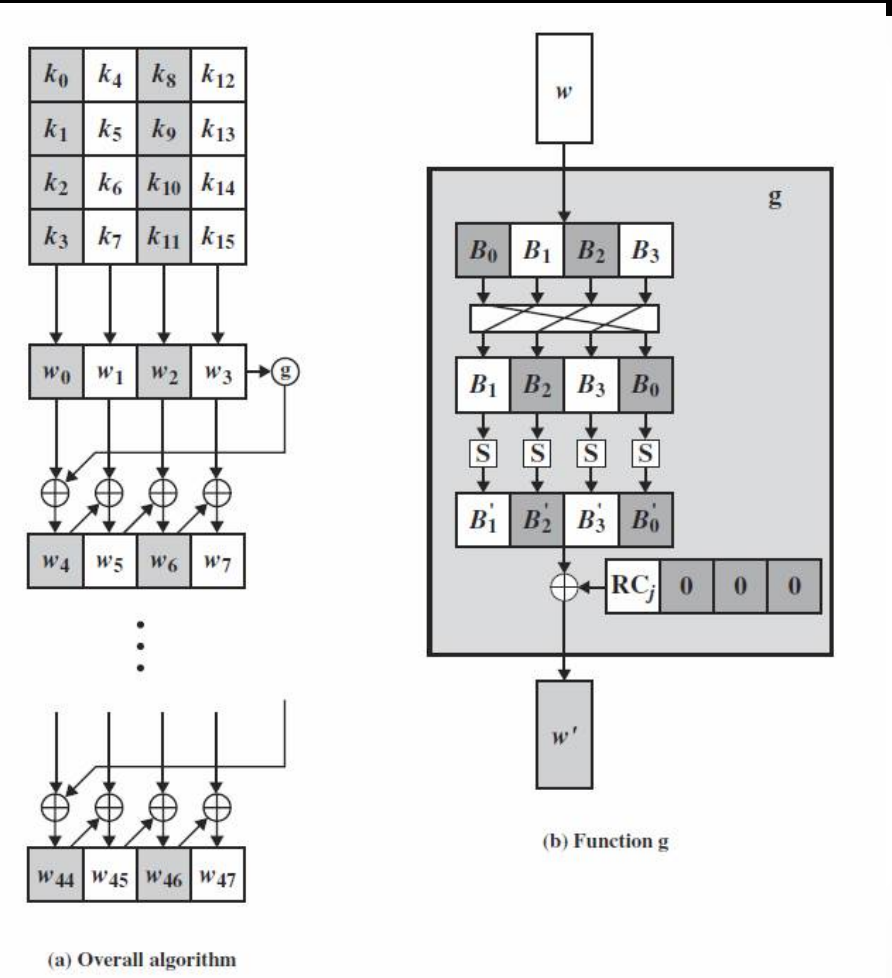
Funciones de transformación de AES



Funciones de transformación de AES



Algoritmo de expansión de clave



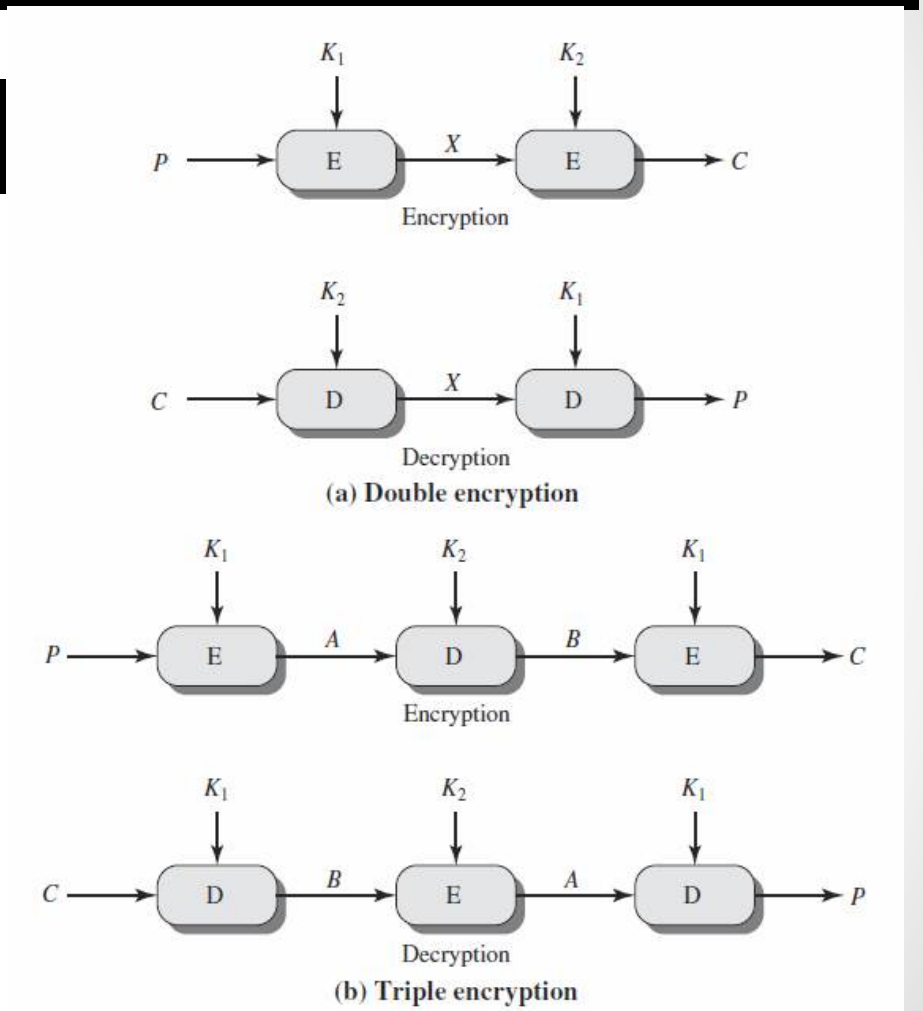
Detalles de implementación AES

- En AES el cifrado no es idéntico al descifrado
  - Se puede transformar el descifrado para que siga la estructura del cifrado, si bien requiere cambiar la expansión de clave
- AES permite la implementación en procesadores de 8bits
  - AddRoundKey -> XOR de bytes
  - ShiftRows -> shift de bytes
  - SubBytes -> tabla de 256 bytes
  - MixColumns -> se puede expresar como XOR y tabla de 256 bytes
- En un procesador de 32bits se pueden expresar las operaciones sobre palabras de 32bits
- Los procesadores modernos incorporan instrucciones nativas AES-NI
  (gran aumento en el rendimiento)
- Es necesario evitar ataques de temporización y de análisis de potencia
  - Estudio de los accesos a caché *(timing attack)*
  - Análisis del calentamiento *(posiciones de acceso a RAM)*

Modos de Operación

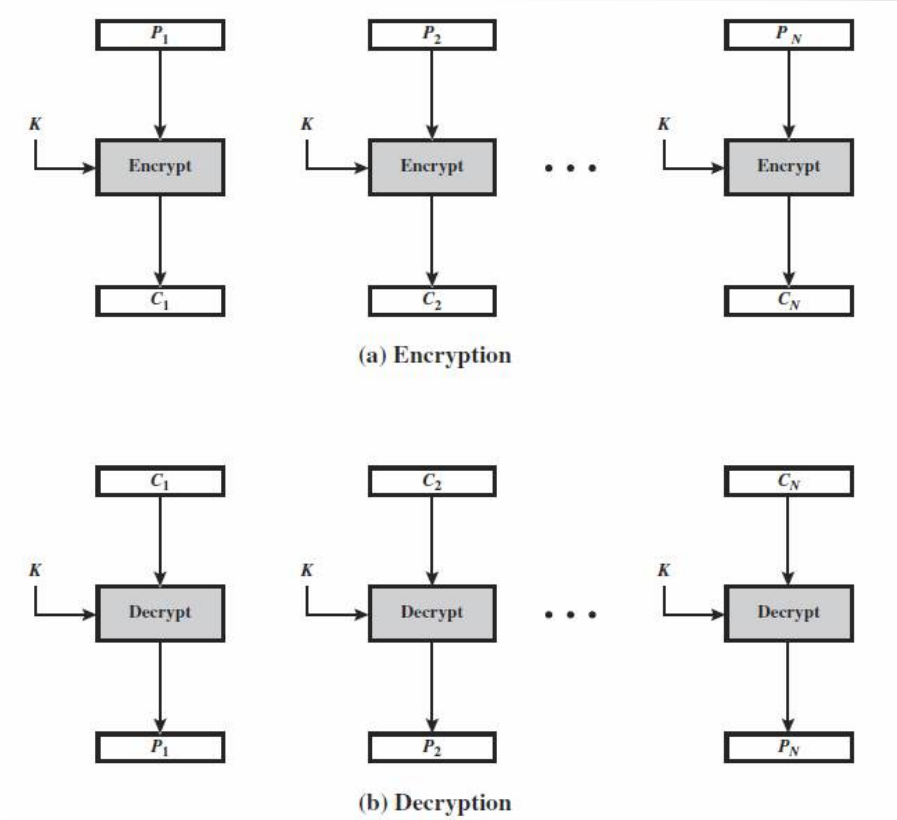
Cifrado múltiple y Triple DES

- Un mismo esquema de cifrado se aplica múltiples veces en secuencia encadenada
- Doble DES
  - Requiere 2 claves
  - Es posible un ataque de hombre en el medio (MITM)
  - Está demostrado que es distinto a DES simple
- Triple DES
  - Requiere 2 o 3 claves
  - Complica el ataque MITM
  - Se usa todavía en banca, etc.



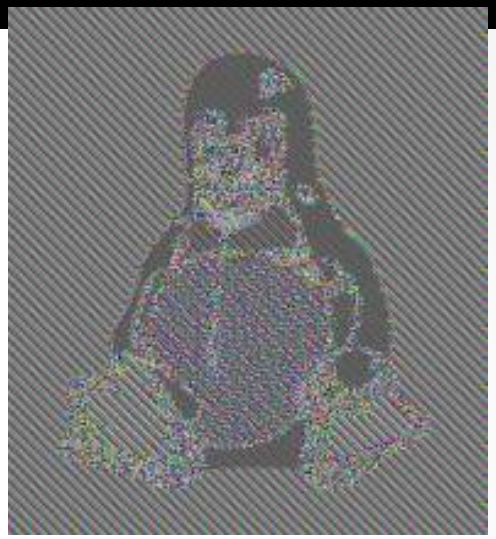
Libro de Código Electrónico (ECB)

- Es el modo de operación más simple
- Se procesa un bloque de texto en claro cada vez y se cifra siempre con la misma clave
- Se llama libro de código porque se podría tabular, dada una clave k, la relación entre cada entrada y salida del cifrador
- No es seguro cuando los bloques de entrada se repiten en el mensaje
- Útil par mensajes muy cortos (claves por ej.)

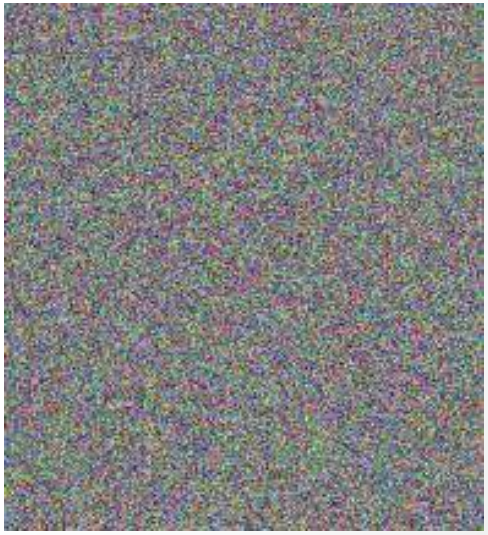


Libro de Código Electrónico (ECB)

Cifrado con ECB

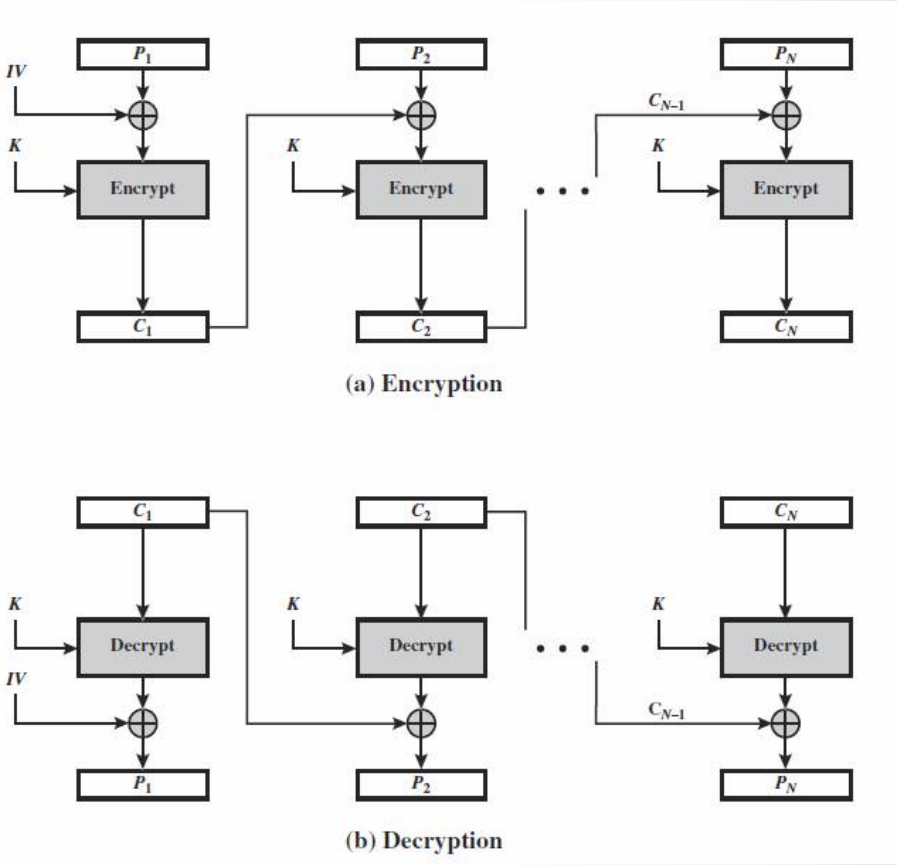


Cifrado en flujo
(o en bloque con otros modos de operación)



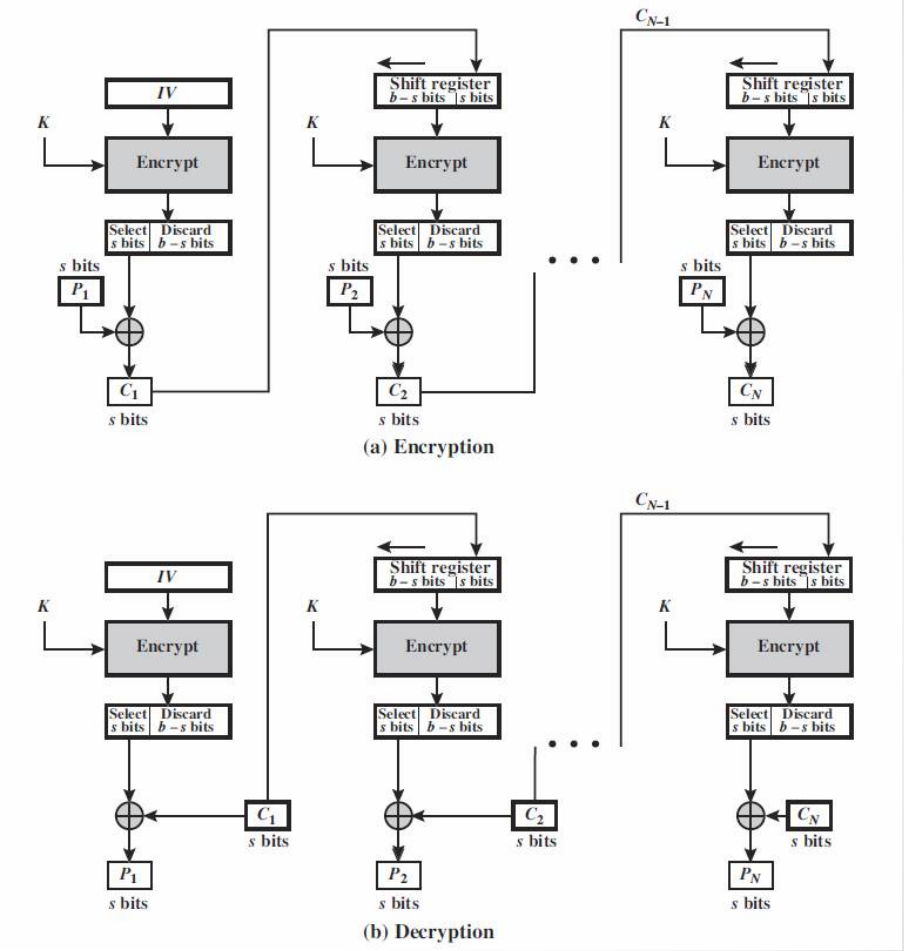
Encadenamiento de bloque (CBC)

- La entrada es el XOR del bloque de texto en claro y del bloque de texto cifrado anterior
- Se parte de un vector de inicialización (IV) que actúa como bloque de texto cifrado inicial
- Al igual que en ECB, es necesario rellenar hasta el tamaño del bloque



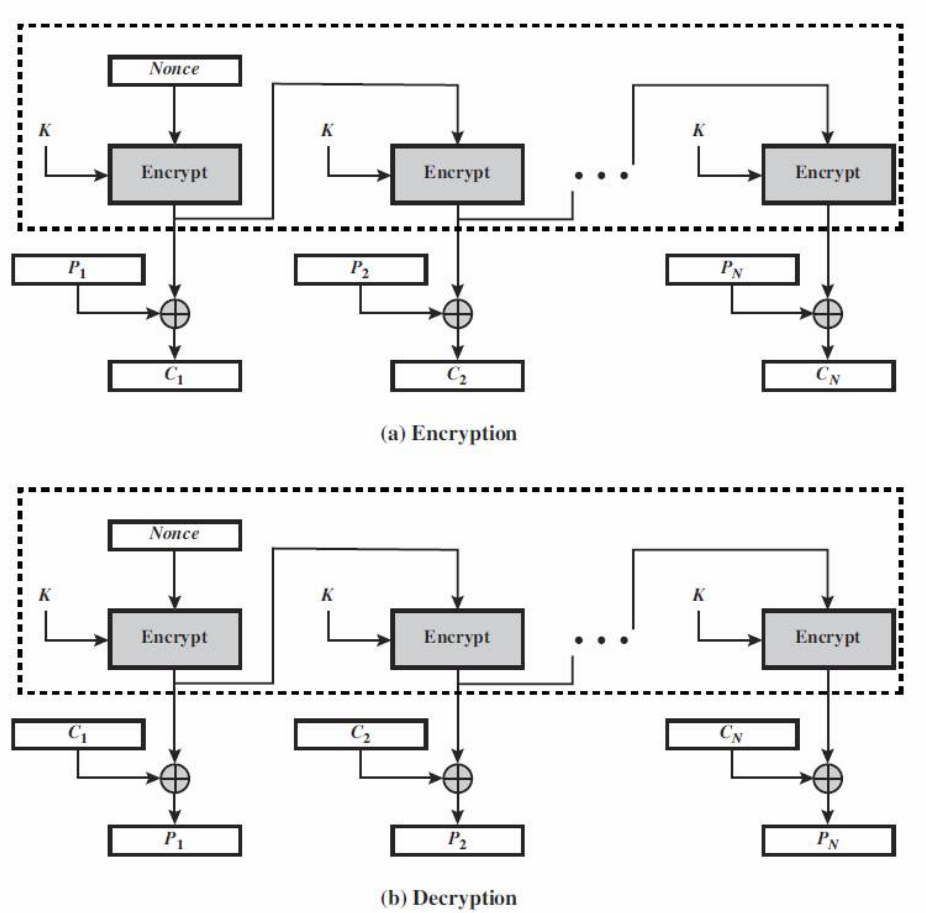
Realimentación de cifrado (CFB)

- Permite convertir un cifrador en bloque en uno en flujo
- La entrada es un vector de inicialización (IV)
- Se hace un XOR entre la salida y el texto en claro, descartando los bits sobrantes
- El resultado se introduce en el registro de entrada, desplazando el contenido a la izquierda
- Reducción de rendimiento



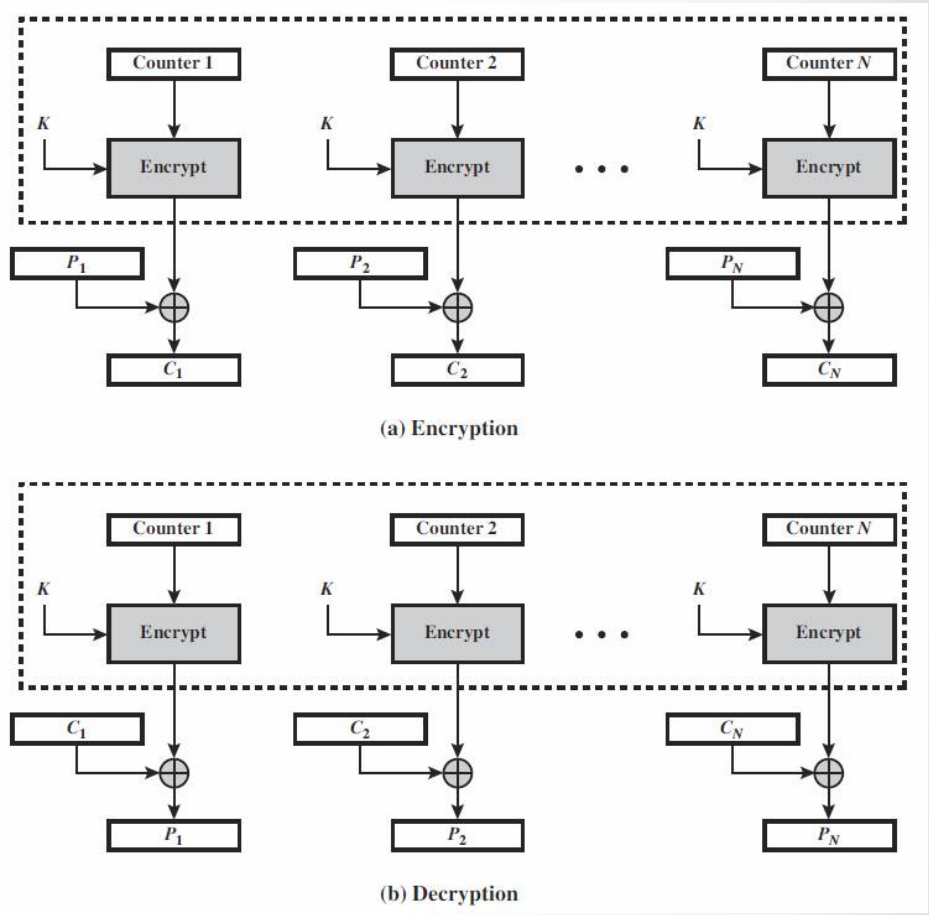
Realimentación de salida (OFB)

- Similar a CFB, pero retroalimentando la salida directa del cifrador en lugar del texto cifrado final
- Como ventaja, evita que los errores de bits se propaguen más allá de un bloque
- La contrapartida es que permite manipular el contenido más fácilmente
- Muy similar a un cifrador en flujo pero cifrando un bloque cada vez



Contador (CTR)

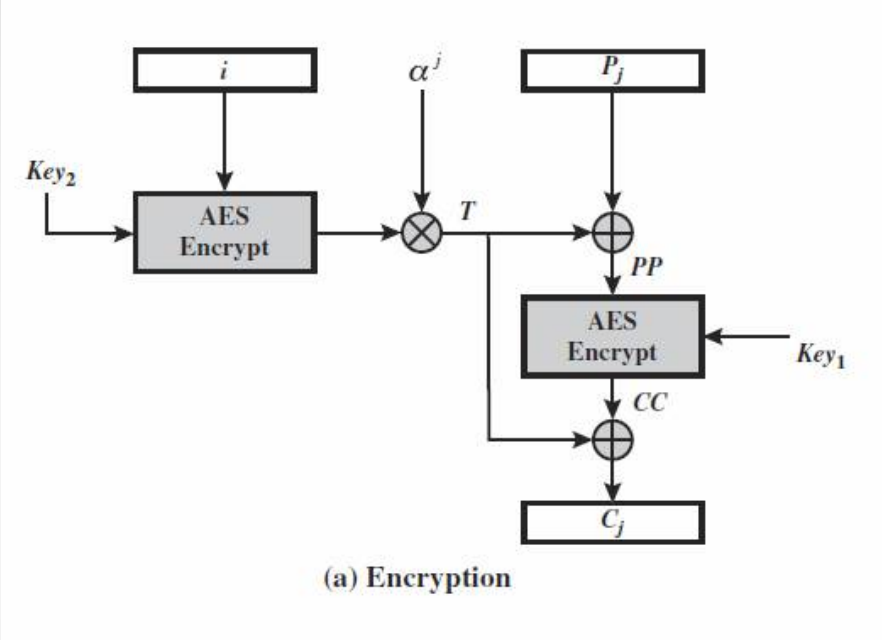
- Se utiliza un contador del tamaño del bloque
- El valor del contador debe ser distinto para cada bloque de texto en claro que se cifre
- Generalmente se inicializa con un IV y se incrementa de uno en uno (mod 2^b)
- Se cifra el contenido, haciendo un XOR del resultado con el texto en claro
- No hay encadenamiento, permite implementaciones en paralelo
- Sólo es necesario el algoritmo de cifrado y no el de descifrado



Xc Cifrad Xo (XT AES)

- Específico para almacenamiento (data at rest)
- Este estándar (IEEE P1619) asume características especiales:
  - El texto cifrado está disponible a un atacante
  - Los datos no se reorganizan dentro del medio de almacenamiento
  - Se accede en bloques de tamaño fijo e independientes entre si (sectores)
  - El cifrado se realiza en bloques de 16 bytes y de forma independiente a los otros bloques (acceso aleatorio)
  - No es necesario más metadata que la localización de cada bloque
  - El mismo texto en claro redunda en textos cifrados distintos para sitios distintos pero siempre se cifra igual en el mismo sitio
  - Un dispositivo que cumpla el estándar puede descifrar los datos cifrados por otro dispositivo estándar

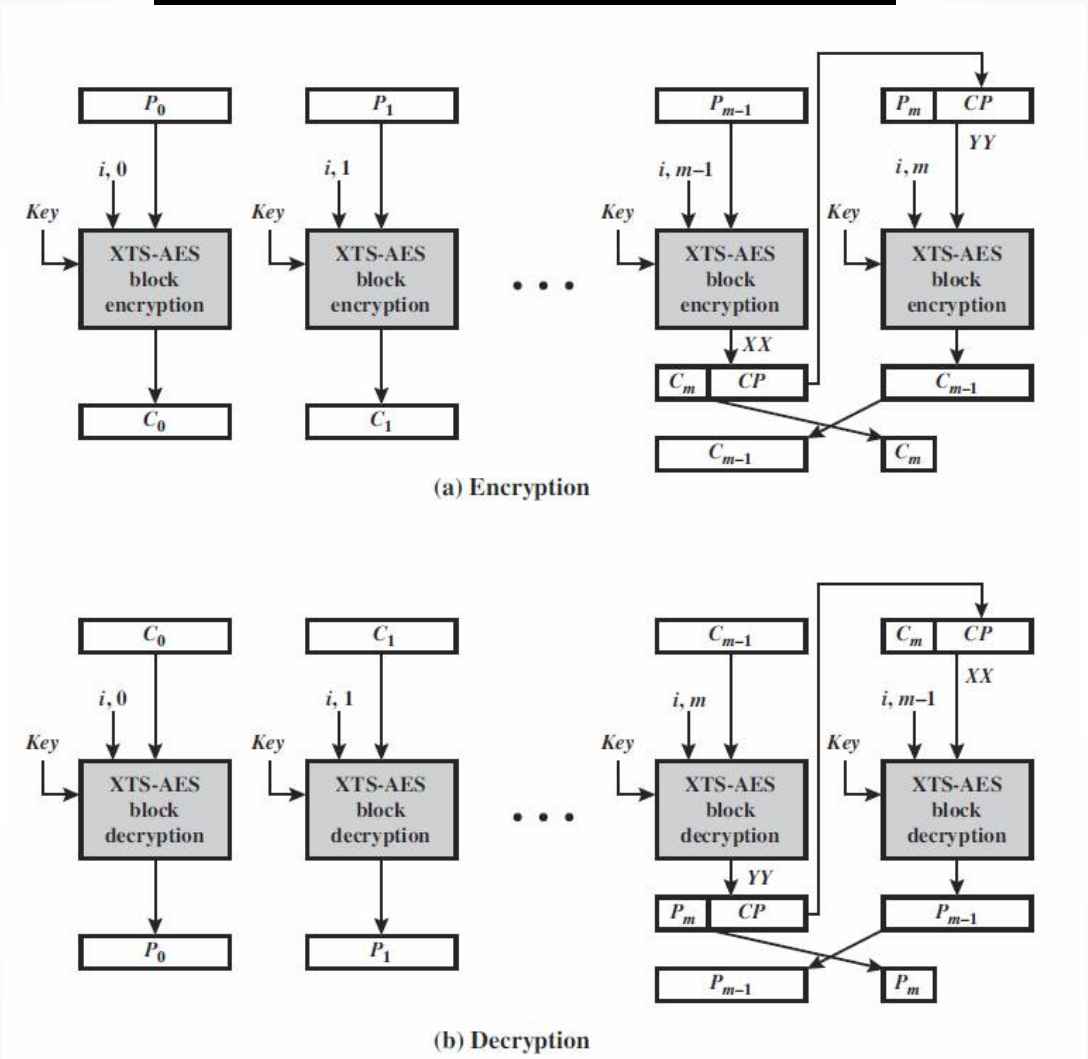
XTS AES en un único bloque



P -> texto en claro
i -> número de sector
j -> bloque dentro del sector
α -> el número 2

No defiende frente a ataques de manipulación o modificación de los datos.

Modo XTS AES


P_0
P_1
P_{m-1}
P_m CP
$i, 0$
$i, 1$
$i, m-1$
i, m
YY
Key
XTS-AES block encryption
XX
C_m CP
C_{m-1}
C_0
C_1
C_{m-1}
C_m
(a) Encryption
C_0
C_1
C_{m-1}
C_m CP
XX
XTS-AES block decryption
YY
P_m CP
P_{m-1}
P_0
P_1
P_{m-1}
P_m
(b) Decryption

Ampliación

Otros materiales

- Se puede consultar el capítulo 10 del libro de Lucena
  (en los materiales de UACloud)

- También se puede consultar el capítulo 7 de
  "Handbook of Applied Cryptography"
  (más avanzado y en inglés)

Cuestiones

- Busca información online acerca de otros cifradores en bloque (ver transparencia número 3). Un buen punto de partida puede ser *Wikipedia*.

- ¿Qué cifrador en bloque elegirías en la actualidad, atendiendo a la seguridad y el rendimiento? ¿Con qué modo de operación?

- ¿Hay situaciones en las que sería preferible un cifrador en bloque a uno en flujo o viceversa?