CRIPTOGRAFIA DE CLAVE SECRETA.

Introducción.

Los sistemas de clave secreta, al contrario que los de clave pública, tienen una única clave que se utiliza tanto para el cifrado como para el descifrado. Este esquema de clave secreta ha sido el predominante a través de la historia y solo ha presentado un serio problema con el advenimiento de las comunicaciones en masa. La necesidad de comunicar una gran cantidad de equipos y de intercambiar las claves entre ellos para la comunicación fue la principal razón del desarrollo de los sistemas de clave pública. Aunque inicialmente estos últimos parecía que iban a eliminar los sistemas de clave secreta, principalmente por dos razones, los sólidos fundamentos matemáticos y la eliminación del problema de intercambio de claves, el tiempo y la experiencia han dejado las cosas en su sitio. Los sistemas de clave pública son empleados, debido fundamentalmente a su falta de rendimiento, como sistemas de intercambio de claves para los sistemas de clave secreta, mucho más rápidos, que son los que se utilizan para cifrar los grandes volúmenes de información que requieren los sofisticados sistemas actuales.

Otra diferencia básica en ambos sistemas, clave pública y privada, es la casi exclusiva utilización por parte de los últimos de los conceptos de difusión y confusión presentados por Shannon. Los sistemas de clave secreta, exceptuando algún caso como el de Pohlig-Hellman, utilizan permutaciones y sustituciones complejas para lograr el secreto. Existen dos métodos de cifrado en clave secreta, el cifrado en bloque y el cifrado en flujo. Los sistemas de cifrado en bloque cifran una cantidad fija de bits de información en cada paso del algoritmo, los sistemas de cifrado en flujo por su parte pretenden simular el one-time pad mediante la utilización de algoritmos de generación de números aleatorios criptográficamente fuertes para generar una clave continua. A continuación se presentan algunos de los más representativos que nos darán una idea de las técnicas utilizadas en ambos.

Sistemas de cifrado en bloque.

Algoritmo de Pohlig-Hellman.

La mayoría de los algoritmos simétricos se basan en los conceptos de confusión y difusión de Shannon y utilizan una sola clave para cifrado y descifrado, sin embargo, el algoritmo de Pohlig-Hellman hace uso de las funciones trampa propias de los algoritmos de clave pública y necesita dos claves, una para cifrado y otra para descifrado, sin embargo ambas deben ser secretas. El algoritmo, muy similar al RSA en concepto, se diferencia básicamente en la elección del módulo, que en este caso no tiene por que ser el producto de dos primos. Se escoge un número n grande de la forma $n = 2n'+1$ y dos números e y d tales que $e.d = 1 \bmod n$, tanto e como d y n deben permanecer secretos. Las operaciones de cifrado C y descifrado D son respectivamente

$$C = M^e \bmod n$$
$$D = C^d \bmod n \,.$$

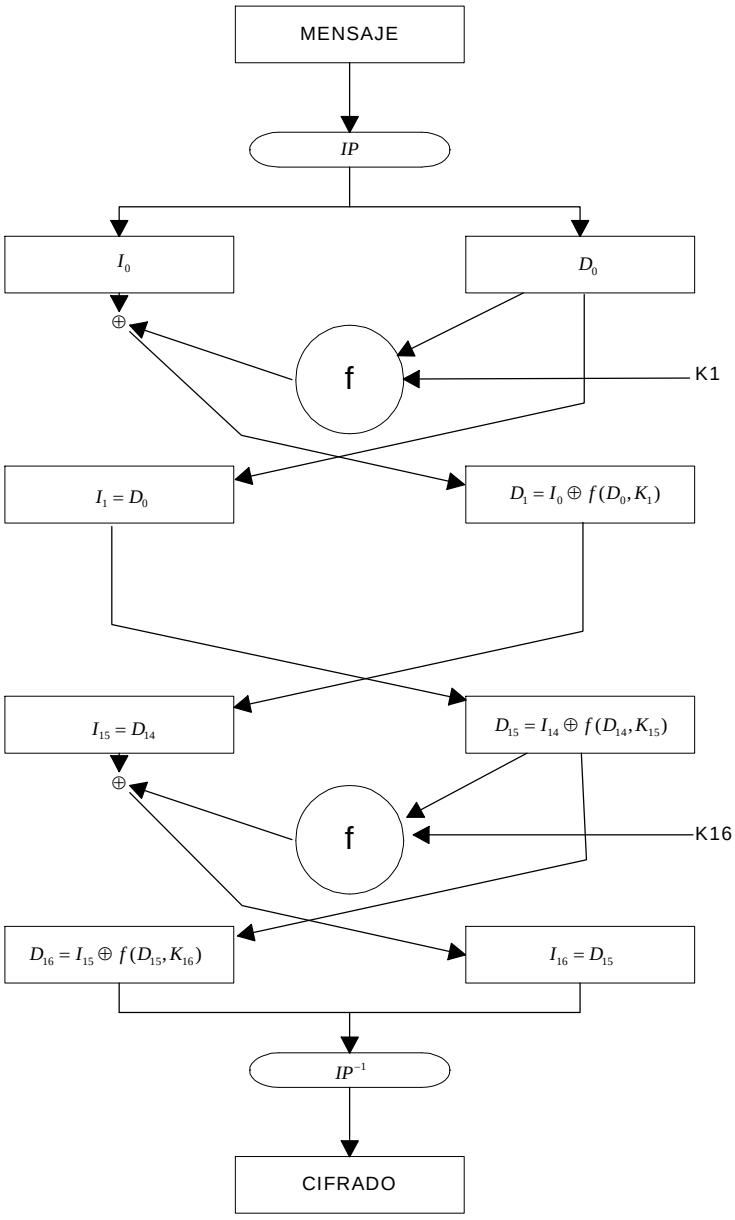
Las operaciones se realizan sobre Campos de Galois, con lo que la seguridad del algoritmo depende de la dificultad de realizar logaritmos discretos en Campos de Galois.

Data Encryption Standard (DES).

El DES o DEA (Data Encryption Algorithm) nació a raíz de una petición del National Bureau of Standards (NBS) en 1.973 de un algoritmo de cifrado para la protección de datos en ordenadores. Entre otros se presentó un algoritmo desarrollado por IBM que fue el finalmente elegido. La descripción del mismo fue publicada el 15 de enero de 1.977 en el Federal Information Processing Standard Publication número 46 (FIPS PUB 46).

El algoritmo consiste en la aplicación consecutiva de 16 pasos más una permutación inicial y una final a los datos a cifrar previamente divididos en bloques de 64 bits. En realidad la clave tiene una longitud de 56 bits, siendo los restantes, hasta llegar a los 64 antes indicados, bits de paridad.

El esquema del algoritmo es el siguiente:

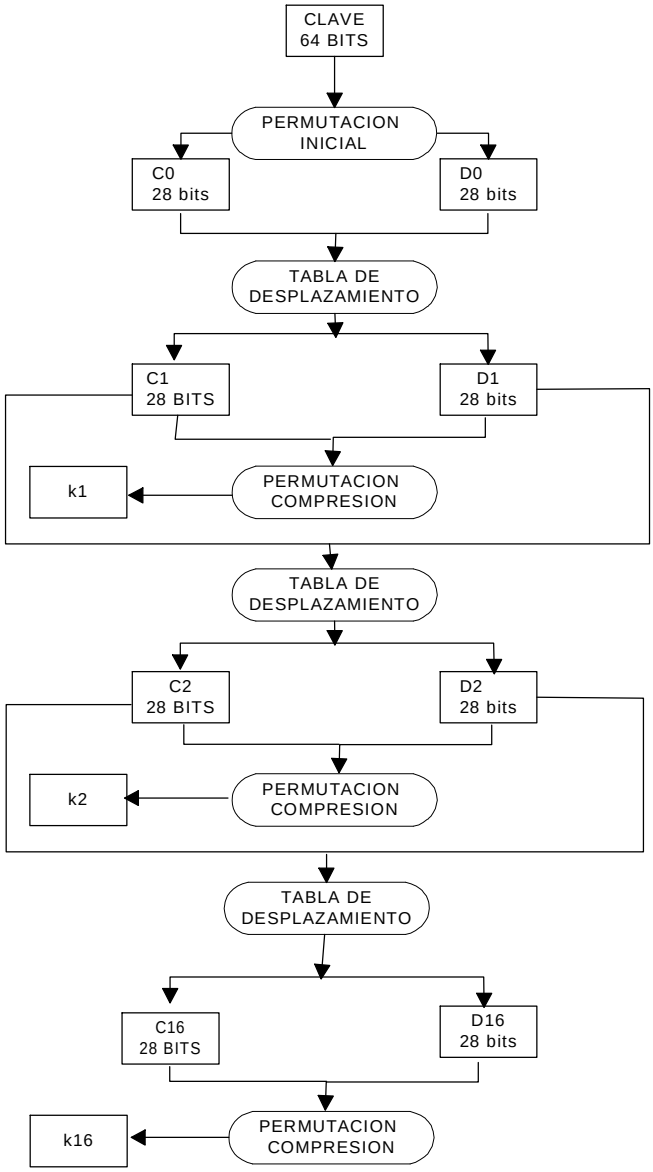


Los 64 bits del mensaje son sometidos a una permutación inicial y posteriormente el resultado es dividido en dos bloques de 32 bits que en el dibujo anterior hemos marcado como

I_0 e I_1. La permutación inicial *IP* procede a la reordenación de los bits según la tabla siguiente.

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

Una vez realizada la permutación inicial, se procede a realizar los 16 pasos de cifrado, en cada uno de ellos se procesará una parte de la clave junto con los bits resultantes del proceso anterior. La clave se somete en cada paso a un proceso de transformación para obtener 48 bits que se combinaran con una expansión de 48 bits de los datos en transformación. El proceso de transformación de la clave se muestra en el dibujo siguiente.



Se aplica a la clave una permutación inicial dada por la tabla KPI y posteriormente se divide en dos mitades que denominamos C y D a las que se le aplica un desplazamiento de bits marcado por la tabla de desplazamientos de clave KDC. Se vuelve a juntar la clave y se le aplica una permutación con compresión que viene dada por la tabla KPC, con lo que se consigue una nueva subclave de 48 bits.

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

Permutación inicial clave KPI

Vuelta i-ésima	Desplazamientos izquierda
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Desplazamientos de clave KDC

Una vez calculados los 48 bits de la subclave se suman O exclusivo con los 48 bits obtenidos de aplicar una permutación con expansión a los 32 bits correspondientes del texto a cifrar. La permutación viene dada por la tabla TPE que simplemente reordena los bits y repite varios de ellos.

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

Tabla de permutación con expansión TPE

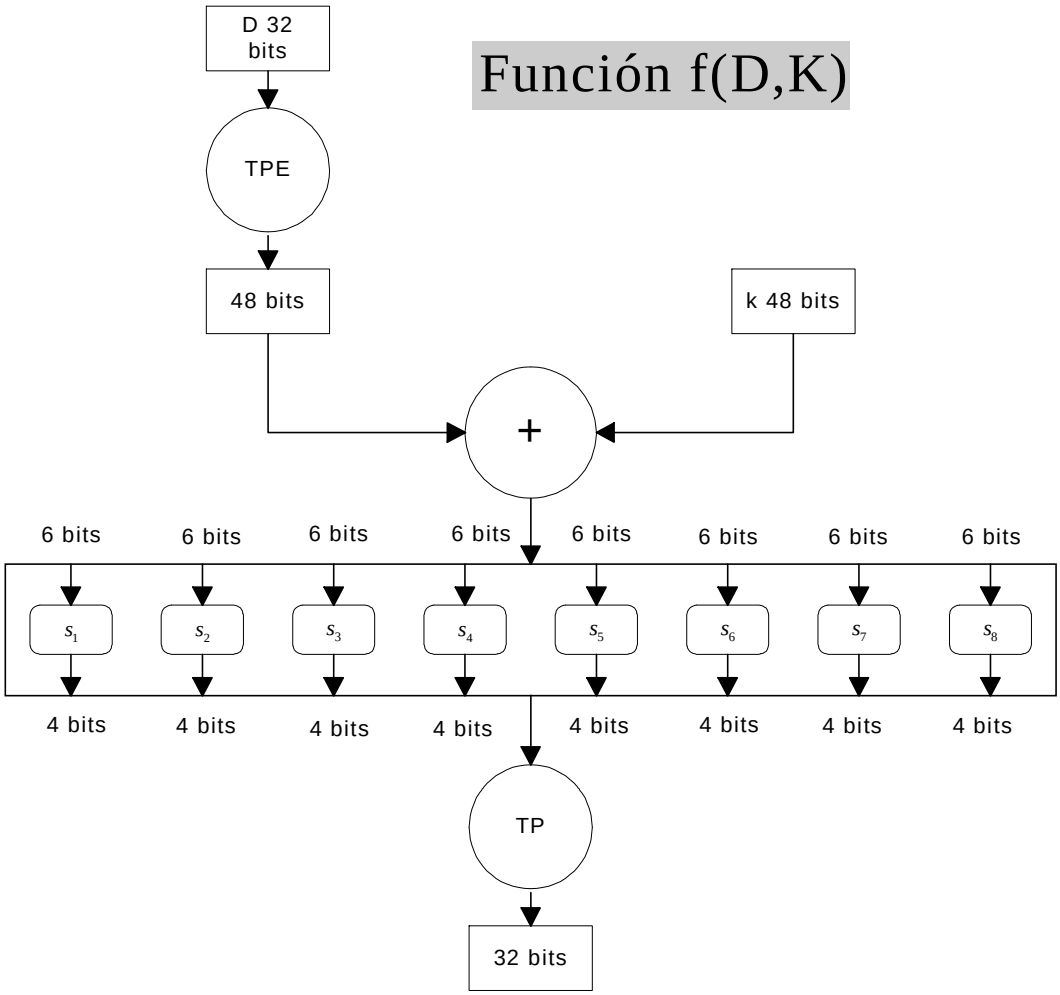
El resultado obtenido de la suma O exclusiva se somete a una sustitución con compresión, posteriormente el resultado se suma con la parte izquierda de los datos de 32 bits. Esta sustitución se realiza mediante las denominadas cajas S que tienen como entrada 6 bits y devuelven 4 como salida. Existen un total de 8 cajas S que son utilizadas 16 veces, una por iteración.

Caja	Fila	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	4	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

El funcionamiento de las cajas S es lo que da fortaleza al DES, éstas están diseñadas de forma que la salida no sea una función lineal de la entrada y que un cambio de un bit en la entrada provoque el cambio de dos bits como mínimo en la salida.

Los 48 bits obtenidos en el paso anterior se dividen en 8 grupos de 6 bits, cada uno de los cuales se utilizará como entrada a la caja S. Para ello se escoge el primer y el último bit del

grupo cuya representación decimal está entre 0 y 3. Este número es lo que se utilizará como indicador de fila de la caja S correspondiente. Los otros cuatro bits dan como resultado un número entre 0 y 15 que se utilizará como indicador de columna. El número que se encuentra en la intersección de ambos en la caja es la representación en decimal del número que sustituirá a la entrada.



Finalmente, los 32 bits de salida de las cajas S son sometidos a una permutación definida por la tabla TP.

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	11	4	25

Tabla TP

El proceso del descifrado en DES es el mismo que el de cifrado, cambiando únicamente el orden de las subclaves y la dirección del desplazamiento en las mismas. Es decir, el orden será $k_{16},..,k_1$ y el desplazamiento a la derecha.

Modos de funcionamiento del DES.

En la publicación del estándar DES [FIP81] se presentan cuatro formas básicas de combinar las entradas y salidas del algoritmo. Estas son:

1) Electronic code book (ECB).
2) Cipher block chaining (CBC).
3) Cipher feedback (CFB).
4) Output feedback (OFB).

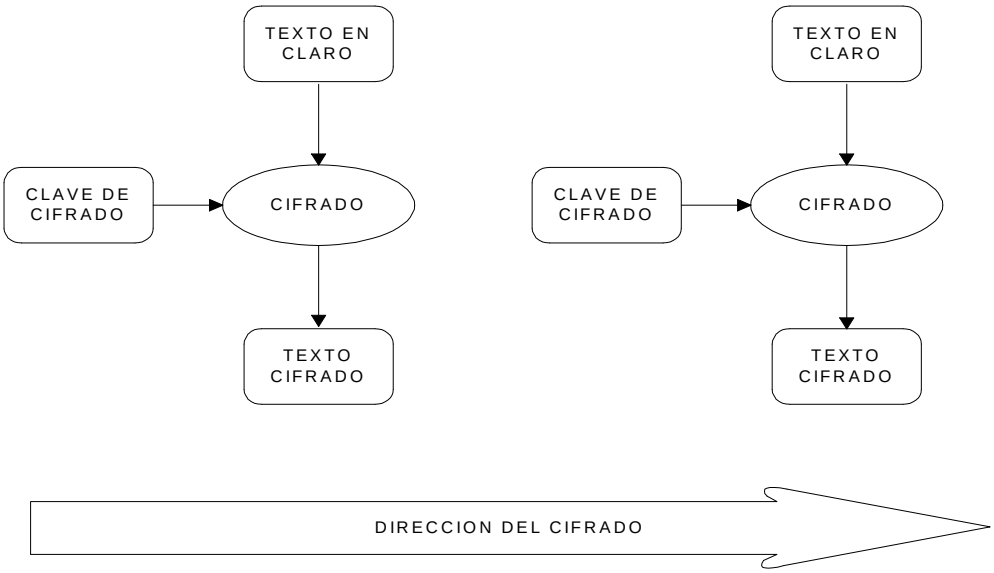
Los cuatro métodos proporcionan diferentes grados de seguridad y adecuación para diferentes necesidades de cifrado, sin embargo, no son privativos del DES y pueden utilizarse en cualquier sistema de cifrado en bloque. Presentamos en detalle cada una de estas estrategias de cifrado.

Electronic code book.

En este modo cada bloque de texto en claro se cifra directamente con la clave, tal como muestra la figura siguiente. Su principal inconveniente es la aparición de repeticiones en el texto cifrado. Estas se deben a la repetición de grupos de caracteres idénticos en el texto en claro. Otro inconveniente es la posibilidad de un ataque activo en el que se introducen trozos de mensajes anteriores grabados con la misma clave, se modifican, o simplemente se eliminan partes del mensaje. Un caso típico es la modificación en una base de datos. Al ser los bloques de tamaño fijo, y al no cifrarse generalmente todos los datos de la base de datos, un usuario podría copiar el campo de importe de una transferencia de un cliente que generalmente hace transferencias de mucho valor, a una transferencia a su cuenta de un importe mucho menor.

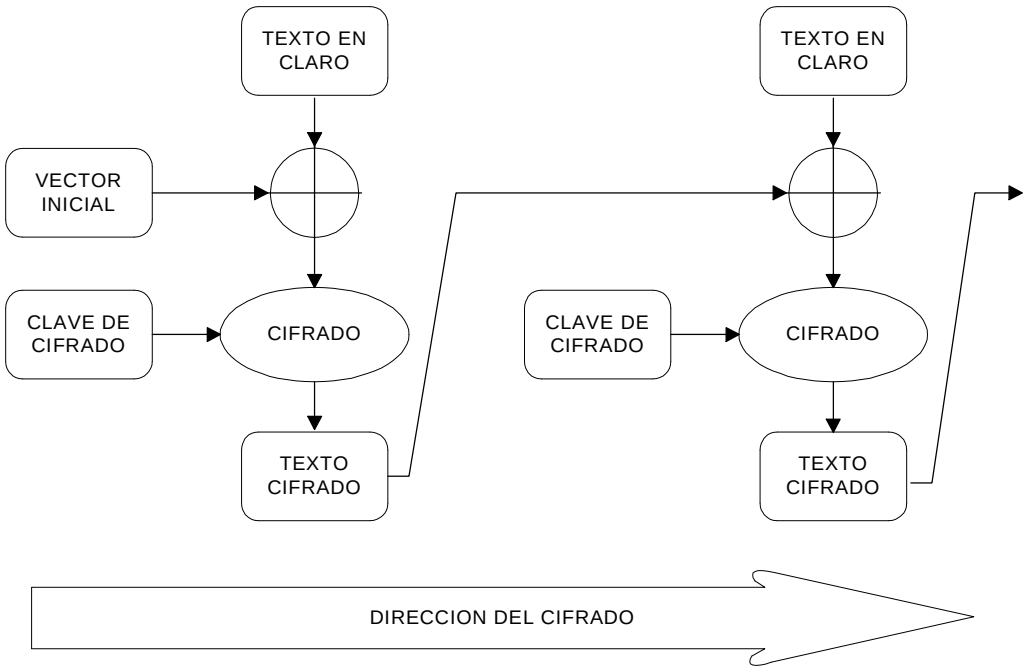
Sea m el mensaje original, k la clave de cifrado, c el mensaje cifrado y C el proceso de cifrado, esquemáticamente tenemos que:

$$c = C_k(m)$$
$$m = C_k^{-1}(c)$$



Cipher block chaining.

En esta forma de cifrado los bloques se combinan, previos al cifrado, con el resultado de cifrar el bloque anterior mediante la operación XOR. Este método evita la aparición de repeticiones y dificulta, aunque no evita, los riesgos de inclusión de mensajes cifrados anteriormente con la misma clave. Para empezar el proceso se utiliza un vector de inicialización, diferente en cada cifrado, que será el que se combinará con el primer bloque de texto en claro.
Cuando se utiliza este esquema, al texto cifrado debe añadirse un bloque complementario correspondiente al vector de inicialización, necesario para empezar el proceso de descifrado. Si denotamos como *Vi* el vector inicial, tenemos que:



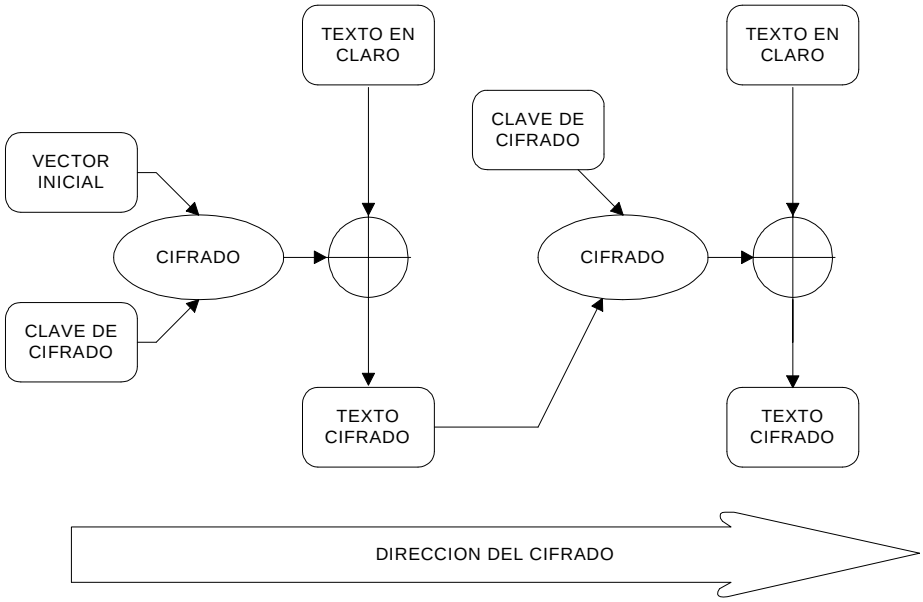
$$c_i = C_k(m_i \oplus c_{i-1}) \quad \text{con } c_o = C_k(Vi)$$
$$m_i = C_k^{-1}(c_i) \oplus c_{i-1} \quad \text{con } m_o = Vi$$

Una característica interesante de este método de cifrado es la posibilidad de limitar la propagación de errores, ya que en el caso de un error, este solo afectará al bloque que lo ha producido y al siguiente, pero no a los demás. Sin embargo, este método también tiene problemas de seguridad, supongamos que todas las transacciones de un banco se cifran con el mismo valor inicial. Si un usuario es capaz de localizar el registro de una transferencia, podría hacer una transferencia importante de dinero a una cuenta y posteriormente borrarla del diario de transacciones sin que esta modificación pudiera ser detectada.

Cipher feedback.

Este método presenta la diferencia con respecto al anterior que en lugar de cifrar el texto en claro, se vuelve a cifrar el cifrado del bloque anterior y posteriormente se combina mediante la operación XOR con el correspondiente bloque de texto en claro. En realidad, lo que se está haciendo es convertir al DES en un algoritmo de cifrado en flujo. De todas las formas de utilización del DES, ésta es la única aceptable para cifrado de Bases de Datos en línea al no obligar a utilizar bloques de tamaño fijo. Los métodos anteriores solo son aceptables en ficheros históricos o de uso secuencial estricto. Al igual que en el caso anterior es necesario un vector de inicialización. La ventaja principal de este método con respecto a los anteriores es la posibilidad de poder cifrar datos con longitud inferior a un bloque con una seguridad aceptable siempre que el tamaño de los mismos sea divisor del tamaño del bloque del algoritmo de cifrado, en este caso 56 bits. Esquemáticamente tenemos que:

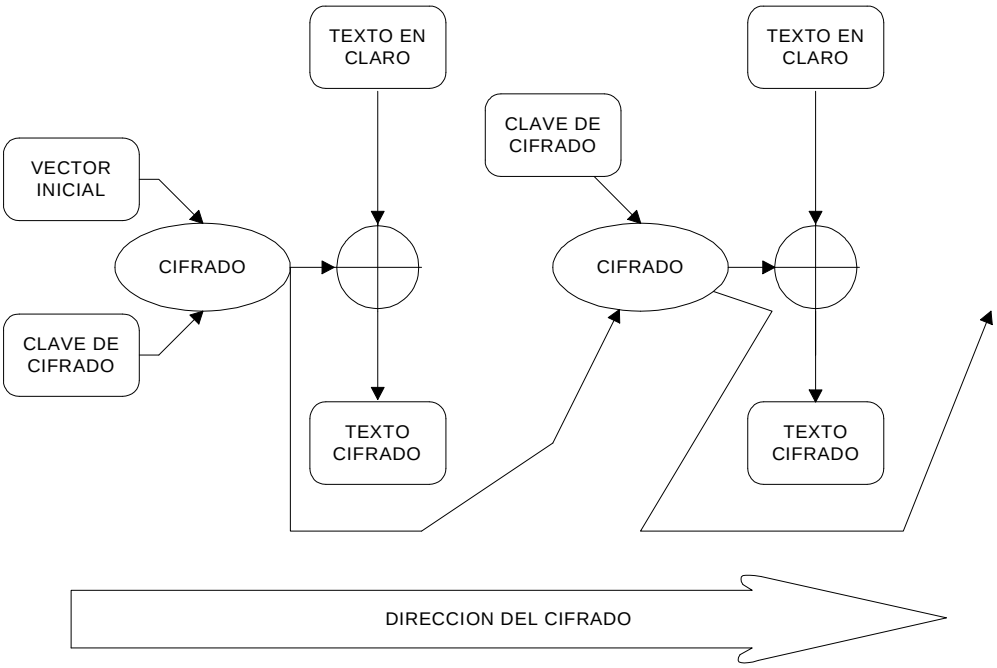
$$c = C_k(C_k(m_{n-1})) \oplus m_n$$
$$m = C_k^{-1}(C_K(m_{n-1})) \oplus c \text{ con } m_0 = Iv$$



Output feedback.

En esta forma de operación se utilizan los bloques cifrados para generar nuevas claves mediante una operación XOR. En este método, similar en concepto al de autoclave, las claves se generan a partir de las claves anteriores sin ninguna intervención del texto tanto cifrado como a cifrar. Al igual que en el modo anterior es necesario un vector de inicialización. La principal ventaja de este método es el de que la clave puede generarse incluso antes de los datos a cifrar estén preparados. Su principal inconveniente es que, al existir una relación directa entre el texto fuente y el cifrado, es posible la sustitución de bloques de texto cifrado si se conocen su contrapartida en fuente.
Esquemáticamente tenemos que:

$$c_i = m_i \oplus s_i$$
$$m_i = c_i \oplus s_i$$
$$s_i = C_k(s_{i-1}) \text{ con } s_0 = Iv$$



Triple DES.

El DES, a pesar de la gran cantidad de ataque realizados contra él y de las suspicacias levantadas por el hecho de haber limitado el tamaño de la clave y la no divulgación de las bases de elección de las cajas S y cajas P, se ha demostrado un sistema sólido y seguro. Los únicos ataques serios contra él, el criptoanálisis diferencial y el lineal, no han pasado de ser ataques teóricos. En el caso del criptoanálisis diferencial, Don Coppersmith miembro del equipo de diseño del algoritmo en los 70 reconoció que conocían este ataque en el momento

de diseñar el algoritmo, muchos años antes de que Biham y Shamir lo desarrollaran y publicaran, y lo hicieron inmune a él[BIH93]. El DES ha caído por pura obsolescencia y mediante ataques por fuerza bruta, no por fallos en el diseño del mismo. Sin embargo, y dado que el cifrado en DES no forma un grupo, es posible cifrar utilizar un cifrado múltiple para aumentar la longitud del espacio de búsqueda. Esta es la idea del triple DES que duplica la clave de cifrado a 112 bits con lo cual su seguridad ante un ataque por fuerza bruta está en estos momentos garantizada, y además es inmune a los ataques por encuentro a medio camino (meet in the middle). La desventaja está en triplicar el número de operaciones de cifrado, sin embargo, incluso así, el algoritmo sigue siendo rápido.

Básicamente el proceso de cifrado es el siguiente. Se escogen dos claves k_1 y k_2 y se cifra el mensaje *M* mediante las tres operaciones sucesivas que se indican a continuación:

$$C = E_{k_1}(E_{k_2}^{-1}(E_{k_1}(M)))$$

El proceso de descifrado consistiría en aplicar las operaciones de descifrado en el orden inverso a las de cifrado.

Otra posibilidad manejada por programas como PGP es la de utilizar el triple DES con tres claves. Esto aumenta la longitud efectiva de la clave a 168 bits con la única desventaja de tener que utilizar tres claves en lugar de dos[STA98].

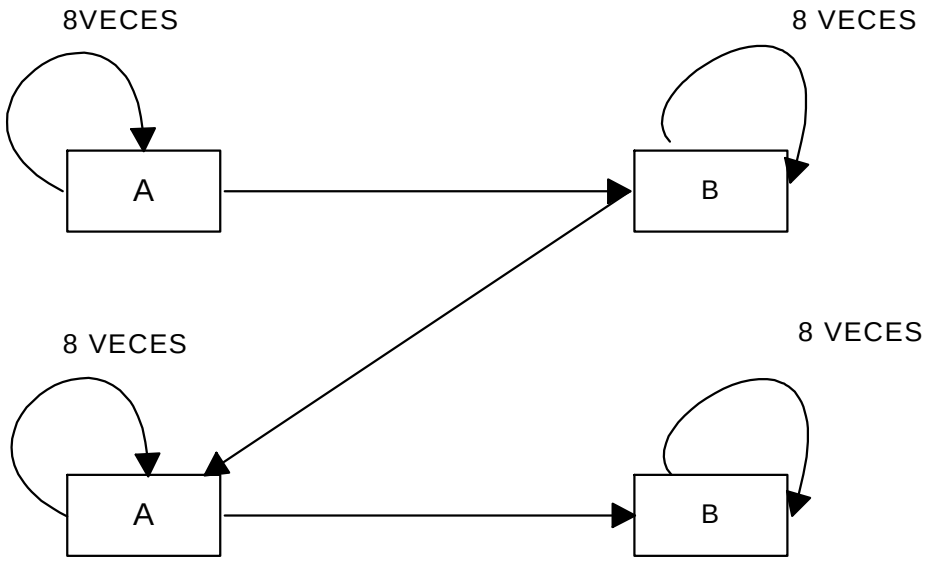
Skipjack.

Es un sistema de clave secreta utilizado como base para implantar los sistemas de clave compartida o de deposito de claves (key escrow) que la Administración norteamericana pretende imponer como estándar de cifrado en Estados Unidos. El sistema se implantaría en hardware, el chip Clipper, y se utilizaría para transferencia de todo tipo de información, con la característica de que en cualquier momento, teóricamente bajo estricto control judicial, podía accederse a la clave de cifrado. Esta estaría dividida en dos partes bajo supervisión de dos organismos diferentes. En un principio se sabía pocos datos de este sistema de cifrado, la Administración no quiso revelar los detalles del diseño, cosa que generó multitud de críticas por parte de los expertos en el tema que apuntaban que de esta manera no se podía validar la seguridad del sistema. Sin embargo, el 23 de junio de 1998 el Departamento de Defensa anuncia la desclasificación del algoritmo de cifrado y del sistema de intercambio de claves (KEA).

A grosso modo el sistema es un sistema de cifrado de clave simple que cifra bloques de 64 bits. El algoritmo utiliza claves de 80 bits con 32 pasos de transposición y soporta los cuatro modos de funcionamiento del DES. El sistema vendrá integrado en un chip que se programará en un compartimento de alta seguridad.

Los sucesores del chip Clipper, Capstone, Fortezza, Keystone, Regent y Krypton incluyen además el Estándar de Firma digital (DSS), el algoritmo de intercambio de claves (KEA) basado en el protocolo de Diffie-Hellman, un exponenciador rápido y un generador de números aleatorios.

El sistema funciona de la siguiente manera, los datos se cifran alternando dos pasos de transformaciones A y B ocho veces cada paso, tal como muestra la figura siguiente.



Las transformaciones A y B son básicamente una permutación de uno de los bloques y un registro de desplazamiento del conjunto con un contador que inicialmente vale 1 y se va incrementando en uno cada paso de los 32 del algoritmo, volviéndose a inicializar en cada bloque a cifrar. Las transformaciones A y B, de las cuales se presenta un esquema a continuación, siguen los siguientes pasos para cifrar y descifrar respectivamente. Denotamos[1] P a la permutación y $\overline{P}$ a la permutación inversa y w_i, w_2, w_3, w_4 a las divisiones de 16 bits del bloque a cifrar, siendo w_i^k el bloque *i* durante la aplicación del paso *k*.

TRANSFORMACION A *CIFRADO* *DESCIFRADO*

$$w_1^{k+1} = P^k(w_1^k) \oplus w_4^k \oplus contador^k$$
$$w_2^{k+1} = P^k(w_1^k)$$
$$w_3^{k+1} = w_2^k$$
$$w_4^{k+1} = w_3^k$$

$$w_1^{k-1} = \overline{P^{k-1}} w_2^k$$
$$w_2^{k-1} = w_3^k$$
$$w_3^{k-1} = w_4^k$$
$$w_4^{k-1} = w_1^k \oplus w_2^k \oplus contador^{k-1}$$

[1] En el original se denomina a la G a la función de permutación.

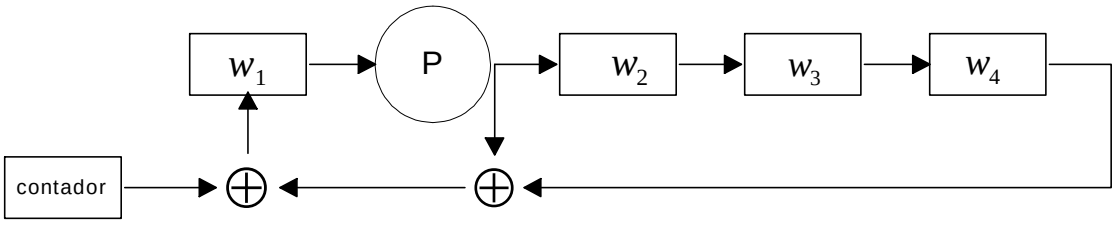
TRANSFORMACION B

$$w_1^{k+1} = w_4^k$$
$$w_2^{k+1} = P^k(w_1^k)$$
$$w_3^{k+1} = w_1^k \oplus w_2^k \oplus contador^k$$
$$w_4^{k+1} = w_3^k$$

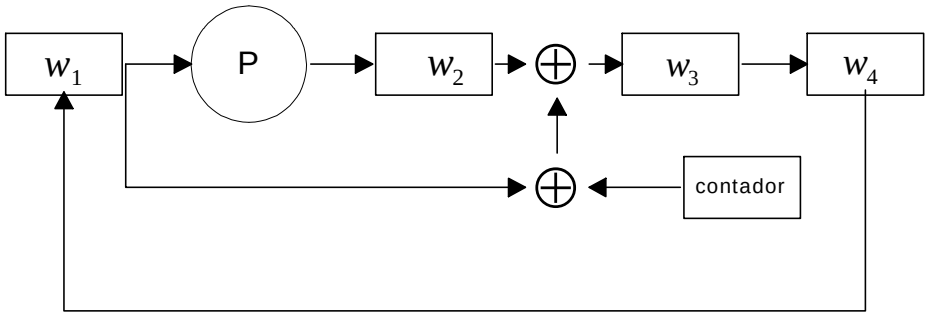
$$w_1^{k-1} = \overline{P^{k-1}} w_2^k$$
$$w_2^{k-1} = \overline{P^{k-1}} w_2^k \oplus w_3^k \oplus contador^{k-1}$$
$$w_3^{k-1} = w_4^k$$
$$w_4^{k-1} = w_1^k$$

Para descifrar se siguen los pasos inversos a los de cifrado. Se empieza aplicando la transformación B de descifrado 8 veces, posteriormente la A, vuelta a la B y finalizamos en la A, siempre utilizando las transformaciones de descifrado. El contador en este caso empieza con el valor 32 y va disminuyendo su valor en 1 en cada paso.

TRANSFORMACION A



TRANSFORMACION B



La permutación *P* es una estructura de tipo Feistel basada en una tabla de sustitución que denominamos F y que presentamos a continuación. La sustitución de un byte se realiza dividiendo el mismo en dos bloques de cuatro bits y utilizando los de la izquierda como fila y los de la derecha como columna.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	A3	D7	09	83	F8	48	F6	F4	B3	21	15	78	99	B1	AF	F9
1	C7	2D	4D	8A	CE	4C	CA	2E	52	95	D9	1E	4E	38	44	28
2	0A	DF	02	A0	17	F1	60	68	12	B7	7A	C3	E9	FA	3D	53
3	96	84	6B	BA	F2	63	9A	19	7C	AE	E5	F5	F7	16	6A	A2
4	39	B6	7B	0F	C1	93	81	1B	EE	B4	1A	EA	D0	91	2F	B8
5	55	B9	DA	85	3F	41	BF	E0	5A	58	80	5F	66	0B	D8	90
6	35	D5	C0	A7	33	06	65	69	45	00	94	56	6D	98	9B	76
7	97	FC	B2	C2	B0	FE	DB	20	E1	EB	D6	E4	DD	47	4A	1D
8	42	ED	9E	6E	49	3C	CD	43	27	D2	07	D4	DE	C7	67	18
9	89	CB	30	1F	8D	C6	8F	AA	C8	74	DC	C9	5D	5C	31	A4
A	70	88	61	2C	9F	0D	B	87	50	82	54	64	26	7D	03	40
B	34	4B	1C	73	D1	C4	FD	3B	CC	FB	7F	AB	E6	3C	5B	A5
C	AD	04	23	9C	14	51	22	F0	29	79	71	7E	FF	8C	0C	E2
D	0C	EF	BC	72	75	6F	37	A1	EC	D3	8E	62	8B	86	10	E8
E	08	77	11	BE	92	4F	24	C5	32	36	9D	CF	F3	A6	BB	AC
F	5E	6C	A9	13	57	25	B5	C3	BD	A8	3A	01	05	59	2A	46

La función *P* puede definirse de la siguiente manera. Sea *k* la clave y k_i el i-ésimo byte de la clave, con $i \in [0,..,9]$ y sea $\|$ la operación de concatenación. Tenemos pues que

$$P^j(w) = P^j(p_1 \| p_2) = p_5 \| p_6 \text{ con } p_i = F(p_{i-1} \oplus k_{4j+i-3}) \oplus k_{i-2}$$
$$\left[P^j(w)\right]^{-1} = \left[P^j(p_5 \| p_6)\right] = p_1 \| p_2 \text{ con } p_{i-2} = F(p_{i-1} \oplus k_{4j+i-3}) \oplus p_i$$

Desde su desclasificación han aparecido algunos ataques al sistema, sin embargo, ninguno representa por el momento un peligro serio [BIH98][BIH98b].

Otros algoritmos.

Existe una gran variedad de algoritmos criptográficos en el mercado, sin embargo, la gran mayoría utilizan los conceptos vistos hasta el momento y no consideramos que sea el propósito de este trabajo la enumeración y descripción de los mismos. Para una descripción detallada de la mayoría de los algoritmos puede verse principalmente [SCH93] y [MEN97].

IDEA.

Fue desarrollado por Xuejia Lai y James Massey en 1990 con el nombre de PES (Proposed Encryption Standard). En 1992 los autores preparan una versión mejorada de forma que fuera inmune al criptoanálisis diferencial que pasaron a denominar IDEA. Se trata de un algoritmo que trabaja con bloques de 64 bits y clave de 128 bits. Se utiliza el mismo algoritmo para el cifrado y el descifrado. Según Schneier [SCH93], se trata del mejor y más seguro algoritmo en bloques disponible al público.

AKELARRE.

Akelarre es un algoritmo de cifrado en bloque desarrollado en España por personal del CSIC[ALV96]. La estructura del algoritmo es similar a la de IDEA con algunas diferencias significativas. La primera es la utilización de subbloques de 32 bits en lugar de los 16 de IDEA. El tamaño de la clave es variable, así como el número de vueltas, aunque los autores recomiendan la utilización de claves de 128 bits y cuatro vueltas. Según los autores el algoritmo está diseñado de forma que sea inmune al criptoanálisis lineal y diferencial. El algoritmo fue forzado en 1997 por Ferguson y Schneier [FER97].

BLOWFISH.

Se trata de un algoritmo de cifrado desarrollado por Bruce Schneier con varias ventajas. La primera de ellas es la de utilizar una clave de longitud variable de hasta 448 bits. Trabaja con bloques de 64 bits y tiene un código muy compacto. El autor acaba de presentar un nuevo algoritmo para la elección del AES denominado TWOFISH.

RIJNDAEL.

Es el ganador del concurso para sustituir al DES como nuevo algoritmo estándar de cifrado. Sus autores son Jin Daemen y Vincent Rijmen. Se trata de un algoritmo de cifrado en bloque que no utiliza la estructura tipo Feistel con longitud variable de bloque y clave. Las longitudes de cifrado pueden ser de 128, 192 y 256 bits. Según los autores el algoritmo se ha optimizado para resistir todos los ataque conocidos y para ser muy rápido y de fácil implementación.

CONTINUARÁ