4.- Cifrado en bloque con clave secreta

4.1.- Características generales
4.2.- DES (Data Encryption Standard)
4.3.- AES (Advanced Encryption Standard)
4.4.- Modos de cifrado en bloque
4.5.- Cifrado Múltiple. Triple DES

Criptografía simétrica

- La criptografía de clave secreta o simétrica se refiere al conjunto de métodos que permiten tener comunicación segura entre las partes siempre y cuando anteriormente se hayan intercambiado la clave correspondiente.

- Ha sido la **más usada en toda la historia** y ha sido implementada en diferentes dispositivos, manuales, mecánicos, eléctricos, hasta los algoritmos actuales que son programables en cualquier ordenador.

- Todos los sistemas **criptográficos clásicos** se basan en **criptografía simétrica**.

Criptografía simétrica

- Generalmente el **algoritmo de cifrado es conocido** por lo que la **fortaleza** del mismo dependerá de su complejidad interna y sobre todo de la longitud de la clave empleada.

- Para que un algoritmo de este tipo sea considerado fiable debe cumplir varios requisitos básicos:
  - **conocido el criptograma** (texto cifrado) **no se pueden obtener de él ni el texto en claro ni la clave**.
  - **conocidos el texto en claro y el texto cifrado** debe **resultar más caro en tiempo o dinero descifrar la clave** que el valor posible de la información obtenida por terceros.

Criptografía simétrica

- El principal problema para este sistema de cifrado consiste en que para cada par de usuarios que quieran establecer comunicación se requiere una clave diferente, es decir, que un usuario de una red debe almacenar tantas claves como personas con las que quiera mantener una comunicación segura.

- Al principio (cuando las redes contaban con pocos usuarios) este hecho no constituía ningún problema, pero actualmente, con la cantidad de usuarios que existen en las redes se convierte en impracticable.

- Otro problema que presentan es el hecho de la distribución de claves y el peligro de que muchas personas deban conocer una misma clave.

4.1 Características generales

- Se denomina cifrado en bloque aquel en el que **se cifra** el mensaje original **agrupando los símbolos en grupos** (bloques) de dos o más elementos.

- Algunos sistemas de cifrado clásicos, como el poligrámico y el de transposición, son ejemplos de *cifrado en bloque*.

- En los sistemas modernos de cifrado en bloque
  - Cada **símbolo** se **cifra** de manera **dependiente de los adyacentes**.
  - Cada **bloque** de símbolos se **cifra** siempre de **igual manera**, independientemente del lugar que ocupe en el mensaje.
  - **Dos mensajes originales iguales**, cifrados con la misma clave, **producen siempre mensajes cifrados iguales**.
  - Para **descifrar parte de un mensaje** no es preciso descifrarlo completamente desde el principio, **basta con hacerlo desde el bloque que interese**.

4.1 Características generales

- Los cifradores en bloque con clave secreta aplican técnicas de **sustitución y transposición**, además de otras operaciones lineales y no lineales.
- Se apoyan en los principios de **confusión y difusión** propuestos por Shannon que se combinan para dar lugar a los denominados cifrados de producto.

  - Recordemos que la **confusión** consiste en tratar de ocultar la relación que existe entre el texto en claro, el texto cifrado y la clave.
  - Por su parte la **difusión** trata de repartir la influencia de cada bit del mensaje original lo más posible entre el mensaje cifrado.

4.1 Características generales

- La mayoría de los algoritmos se basan en diferentes capas de sustituciones y permutaciones, estructura que denominaremos **red de sustitución-permutación**.

- En muchos casos el criptosistema no es más que **un paso simple de sustitución-permutación repetido n veces**, como ocurre con **DES**.

4.1 Características generales

- En muchos casos, los cifrados en bloque se componen de cuatro elementos:
  - Transformación inicial.
  - Una función criptográficamente débil iterada *r* veces o «**vueltas**» (« **rondas** »).
  - Transformación final.
  - Algoritmo de expansión de clave.

4.1 Características generales

- **La transformación inicial** puede tener una o dos funciones:
  - La primera consiste simplemente en **distribuir aleatoriamente los datos de entrada** (para ocultar bloques de datos de todo ceros o unos, etc.), **careciendo de significación criptográfica** si no depende de la clave.
  - La segunda función, solamente presente en algunos criptosistemas, **tiene significación criptográfica**, **dificultando ataques por análisis lineal o diferencial**; en este caso es función de la clave.

4.1 Características generales

- **Las vueltas intermedias**:
  - Consisten en una función complicada de los datos y la clave, por lo general no lineal.
  - **No deben formar grupo**, para que el conjunto de varias pasadas sucesivas con sus subclaves correspondientes no sean equivalentes a una pasada única con una subclave diferente.
- La **transformación final** sirve para que las operaciones de cifrado y descifrado sean simétricas.

4.1 Características generales

- El **algoritmo de expansión de clave** tiene por objeto convertir la clave de usuario en un **conjunto de subclaves** que pueden estar constituidas por varios cientos de bits en total.
  - **Conviene que sea unidireccional** y que el conocimiento de una o varias subclaves intermedias no permita deducir las subclaves anteriores o siguientes.
  - Además, se ha de cuidar que las **subclaves producidas no constituyan un pequeño subconjunto monótono** de todas las posibles.

4.1 Características generales

ALGORITMOS MÁS UTILIZADOS

- AES
- DES (Triple DES)
- Serpent
- Blowfish, Twofish

OTROS ALGORITMOS

- Camellia
- CAST-128
- IDEA
- RC2, RC5, RC6
- SEED
- ARIA
- Skipjack
- TEA, XTEA

4.1 Características generales

- **Blowfish** fue creado por Bruce Schneier, autor del libro Applied Criptography (considerado por muchos como la "biblia" en cuestiones de criptografía). Utiliza claves de hasta 448 bits y, hasta el momento, ha resistido con éxito todos los ataques. Por ello y por su estructura se le considera uno de los algoritmos más seguros, a pesar de lo cual no se utiliza masivamente. Su autor no ha patentado el método para que pueda ser empleado sin limitaciones.

  La versión más actual es **Twofish**, que llegó a la ronda final del concurso AES del NIST (quedó tercero, tras Rijndael y Serpent). Cifra bloques de 128 bits con claves de hasta 256 bits.

- **Serpent** fue diseñado por Ross Anderson, Eli Biham y Lars Knudsen y quedó finalista en el concurso Advanced Encryption Standard del NIST, tras Rijndael que fue el ganador. Cifra bloques de 128 bits con claves de 128, 192 o 256 bits.

4.1 Características generales

- **DES** En enero de 1977 la *National Bureau of Standards (NBS)* de Los Estados Unidos de América publicó su *Federal Information Processing Standard* con el título de *Data Encription Standard*, en él se exponía el funcionamiento de un algoritmo de cifrado estándar que debía ser utilizado por todas las Agencias Federales para la protección criptográfica de datos informáticos de naturaleza reservada pero **no secreta**.

- DES es un algoritmo de cifrado en bloque de los denominados redes Feistel. La longitud de bloque es de 64 bits y la de clave es 56 bits.

- DES ha sido el estándar utilizado mundialmente durante más de 25 años, generalmente en la banca. Hoy presenta signos de envejecimiento y ha sucumbido a los diversos criptoanálisis que contra él se vienen realizando. No obstante se sigue utilizando (TDES).

- **TripleDES** cifrado múltiple, *EDE* $c = E_{k_1}\left(D_{k_2}\left[E_{k_1}(m)\right]\right)$

4.1 Características generales

- **AES.** El 2 de octubre de 2000 el NIST (*National Institute for Standards and Technology*) anunciaba oficialmente la adopción del algoritmo ***Rijndael*** como nuevo Estándar Avanzado de Cifrado (**AES**) para su empleo en aplicaciones criptográficas no militares, culminando así un proceso de mas de tres años, encaminado a proporcionar a la comunidad internacional un nuevo algoritmo de cifrado potente, eficiente y fácil de implementar.

- La palabra ***Rijndael*** es un acrónimo formado por los nombres de sus dos autores, los belgas
  - *Vincent Rijmen* y *Joan Daemen*.

- Su interés radica en que todo el proceso de selección, revisión y estudio tanto de este algoritmo como de los restantes candidatos, se ha efectuado de forma pública y abierta, por lo que, prácticamente por primera vez, toda la comunidad criptográfica mundial ha participado en su análisis, lo cual convierte a ***Rijndael*** en un algoritmo perfectamente digno de la confianza de todos.

4.1 Características generales

- Los tres aspectos básicos sobre los que se ha diseñado **AES** son los siguientes:
  - Resistencia contra todo tipo de ataque conocido hasta ese momento.
  - Eficiencia computacional en un amplio abanico de plataformas, tanto hardware como software (optimizado para 32 bits).
  - Simplicidad de diseño.

- AES es un sistema de cifrado por bloques cuya longitud de bloque es de 128 bits, diseñado para manejar longitudes de clave variables: 128, 192 y 256 bits.

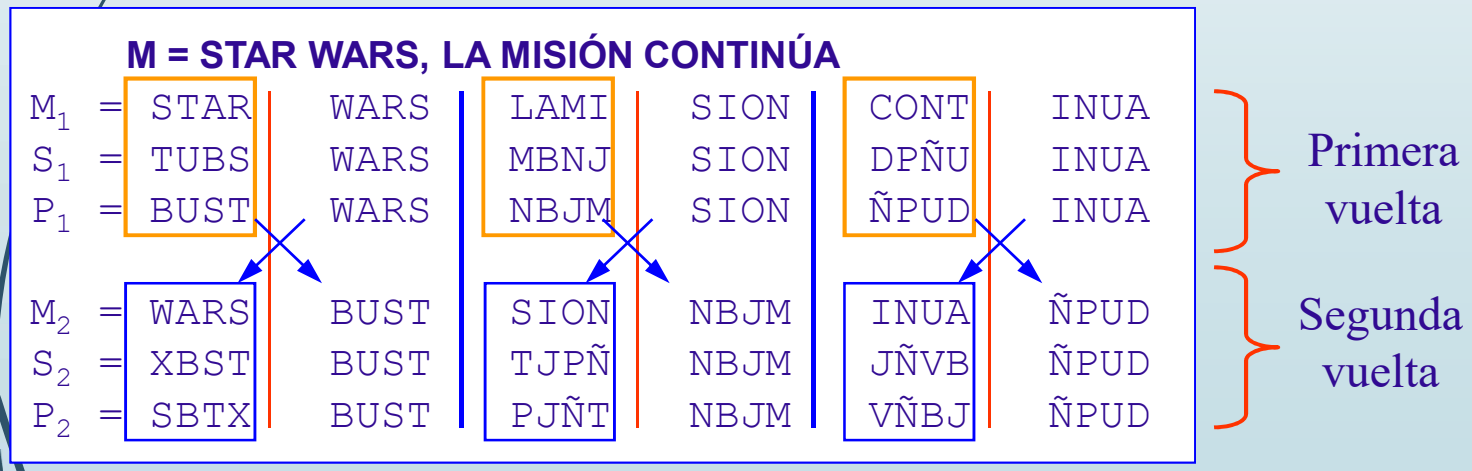
Cifrado tipo Feistel

Se denomina cifradores ***tipo Feistel*** a aquellos en los que el bloque de datos se divide en dos mitades y en cada vuelta de cifrado se trabaja, alternada-mente, con una de las mitades

EJEMPLO: 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 S: Desplazamiento +1 mod 27

Permutación P: σ=(3 2 4 1)



4.2 DES (Data Encryption Standard)

- En enero de 1977 la *National Bureau of Standards (NBS)* de Los Estados Unidos de América publicó su *Federal Information Processing Standard* con el título de *Data Encription Standard*, en él se exponía el funcionamiento de un algoritmo de cifrado estándar que debía ser utilizado por todas las Agencias Federales para la protección criptográfica de datos informáticos de naturaleza reservada pero no secreta.

- Este método fue el resultado de la investigación realizada por IBM durante los años 1968 a 1975 inspirada en el sistema de Horst Feistel denominado *Lucifer*, consistente en una composición de diferentes transformaciones tal como proponía Shannon en su artículo "*Communication Theory of Secrecy Systems*" (Octubre 1949).

4.2 DES (Data Encryption Standard)

- DES es un algoritmo de cifrado en bloque de los denominados **redes Feistel**. La longitud de bloque es de 64 bits (ocho símbolos ASCII) y la de clave es 56 bits.

- DES ha sido el estándar utilizado mundialmente durante más de 25 años, generalmente en la banca.
  - Hoy no es muy utilizado dado que ha sucumbido a los diversos ataques que contra él se han realizando.
  - No obstante se sigue utilizando (TDES).

4.2.1 Introducción y descripción

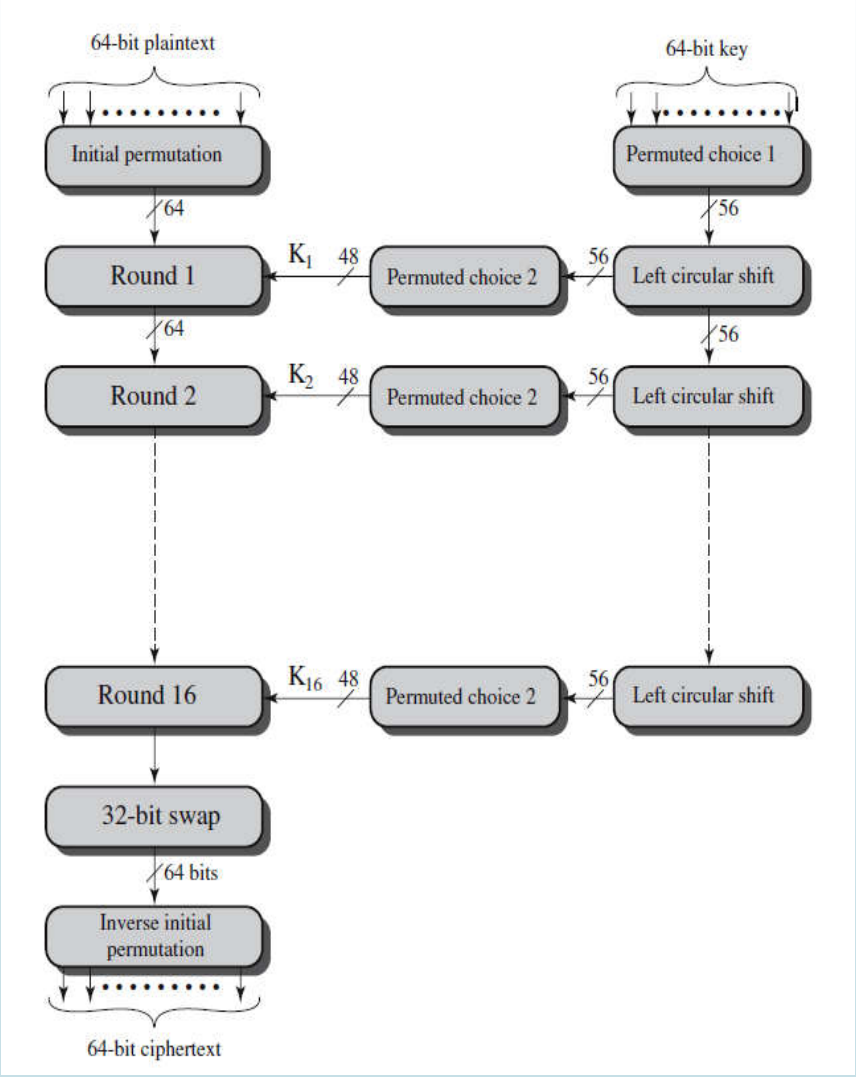
- La NSA fue la que impuso la longitud de clave del DES, que es bastante modesta y que la hace desaconsejable con el actual desarrollo de la informática.

- DES es un algoritmo de cifrado en bloque; la longitud de bloque es de 64 bits (ocho símbolos ASCII); la longitud de la clave es de 56 bits, lo que equivale a que existan

$$2^{56} = 7'2 \cdot 10^{16} \text{ claves diferentes.}$$

4.2.1 Introducción y descripción

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



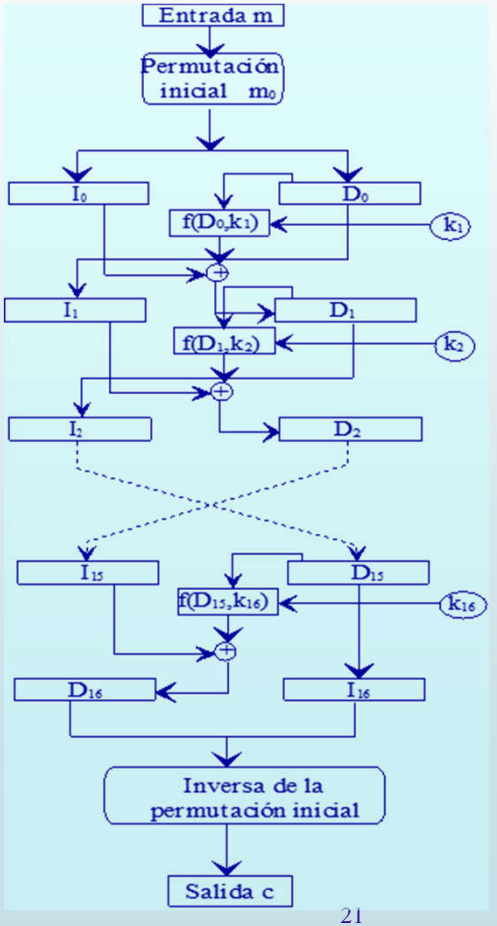
4.2.1 Introducción y descripción

- El algoritmo es válido para el cifrado y descifrado de bloques de 64 bits mediante una clave de 56 a la que se añaden 8 de paridad.
- Si $m=m_1m_2...m_{64}$ es un bloque de 64 bits, se realiza una permutación inicial

$$m_0=m_{58}m_{50}...m_7$$

utilizando la siguiente tabla

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
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



4.2.1 Introducción y descripción

- Este bloque m_0 es dividido en dos subbloques de 32 bits cada uno

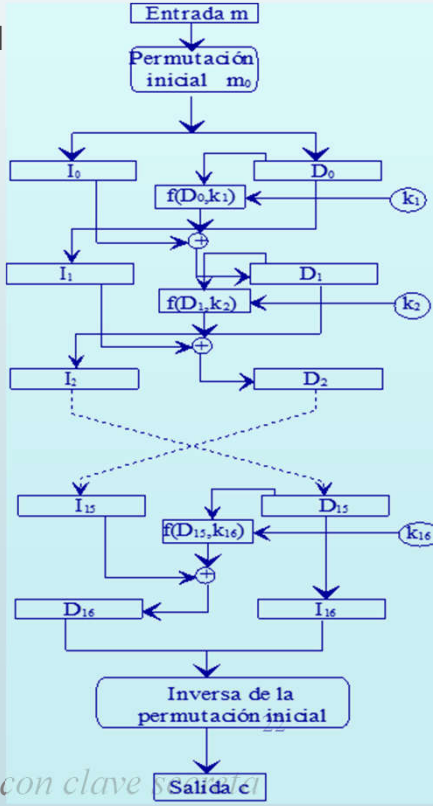
$$I_0=m_{58}m_{50}...m_{16}m_8 \text{ y } D_0=m_{57}m_{49}...m_{15}m_7$$

- A continuación se realizan sobre estos dos bloques **16 transformaciones** que combinan **sustituciones y transposiciones**.

- Al bloque resultante de concatenar D_{16} e I_{16} se le aplica la permutación inversa a la inicial, dada en la tabla,

40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

para obtener el bloque cifrado c, de 64 bits.



4.2.2 La función f y las cajas S_i

- Entre la permutación inicial y la final el algoritmo realiza 16 iteraciones que describiremos a continuación.

- Denotemos por

$$T_i=t_1...t_{32}\ t_{33}...t_{64}= I_i\ D_i$$

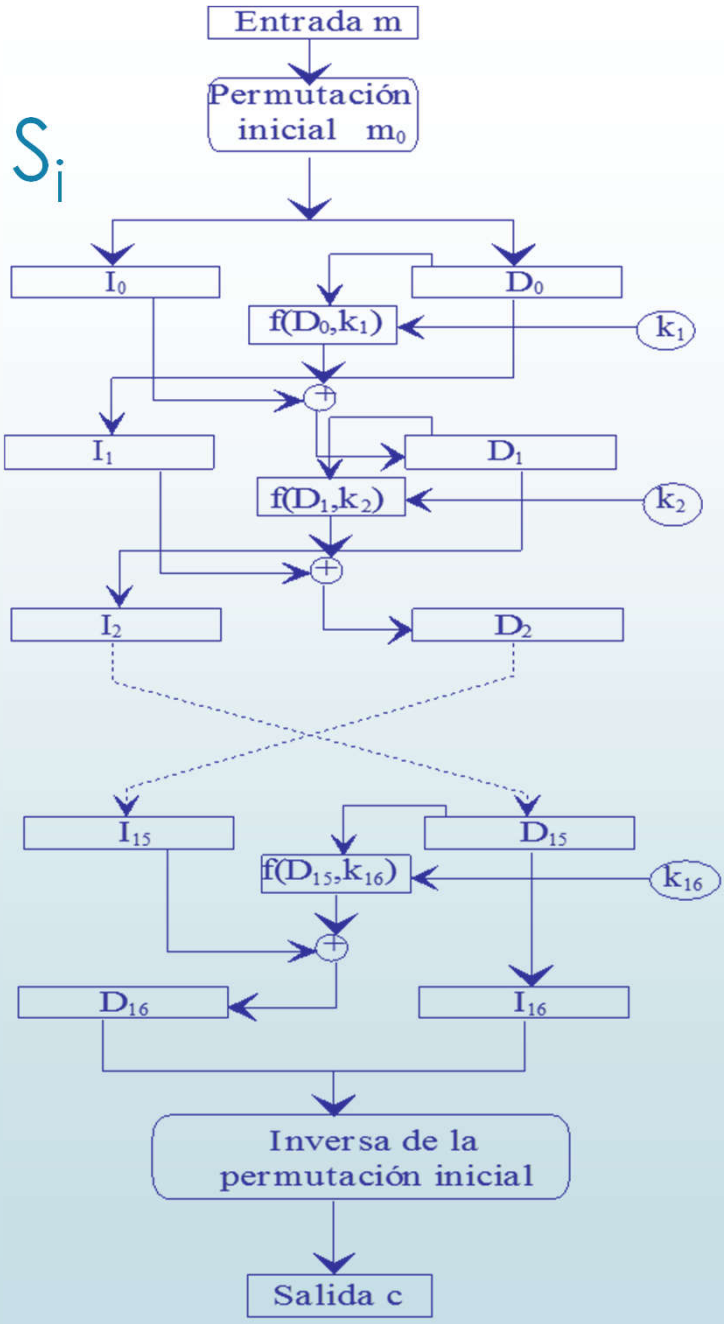
al resultado obtenido tras la iteración i-ésima, donde I_i representa los 32 primeros bits de T_i y D_i los 32 últimos, esto es

$$I_i=t_1...t_{32} \quad y \quad D_i=t_{33}...t_{64}$$

- Se define

$$I_i=D_{i-1} \quad y \quad D_i=I_{i-1}\oplus f(D_{i-1},k_i) \qquad i=1,2,...16$$

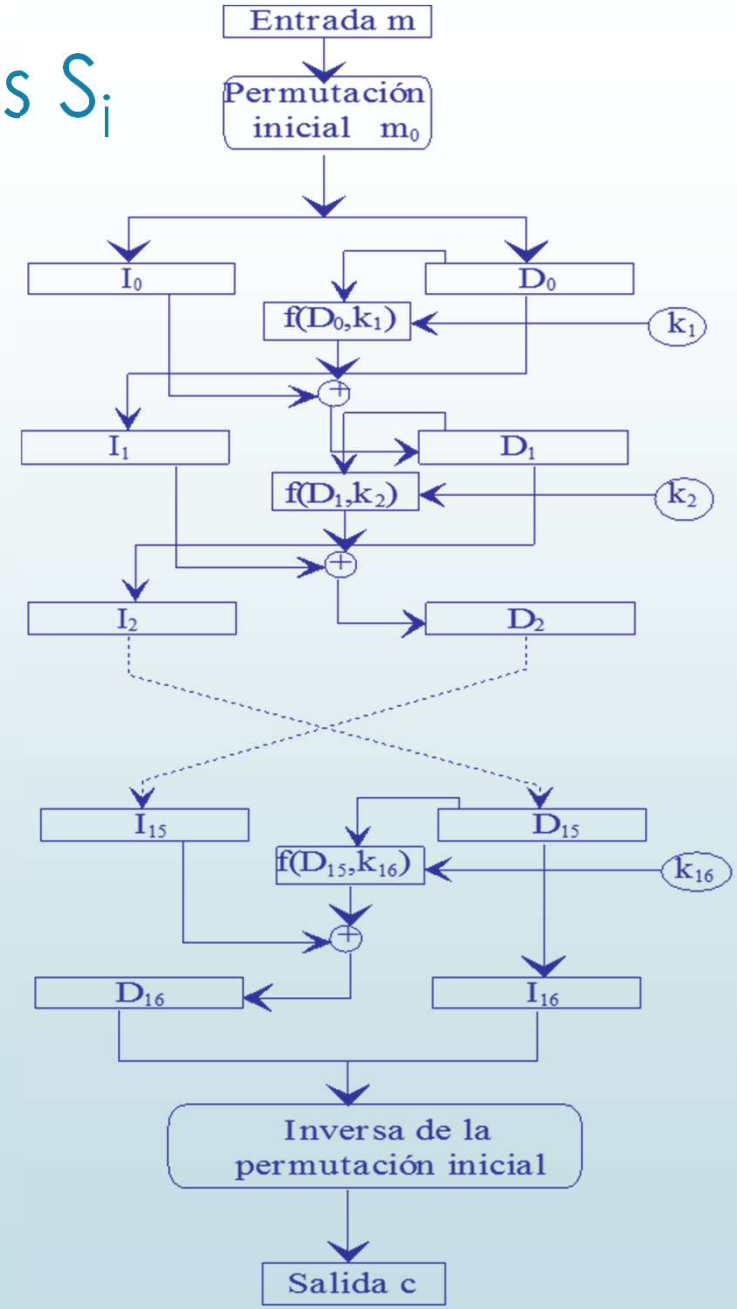
donde $\oplus$ es la operación OR-exclusiva y k_i es una subclave de 48 bits que se obtiene a partir de la clave original k.



4.2.2 La función f y las cajas S_i

- Se puede observar que en la última iteración las dos mitades, izquierda y derecha, no son cambiadas, siendo el bloque $T_{16}=D_{16}\,I_{16}$ al que se le aplica la permutación final. Esto es necesario para que el algoritmo sea válido para cifrar y descifrar.
- La función f transforma, mediante la subclave k_i, los bloques D_i en los bloques de 32 bits $f(D_{i-1},k_i)$.
- Para ello, en primer lugar, se produce una **expansión E de los 32 bits de D_{i-1} para obtener un bloque de 48 bits**, de tal manera que si numeramos los bits de D_{i-1} de 1 a 32

$$D_{i-1}=d_1d_2...d_{32}$$

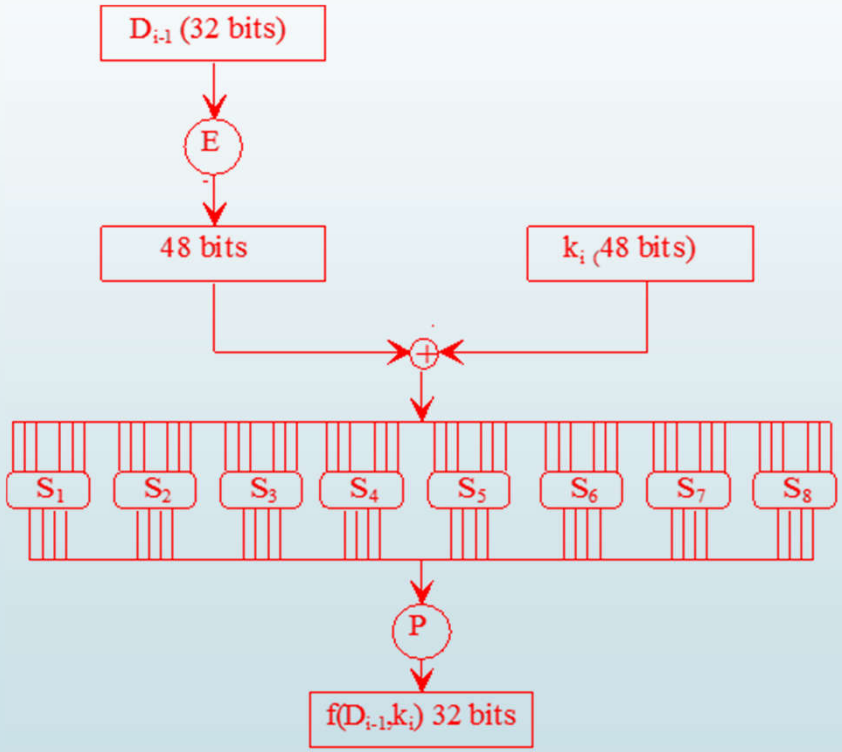


Cifrado en bloque con clave secreta

4.2.2 La función f y las cajas S_i

- La expansión se obtiene permutando dichos bits de acuerdo con el orden de la tabla

32	01	02	03	04	05
04	05	06	07	08	09
08	09	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	01



esto es

$$E(D_{i-1})=d_{32}d_1d_2...d_{32}d_1$$

- Una vez obtenido el bloque de 48 bits se efectúa la operación $\oplus$ entre $E(D_{i-1})$ y la subclave k_i. El resultado es dividido en 8 bloques B_i de 6 bits cada uno

$$E(D_{i-1}) \oplus k_i=B_1B_2...B_8$$

4.2.2 La función f y las cajas S_i

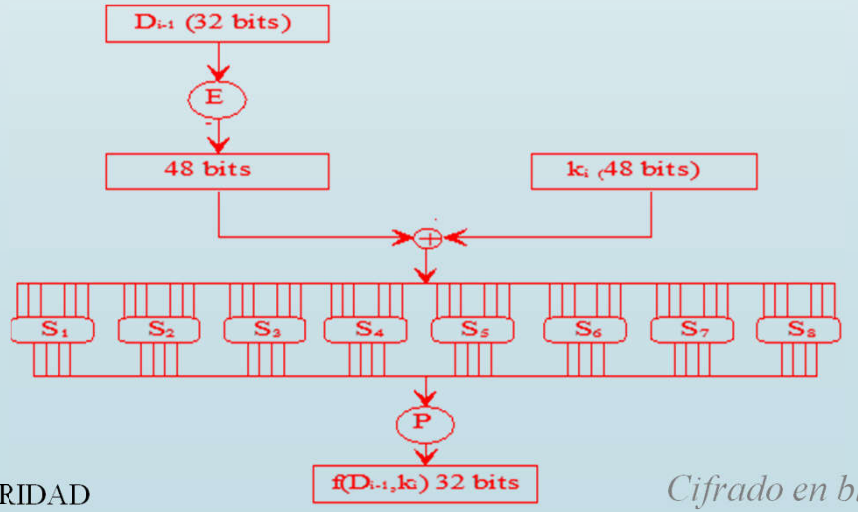
- Cada uno de estos bloques B_i es utilizado como entrada para una **función de selección-sustitución S_i** que tiene como salida un bloque de 4 bits, de manera que si un bloque

$$B_i=b_1b_2b_3b_4b_5b_6$$

la salida se obtiene de la siguiente manera:

  - **$b_2b_3b_4b_5$** representa un número binario cuyo valor decimal está comprendido entre **0 y 15**, este selecciona una **columna en la tabla**.
  - **b_1b_6** representa un número binario cuyo valor decimal está comprendido entre **0 y 3**. Este número selecciona **una fila de la tabla**.
  - La salida de **$S_i(B_i)$** es la expresión binaria del **número seleccionado en la tabla**.

- Se obtienen así **8 bloques de 4 bits** que se concatenan.



4.2.2 La función f y las cajas S_i

S1	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	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

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

S3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

S4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	9	6	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

S5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

S6	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

S7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

S8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

No lineal y unidireccional *(hay cuatro soluciones de entrada para cada salida)*

4.2.2 La función f y las cajas S_i

S1	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	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

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

S3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

S4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	9	6	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

S5																15
0																11
1																8
2																6
3																13

S7																15
0																7
1																2
2																8
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

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

Para seleccionar la columna tomamos los bits centrales: 0110 = 6

La caja S_2 indica una salida igual a 13 = 1101

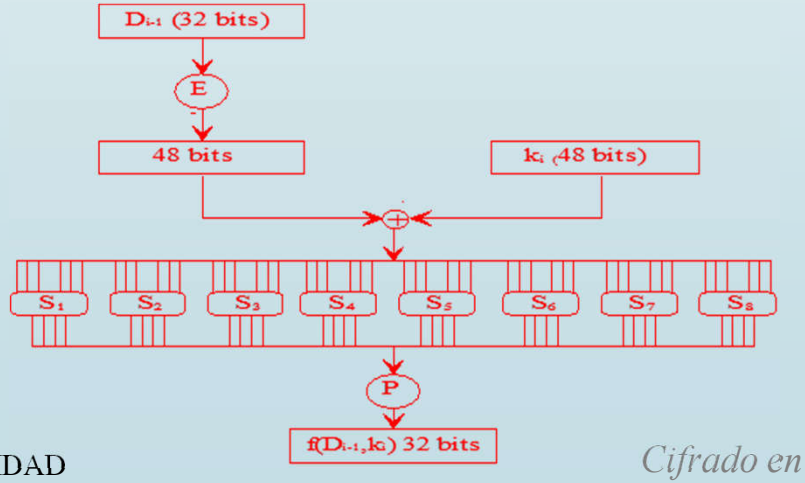
4.2.2 La función f y las cajas S_i

- Finalmente la salida de la función f se obtiene aplicando la permutación P

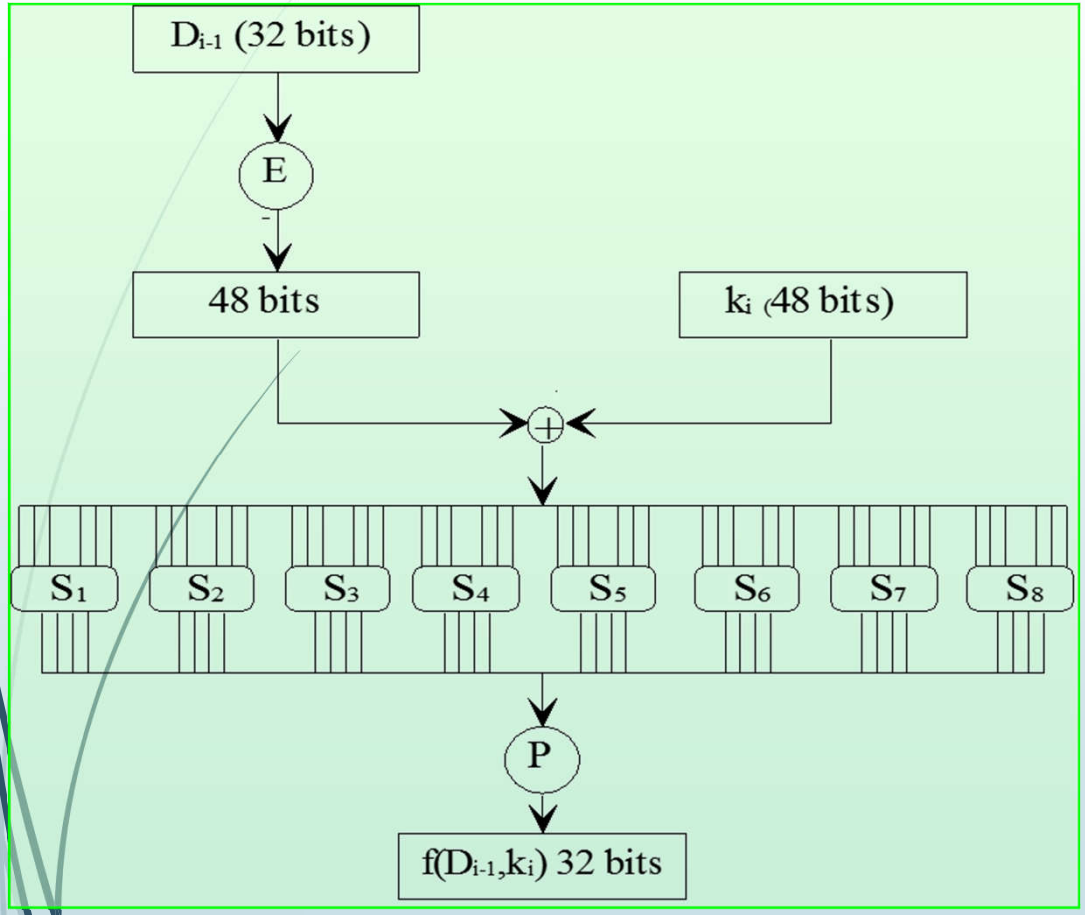
16	07	20	21
29	12	28	17
01	15	23	26
05	18	31	10
02	08	24	14
32	27	03	09
19	13	30	06
22	11	04	25

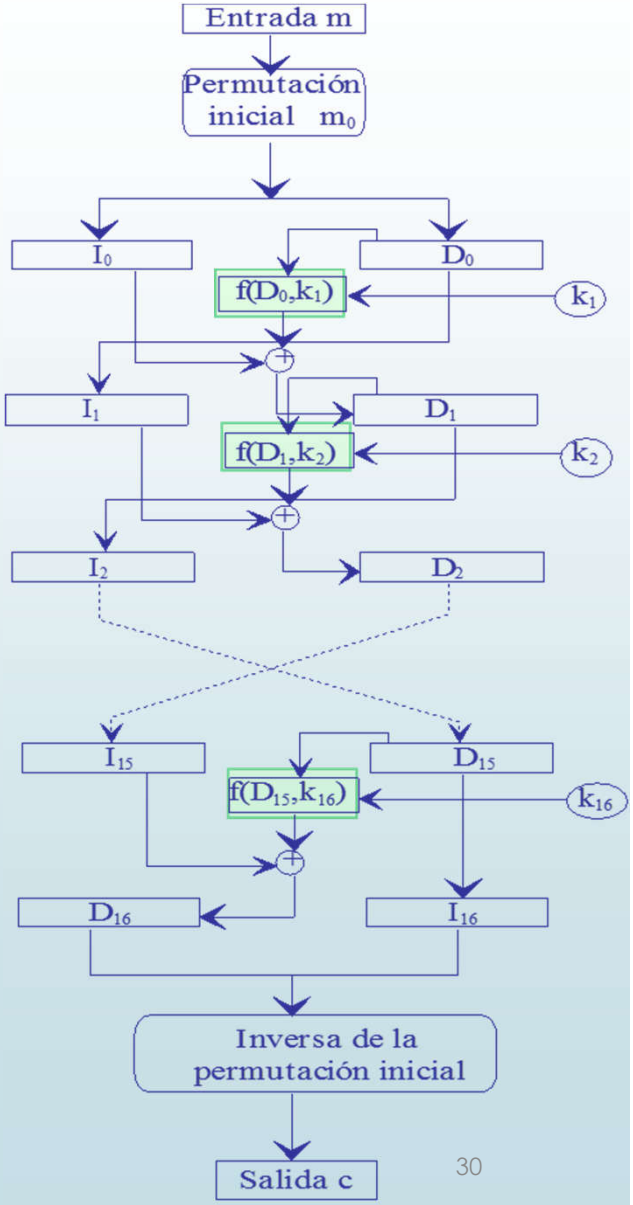
sobre el bloque de 32 bits obtenido, esto es

$$f(D_{i-1},k_i) = P[S_1(B_1)\ S_2(B_2)\ S_3(B_3)\ S_4(B_4)\ S_5(B_5)\ S_6(B_6)\ S_7(B_7)\ S_8(B_8)]$$



4.2.2 La función f y las cajas S_i

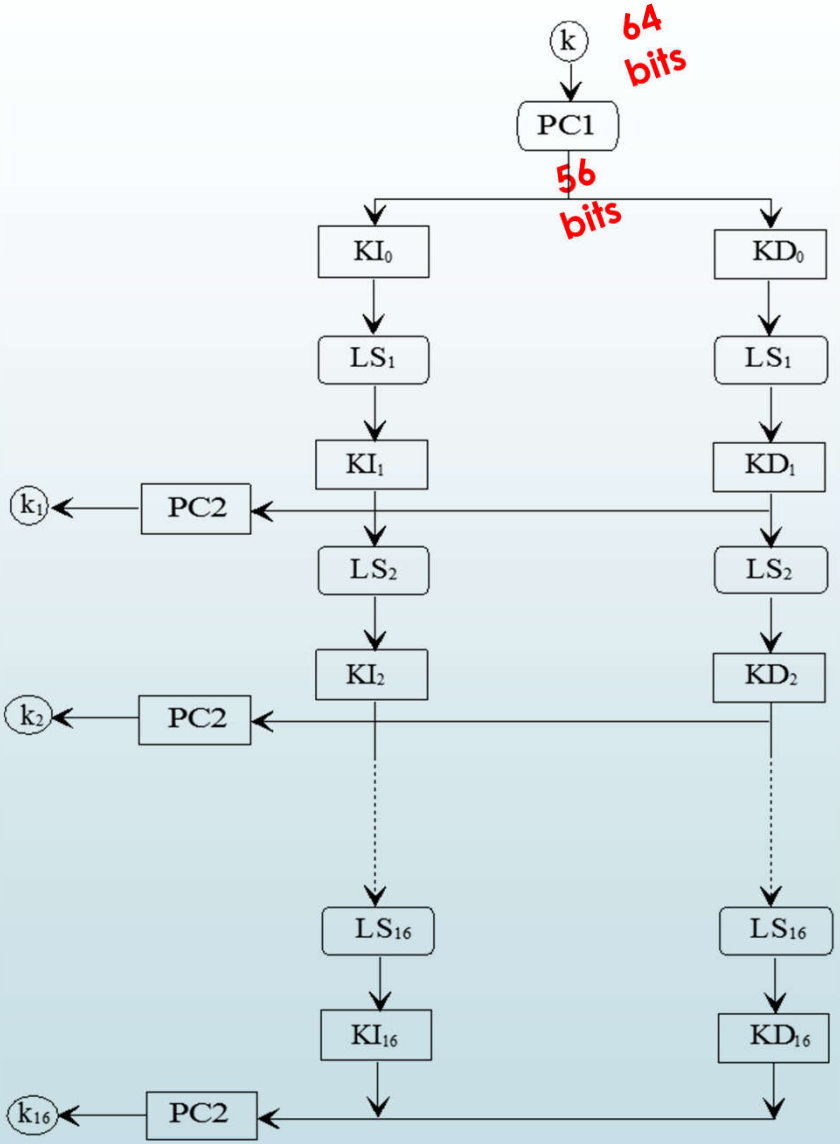




4.2.3 Cálculo de las subclaves k_i

- La clave k **tiene inicialmente 64 bits**. Los que ocupan las posiciones múltiplo de ocho: 8,16,24,32,40,48,56,64, controlan la paridad de los siete anteriores.
- En cada iteración se obtiene una subclave **k_i a partir k**.
- En primer lugar se eliminan los ocho bits de paridad de la clave y se aplica la permutación PC1 (permuted choice 1)

57	49	41	33	25	17	09
01	58	50	42	34	26	28
10	02	59	51	43	35	27
19	11	03	60	52	44	36
63	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



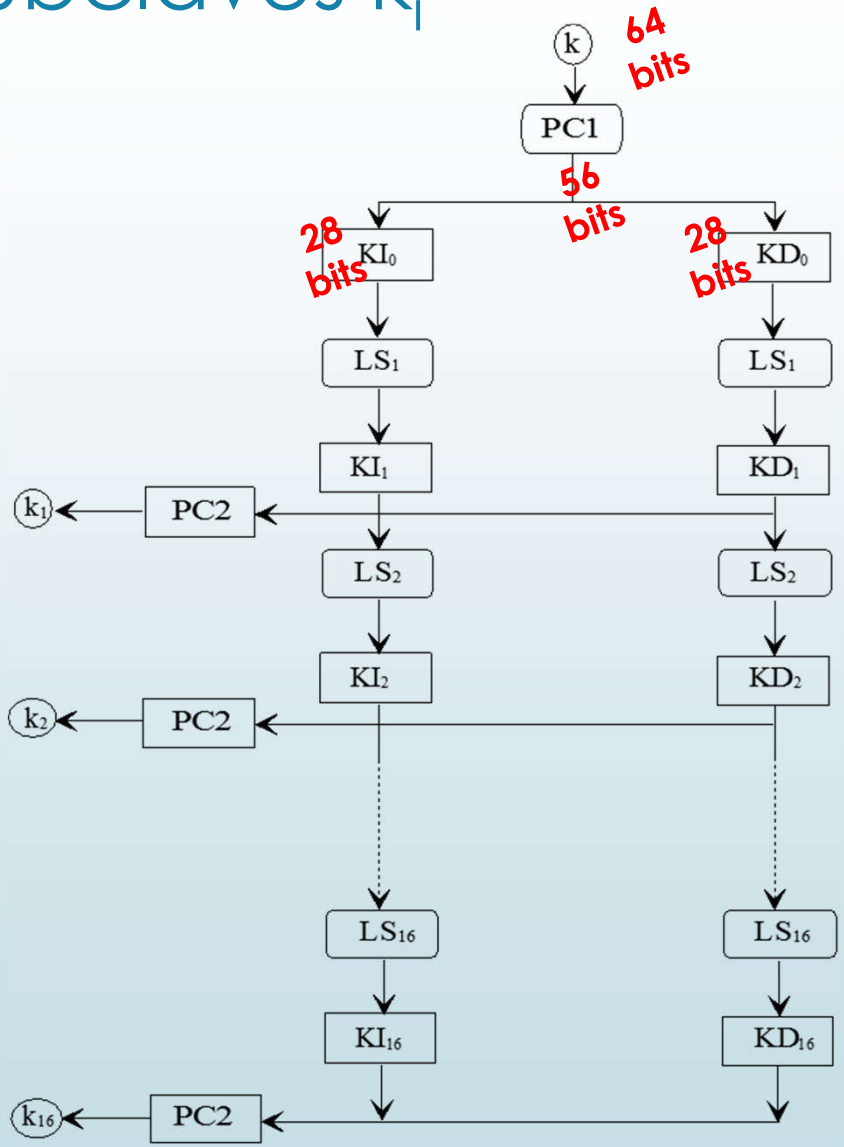
4.2.3 Cálculo de las subclaves k_i

- Una vez efectuada la permutación, los 56 bits se **dividen** en dos bloques de 28 cada uno, KI_0 y KD_0

$$PC1(k) = KI_0\ KD_0$$

- La obtención de los bloques siguientes, **KI_i KD_i se hace siempre a partir del bloque anterior KI_{i-1} KD_{i-1}** efectuando cada mitad una rotación a la izquierda que depende de cada iteración de acuerdo con

I	bits	I	bits
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



4.2.3 Cálculo de las subclaves k_i

- Se tiene

$$KI_i=LS_i(KI_{i-1}) \qquad KD_i=LS_i(KD_{i-1})$$

donde LS_i representa el desplazamiento a **izquierda** a efectuar en la iteración i.

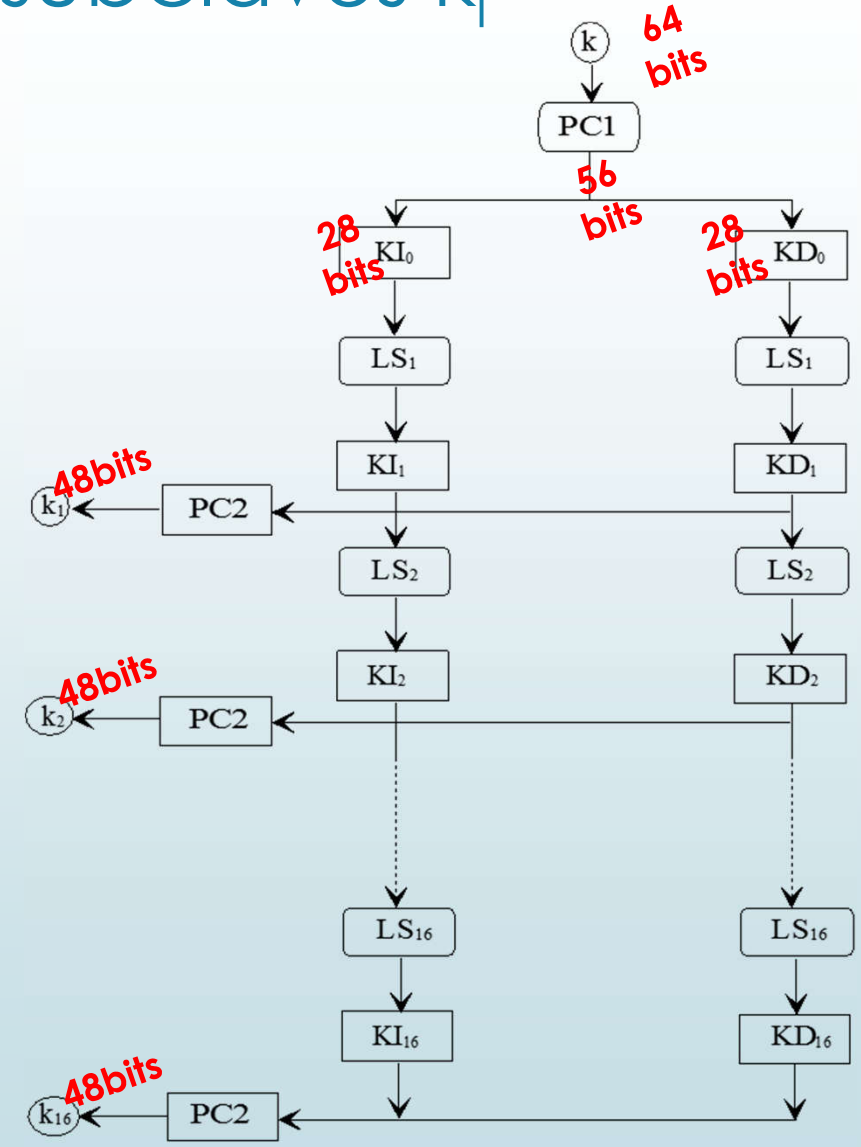
- La subclave k_i viene dada por

$$k_i=PC2(KI_i||KD_i)$$

siendo PC2 la permutación dada en la tabla

14	17	11	24	01	05
03	28	15	06	21	10
23	19	12	04	26	08
16	07	27	20	13	02
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 y 54



4.2.4 Descifrado de DES

- Este algoritmo, como ya se ha comentado con anterioridad, es válido para descifrar los criptogramas.
- La única variación estriba en la utilización de las subclave k_i que se efectúa en orden inverso, o sea, k_{16} se utiliza en la primera iteración, k_{15} en la segunda y así sucesivamente; esto es así porque la iteración final es la inversa de la inicial. Se tiene además

$$D_{i-1}=I_i\ ,\ I_{i-1}=D_i \oplus f(I_i,k_i)$$

- Para la obtención de las distintas subclaves se puede utilizar únicamente k_{16} y aplicar las rotaciones en orden inverso, de acuerdo con la tabla

I	bits	I	bits
1	0	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

4.2.5 Propiedades y características de DES

- ***Dependencia entre símbolos***: Cada bit del texto cifrado es una función compleja de **TODOS** los bits de la clave y **TODOS** los bits del texto original (por bloques).

- ***Cambio de los bits de entrada***: Un cambio de un bit en el mensaje original produce el cambio del *50 %*, aproximadamente, de los bits del bloque cifrado.

- ***Cambio de los bits de clave***: Un cambio en un bit de la clave produce, aproximadamente, el cambio de la mitad de los bits del bloque cifrado.

4.2.5 Propiedades y características de DES

- ***Claves débiles***: Existen cuatro claves **«débiles»** que producen un mensaje cifrado fácil de descifrar, porque todas las claves parciales k_1 a k_{16} son iguales. Existen 28 claves **«semidébiles»** que producen un mensaje cifrado fácil de descifrar, porque producen sólo dos o cuatro subclaves parciales diferentes. Cuando se elige una clave al azar, es preciso asegurarse de que no se ha producido una de estas claves.

Clave	Clave tras aplicar PC1
0101010101010101	00000000000000
1F1F1F1F0E0E0E0E	0000000FFFFFFF
E0E0E0E0F1F1F1F1	FFFFFFF0000000
FEFEFEFEFEFEFEFE	FFFFFFFFFFFFFF

Clave	Clave tras aplicar PC1
01FE01FE01FE01FE	AAAAAAAAAAAAAA
E01FE01FE01FE01	55555555555555
1FE01FE00EF10EF1	AAAAAAA5555555
E01FE01FF10EF10E	5555555AAAAAAA
01E001E001F101F1	AAAAAAA0000000
E001E001F101F101	55555550000000
1FFE1FFE0EFE0EFE	AAAAAAAFFFFFFF
FE1FFE1FFE0EFE0E	5555555FFFFFFF
011F011F010E010E	0000000AAAAAAA
1F011F010E010E01	00000005555555
E0FEE0FEF1FEF1FE	FFFFFFFAAAAAAA
FEE0FEE0FEF1FEF1	FFFFFFF5555555

4.2.6 Seguridad de DES

- En 1976, antes de que el *National Bureau of Standards* aprobara el proyecto DES se encargaron dos estudios sobre la seguridad del algoritmo. **Diffie, Hellman y otros encontraron ciertas debilidades en el mismo**.

- El tamaño de la clave, 56 bits, hace que el sistema sea vulnerable dado que el conjunto de claves resulta demasiado pequeño:

2^{56} posibilidades.

4.2.6 Seguridad de DES

- Martin Hellman en su artículo "*DES will be totally insecure within ten years*" (1979), aseguraba que un millón de procesadores trabajando en paralelo, cada uno de ellos procesando un millón de claves por segundo, recorrerían todo el espacio de claves en poco más de veinte horas.
  - Como es probable que, en promedio, sólo se tenga que buscar en la mitad del conjunto de claves, un ataque al criptosistema con una máquina de tales características, conociendo un texto en claro y su correspondiente criptograma necesitaría de **poco más de diez horas** para encontrar la clave

$$\frac{2^{56}}{3'6\cdot 10^{15}} = \frac{7'2\cdot 10^{16}}{3'6\cdot 10^{15}} = 20$$

4.2.6 Seguridad de DES

- El **18 de junio de 1997**, un esfuerzo coordinado a través de Internet, en respuesta a un desafío de RSA Data Security, Inc. permitió el descifrado de un mensaje cifrado con una clave DES de 56 bits utilizando la **fuerza bruta**.

- El tiempo invertido desde que comenzó el análisis de las posibles claves hasta que se consiguió descifrar el mensaje fue de **cuatro meses**.

- Sin embargo, se trataba de un ataque no demasiado práctico, puesto que se tardaron cuatro meses en "reventar" un único mensaje.

4.2.6 Seguridad de DES

- RSA Data Security, Inc. lanzó una nueva serie de desafíos, estableciendo premios, de modo que estos sólo se ganaran si el tiempo empleado en reventar una clave era inferior al 25% del empleado en el desafío exitoso anterior.

- El **13 de enero de 1998** se lanzó el **primero de los dos desafíos** conocidos como **DES-II Challenges**.

- La organización distributed.net utilizó el esfuerzo cooperativo de cientos de ordenadores (unos 50.000) a través de Internet, para reventar la clave utilizando la fuerza bruta.

- Tardó **39 días** en conseguir sus propósitos.

4.2.6 Seguridad de DES

- La puntilla llegó el **17 de julio de 1998**. Tras el lanzamiento del **segundo desafío** (el 13 de julio de 1998) y, con el fin de demostrar la inseguridad de DES, la Electronic Frontier Foundation anunció la construcción de una máquina especializada diseñada para reventar mensajes cifrados con DES

  Esta máquina, bautizada apropiadamente como **DES Cracker**, fue construida por menos de 250.000 dólares (unos 40 millones de pesetas), y fue capaz de ganar fácilmente el segundo desafío, tardando apenas tres días.

- Los detalles de diseño de esta máquina se encuentran completamente documentados en el libro publicado por la propia EFF y O'Reilly and Associates, titulado "*Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design*".
  - **Curiosamente, la exportación de la propia máquina no está permitida por las leyes de EE.UU. El libro, sin embargo, puede ser comprado libremente y utilizado para la implementación del DES Cracker en cualquier sitio**.

4.2.6 Seguridad de DES

19-01-1999 - DES III

- En un esfuerzo conjunto de distributed.net y EFF con el mítico DES Cracker, se consiguió romper el estándar de cifrado en menos de un día:
  - En el **tiempo récord de 22 horas y 15 minutos** el **desafío DES III** sucumbió ante el poder de unos 100.000 PCs y un super-ordenador especialmente diseñado por la EFF.
  - El equipo combinado comprobó más de **245.000 millones de claves cada segundo**.
- En esta ocasión, el escrito tras la clave ocultaba la siguiente misiva ***"See you in Rome (second AES Conference, March 22-23, 1999)***".

4.2.6 Seguridad de DES

- La resolución en este tiempo récord vino a demostrar muchas cosas en el mundo de la informática y la seguridad.
  - En primer lugar la debilidad del algoritmo DES y la necesidad de buscar un nuevo **sustituto** de forma inmediata y
  - por otro lado la efectividad del uso de la **computación distribuida** para la resolución de problemas que requieran una gran potencia de cálculo.

4.2.6 Seguridad de DES

- En 1996, el Instituto Nacional de Estándares y Tecnología (*National Institute of Standards and Technology*, NIST) dio los primeros pasos para la consolidación de **un Estándar de Cifrado Avanzado** (*Advanced Encryption Standard*, AES).

- Su objetivo fue desarrollar una especificación para encontrar un algoritmo de cifrado que sustituyese a DES de manera que el nuevo algoritmo fuese capaz de proteger la información sensible de los ciudadanos y del gobierno hasta bien entrado el siglo XXI.