



# Cifrado en bloque

# Principios de cifradores en bloque

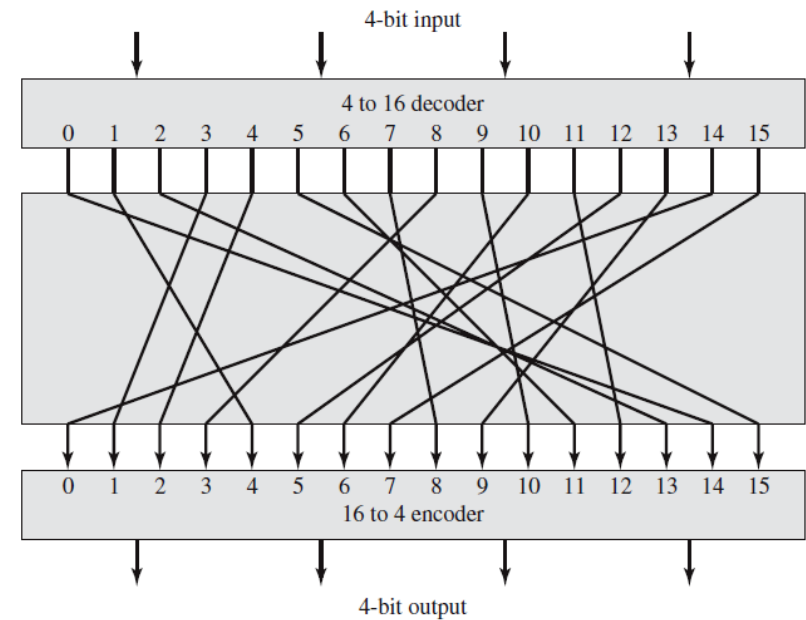
...

# 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 de DES o AES)*
- **RC5 [1994] / RC6 [1998]**  
*(algoritmos de Rivest; son cifradores en bloque al contrario que RC4 y no han tenido uso en la práctica)*
- **Blowfish [1993] / Twofish [1998]**  
*(algoritmos de Schneier; se han empleado en el mundo del software libre y forman la base de bcrypt)*

# Motivación para la estructura 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 estructura Feistel

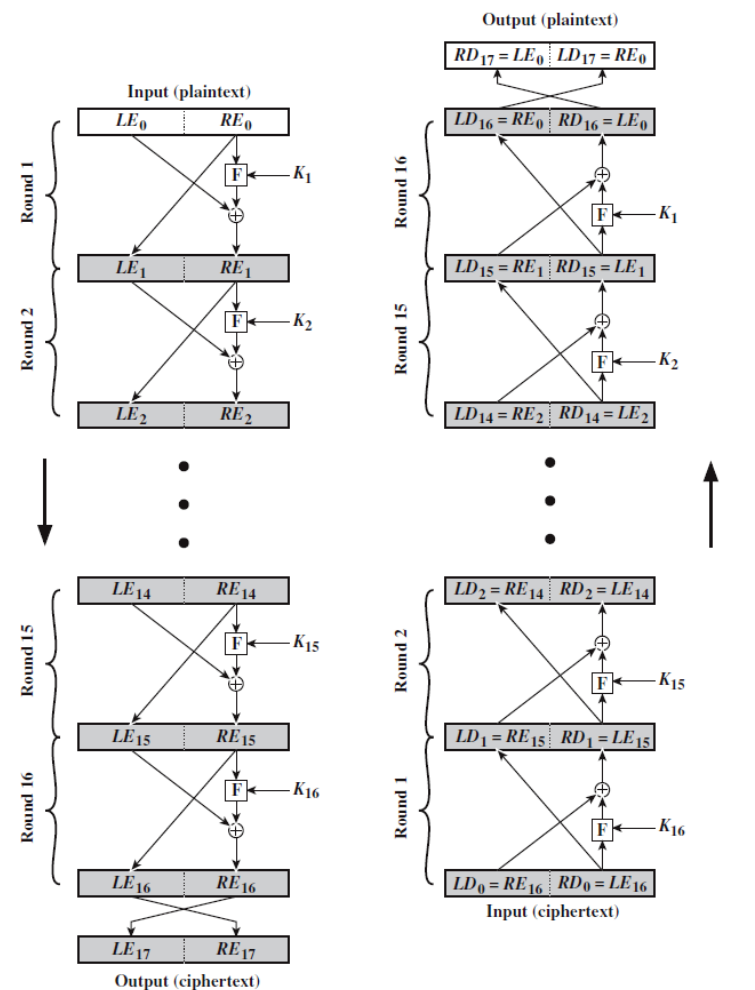
- Problemas:
  - Si se utiliza un **bloque pequeño**, es susceptible de **ataque estadístico**
  - Una **sustitución generalizada** para un bloque grande es **impráctica** desde el punto de vista de la **implementación** y el **rendimiento**
  - En dicha implementación, la **clave** consiste en la **sustitución** concreta
- Feistel indica que es necesaria una ***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 cifrado Feistel

- El bloque de texto en claro se divide en **dos mitades**
- Estas mitades pasan por una **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
- Cada **ronda** es estructuralmente **idéntica** pero parametrizada por una **subclave única**
- Esta estructura es una **forma particular** de la red de sustitución y permutación (SPN) de Shannon



# El cifrado Feistel

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



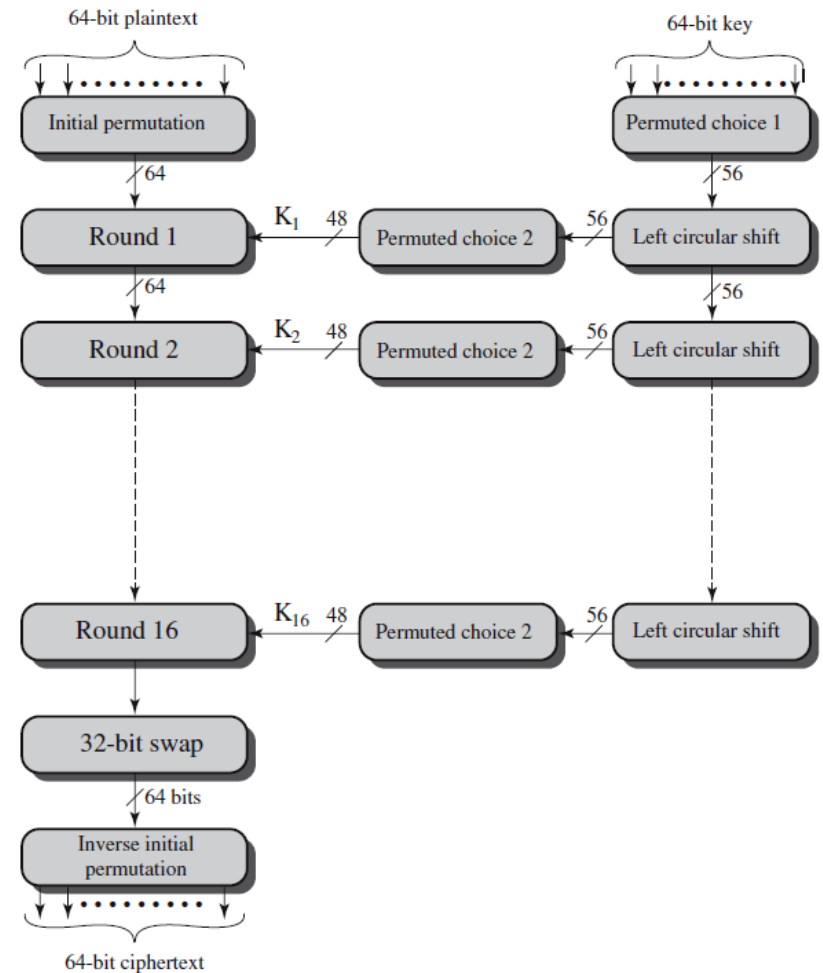
# El Estándar de Cifrado de Datos (DES) ...

# 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



# 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.

# Criptografía

## Criptografía 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

## Criptografía 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*)

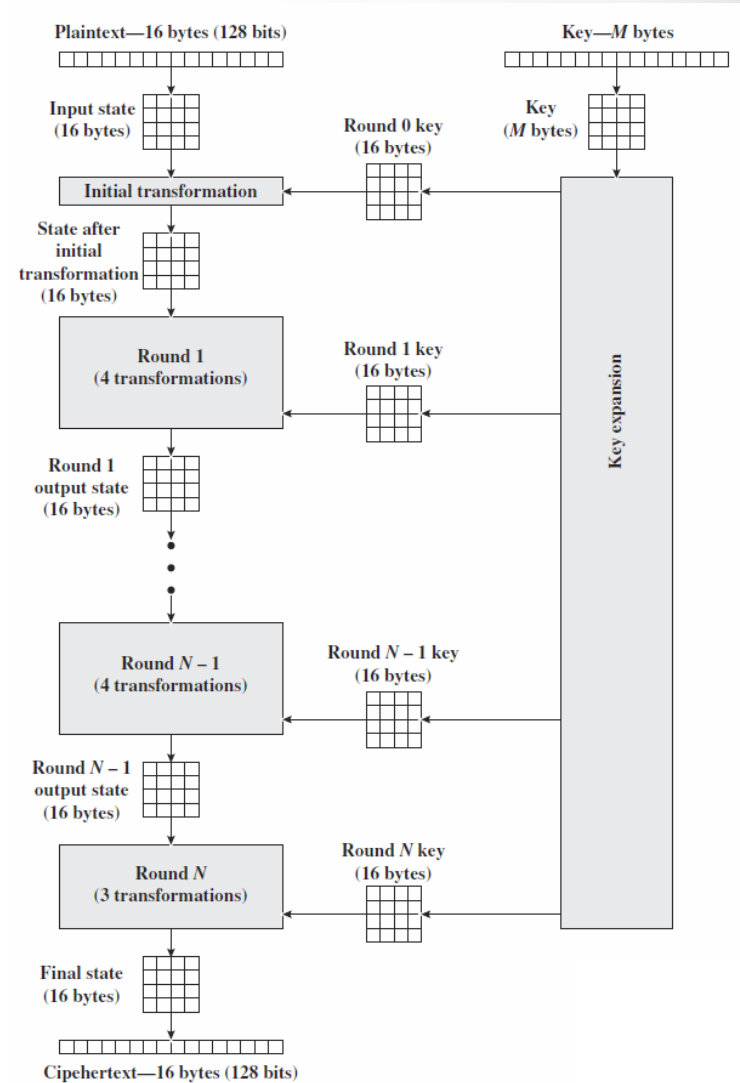
# Aritmética de cuerpos finitos

- 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 ( $\text{mod } 2^n$ ) 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



# Detalles de implementación AES

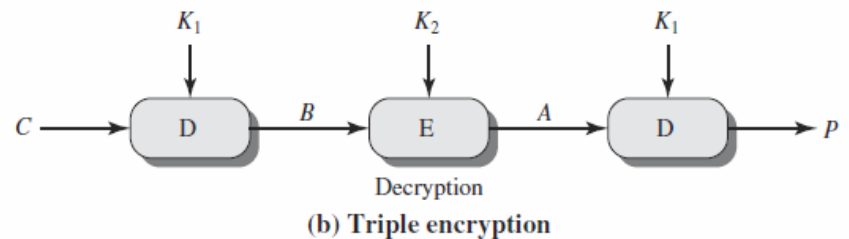
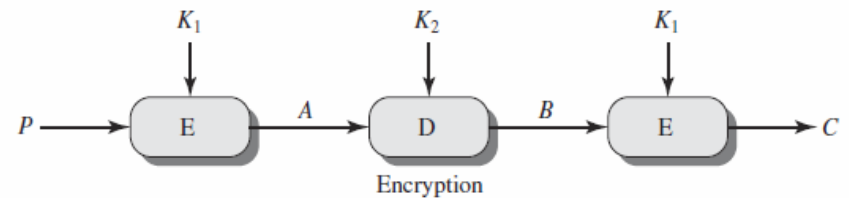
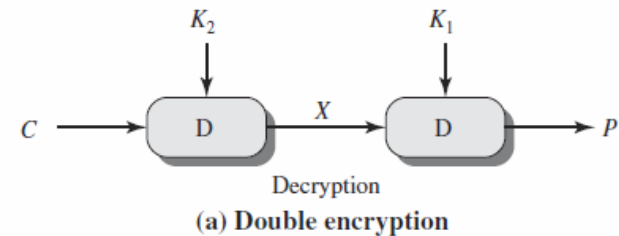
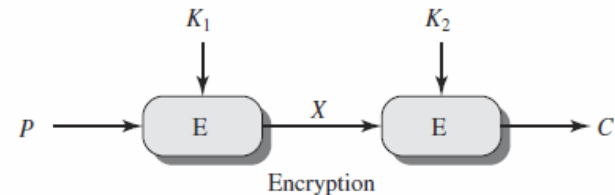
- En AES el cifrado no es idéntico al descifrado
  - Se puede transformar el descifrado para que siga la estructura del cifrado, si bien requiere cambiar la expansión de clave
- AES permite la implementación en procesadores de 8bits
  - AddRoundKey -> XOR de bytes
  - ShiftRows -> shift de bytes
  - 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

...

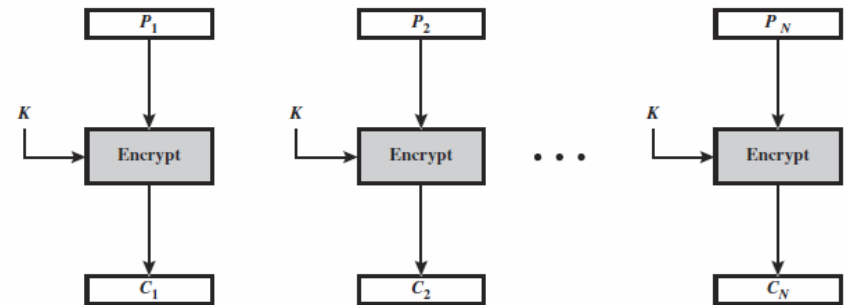
# Cifrado múltiple y Triple DES

- 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.

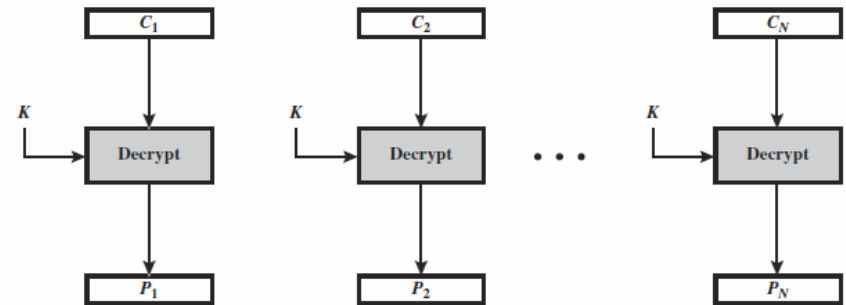


# Libro de Código Electrónico (ECB)

- 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 para **mensajes muy cortos** (claves por ej.)



(a) Encryption



(b) Decryption

# Libro de Código Electrónico (ECB)

Cifrado con ECB

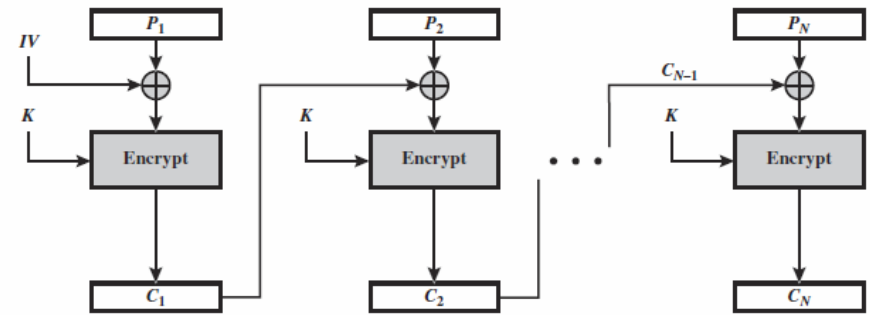


Cifrado en flujo  
(o en bloque con otros  
modos de operación)

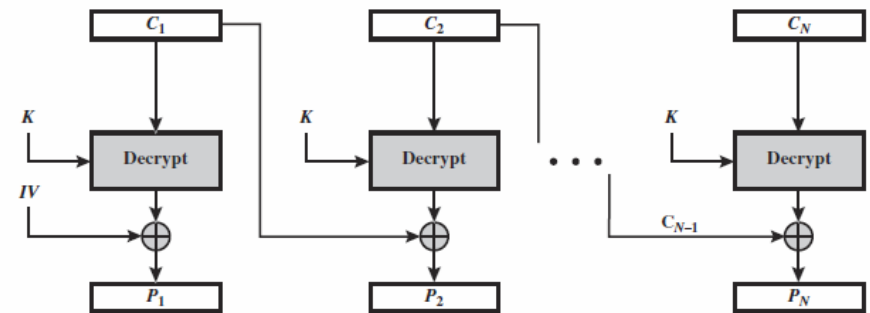


# Encadenamiento de bloque (CBC)

- 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



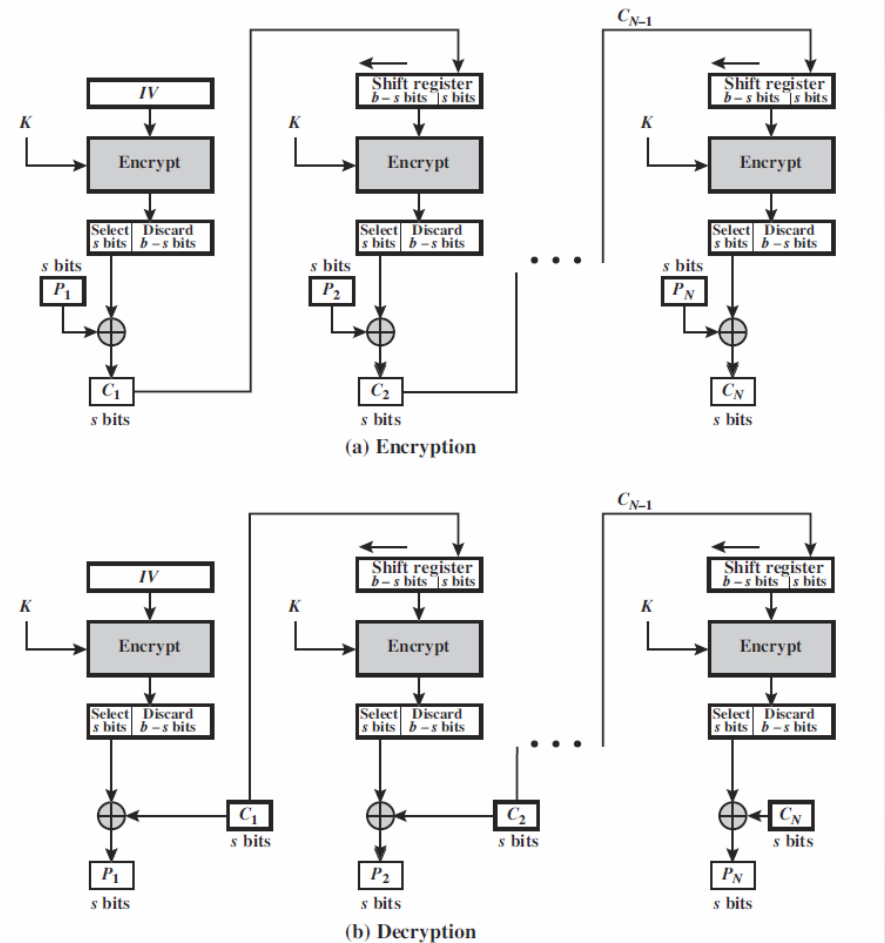
(a) Encryption



(b) Decryption

# Realimentación de cifrado (CFB)

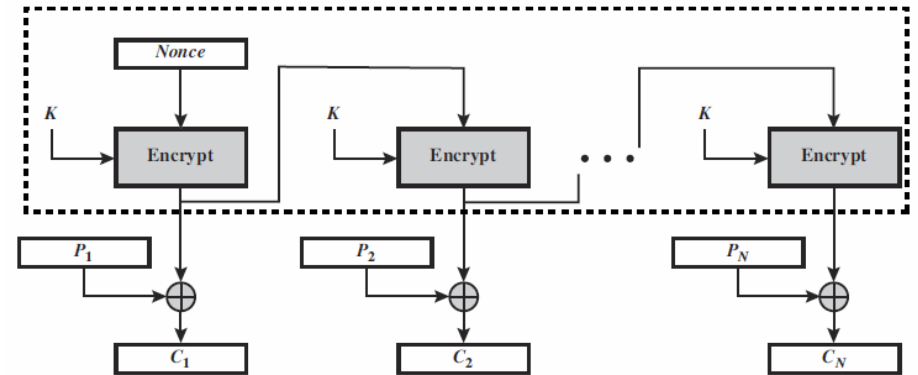
- Permite convertir un cifrador en bloque en uno en flujo
- La entrada es un vector de inicialización (IV)
- Se hace un XOR entre la salida y el texto en claro, descartando los bits sobrantes
- El resultado se introduce en el registro de entrada, desplazando el contenido a la izquierda
- Reducción de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



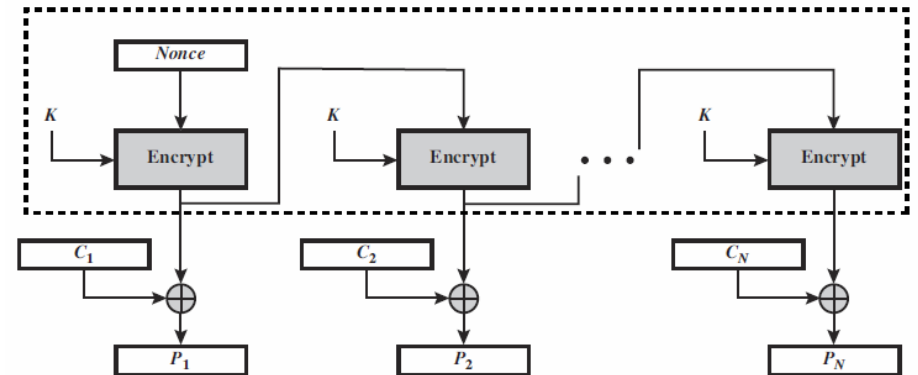


# Realimentación de salida (OFB)

- 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



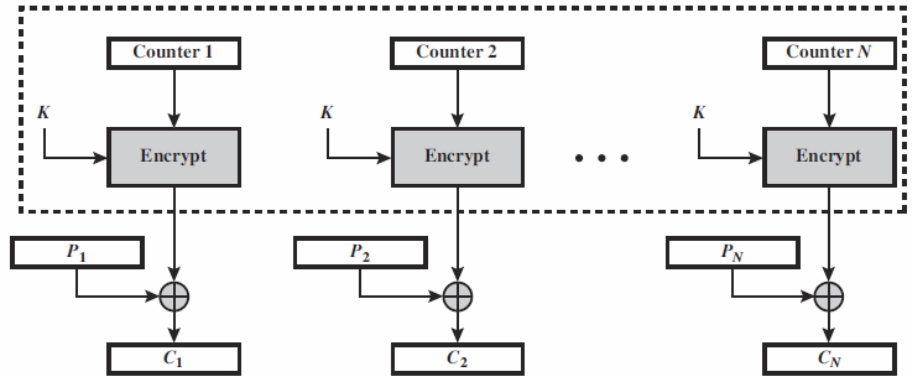
(a) Encryption



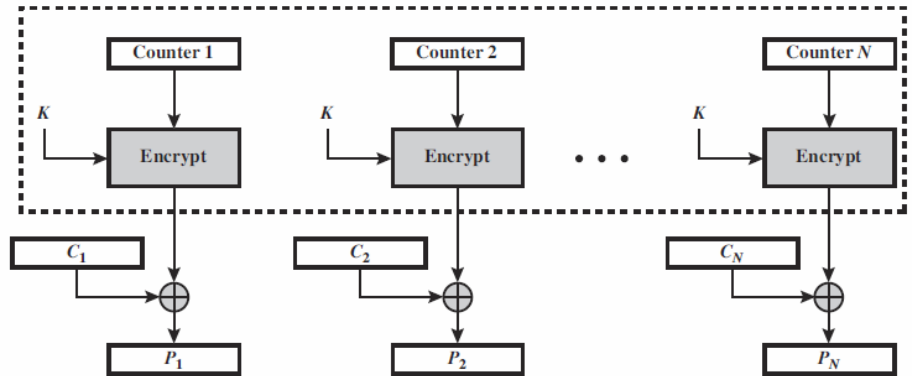
(b) Decryption

# 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



(a) Encryption

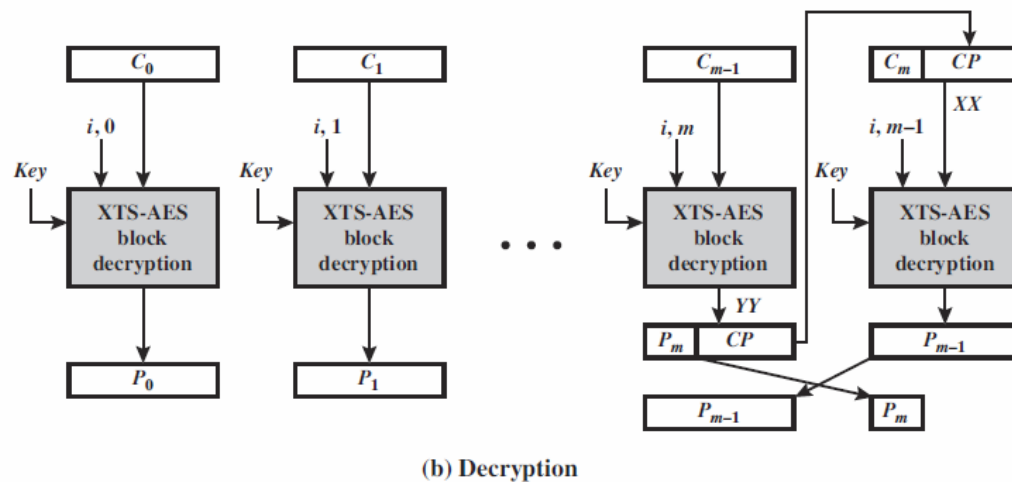
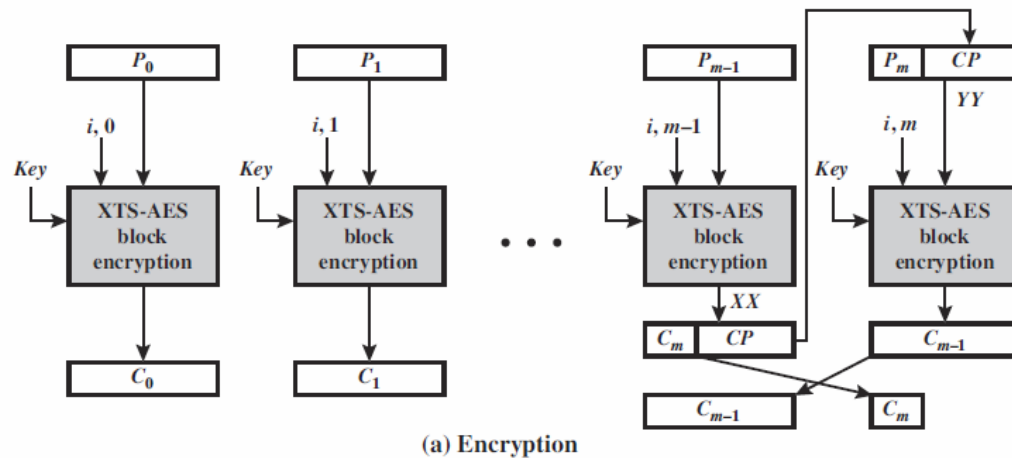


(b) Decryption

# Xor-Cifrado-Xor (XTS-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 sí (sectores)
  - El cifrado se realiza en bloques de 16 bytes y de forma independiente a los otros bloques (acceso aleatorio)
  - No es necesario más metadatos 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

# Modo XTS-AES



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

## Cuestiones

- *Demostración de rendimiento.*
- ¿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?