Criptografía y Seguridad en Redes

v.2015

Seminario #3

Prof.Ing.Miguel SOLINAS

mksolinas@gmail.com

Agenda

- *Introducción*
- *Criptografía histórica*
- *Fundamentos teóricos*
- *Cifra de Bloque Moderna*

Clasificación Criptosistemas Clásicos

SUSTITUCIÓN

Monoalfabética

- Monográfica
  - Alfabeto Estándar (Cesar)
  - Alfabetó Mixto (Otros)
- Poligráfica
  - Diagrámica (Playfair)
  - N-Grámica (Hill)

Polialfabética

- No Periódica (Vernam)
- Periódica
  - Lineales
    - Alfabeto Estándar (Vigenere)
    - Alfabeto Mixto (Otros)
  - Progresivos (Enigma)

TRANSPOSICIÓN

- GRUPOS (Escítalo)
- Series
- Filas
- Columnas

Método de Cifra

- De Flujo (RC5)
- De Bloque
  - Clave Secreta
    DES; IDEA; AES
  - Clave Pública
    - Exponenciación (RSA)
    - Suma / Producto (Curvas Elípticas)

Cifra de Bloque Moderna

Una gran parte de los algoritmos de cifra simétrica operan dividiendo el mensaje, que se pretende codificar, en bloques de tamaño fijo y aplican sobre cada bloque una combinación más o menos compleja de operaciones de confusión — sustituciones — y difusión — transposiciones —.

Estos algoritmos se denominan, en general, cifrados por bloques.

Cifra de Bloque Moderna

Motivación para esta estructura

Una cifra de bloque opera sobre un bloque de texto plano de n *bits y produce un bloque de cifra de* n *bits.*

Hay 2^n *posibles bloques de texto plano diferentes, cada uno produciendo un bloque de cifra único.*

Tal transformación la llamamos reversible o no singular.

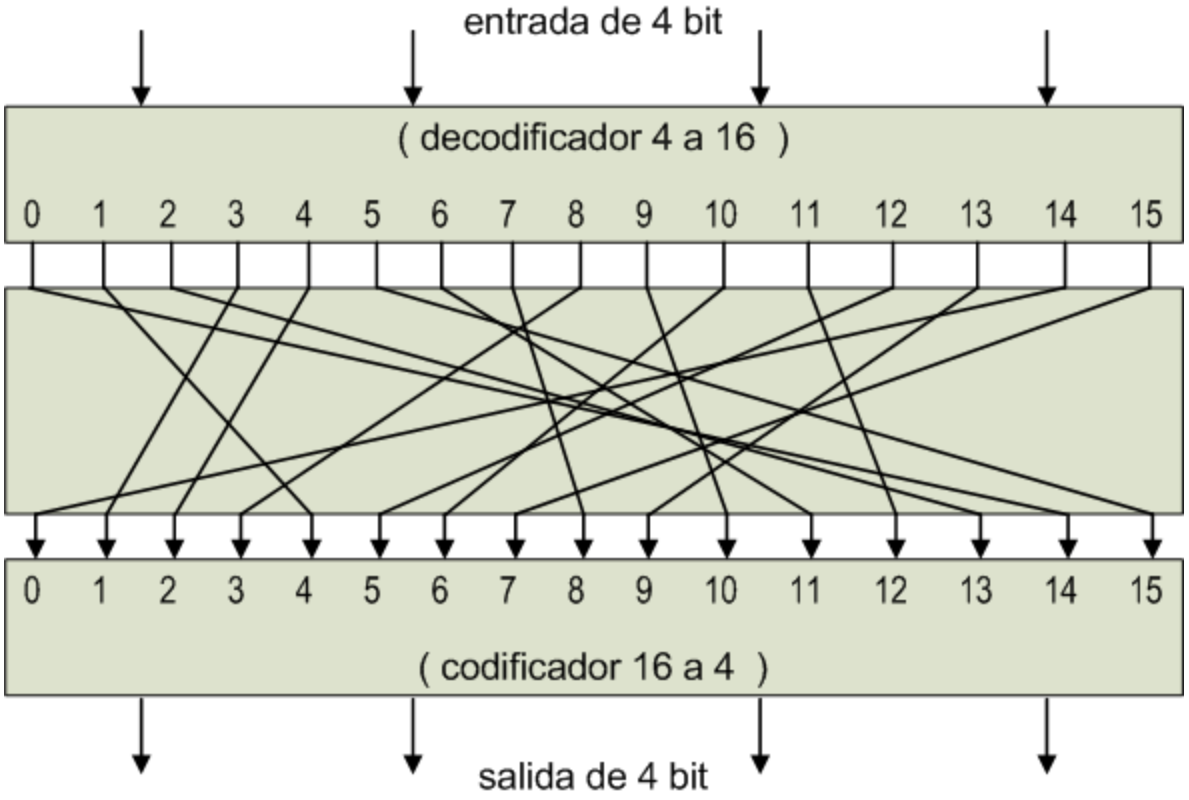
Ejemplo de mapeos para $n = 2$*:*
Cuántas transformaciones reversibles diferentes tenemos..?

$$2^n! = 24$$

Reversible		Irreversible	
T.Plano	Cifra	T.Plano	Cifra
00	11	00	11
01	10	01	10
10	00	10	01
11	01	11	01

Cifra de Bloque Moderna

Ejercicio: ¿ cuántos bits se necesita para codificar la clave en este caso ?



Cifra de Bloque Moderna

*En general para una cifra de bloque, de sustitución, de **n** bits voy a necesitar **$nx2^n$** bits para expresar la clave..!*

$$n = 64$$

$$n2^n = 64 \cdot 2^{64} = 2^{70} \approx 10^{21} \text{bits..!}$$

En consideración de estas dificultades FEISTEL diseñó su cifra..!

Cifra de Bloque Moderna

Established 1845 **SCIENTIFIC AMERICAN** May 1973 Volume 228 Number 5

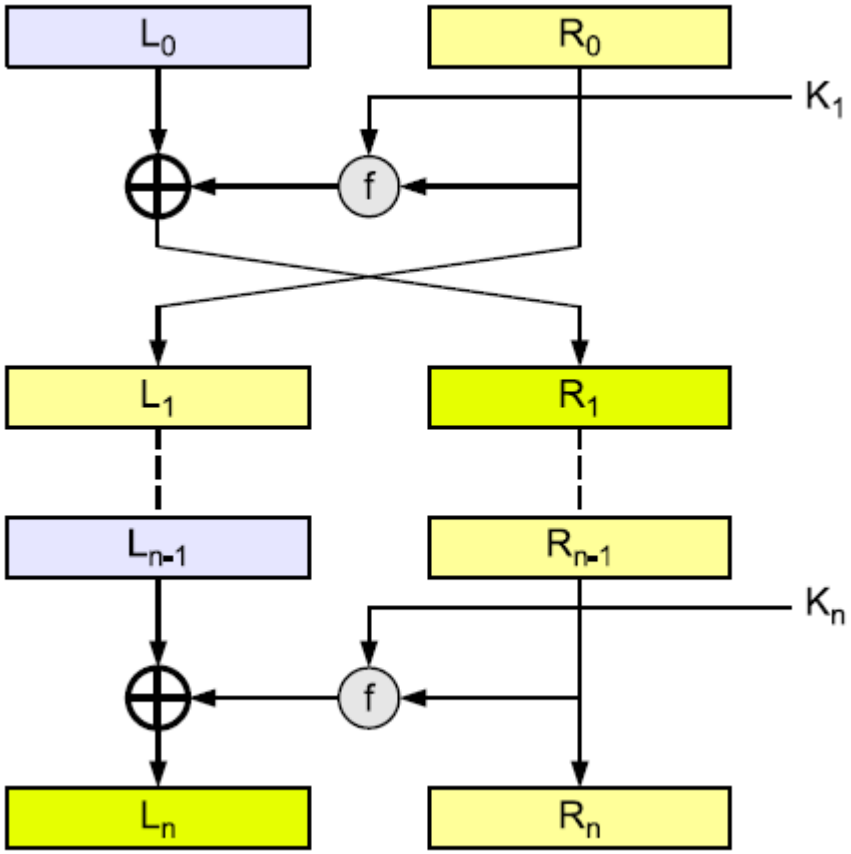
Cryptography and Computer Privacy

Computer systems in general and personal "data banks" in particular need protection. This can be achieved by enciphering all material and authenticating the legitimate origin of any command to the computer

by Horst Feistel

Cifra tipo Feistel

Estructura de una red de Feistel



Para $0 < i < n$

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

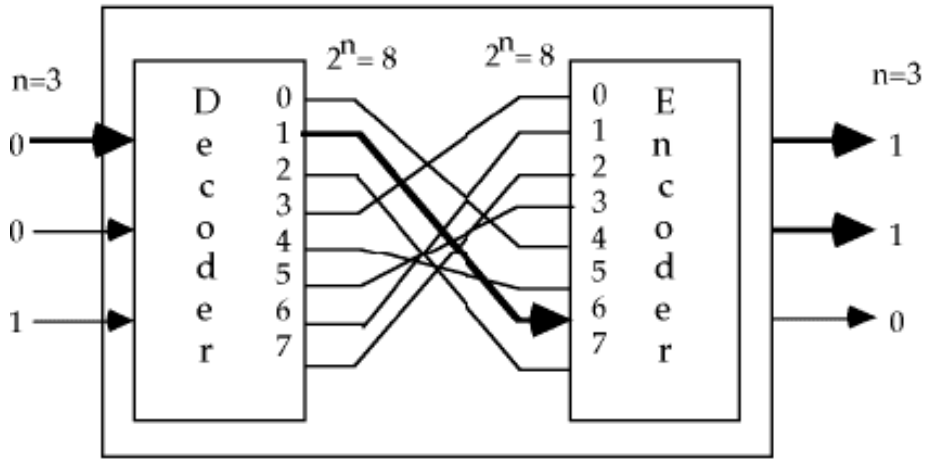
$$L_n = L_{n-1} \oplus f(R_{n-1}, K_n)$$

$$R_n = R_{n-1}$$

Cifra tipo Feistel

Estructura de una cifra Feistel (**S-box**)

- Una palabra binaria es reemplazada por alguna otra palabra binaria
- La función de sustitución forma la clave
- Para una palabra de ***n*** bits, la clave tiene 2^n y existen $2^n!$ transformaciones diferentes
- Por cada "***n***" bits, tenemos una salida



Cifra tipo Feistel

Estructura de una cifra Feistel (**S-box**)

Objetivo: efecto de confusión o avalancha (Webster & Tavares)

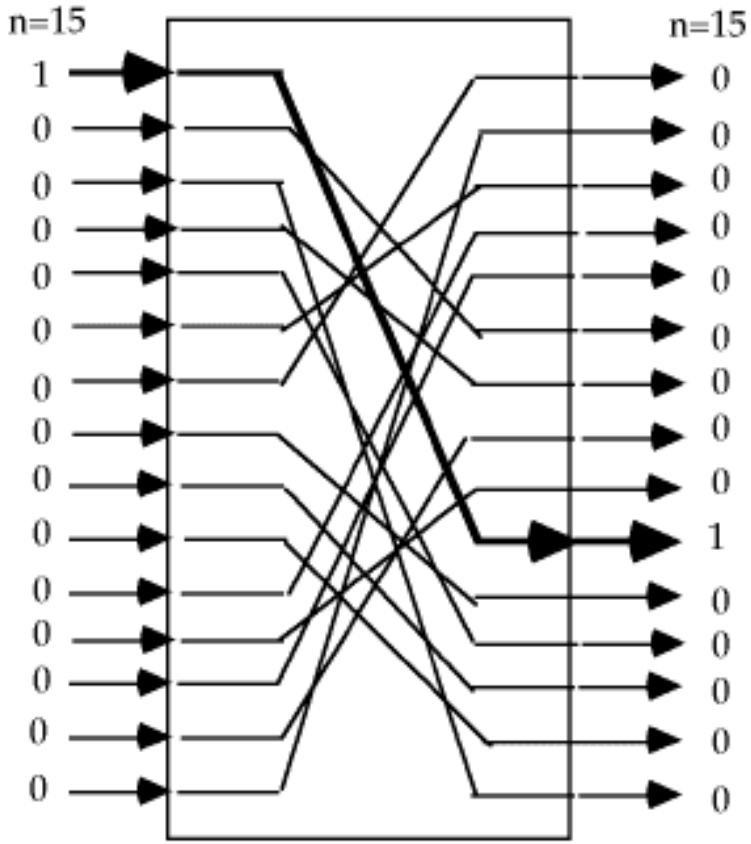
- El cambio en un bit en la entrada debe resultar en cambios en ≈ la mitad de los bits de salida. Cómo medir esto ?
- Se observa un bit de la entrada por vez y para todos los posibles valores del resto de bits de la entrada, se comparan las salidas creadas cuando ese bit cambia 0/1. El número de bits a la salida que difieren con la entrada, en promedio, deberían ser aproximadamente la mitad.
- Pequeños cambios en la entrada asegura grandes cambios en la salida.
- Lo cual significa que si un atacante cree estar próximo a descifrarla, el resultado no será tan cercano como supone

Ej.: DES usa bloques de 64 bits de entrada, cuando cambie tan sólo 1 bit, aproximadamente 32 bits a la salida deberían cambiar.

Cifra tipo Feistel

Estructura de una cifra Feistel (**P-box**)

- A una palabra binaria se le reordenan sus bits (permutan).
- La clave la forma el reordenamiento.
- Para n bits, existen n! claves.
- La relación crece mas lentamente y es menos segura que la sustitución.
- Es equivalente a un cruce de cables en la práctica.
- Mas duro de implementar en software.



Cifra tipo Feistel

Estructura de una cifra Feistel (**P-box**)

Objetivo: efecto difusión o integridad (Webster & Tavares)

- Cada bit a la salida es una función compleja de **todos** los bit de la entrada.
- Para “medir” la integridad, se observa un bit de la salida por vez. Entonces para cada bit a la entrada, encontrar un par de entradas, difiriendo sólo en el bit bajo estudio, las que resulten en un cambio en el bit de salida. Esto significará que todo bit de entrada tiene influencia sobre cada bit de salida.
- Asegura que cada bit a la salida depende de la mayor cantidad de bits a la entrada.
- De modo que un atacante está inhabilitado para “dividir y triunfar” sobre esta cifra.

Cifra tipo Feistel

Estructura de una cifra Feistel (**P-box & S-Box**)

Redes de sustitución & permutación (avalancha & integridad)

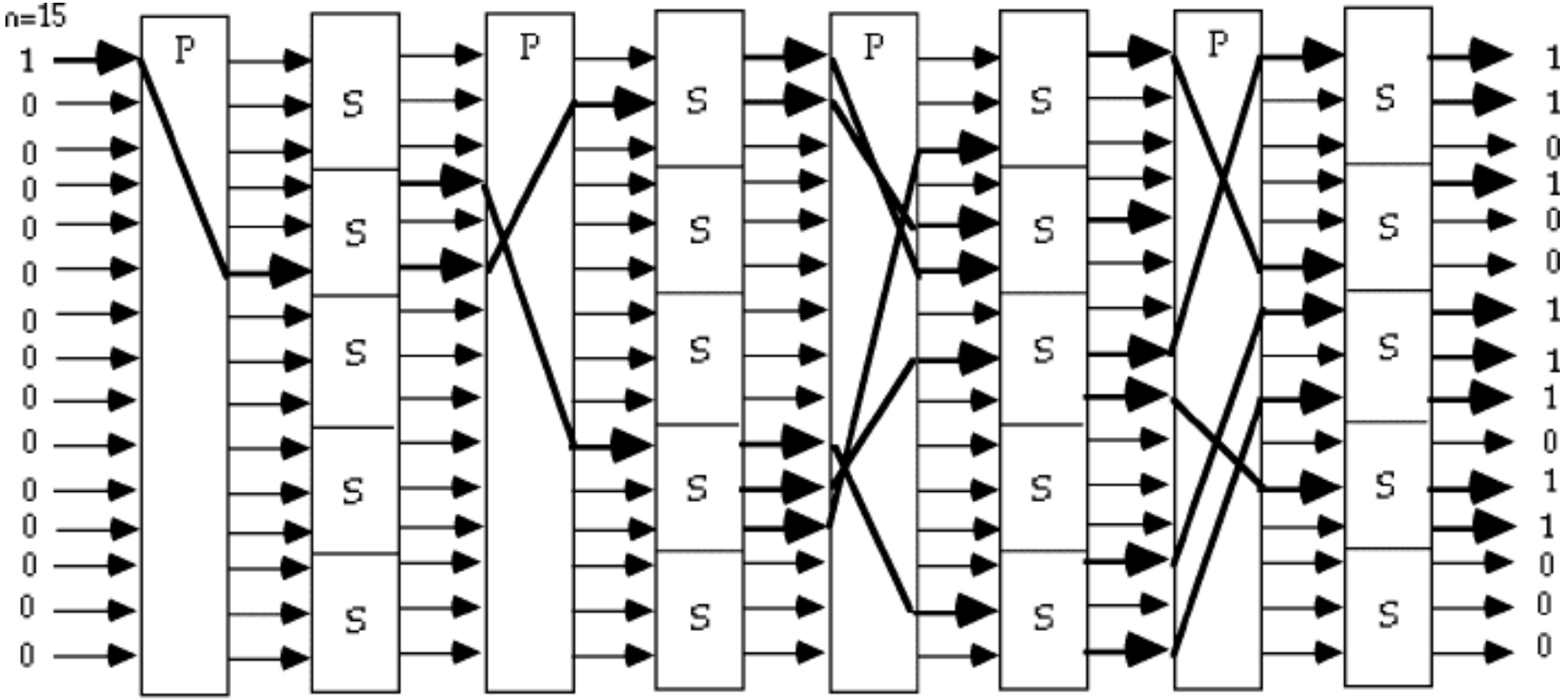


Fig 2.3 - Substitution-Permutation Network, with the Avalanche Characteristic

Cifra tipo Feistel

Parámetros a considerar en el diseño de la cifra Feistel

Tamaño de bloque: incrementando su tamaño se incrementa la seguridad pero ralentiza la cifra.

Tamaño de la clave: incrementando su tamaño mejora la seguridad, hace muy dura una búsqueda exhaustiva de clave, pero también ralentiza la cifra.

Número de rondas: incrementando su valor mejora la seguridad pero hace muy lenta la cifra.

Generación de sub-claves: mejora la complejidad y hace muy duro su criptoanálisis pero hace muy lenta la cifra.

Funciones de ronda: Ídem

Cifra tipo Feistel

Algunas generalidades sobre el diseño de la cifra

Un buen diseño logra efecto avalancha, integridad e impredecibilidad.

Un mal diseño pierde azarosidad y tiene mucho de predecibilidad.

Muchas cifras han sido quebradas, incluidos productos comerciales como WordPerfect, pkzip y cifras de teléfonos móviles actuales.

La razón: los expertos han perdido de vista estos criterios básicos.!!

Lo mejor es usar cifras testeadas y probadas a partir de la experiencia.

Cifra tipo Feistel

LUCIFER: 1er.ejemplo práctico, conocido, de cifra de S-P.

- Fue desarrollado por H.Feistel en los laboratorios de IBM
- H.Feistel, "***Cryptography and Computer Privacy***", Scientific American, Vol 228(5), May 1973, pp 15-23. Donde brinda una descripción superficial de su trabajo.
- La referencia mas detallada corresponde a Arthur Sorkin, "***Lucifer, A Cryptographic Algorithm***", Cryptologia, Vol 8(1), Jan 1984, pp 22-41, with addenda in Vol 8(3) pp260-261. Contiene una descripción detallada y su implementación en Fortran.

Cifra tipo Feistel

LUCIFER

- Cifra tipo Feistel con bloque de datos de 128 bit y clave de 128 bits..!!
- Subclaves de cada ronda son tomadas de la parte izquierda de la clave
- La clave es rotada a la izquierda 56 bits de modo de usar todos los bits de la clave

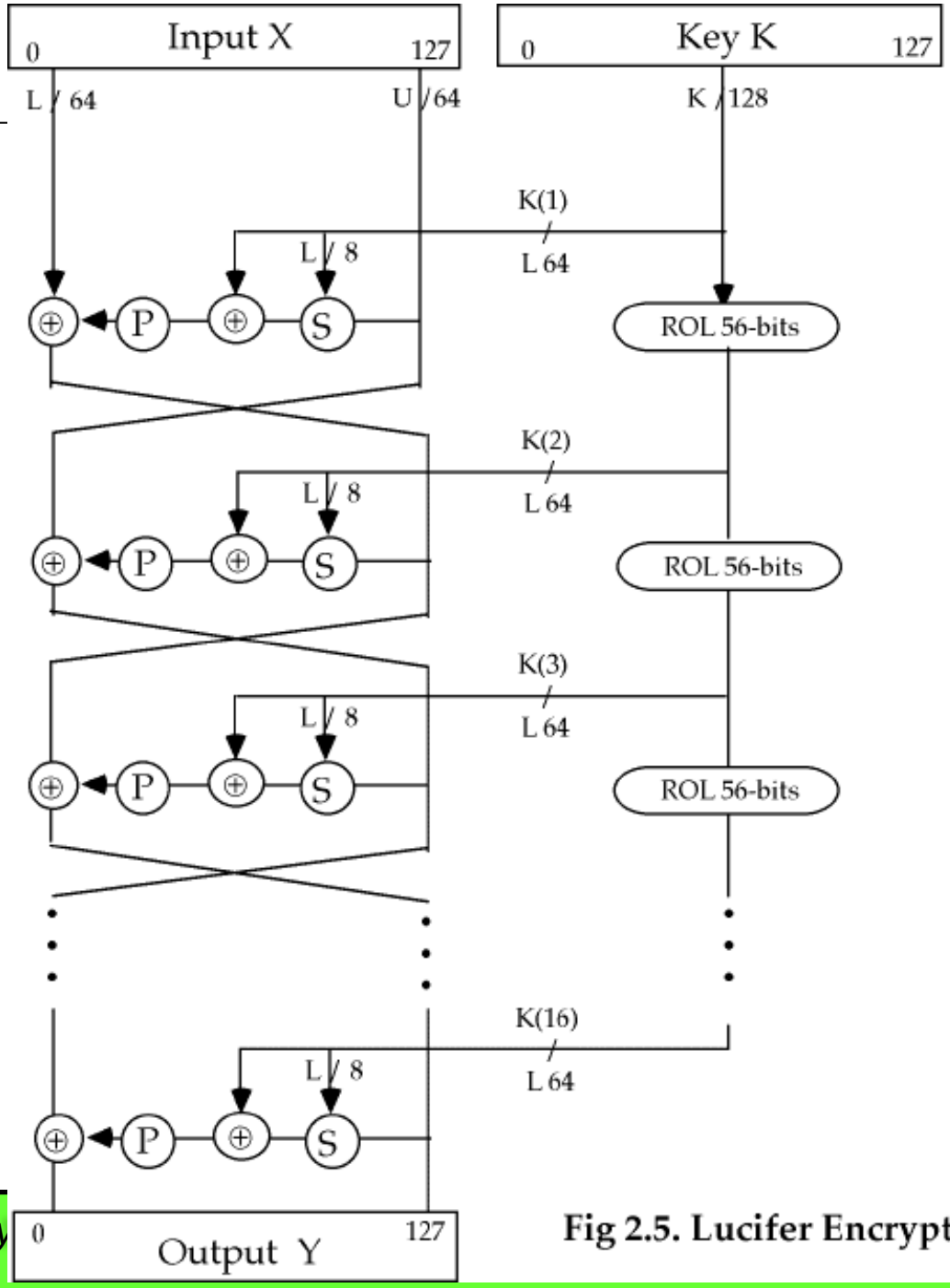


Fig 2.5. Lucifer Encryption

Cifra tipo Feistel

Cómputo de los datos en Lucifer

- Hay un proceso de cálculo en esta cifra que tiene 16 rondas, y usa la sub-clave y la mitad derecha de la entrada para alimentar a una función de ronda ***f.***
- Para luego XORear la salida con la mitad izquierda, previo intercambiar parte derecha e izquierda.

Estructura S-P
de la función ***f*** :

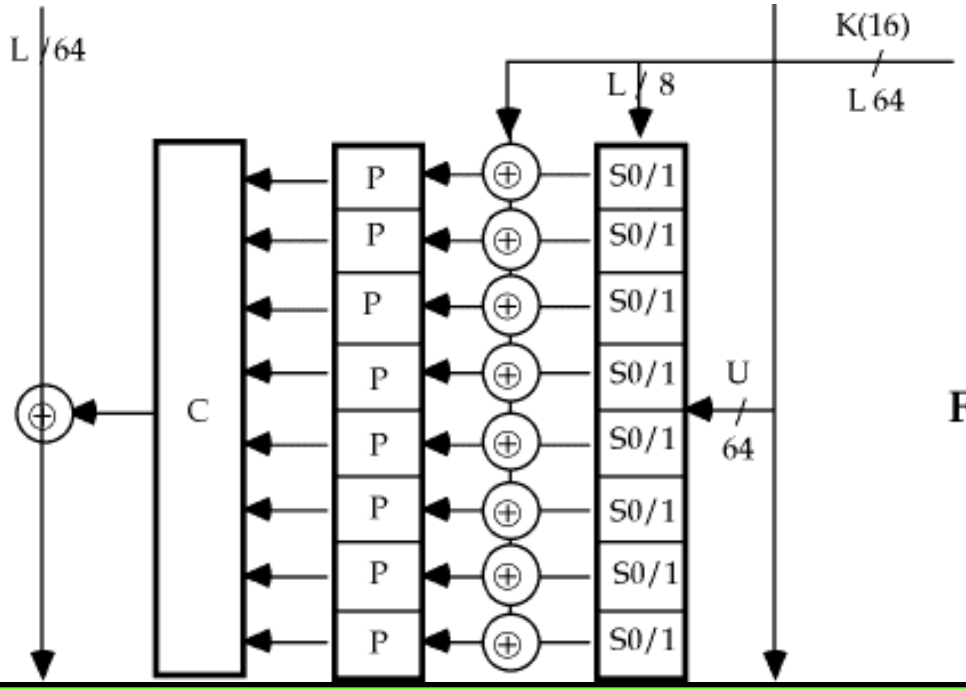


Fig 2.6 - Lucifer Function f

Cifra tipo Feistel

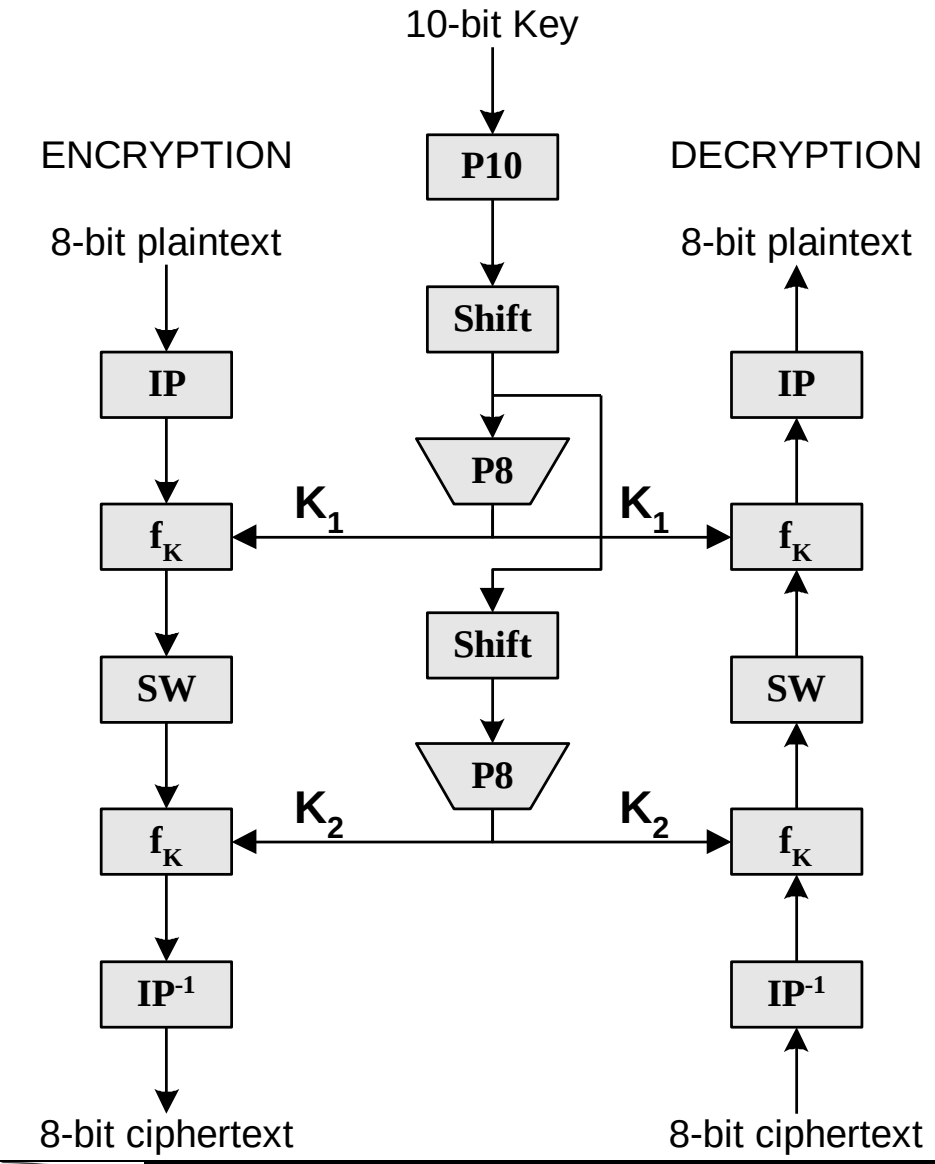
Conclusión y Seguridad de Lucifer

- No ha sido analizada exhaustivamente.
- Actualmente puede ser quebrada con criptoanálisis diferencial.
- No se utiliza.
- Su principal aporte es como predecesora de DES.

S-DES

Small DES

S-DES



$$IP^{-1} \circ f_{K2} \circ SW \circ f_{K1} \circ IP$$

$$TC = IP^{-1}(f_{K2}(SW(f_{K1}(IP(TP)))))$$

Con

$$K_1 = P8\,(\,Shift\,(\,P10\,(\,key\,)\,)\,)$$

$$K_2 = P8\,(Shift\,(Shift\,(P10\,(key)\,)\,)\,)$$

$$TP = IP^{-1}(f_{K1}(SW(f_{K2}(IP(TC)))))$$

S-DES

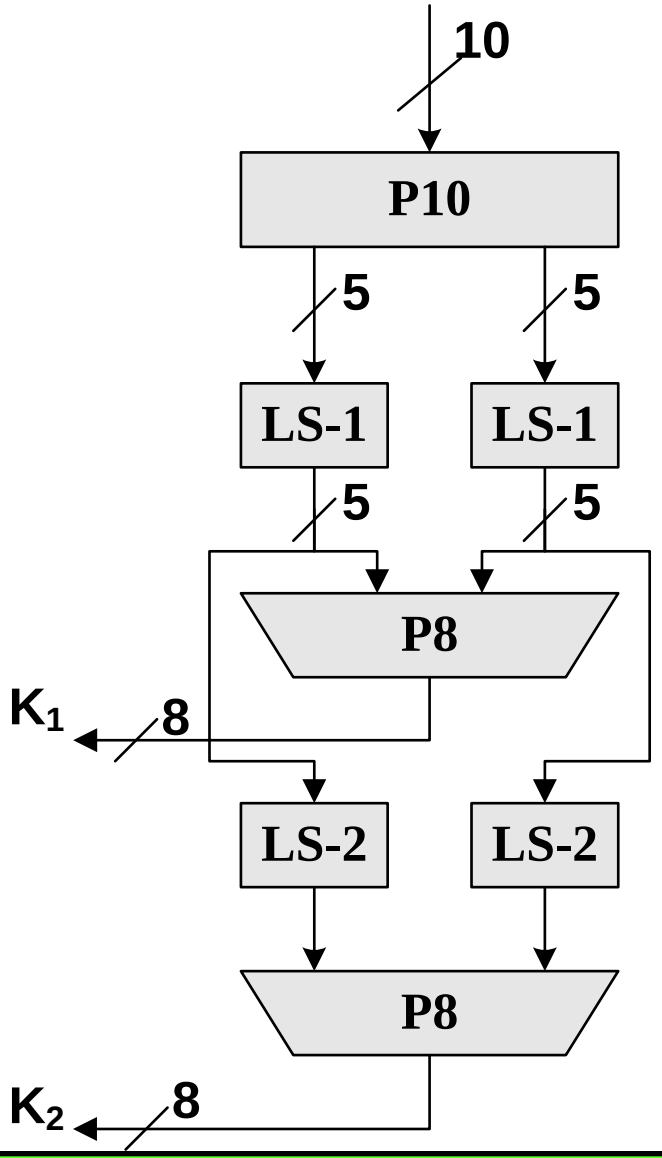
Generación de clave

P10$(k_1,k_2,k_3,k_4,k_5,k_6,k_7,k_8,k_9,k_{10})$ =

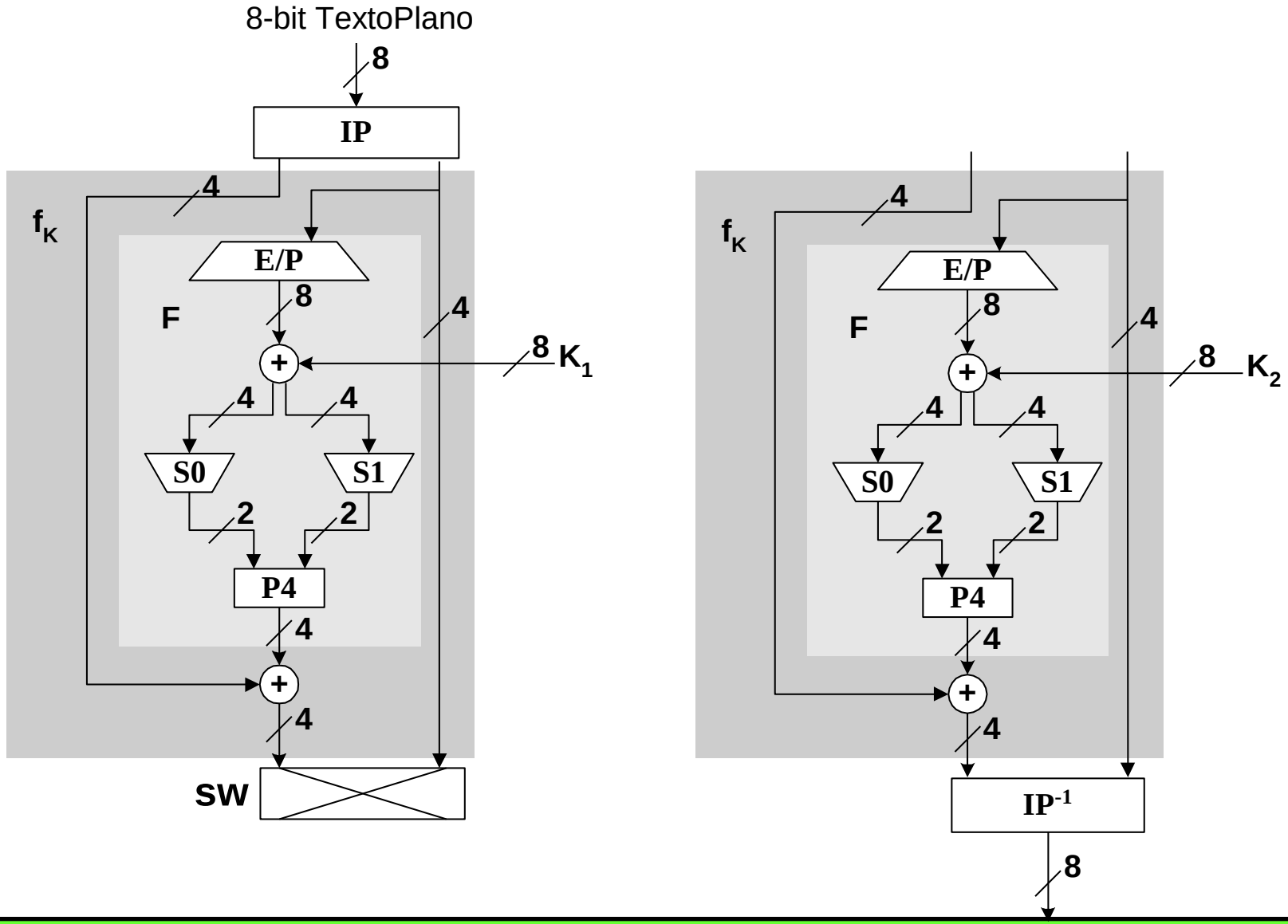
$(k_3,k_5,k_2,k_7,k_4,k_{10},k_1,k_9,k_8,k_6)$

P10									
3	5	2	7	4	10	1	9	8	6

P8							
6	3	7	4	8	5	10	9



S-DES


8-bit TextoPlano
8
IP
4
f_K
E/P
F
8
4
8
K_1
4
4
S0
S1
2
2
P4
4
4
SW
4
f_K
E/P
F
8
4
8
K_2
4
4
S0
S1
2
2
P4
4
4
IP^{-1}
8
8-bit TextoCifrado

S-DES

IP							
2	6	3	1	4	8	5	7

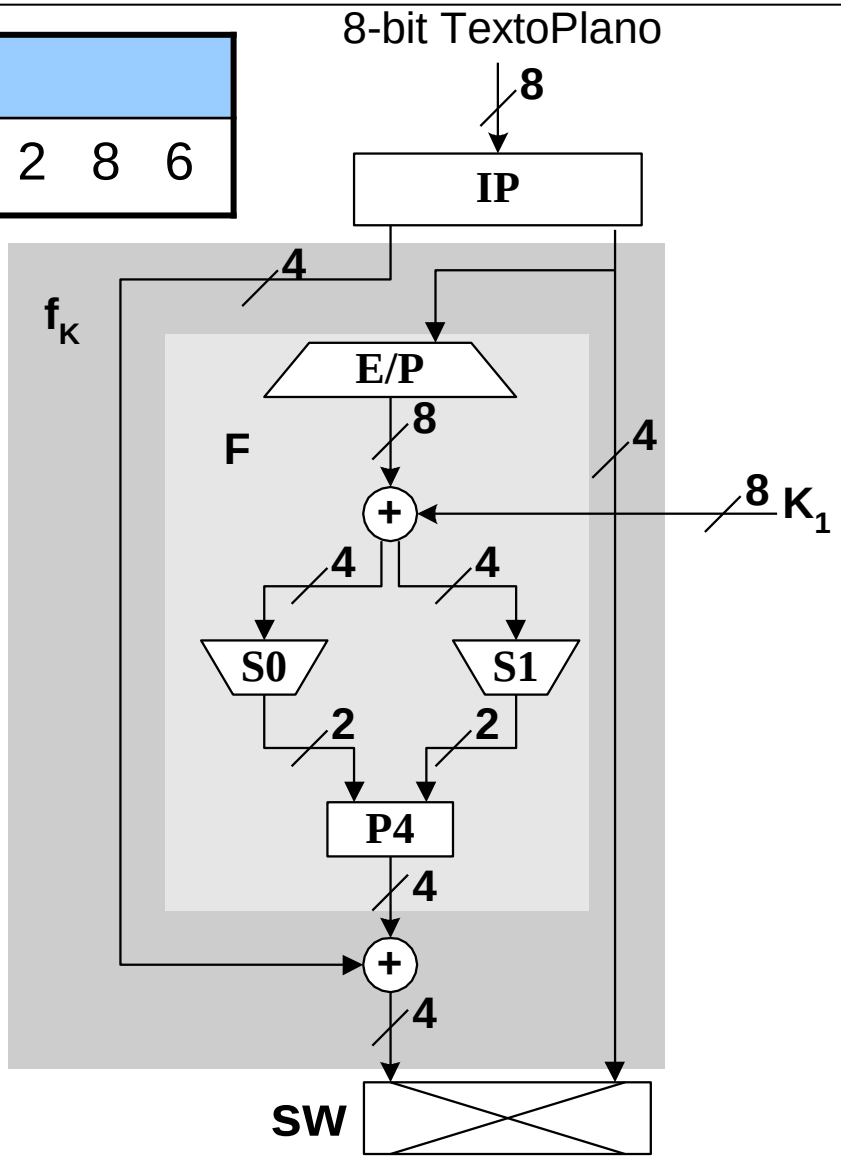
IP^{-1}							
4	1	3	5	7	2	8	6

$$f_K(L,R) = (L \oplus F(R,SK),\ R)$$

E/P							
4	1	2	3	2	3	4	1

$n_4 + k_{11}$	$n_1 + k_{12}$	$n_2 + k_{13}$	$n_3 + k_{14}$
$n_2 + k_{15}$	$n_3 + k_{16}$	$n_4 + k_{17}$	$n_1 + k_{18}$

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



S-DES

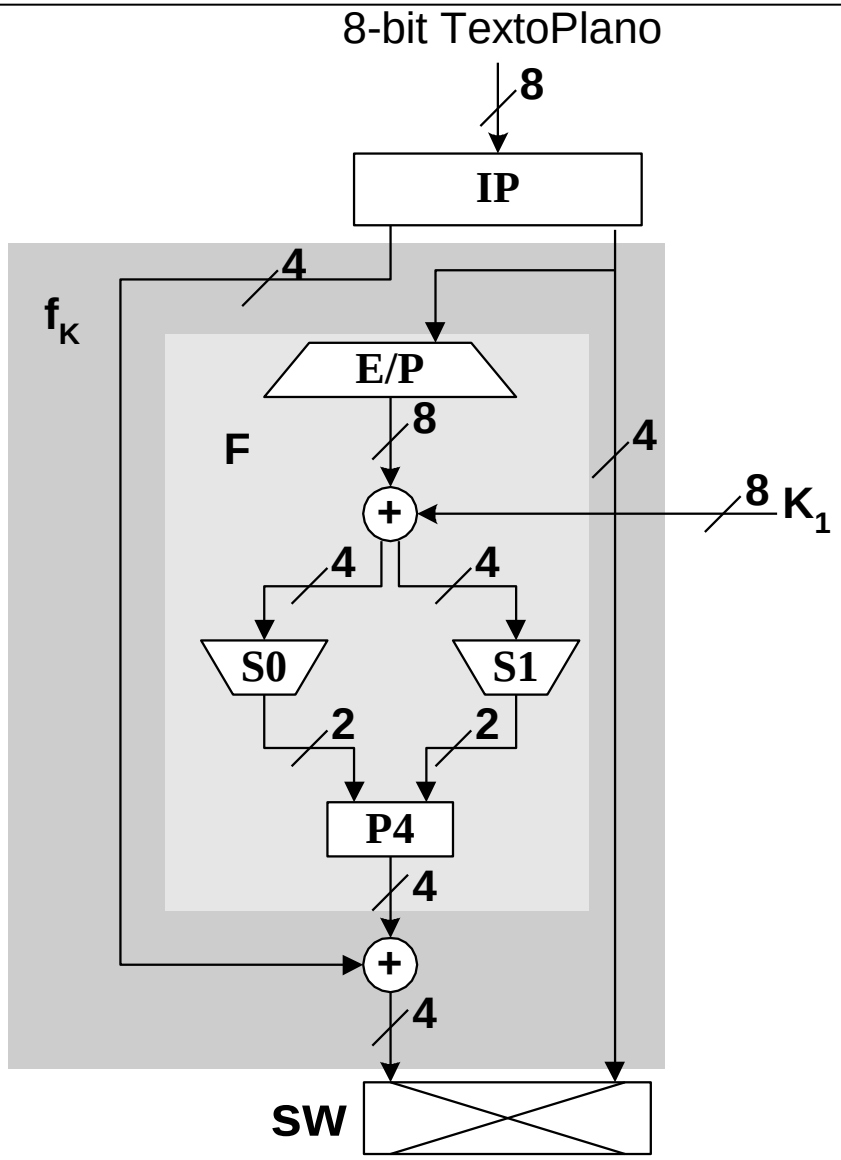
$n_4 + k_{11}$	$n_1 + k_{12}$	$n_2 + k_{13}$	$n_3 + k_{14}$
$n_2 + k_{15}$	$n_3 + k_{16}$	$n_4 + k_{17}$	$n_1 + k_{18}$

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

$$S1 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{pmatrix}$$

P4			
2	4	3	1

$$S0 = \begin{pmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{pmatrix}$$



DES

Data Encryption Standard

DES

La *NSA "National Security Agency"* fue creada por el *Harry Truman* a comienzos de la década del 50'.

Depende del *DoD* y centraliza todo lo referente a la Seguridad, tanto interna como externa y para ello cuenta con el equipamiento más sofisticado del mundo y el mayor equipo de matemáticos especializados en criptografía.

Trabaja en estrecha colaboración con el *NIST "National Institute of Standards and technology"*, dependiente del *Departamento de Comercio*.

Entre ambos regulan el uso, control y exportación del software y del hardware vinculado a la *Privacidad y Seguridad*.

NSA: http://www.nsa.gov/

NIST: http://www.nist.gov/index.html

NIST Virtual Library: http://csrc.nist.gov/publications/PubsSPs.html

DES

Estas dos organizaciones suelen tener posiciones encontradas con la comunidad internet, especialmente en asuntos concernientes a la privacidad y a las *Libertades Civiles*.

Aconsejo dar un vistazo no sólo a éstos sitios oficiales del gobierno de USA sino también a los sitios "rebeldes".

Por ejemplo, el de la *EFF "Electronic Frontier Foundation"* http://www.eff.org a fin de tener un panorama global de los intereses económicos, políticos y sociales en juego.

Comentar artículo:

https://www.eff.org/deeplinks/2013/08/illustration-how-nsa-misleads-public-without-actually-lying

DES

15 Mayo	1973	NBS 1er llamado a la construcción de un algorimto estándar de encripción
27 Agosto	1974	NBS 2do llamado..!!
17 Marzo	1975	DES es publicado en los Federal Register para su comentario.
Agosto	1976	First Workshop on DES
Septie.	1976	Second Workshop. Se discuten fundamentos matemáticos de DES.
Noviem.	1976	DES es aprobado como estándard.
15 Enero	1977	DES es publicado com el estándar FIPS PUB 46.
	1983	DES es reafirmado por primera vez..!!
	1986	Videocipher II, un codificador de TV satelital basado en DES. Usado por HBO..!
22 Enero	1988	DES es reafirmado por segunda vez. FIPS PUB 46-1 reemplaza a la PUB 46.
Julio	1990	Biham y Shamir descubren el criptoanálisis diferencial y lo aplican sobre DES15.
	1992	Biham y Shamir informan el primer ataque teórico con una complejidad menor que un ataque de fuerza bruta: el criptoanálisis diferencial. De todos modos requiere 2^{47} textos planos elegidos.

DES

30 Diciem	1993	DES es reafirmado por tercera vez, ahora con la FIPS 46-2
	1994	Se realiza el primer criptoanálisis experimental de DES usando criptoanálisis lineal. (Matsui, 1994).
Junio	1997	Mediante el proyecto DESCHALL se rompe un mensaje encriptado con DES por primera vez en público.
Julio	1998	Los Crakers de la EFF rompen una clave de DES en 56 horas (Deep Crack).
Enero	1999	Deep Crack junto a distribuited.net rompen una clave de DES en 22 hs 15m.
25 Octub	1999	DES reafirmado por cuarta vez en FIPS 46-3, se prefiere Triple DES. http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
26 Novie	2001	Se publica AES en FIPS 197
26 Mayo	2002	Se hace efectivo el uso del nuevo estándar AES
26 Julio	2004	Se propone el retiro de los Federal Register mediante FIPS 46-3
19 Mayo	2005	NIST retira la FIPS 46-3
15 Marzo	2007	La máquina paralela basada en FPGA COPACOBANA de University of Bockum and Kiel, Germany, rompe DES en 6,4 días con un costo de hardware de 10.000 $.

DES

Hay varios puntos álgidos en el diseño de DES

Si bien el estándar es de dominio público, no así los criterios de diseño.

Hubo grandes discusiones particularmente en lo que se refiere a la longitud de la clave de 56 bits y al diseño de los S-Box.

W.Diffie, M.Hellman "Exhaustive Cryptanalysis of the NBS Data Encryption Standard" IEEE Computer 10(6), June 1977, pp74-84

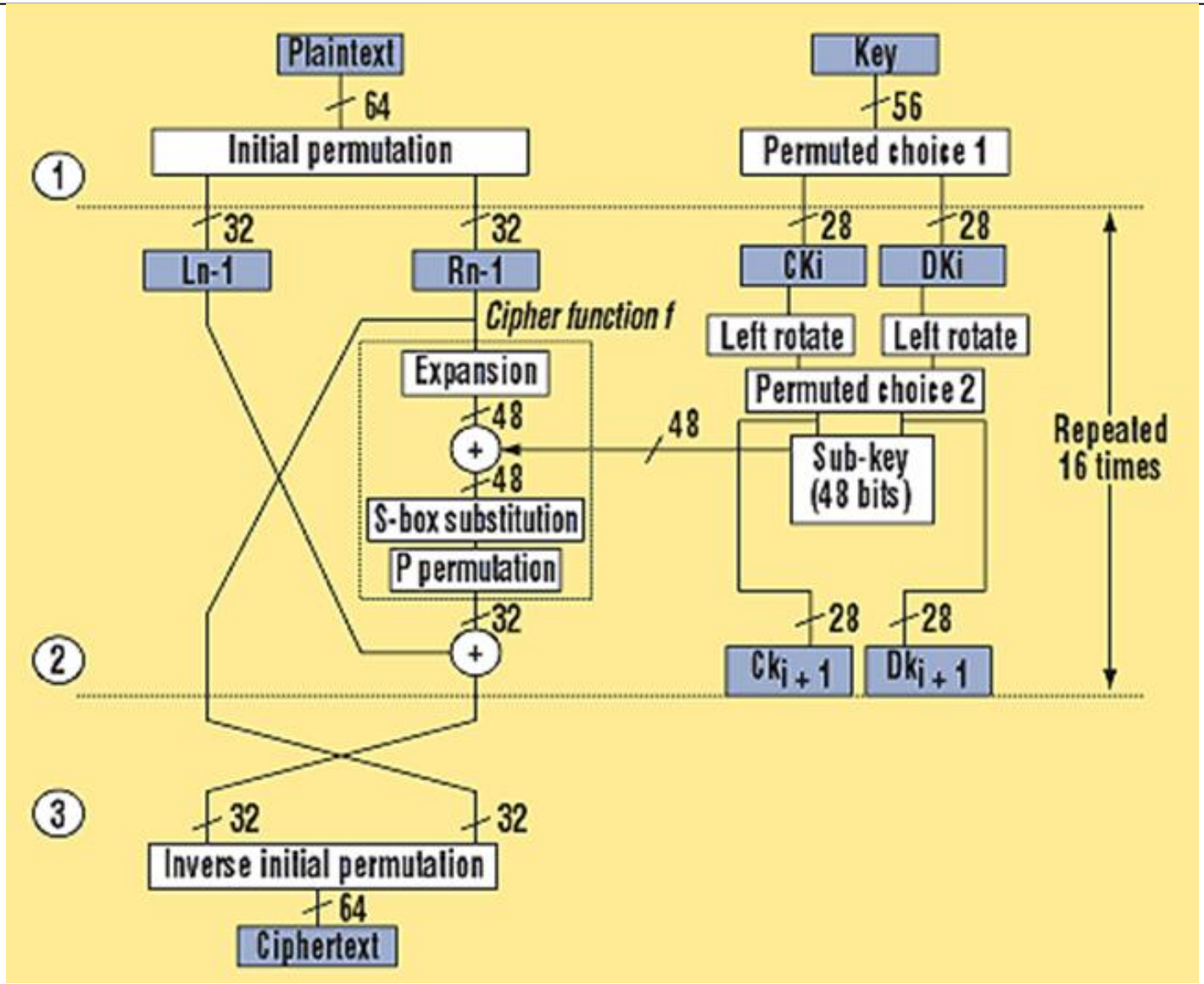
M.Hellman "DES will be totally insecure within ten years" IEEE Spectrum 16(7), Jul 1979, pp 31-41

Descripción del algoritmo de encripción DES

El proceso de cifrado usado por DES consiste de:

- Una permutación inicial IP.
- 16 rondas de cálculo con una función ***f*** dependiente de una clave compleja.
- Una permutación final, la inversa de IP.

DES


Plaintext
64
Initial permutation
32
32
Ln-1
Rn-1
Cipher function f
Expansion
48
48
S-box substitution
P permutation
32
Key
56
Permuted choice 1
28
28
CKi
DKi
Left rotate
Left rotate
Permuted choice 2
48
Sub-key
(48 bits)
Repealed
16 times
28
28
Ckj + 1
Dkj + 1
1
2
3
32
32
Inverse initial permutation
64
Ciphertext

Obtención de las sub claves de cada ronda

Se obtienen a partir del siguiente proceso sobre la clave de entrada:

1. Una permutación PC1 de los 56 bits originales en dos grupos de 28 bits.

2. 16 rondas consistentes en:
   - Selección de 24 bits de cada mitad.
   - Permutación PC2 para usar el resultado en la función f.
   - Rotación de cada mitad por separado 1 o 2 posiciones dependiendo de la ronda, que puede expresarse por K_S.

3. Funcionalmente puede ser descripto como sigue:

$$SK_i = PC2\ (K_S\ (\ PC1\ (Key),\ i\ )\ )$$

PC1:

- Objetivo: seleccionar 56 de los 64 bit de la clave.
- Cada 8 bit se descarta uno (asume ser bit de paridad).
- PC1 parte la clave en dos mitades (C y D)
- Para DES, de 1 a 32 (izq, MSB) / 32 a 64 (der, LSB)

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

Mitad C

Mitad D

PC2:

- Objetivo: seleccionar 48 bits.
- Usados en c/ronda del procesamiento de los datos de DES
- La mitad C provee bits para S1-S4, la mitad D para S5-S8.

14	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

Mitad C
bits 1 – 28

Mitad D
bits 29 – 56

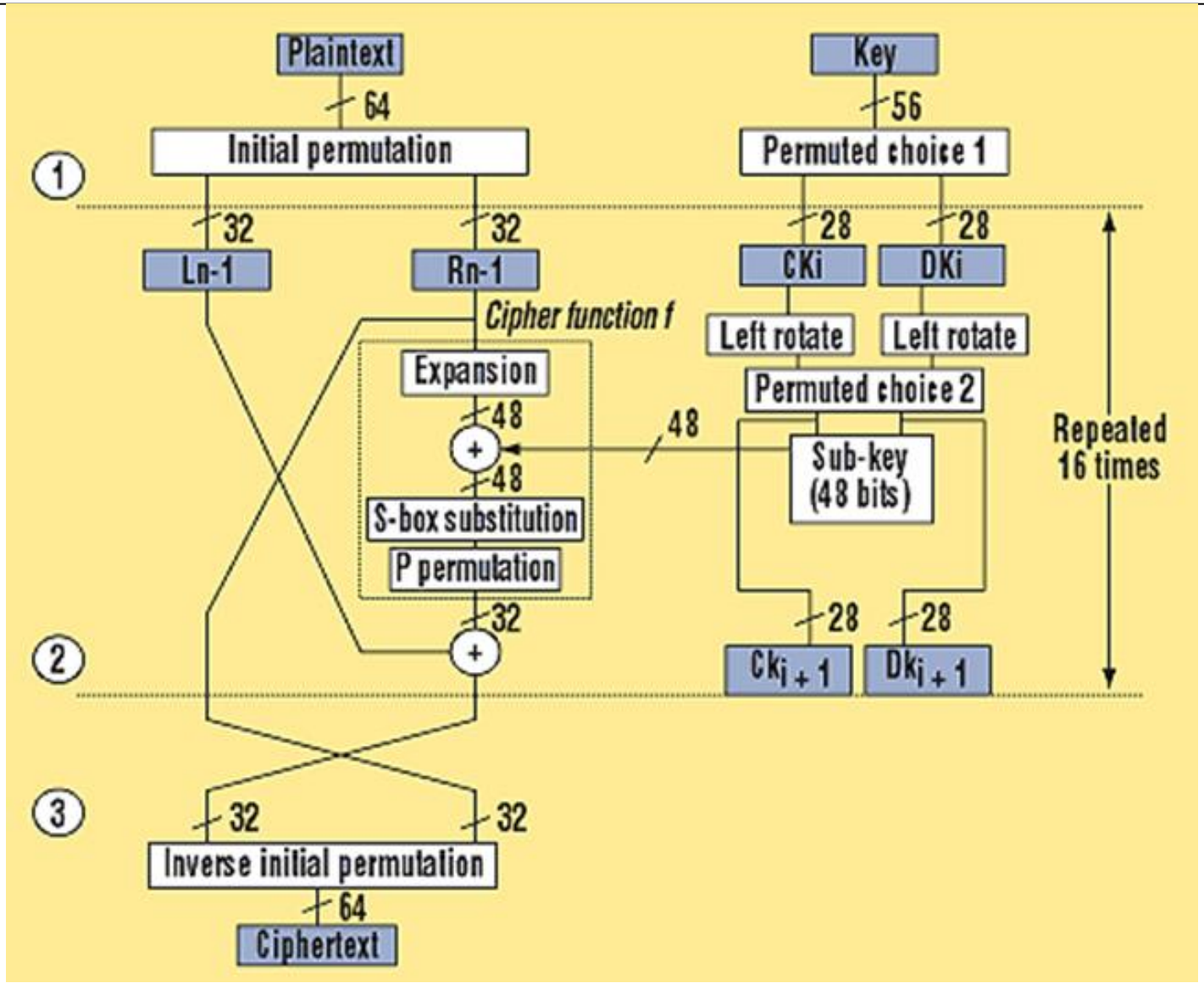
Programación de rotaciones de la clave

Las rotaciones de clave, K_S están especificadas por:

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
K_S	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1
Total	1	2	4	6	8	10	12	14	15	17	19	21	23	25	27	28

DES

Tratamiento de los datos



DES

Permutación Inicial IP: Primer paso del cómputo de los datos.

- Objetivo: reordenar los bits de entrada.
- Bit pares del lado izquierdo **LH** y bits impares del lado derecho **RH**.
- Una estructura bastante regular (fácil de implementar en h/w)
- No mejora en general la seguridad de la cifra, solo la hace mas compleja.

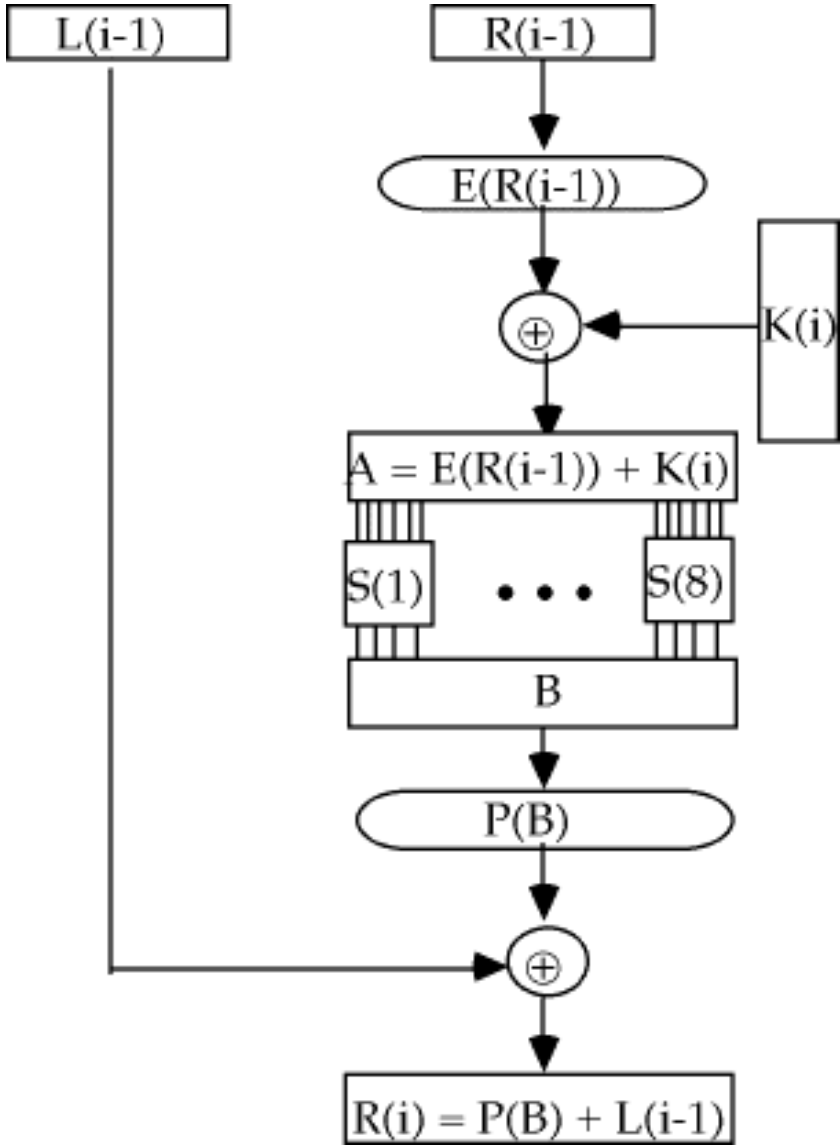
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

DES

Función f

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus P(S(\boldsymbol{E(R_{i-1})} \oplus \boldsymbol{SK_i}))$



Función de expansión E

- Expande R_{i-1} de los datos de entrada de 32 bit a 48
- Duplica algunos bits y parte la entrada en 8 grupos de 4 bits
- Luego duplica bit de cada lado para formar grupos de 6 bits.
- Provee un efecto de autoclave para atacar las S-box.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Suma con la clave

- Los 48 bits expandidos son XOReados con los 48 de la clave seleccionados por PC2
- Objetivo: la clave tiene influencia sobre el resultado de cada ronda.

DES

Bloques de sustitución S

- Hay 8 S-box donde cada una mapea 6 bits en 4 bits.
- Cada S-box se implementa mediante 4 box mas pequeñas de 4 bits
- Los bits externos 1 & 6 (row bits) seleccionan una de las 4 filas
- Los bits internos 2 - 5 (col bits) seleccionan la columna
- Resultando en 8 lotes de 4 bits que resultan en 32 bits a la salida

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

DES

Box	Row	Column 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	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
S_3	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Permutación P

- Objetivo: difundir la salida de las S-box sobre las diferentes entradas de las S-box de la próxima ronda.
- Los valores tienen un patrón.
- Asegura que cada salida de la S-box afecte tanto a los bit de columnas como de filas.

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	1	4	25

Permutación final

- Después de 16 rondas, una permutación final entrega la salida del cómputo DES.
- Es la inversa de la permutación IP.
- En el proceso de desencripción desanda IP.

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

Validación

Test 1:

K: 5b5a57676a56676e

P: 675a69675e5a6b5a

C: 974affbf86022d1f

Test 2:

K: 1c587f1c13924fef

P: 63fac0d034d9f793

C: e4c70d01ea89efc5

¿ Modos de uso de una cifra de bloque ?

Modos de uso

Modos de uso

Con la cifra de bloques se encripta un bloque de información de tamaño fijo.

DES encripta una bloque de 64 bits de datos usando una clave de 56 bits.

¿ Cómo usar los algoritmos de encripción en la práctica ?

Es usual tener bloques de datos de mayor longitud.

La solución son los **Modos de Uso**.

Son recomendaciones transformadas en estándares.

Modos de uso

Electronic Codebook (ECB)

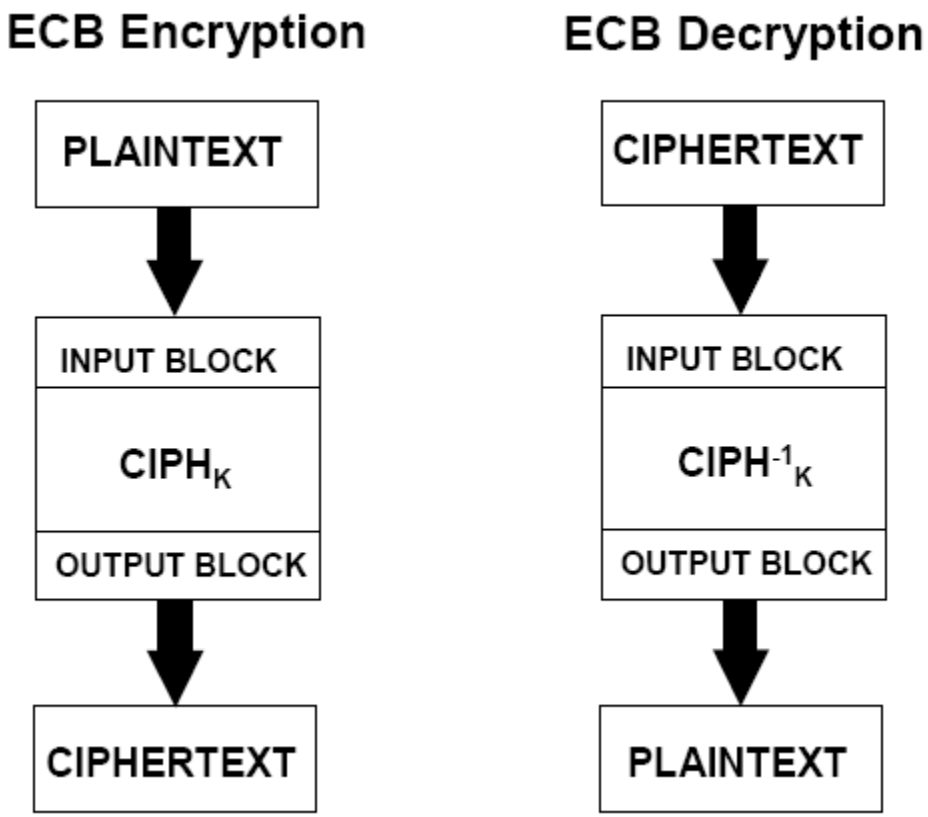
- El mensaje es roto en bloques independientes, que luego son encriptados de forma independiente.
- Cada bloque es un valor, el cual es sustituido tal cual se hacía con los libros de códigos. De allí su nombre..!
- Cada bloque es independiente de todos los otros.

$$C_j = CIPH_K(P_j) \qquad \text{for } j = 1 \ldots n.$$

$$P_j = CIPH_K^{-1}(C_j) \qquad \text{for } j = 1 \ldots n.$$

Modos de uso

Electronic Codebook (ECB)



Modos de uso

Ventajas y Limitaciones de ECB

- Las repeticiones en el mensaje pueden reflejarse en el texto cifrado..!
- Si la cifra se alinea con el mensaje, particularmente con datos tales como gráficos, o mensajes que cambian muy poco, volvemos a una situación del problema del libro de códigos.
- Es realmente útil, sólo cuando se transmiten unos pocos bloques de datos.

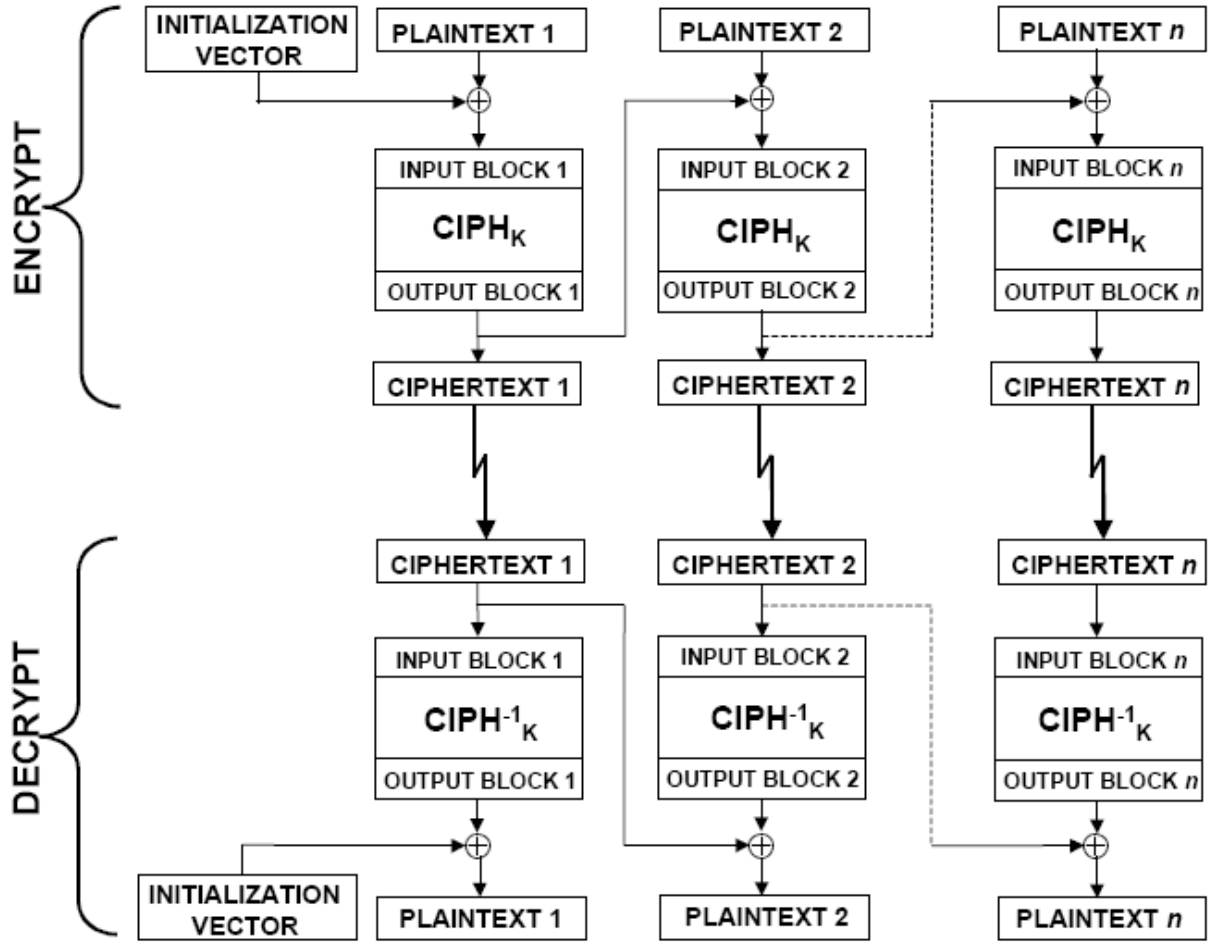
Modos de uso

Cipher Block Chaining (CBC)

- El mensaje es nuevamente roto en bloques.
- Pero ahora se encadenan todos juntos en el proceso de encripción.
- Bloques de texto cifrado se encadenan con texto plano, de allí su nombre.
- Usa un Initial Vector (IV) para comenzar el proceso..!

Modos de uso

Cipher Block Chaining (CBC)



$C_1 = CIPH_K(P_1 \oplus IV);$
$C_j = CIPH_K(P_j \oplus C_{j-1})$

$P_1 = CIPH_K^{-1}(C_1) \oplus IV;$
$P_j = CIPH_K^{-1}(C_j) \oplus C_{j-1}$

Modos de uso

Ventajas y Limitaciones de CBC

- Cada bloque de cifra depende de todos los bloques de mensajes anteriores a él.
- Es el modo de uso mas usualmente usado cuando se dispone de los datos por adelantado.
- De este modo, un cambio en el mensaje no solo afecta al bloque de cifra obtenido, sino también a los sucesivos bloques.
- Para comenzar se necesita un **Initial Value** (IV) el cual debe conocerse por ambas partes (emisor y receptor)
- Si el IV es enviado sin cifrar, un atacante puede cambiar bits en el primer bloque y cambiar IV para ocultarlo

Modos de uso

Ventajas y Limitaciones de CBC

- Otro punto a considerar es que en el final el mensaje debe contemplar la posibilidad de un bloque mas corto.

  Este o se rellena, del siguiente modo

  Ej.: [b1 b2 b3 0 0 0 0 5]

  3 bytes de datos, luego 5 bytes pad + cuenta

  O se completa con datos sin sentido, desdoblando los últimos dos bloques.

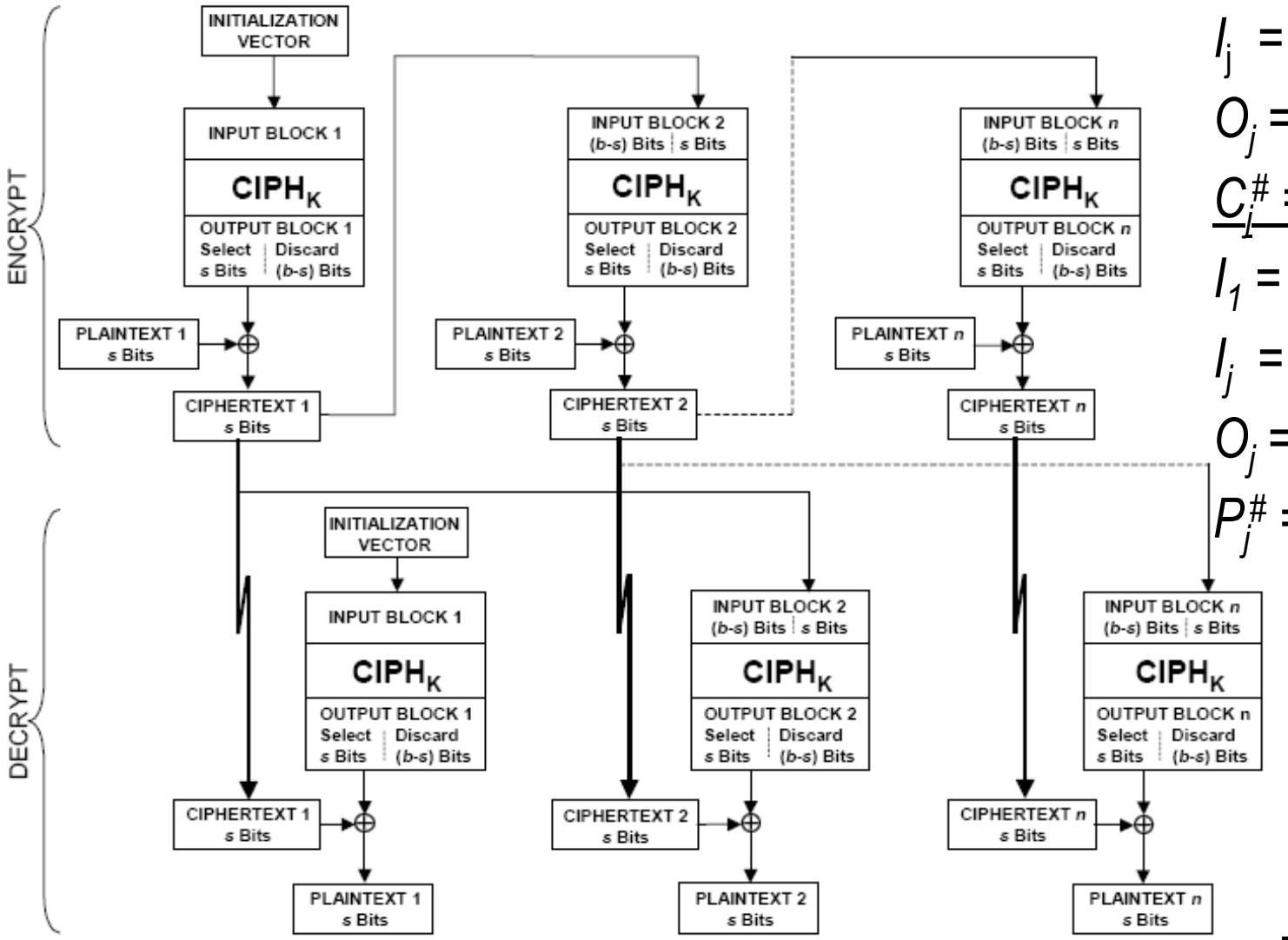
Modos de uso

Modo Cipher Feed Back (CFB)

- El mensaje es tratado como una stream de bits que se suman a la salida del bloque de cifra, y realimentados en la siguiente etapa (de allí su nombre).
- El estándar permite realimentar s =1, 8, 64 ó 128.
- Luego esto de nombre como CFB-1, CFB-8, CFB-64,… etc
- En la práctica es muy frecuente realimentar los 64 bits (CFB-64).
- Este modo requiere que el IV sea un “nonce” (*number used once*) para cada ejecución del modo dada una clave.

Modos de uso

Modo Cipher Feed Back (CFB)



$I_1 = IV;$

$I_j = LSB_{b-s}(I_{j-1}) \mid C^{\#}_{j-1}$

$O_j = CIPH_K(I_j)$

$C^{\#}_j = P^{\#}_j \oplus MSB_s(O_j)$

$I_1 = IV;$

$I_j = LSB_{b-s}(I_{j-1}) \mid C^{\#}_{j-1}$

$O_j = CIPH_K(I_j)$

$P^{\#}_j = C^{\#}_j \oplus MSB_s(O_j)$

Modos de uso

Ventajas y Limitaciones de CFB

- Es mucho mas apropiado usar este modo cuando los datos arriban de a bits/bytes.
- Es el modo mas común de los “stream modes”.
- La cifra se usa en modo encripción en ambos extremos..!
- Obviamente, que un error (ruido) en un bloque se propaga sobre el resto de los bloques..!

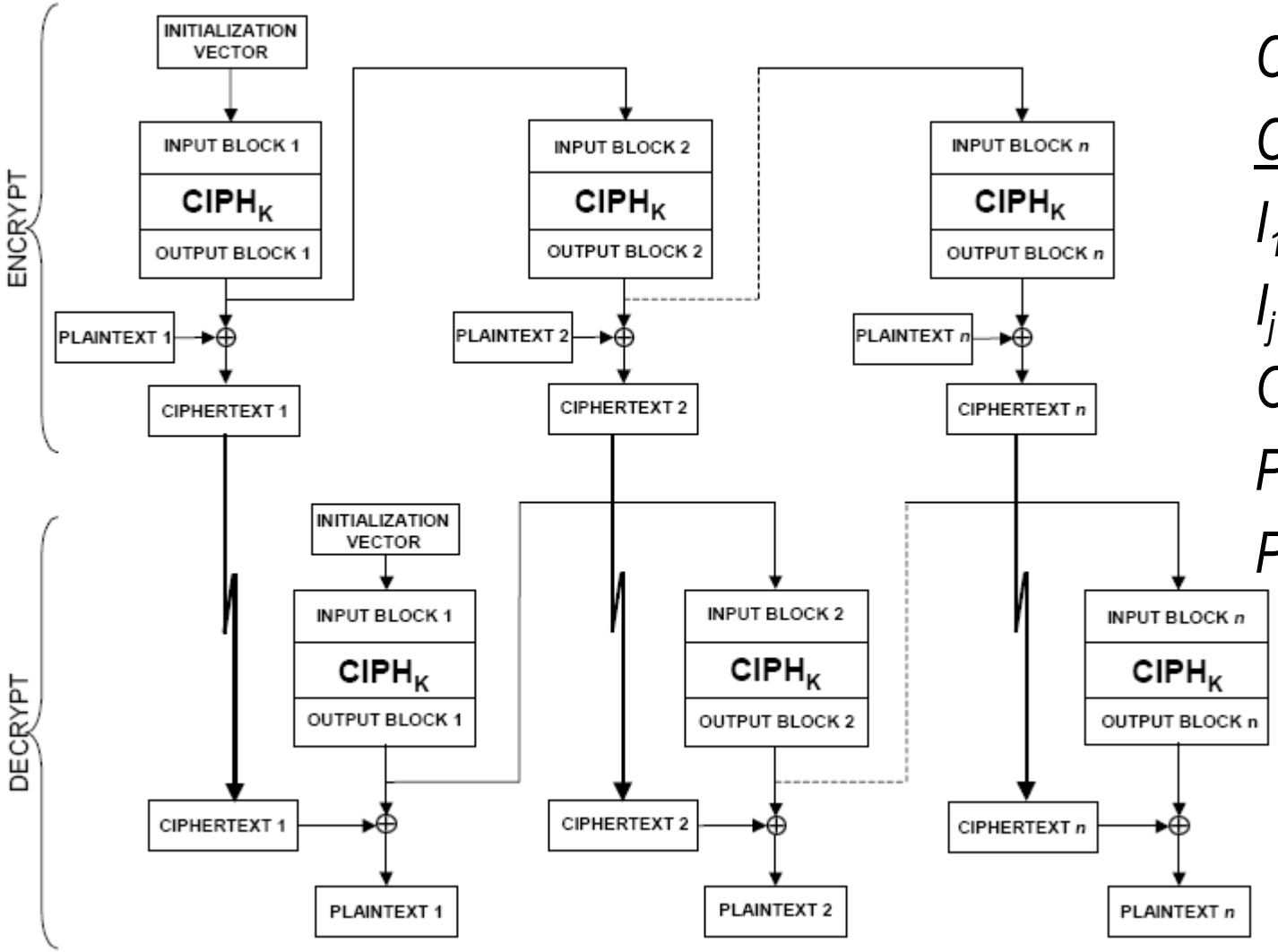
Modos de uso

Modo Output Feed Back (OFB)

- El mensaje es tratado como una stream de bits..!
- El bloque de cifra se suma al mensaje y la salida es entonces realimentada (de allí su nombre).
- En este modo, la realimentación es independiente del mensaje.
- Requiere que el IV sea un “nonce” (*number used once*) para cada ejecución del modo dada una clave.

Modos de uso

Modo OFB - Encripción



$I_1 = IV;$

$I_j = O_{j-1}$

$O_j = CIPH_K(I_j)$

$C_j = P_j \oplus O_j$

$C_n^* = P_n^* \oplus MSB_u(O_n)$

$I_1 = IV;$

$I_j = O_{j-1}$

$O_j = CIPH_K(I_j)$

$P_j = C_j \oplus O_j$

$P_n^* = C_n^* \oplus MSB_u(O_n)$

Modos de uso

Ventajas y Limitaciones de OFB

- Se usa cuando la realimentación de un error es un problema o cuando el proceso de encripción debe realizarse antes que la totalidad del mensaje este disponible.
- Es similar a CFB superficialmente, pero la realimentación se realiza desde la salida del bloque de cifrado y es independiente del mensaje.
- Nunca se re usa la misma secuencia key+IV..!!!
- Emisor y receptor deben permanecer en sincronismo y se debe tener algún método de recuperación para asegurar esto.
- Si bien en el estándar se específico una realimentación de m bits, luego se rectificó a 64.

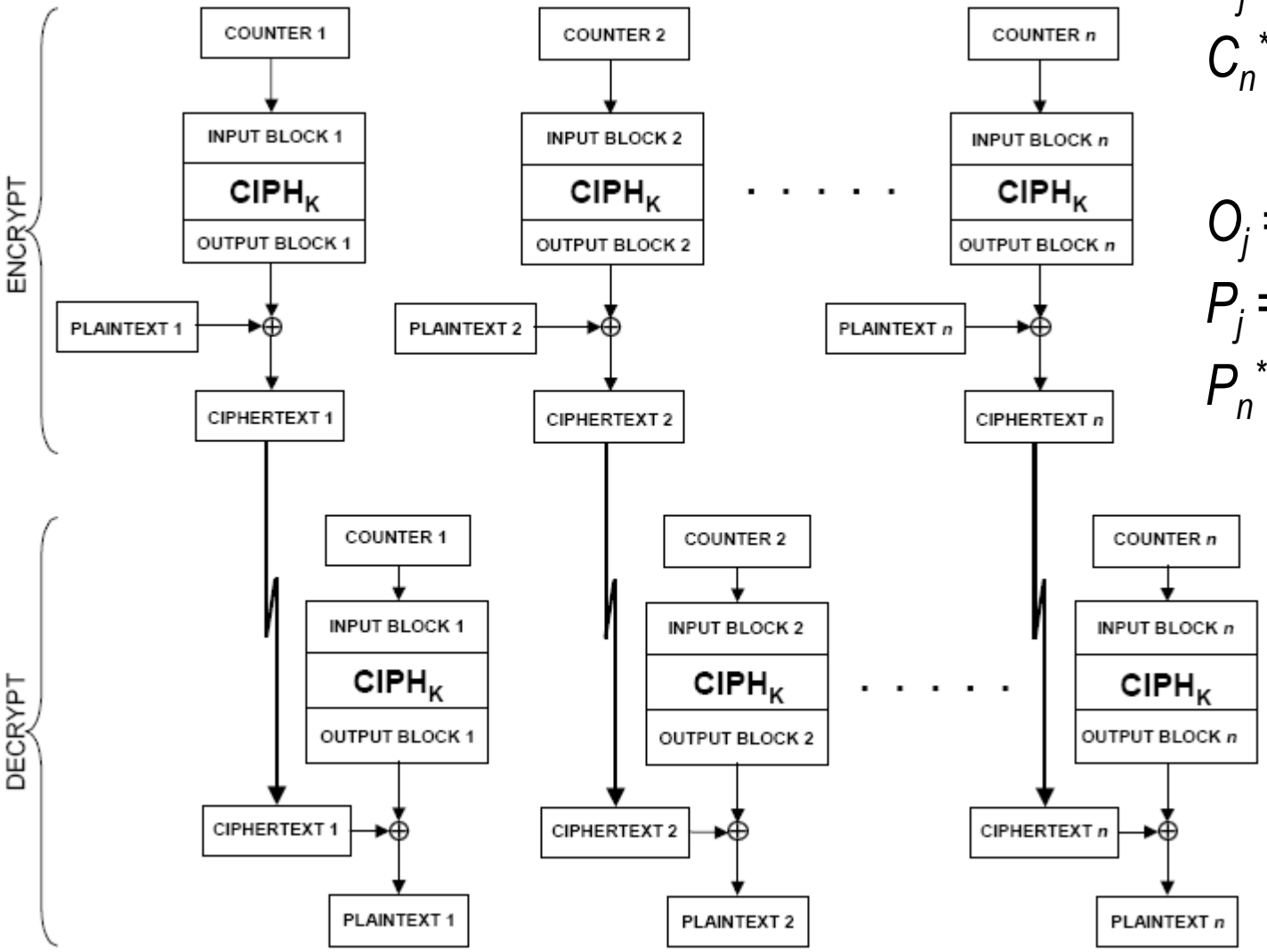
Modos de uso

Counter Mode (CTR)

- El mensaje es tratado como un conjunto de bloques asociados a valores de un contador.
- La secuencia de conteo debe presentar la propiedad de que c/bloque en la secuencia sea diferente de cualquier otro bloque.
- Esta condición no está restringida a un único mensaje: a lo largo de todos los mensajes que son encriptados con una clave, todos los valores contados deben ser distintos.
- Los valores del contador para un mensaje dado, son $T_1, T_2, \ldots, T_n$.

Modos de uso

Counter Mode (CTR)



$O_j = CIPH_K(T_j)$

$C_j = P_j \oplus O_j$

$C_n^* = P_n^* \oplus MSB_u(O_n)$

$O_j = CIPH_K(T_j)$

$P_j = C_j \oplus O_j$

$P_n^* = C_n^* \oplus MSB_u(O_n)$

Modos de uso

Ventajas y Limitaciones de CTR

- Tanto el proceso de encripción como de desencripción puede realizarse en paralelo.
- Cualquier bloque puede ser recuperado en forma independientemente del resto, si se conoce el valor de T_i.

Modos de uso

Aspectos adicionales

- Rellenado de último bloque
- Generación de bloques de conteo
- Generación de Vectores de Inicialización
- Propiedad de los errores de un bit en alguno de los bloques
- En la documentación podrán encontrar valores para testing de c/u de estos modos para AES.

Cifra de Bloque Moderna

Bibliografía:

- Kam, J.B., and Davida, G.I.; “Structured Design of Substitution Permutation Encryption Networks”. IEEE Transactions on Computers, Vol 28, No. 10, 747 (1979).
- Feistel, H.; “Cryptography and Computer Privacy”. Scientific American, Vol. 228, No. 5, 15 (1973).
- Webster A.F., S.E. Tavares; "On the Design of S-Boxes"; Department of Electrical Engineering; Queen's University; Kingston; Canada.
- Recommendation for Block Cipher Modes of Operation, Methods and Techniques; NIST; 2001. http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf
- S-Box Design: A Literature Survey; (http://www.ciphersbyritter.com/RES/SBOXDESN.HT)


GRACIAS!