Fundamentos de criptografía
Criptografía simétrica

Garantía y Seguridad de la Información.

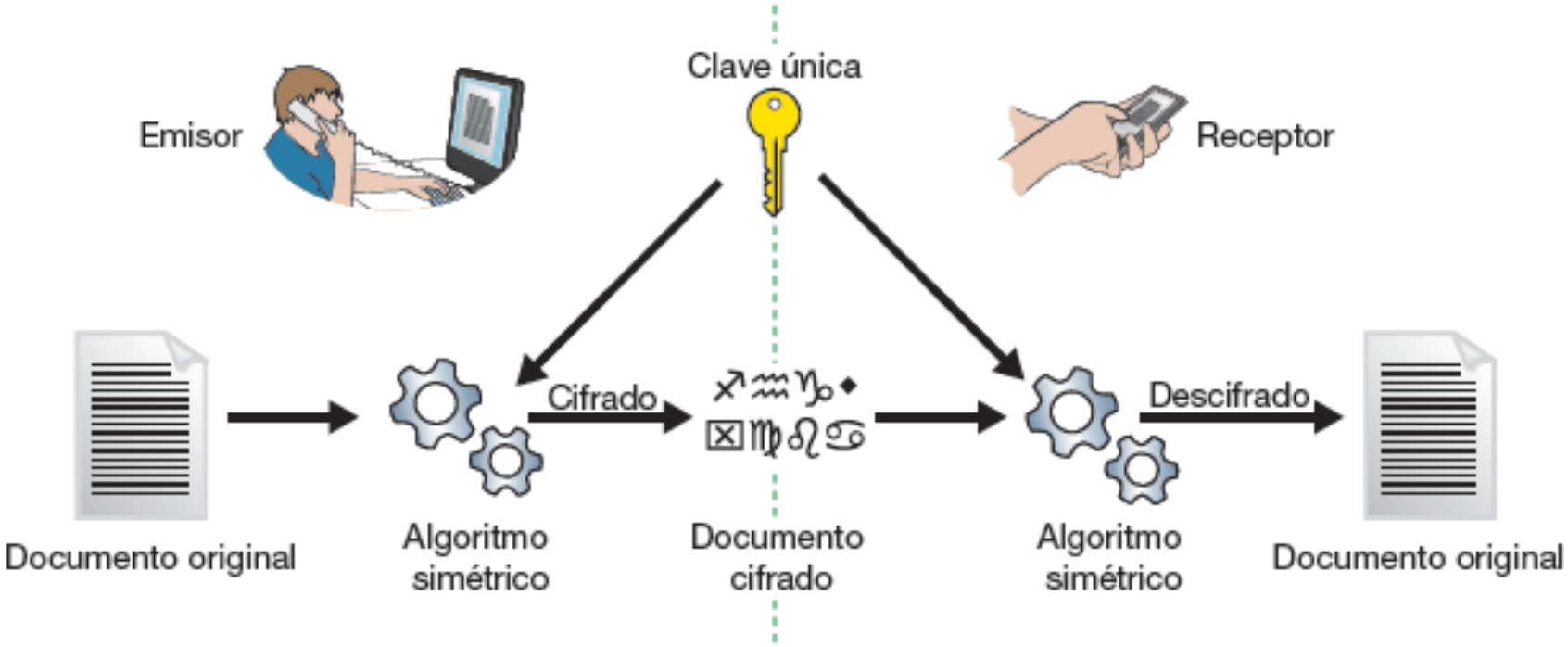
Requisitos de un criptosistema moderno

- El canal es público
  - Debemos asumir que el enemigo captura mensajes
- La seguridad debe ejercerse mediante la clave
  - Los algoritmos son públicos
  - El algoritmo será analizado por el criptoanalista
- El secreto de la comunicación se basa en la posesión de un secreto: la clave

Clases de transformaciones criptográficas

- Transformaciones **reversibles**:
  - resulta **computacionalmente** muy costoso deducir el mensaje original salvo si se dispone de la clave.
  - algoritmos simétricos.
- Transformaciones **irreversibles**:
  - a partir del mensaje original se produce otro, pero resulta **imposible** conocer el original.
  - mediante funciones especializadas como los algoritmos Hash.

Algoritmos simétricos



Algoritmos simétricos

- Utilizan la **misma clave** para cifrar y descifrar.
- El funcionamiento es simple:
  - el emisor toma un documento y le aplica el algoritmo, usando la clave única, que también conoce el receptor.
  - el receptor recibe el documento cifrado, le aplica el mismo algoritmo con la misma clave, pero ahora en función de descifrar.
- Si el documento cifrado no ha sido alterado en el camino y la clave es la misma, el resultado será el documento original
- Son sencillos de utilizar y resultan bastante **eficientes**

Algoritmos simétricos

- ¿Cómo se transforma el texto en claro en texto cifrado?
  - sustitución, cada elemento del texto claro (bit, letra, grupo de bits o letras) se sustituye por otro diferente
  - transposición, los elementos del texto claro se reordenan
    - todas las operaciones deben ser reversibles

Confusión

Difusión

Algoritmos simétricos

- ¿Cómo se procesa el texto en claro?
  - un bloque de elementos cada vez, produciendo un bloque de salida por cada bloque de entrada.
  - procesa los elementos de entrada continuamente, produciendo la salida de un elemento cada vez.

Cifrado de bloque

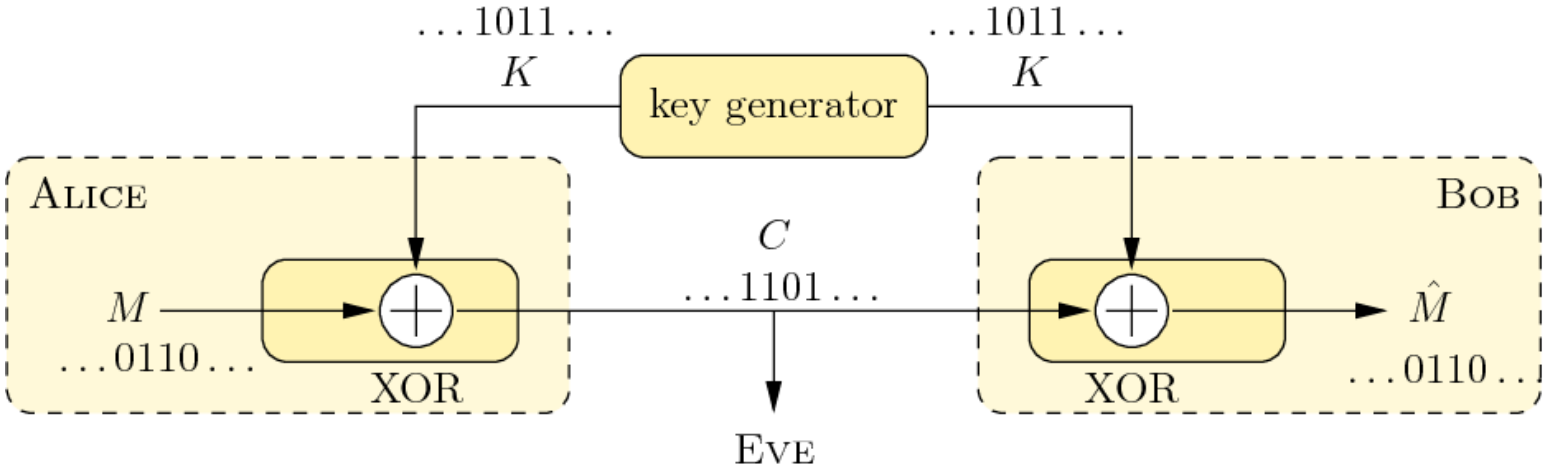
Cifrado de flujo

Cifrado de flujo

- El cifrado de todo el mensaje se realiza bit a bit (carácter a carácter).
- Se utilizan claves muy largas que sirven tanto para cifrar como para descifrar.
- Estas claves pueden
  - estar predeterminadas (One-Time Pad (OTP), Libreta de un solo uso (LSU))
  - generarse usando un generador de claves pseudoaleatorias

Cifrado de Vernam (OTP-LSU)

- C = M⊕K, donde K tiene tantos bits como C y M
- M = C⊕K
- La clave sólo sirve para una vez.



Cifrado de Vernam

- El mensaje, M
  - XOR
- La clave, K: *flujo de datos aleatorio o pseudoaleatorio del mismo tamaño*
- Texto cifrado, C.

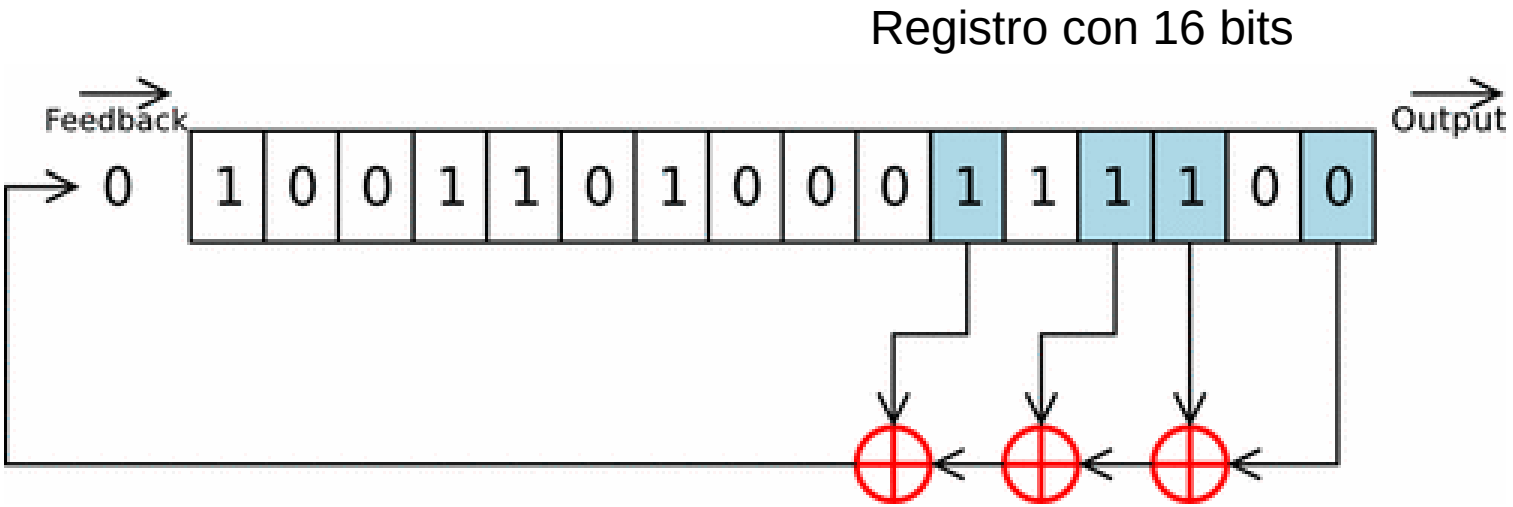
- Si la secuencia con la que se mezcla el mensaje es realmente aleatoria y se usa sólo una vez, es un *secreto perfecto*.
  - Deberemos tener un canal seguro de distribución para claves arbitrariamente largas

Cifrado de Vernam

- Es el mejor sistema siempre que:
  - Podamos tener una clave K completamente impredecible (aleatoria)
  - No la usemos más de una vez por cada encriptación
  - Tengamos un canal seguro de distribución para claves arbitrariamente largas
- En la práctica:
  - utilizamos OTP con K pseudoaleatoria.

OTP con K pseudoaleatoria

- Una clave fiable
  - utilizar un **LFSR** (Linear Feedback Shift Register), o Registro de Desplazamiento con Retroalimentación lineal.



https://upload.wikimedia.org/wikipedia/commons/9/99/Lfsr.gif

OTP con K pseudoaleatoria

- Registro de Desplazamiento con Retroalimentación lineal (LFSR)
  - Conjunto de bits que tienen la capacidad de moverse a través de las celdas en las que residen.
  - Al hacerlo, **el último bit sale del registro** y pasará a formar parte de la clave.
  - La primera celda, **que se quedaría vacía**, se rellena con un bit que es el resultado de una **operación lógica** con ciertos bits residentes en el registro.
    - periodicidad del registro
    - semilla

Cifrado de flujo. Ejemplo

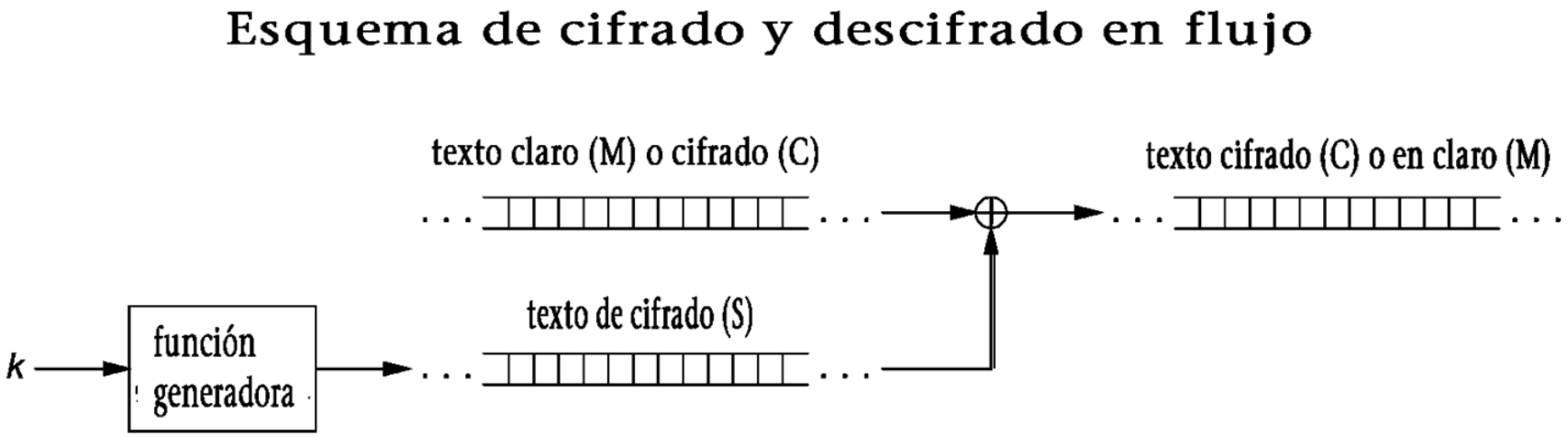
- Texto en claro 1111 1111 0011 101…
- Generador de flujo de clave
  - $s_{i+4} = (s_i + s_{i+1}) \; mod \; 2, i \geq 1$
  - La semilla es k = 1000

- Calcular el flujo de clave.

  1000 1001 1010 111…
- Calcular el criptograma.

  0111 0110 1001 010…

Cifrado de flujo. Fortaleza



El registro tiene 2^n estados posibles. La secuencia clave generada tiene un periodo máximo de 2^n-1 bits.

RC4 (Ron´s Code 4)

- Cifrado de flujo.
- Diseñado por Ronald Rivest en 1987 y publicado en Internet por un remitente anónimo en 1994.
- El sistema de protección WEP (*Wired Equivalent Privacy*) que incorpora el estándar IEEE 802.11 para tecnología LAN inalámbrica utiliza este criptosistema de cifrado en flujo.
- También incorporado en TLS/SSL

RC4 (Ron´s Code 4)

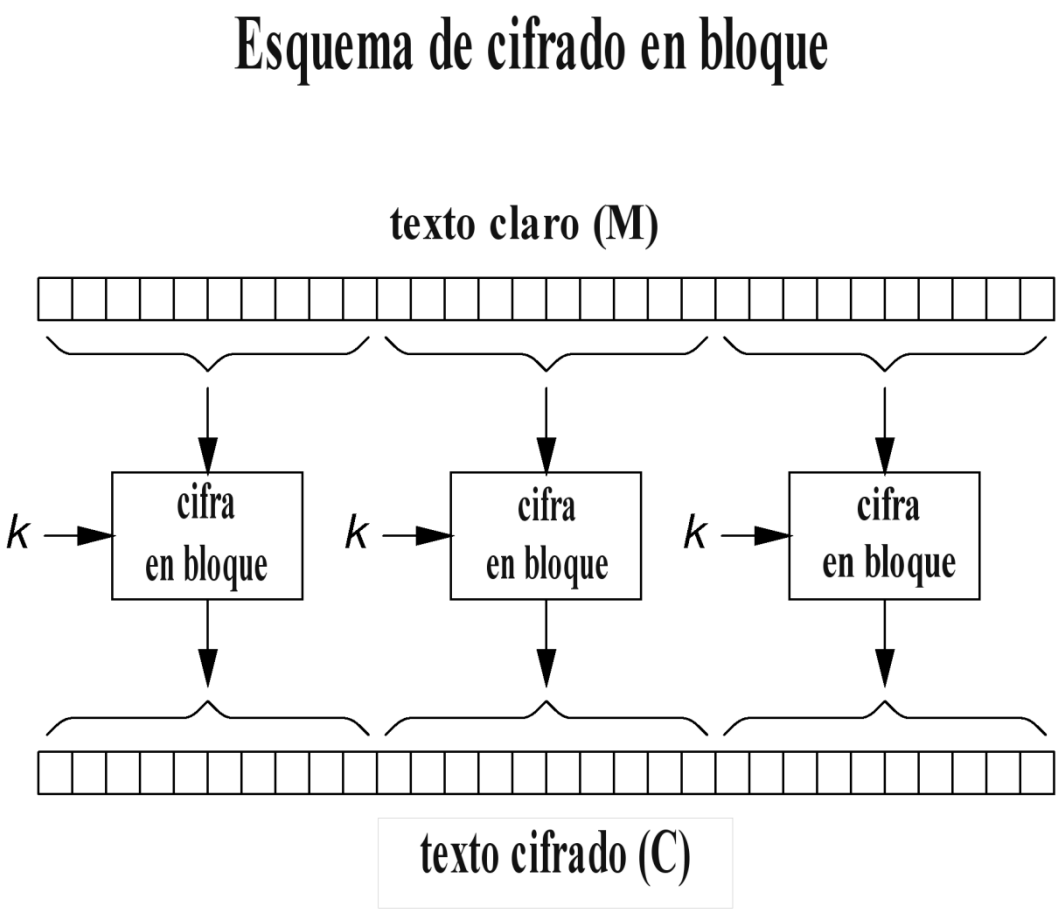
- Genera un flujo pseudo-aleatorio de bits que se combina con el texto plano
  - usando la función XOR
- Para generar el flujo, el algoritmo tiene un estado interno secreto y utiliza dos funciones.
  - Un algoritmo para las claves
    - Key scheduling algorithm o KSA
  - Un algoritmo de generación pseudo-aleatoria
    - Pseudo-random generation algorithm o PRGA

Ver: ¿Cómo funciona el algoritmo RC4?

Cifrado en bloque

- El texto en claro de *L* bits se divide en bloques de *b* bits.
- Se cifra cada bloque.
  - Si *L* no es múltiple de *b*, se agregan bits adicionales
    - indicar cuántos bits había realmente en el mensaje original
    - el último byte del último bloque, número de bites que se han añadido
- El descifrado se realiza bloque a bloque.

Cifrado en bloque

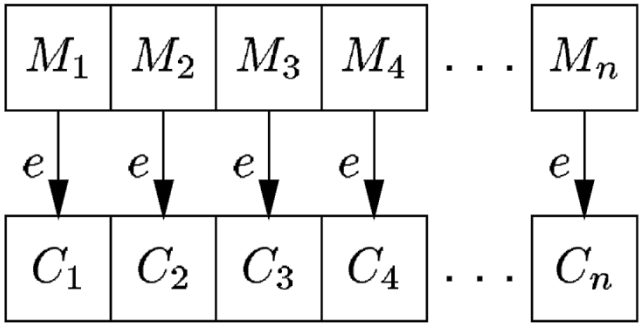


Cifrado en bloque

- Modos de operación más habituales son:
  - Modo "Electronic Codebook" (ECB), para bloques aislados de tamaño pequeño.
  - Modo "Cipher Block Chaining" (CBC), para streams de datos.
  - Modo "Cipher Feedback" (CFB), para streams de texto.
  - Modo "Ouput Feedback" (OFB), para bloques aislados, pero que se suceden continuadamente.
  - Modo "CTR", igual que OFB.

Modo ECB (Electronic Codebook)

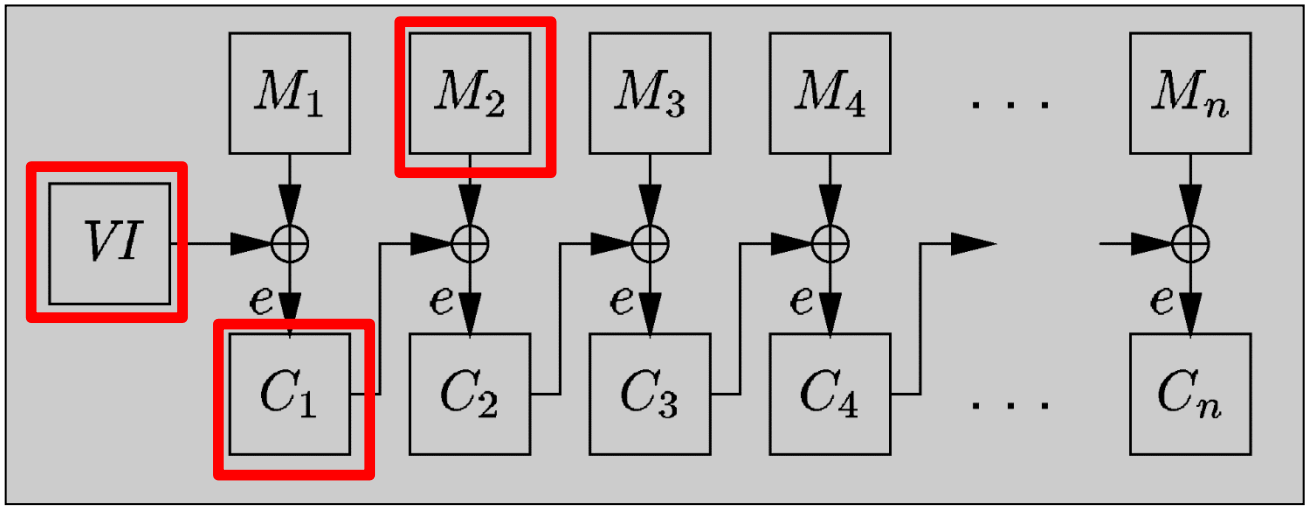
- Subdivide el texto a codificar en bloques del tamaño fijo y se cifran todos ellos empleando la misma clave.



- Ventajas:
  - Permite codificar bloques independientemente de su orden.
- Desventajas:
  - Si el mensaje presenta patrones que se repiten, el texto cifrado también.
  - Puede sufrir una sustitución de bloques.

Modo CBC (Cipher Block Chaining)

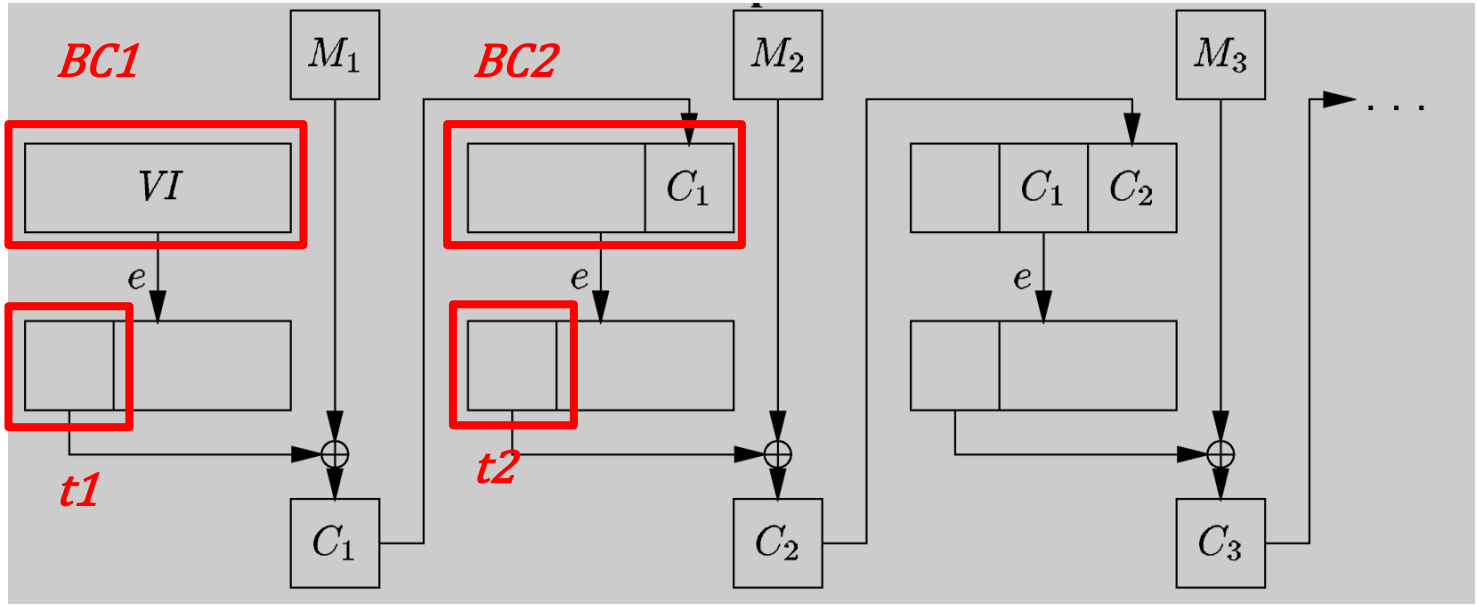
- La entrada al algoritmo de cifrado es (el bloque de texto plano a cifrar) XOR (bloque de texto cifrado precedente).
- Con un vector de inicialización (VI) conocido por ambos.



- Ventaja:
  - Protege respecto a la sustitución de bloques.
- Con un error en la transmisión de C_j, perdemos M_j y M_{j+1}

Modo CFB (Cipher Feedback)

- El algoritmo de cifrado se usa para generar una secuencia de **bloques** de **clave** (BC), partiendo de un VI.
- El texto plano se cifra haciendo un XOR entre el bloque de texto plano y el bloque de clave generado.



Modo CFB (Cipher Feedback)

- *VI*, texto plano M (u bloques de longitud r)
- Para cifrar cada bloque

  $BC_1 = VI$

  Para $1 \leq j \leq u$ hacer:

  $O_j = e(BCj)$ *encriptamos BCj

  Sea tj, cadena con los r bits más significativos de O_j

  $C_j = Mj \oplus tj$

  BC_{j+1} = eliminar los primeros r bits de BC_j y sumar C_j

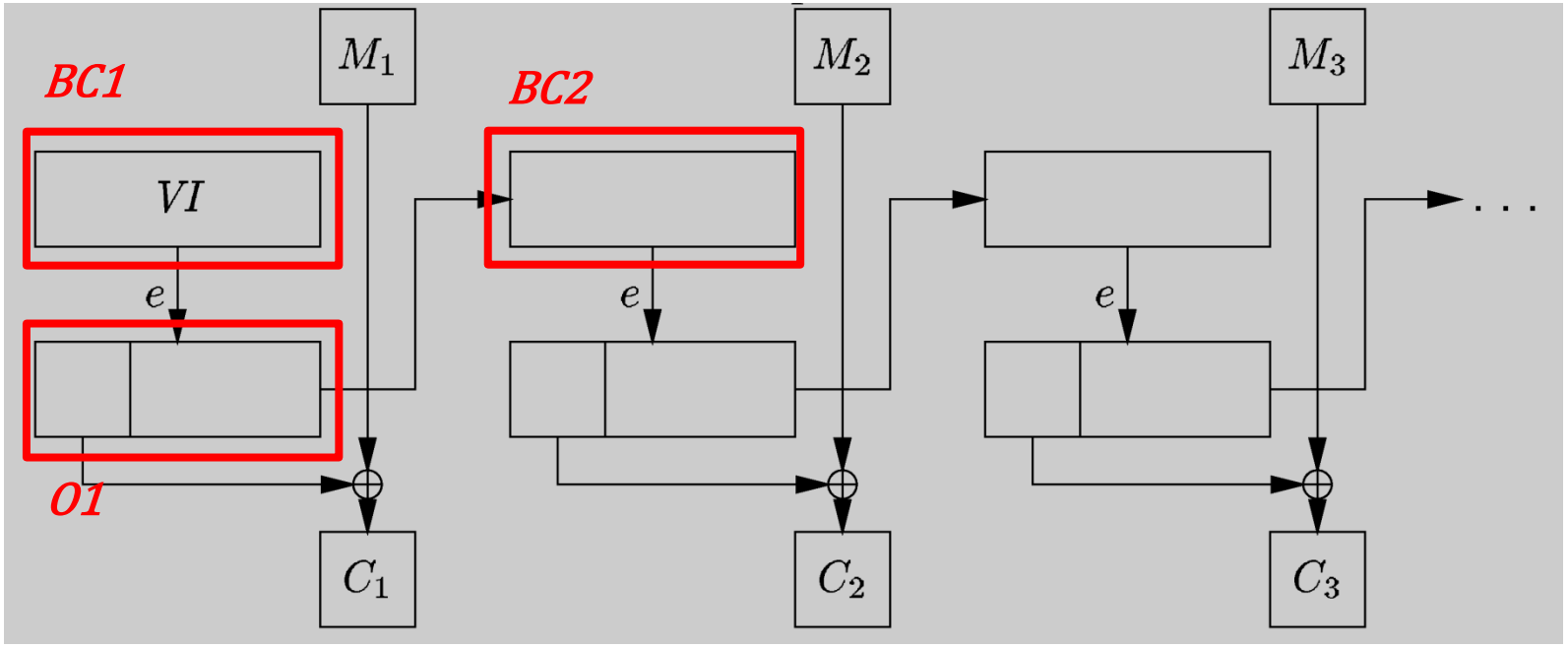
- El texto cifrado queda formado por la secuencia $C_1, C_2, \ldots C_u$

Modo CFB (Cipher Feedback)

- La transmisión es más rápida.
- Hay que usar el algoritmo de cifrado muchas más veces
  - los bloques de M son mas pequeños
- El tamaño del bloque, r, es un valor de conveniencia entre las velocidades de computación y de transmisión.
- Un error en un sub-bloque estropea la labor de descifrado mientras ese tramo forma parte del vector BC_j

Modo OFB (Output Feedback)

- Opera como el CFB pero el vector auxiliar se actualiza con el resultado obtenido del algoritmo de cifrado.
- El texto cifrado no se utiliza



Modo OFB (Output Feedback)

- *VI*, texto plano M (u bloques de longitud r)
- Para cifrar cada bloque

  $BC_1 = VI$

  Para $1 \leq j \leq u$ hacer:

  $O_j = e(BCj)$ *encriptamos BCj

  Sea tj, cadena con los r bits más significativos de O_j

  $C_j = Mj \oplus tj$

  $BC_{j+1} = O_j$

- El texto cifrado queda formado por la secuencia $C_1, C_2, \ldots C_u$

Modo OFB (Output Feedback)

- No se propagan los errores de transmisión
  - si ocurre un error en un bit de C_1, ese error sólo afecta al valor de M_1.
- La manipulación del texto cifrado resulta más sencilla que en el modo CFB

Elección de un modo

- ECB es el más simple, sencillo y rápido.
- Es el más débil: vulnerable a ataques de repetición y criptoanálisis.
  - Sólo recomendable para encriptar datos breves, como las claves criptográficas.
- Para el texto en claro es mejor CBC, CFB u OFB:
  - CBC es el mejor para archivos
  - CFB es la mejor elección para encriptar cadenas de caracteres donde se envía un carácter cada vez (por ejemplo en aplicaciones tipo terminal).
  - OFB se usa en sistemas síncronos de alta velocidad donde no puede haber propagación de errores.

Algoritmos simétrico de bloques

- Maximizar los dos conceptos de Shannon, confusión y difusión
  - Minimizar el conocimiento del atacante sobre las propiedades estadísticas del texto.
  - Manipular el texto para que sea ininteligible e irreversible.

Algoritmos simétricos de bloques

- Criterios para alcanzar difusión y confusión
  - Operación or-exclusiva (50% de tener 0 ó 1)
  - Operación de desplazamiento o rotación (trasposición)
  - Operaciones de *no-linealidad*
  - Mezcla de las operaciones anteriores
  - Número de repeticiones del conjunto de operaciones

Or-exclusiva, desplazamiento o rotación

- Sencillez, rapidez en la operación
- 50% de probabilidad de que la entrada sea 0 ó 1
- Mecanismos de trasposición de bits

No-linealidad. S-Box

- Toma *m* de bits de entrada y los transforma en *n* bits de salida
- Se construye una tabla en la que dada una entrada se obtiene una salida
  - puede provenir de varias entradas

- Elemento fundamental para la seguridad del algoritmo
- Basada en funciones matemáticas no lineales

Box 6x4: S_1 en DES

Tenemos 6 bits de entrada: 2 bits externos, 4 bits internos.
Con 2 bits, disponemos de 4 filas.
Con 4 bits, disponemos de 16 columnas.
En cada fila se coloca una posible permutación de 0-15 (4 bits).

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	6	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Los bits externos (1 y 6) de la entrada se usan para seleccionar la fila.
Los 4 bits internos de la entrada para seleccionar la columna.

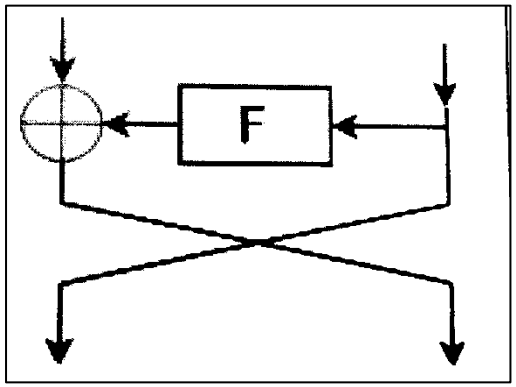
S_1 (101010) = 6 = 0110

No-linealidad. S-Box 6x4. S_5

S		4 bit de entrada internos														
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Bits externos	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1100	0011	1001	1000	0110
	10	0100	0010	0001	1011	1100	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1100	0100	0101	0011

Los bits externos (1 y 6) de la entrada se usan para seleccionar la fila.
Los 4 bits internos de la entrada para seleccionar la columna.
La entrada "**0**11011**1**" tiene como salida "1001".

Estructura de mezcla. Red Feistel



- Se parte un bloque de N bits en dos.
- La parte derecha sale como nueva parte izquierda.
- La nueva parte derecha se obtiene:
  - Hacer or-exclusivo de la entrada izquierda, con la parte derecha modificada según una función F.

Estructura de mezcla. Red Feistel

- F realiza funciones de no-linealidad, desplazamientos, or-exclusivos, … para conseguir *confusión* y *difusión*
- Y para conseguir *seguridad*, se repiten estas estructuras de operaciones varias veces.

- *Seguridad condicional: seguro frente a los ataques conocidos*
- *Seguridad computacional: resistente a la fuerza bruta*

Algoritmos simétricos de bloques

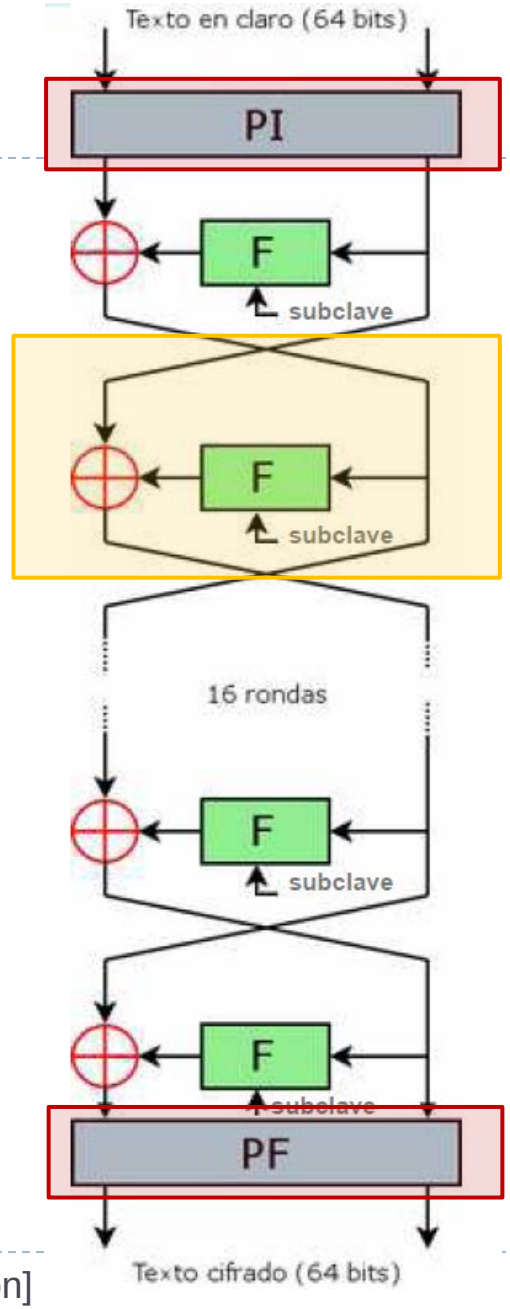
- DES (Digital Encryption Standard).
  - Primer algoritmo simétrico de uso comercial (1976). Deja de ser un estándar público en 2005.
- Triple DES (3DES)
  - Reemplazo para DES. Utiliza una clave de 112 ó de 168 bits.
- AES (Advanced Encryption Algorithm).
  - Reemplazo oficial para DES. Utiliza claves de 128, 196 o 256 bit.
- IDEA (International Data Encryption Algorithm).
  - Es de libre distribución y no comercial.

Algoritmo DES

- Algoritmo de cifrado en bloques
- Cifra bloques de 64 bits utilizando una clave de 64 bits
  - 8 bytes con paridad, o sea 7 bytes útiles (56 bits)
- Produce bloques cifrados de 64 bits.
- Realiza 16 rondas (iteraciones)
  - En cada ronda utiliza una clave "nueva" generada a partir de la clave inicial.

Esquema general

- Existe una permutación inicial y una final del texto
- Hay 16 rondas que
  - Divide el bloque en dos mitades
  - Pasa una mitad por una función F en una red Feistel
  - Mezcla ambas mitades con un XOR
  - Rota ambas mitades.



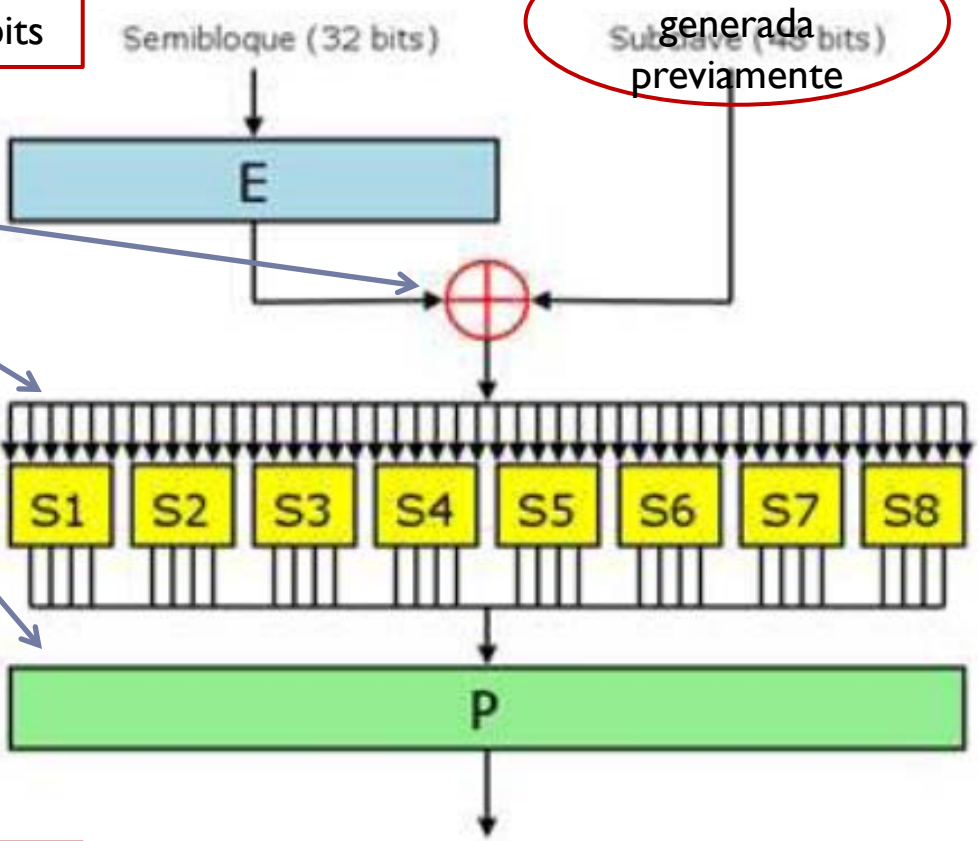
Función F

Trabaja con bloques de 32 bits y consta de 4 pasos:

- *Expansión*: el bloque de 32 bits se expande a 48 bits
  - *permutación de expansión (*duplica algunos bits)
- *Mezcla*: el resultado se combina con una *subclave*
  - mediante XOR.
- *Sustitución*: se hacen ocho trozos de 6 bits y se procesan en las *S-Box*
  - ocho S-Box con 6 bits de entrada y 4 bits de salida (*trasformación no lineal*)
- *Permutación*: los 32 bits de salida se reordenan (permutación fija, P-Box)

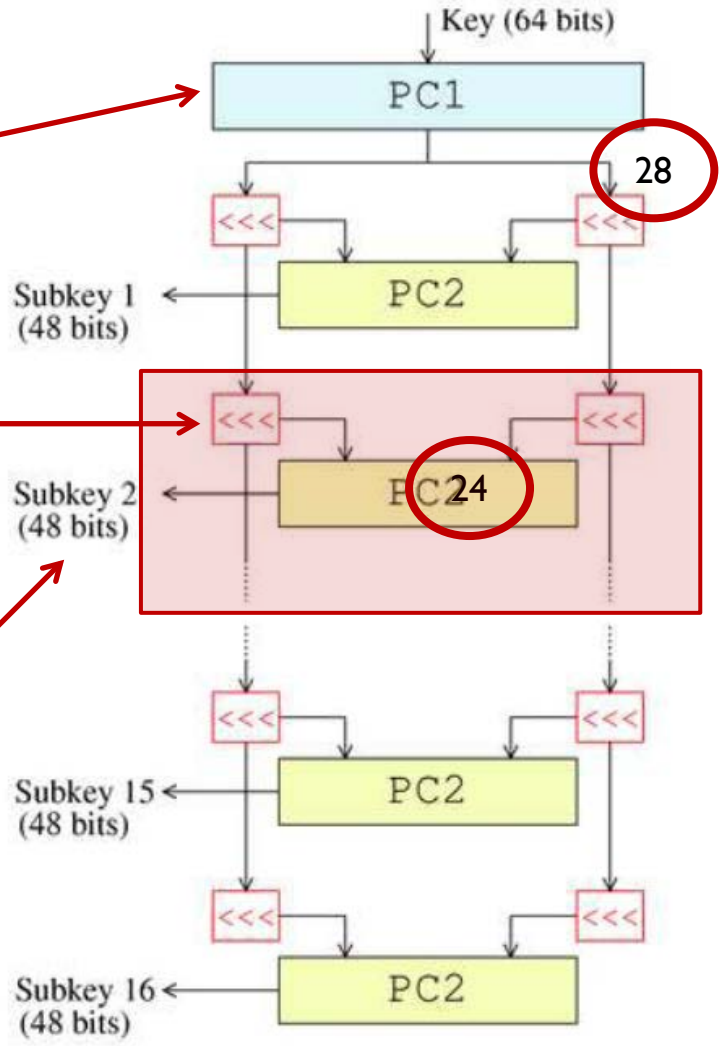
Función F

- *Expansión*
- *Mezcla*
- *Sustitución*
- *Permutación*

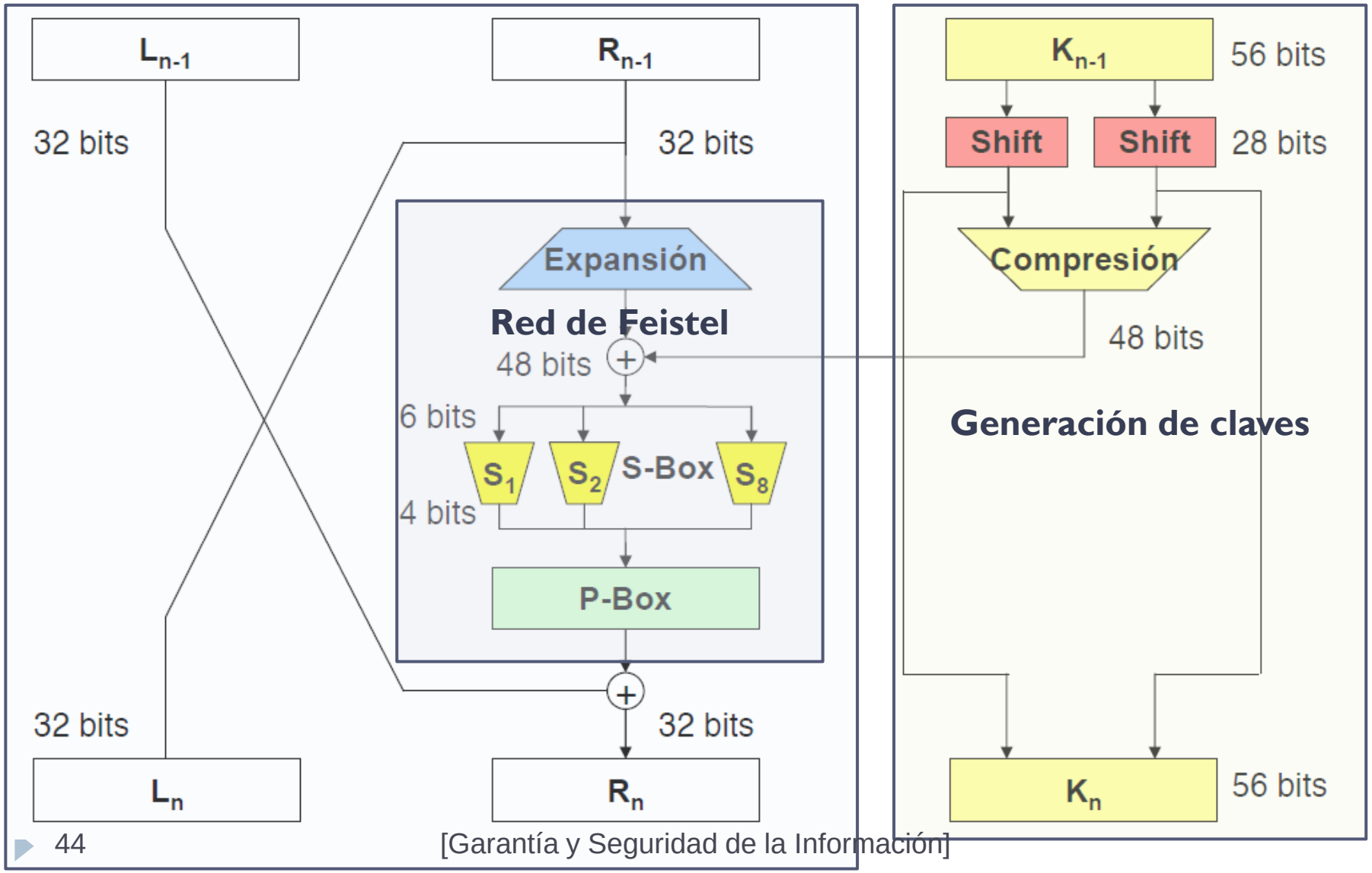


Generación de subclaves

- En *PC1* se permutan y eligen 56 bits de la clave inicial.
  - los 8 bits restantes se descartan o se usan como paridad.
- Los 56 bits se dividen en dos mitades de 28 bits.
- En cada ronda,
  - cada mitad se desplaza hacia la izquierda uno o dos bits
  - en PC2
    - se permutan las mitades
    - se seleccionan 24 bits de cada mitad
  - La subclave tiene 48 bits.



Funcionamiento



Efecto avalancha en DES

- Un pequeño cambio en el texto a cifrar o en la clave produce un cambio significativo en el texto cifrado
- Es una evidencia del alto grado de difusión y confusión.
  - Un cambio de 1 bit en el texto a cifrar afecta a una media de 34 bits en el texto cifrado.
  - Un cambio en 1 bit de la clave afecta a una media de 35 bits en el texto cifrado.

Algoritmo DES

- Para descifrar basta con usar el mismo algoritmo empleando el orden inverso.
- *Ventajas* del algoritmo:
  - Es muy rápido y fácil de implementar.
- *Desventajas*:
  - La clave es de 56 bits
  - Demasiado corta. Se pueden llevar a cabo ataques por fuerza bruta.

Ver: ¿Cómo funcionan los algoritmos DES y 3DES?

Variantes del DES

- Se han propuesto variantes.
- DES Múltiple:
  - aplicar varias veces el algoritmo DES con diferentes claves al mensaje original.
- El más común de todos ellos es el *Triple-DES*.
  - Tres veces, tres claves distintas.
  - Clave de 168 bits
  - Cifra bloques de 64 bits

Algoritmo de Rijndael (AES)

- En 1997, el NIST convocó al desarrollo de un nuevo algoritmo con los requisitos:
  - Algoritmo de cifrado simétrico.
  - Algoritmo de cifrado en bloques.
  - Manejo de bloques de 128 bits (16 bytes)
  - Soporte de manejo de claves de diferente longitud.
  - Claves de 128, 192 y 256 bits.
- Se aceptó en 2001, para sustituir al DES.

AES

- Opera con bloques y claves de longitudes variables
  - 128, 192 ó 256 bits
  - Puede extenderse a múltiplos de 32 bits
    - Más fortaleza, más uso futuro.
- Permite una implementación eficiente y rápida en software y hardware.
  - adecuado para la casi totalidad de las aplicaciones actuales

Ver: ¿Cómo se cifra con el algoritmo AES?