Reglas:

1. Dos letras repetidas se separan por una x. Ejemplo MALLA sería MA LX LA, si la palabra tiene un número impar de letras se coloca una x al final.
2. Si las dos letras se encuentran en el mismo fila de la matriz, cada una de ellas se sustituye con la letra que esté a su derecha. Por ejemplo, ED se cifra como DM.
3. Si las dos letras se encuentran en la misma columna, cada una de las letras se sustituye por la letra debajo de ella. Por ejemplo, AV se cifra HM
4. En otro caso, la primera letra de la pareja se sustituye por la que este en la intersección de su misma fila y la columna de la segunda letra, la segunda letra se sustituye por la que este en la intersección de su misma fila y la columna de la primera letra. Por ejemplo: BT se cifra FR

COMO ESTAS = CO MO ES TA SX se cifraría como BN NM NT FQ NN

- Cifrado Atbash
- Cifrado de Polybios.

b) ***sustitución polialfabética:*** el criptograma del texto claro puede ser diferentes dependiendo de la clave que se utilice para cifrar, por lo que se dice que existen múltiples alfabetos de cifrado, de ahí el nombre de sustitución polialfabética.

- Cifrado de Alberti
- Cifrado por desplazamiento
- Cifrado de Vigenère
- Cifrado de Vernam
- Cifrado One-Time Pad

1.3.4 Técnicas de Transposición

Una alternativa de cifrado es realizar algún tipo de permutación en las letras de texto plano. Esta técnica se conoce como cifrado de transposición, las unidades que se permutan pueden ser de una letra, pares o trios. (Ejemplo: la escitala). Más que usarlas individualmente se superponen a otras técnicas para mejorar la seguridad. Las 'unidades de texto' pueden ser de una sola letra, pares de letras, tríos de letras o combinaciones de lo anterior, normalmente el algoritmo se basaba en un diseño geométrico o el uso de artilugios mecánicos (Ej escítala). algoritmo y clave son un conjunto indivisible.

- Cifrado por Escritura Inversa

Se escribe la palabra al revés, por ejemplo, la cadena: *"Hola mi nombre es Pepa"* se cifra como "aloH im erbmon se apeP". Puede emplearse para palabras sueltas o mensajes completos.

- Cifrado por Transposición Columnar Simple

Este es un cifrado con forma de columna, el mensaje original estará limitado a un rectángulo, de izquierda a derecha y de arriba hacia abajo. Después, se escoge una clave para asignar un número a cada columna del rectángulo para determinar el orden. El número correspondiente a la letra de la clave estará determinado por su posición en el alfabeto, por ejemplo. A es 1, B es 2, C es 3, etc. Por ejemplo, si la clave es CAT y el mensaje es "THE SKY IS BLUE", el proceso sería el siguiente:

C	**A**	**T**
3	**1**	**20**
T	H	E
S	K	Y
I	S	B
L	U	E

Tomando las letras en orden numérico se forma el mensaje, la columna debajo de la A primero, después la columna de C y por último la columna de T, el mensaje cifrado será: HKSUTSILEYBE

- Cifrado por transposición columnar doble

Considerada, por mucho tiempo, la forma más segura y compleja. Su operación es idéntica a la transposición columnar simple, pero tras una primera transposición, se realizaba una segunda, empleando o no, la misma clave.

- Cifrado por transposición interrumpida

Diversos puntos de la matriz de cifrado, conocidos por emisor y receptor de la codificación, quedan vacíos, ello altera la serie y distorsiona la lógica de la transposición. Uno de los más conocidos fue propuesto por el general Luigi Sacco que establecía diferentes longitudes para cada una de las líneas a cifrar, extendiéndose hasta encontrar la columna con el número correspondiente a la línea (así la primera línea llega hasta el número 1 y la quinta hasta el 5). Y donde cada columna se leía ignorando los huecos. Por ejemplo John Savard propone la clave CONVENIENCE y el texto claro *"Here is a secret message enciphered by transposition"*.

C	**O**	**N**	**V**	**E**	**N**	**I**	**E**	**N**	**C**	**E**
1	**10**	**7**	**11**	**3**	**8**	**6**	**4**	**9**	**2**	**5**
H										
E	R	E	I	S	A	S	E	C	R	
E	T	M	E	S						
S	A	G	E	E	N	C	I			
P	H	E	R	E	D	B	Y	T	R	A
N	S	P	O	S	I	T				
I	O	N								

El texto cifrado será HEESPNI RR SSEES EIY A SCBT EMGEPN ANDI CT RTAHSO IEERO.

- Cifrado AMSCO

Creado por A. M. Scott, es una transposición columnar en la que los elementos del texto cifrado son escogidos, alternativamente, como letras simples y pares de letras. Así, la palabra MATEMÁTICAMENTE será cifrado con la clave COPLA como sigue:

C	**O**	**P**	**L**	**A**
2	**4**	**5**	**3**	**1**
M	AT	E	MA	T
IC	A	ME	N	TE

El texto cifrado será: TTE MIC MAN ATA EME

- Cifrado por transposición por ruta

Consisten en repartir el mensaje a cifrar en una figura geométrica, normalmente un paralelepípedo y definir una ruta para leer el mensaje. La complejidad de la ruta determinará la fuerza del cifrado. Pueden establecerse distintas rutas, por ejemplo, para la palabra EUFORICAS se puede ordenar en tres columnas de igual longitud.

E	U	F
O	R	I
C	A	S

Puede leerse como una espiral hacia el centro EUFISACOR, como una espiral desde el centro ROCASIFUE, de modo diagonal ERS UI F C OA, que está inscrito en un cilindro, ERS UIC FOA, la primera columna en descendente, la segunda en ascendente y la tercera en descendente: EOC ARU FIS.

Otros algoritmos de transposición o codificación son:

- Cifrado por permutación de grupos
- Cifrado por permutación por series
- Cifrado por rejillas criptográficas

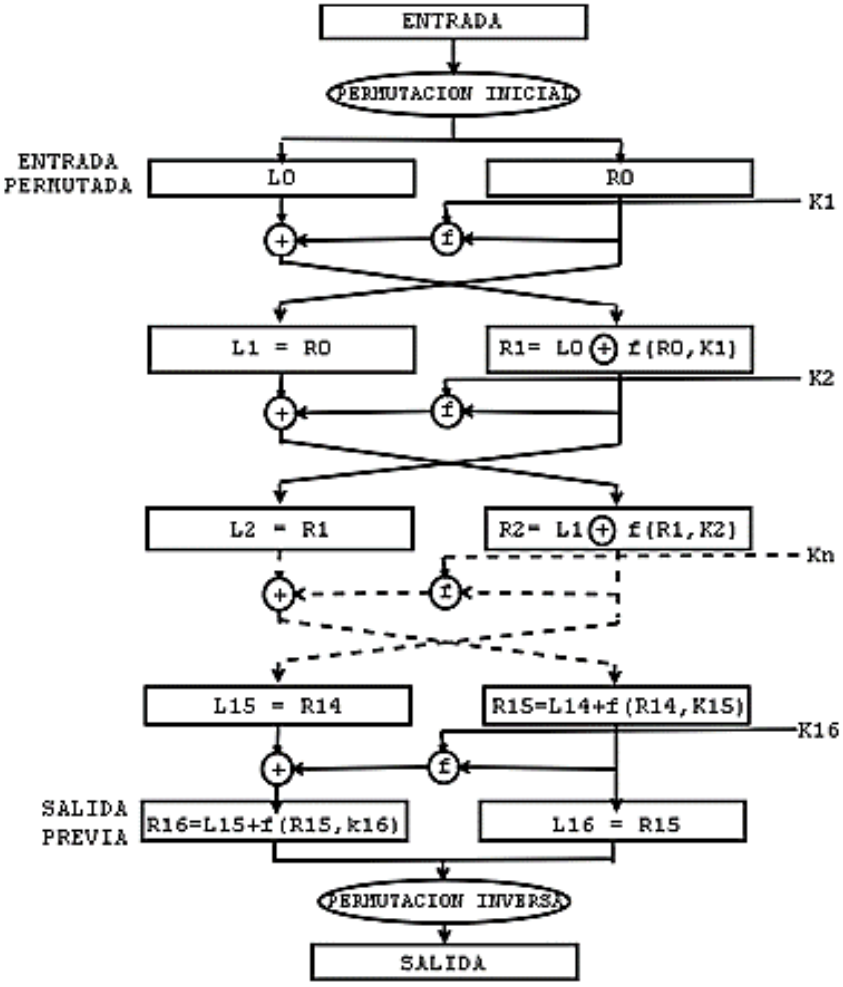
1.3.5 Algoritmos de Criptografía Simétrica

a) DES (Data Encription Standard o Estándar de Encriptación de Datos) o DEA (Data Encription Algorithm)

   Es el algoritmo de cifrado simétrico más empleado, es un algoritmo de cifrado por bloques de 64 bits, con clave de 56 bits (se eliminan 8 bits de paridad), fue diseñado para ser implementado en hardware y es usado en comunicaciones. Emisor y receptor deben conocer la clave secreta, la que se intercambia por algoritmos de clave pública. Se usa para encriptar y desencriptar mensajes, generar y verificar códigos de autentificación de mensajes (MAC) y para encriptación de un sólo usuario (por ejemplo, guardar un archivo en disco), los ataques se realizan con hardware específico que con fuerza bruta descubran una clave en pocos días por el tamaño de la clave.

Las fases del algoritmo son las siguientes:

1) fraccionamiento del texto en bloques de 64 bits (8 bytes)
2) permutación inicial de los bloques
3) partición de los bloques en dos partes: izquierda y derecha, *L y R* respectivamente
4) fases de permutación y de sustitución repetidas 16 veces (rondas)
5) reconexión de las partes izquierda y derecha, seguida de la permutación inicial inversa.



- **Permutación inicial** cada bit de un bloque está sujeto a una permutación inicial, representada mediante la siguiente matriz de permutación inicial (PI):

PI								
	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

- **División en bloques de 32 bits** el bloque de 64 bits se divide en dos bloques de 32 bits llamados L y R por Left y Right. El estado inicial de estos bloques se denomina L_0 y R_0:

L_0							
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

R_0							
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

  L_0 contiene los bits en posición par en el mensaje inicial y R_0 contiene los bits en posición impar.

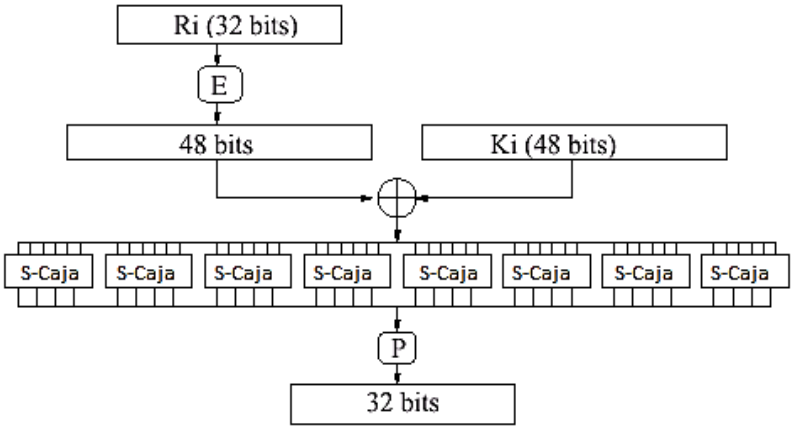
- **Fases de permutación y de sustitución repetidas o rondas** los bloques L_n y R_n sufren transformaciones iterativas llamadas *rondas*, cada una de las 16 iteraciones realiza transformaciones y sustituciones, el resultado de cada iteración T_i, es la concatenación de las partes L_i y R_i, es decir, $Ti = L_i \,.\, R_i$ $(1 \leq i \leq 16)$. Para cada uno de estos pasos se verifica que:

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

  K_i es la clave para la iteración *i-ésima*
  f es una función de cifrado

  K_i es una clave de 48 bits generada en cada ronda por una función generadora de claves definida como $K_i = K\,S(i, K)$ siendo K la clave externa de 64 bits y S es una función de sustitución generada a partir de la clave externa K y dependiente del número de iteración i, que se explicará más adelante.

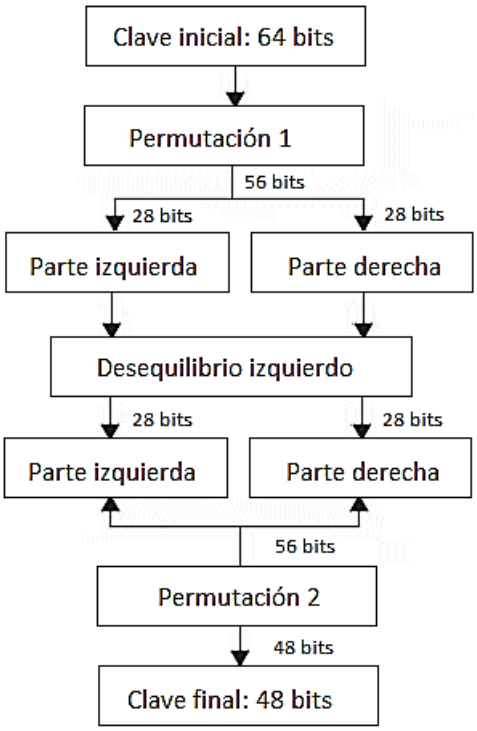
  La <u>función de cifrado f</u> transforma los 32 bits del bloque R_{i-1} mediante la subclave Ki, en los 32 bits de $P=f(R_{i-1},K_i)$ según el siguiente esquema:



  Aquí <u>E es una tabla de expansión</u> que transforma R_i de 32 a 48 bits según se muestra:

E						
	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

La otra entrada a la función OR-exclusiva es la clave Ki obtenida a partir de una función de generación de claves, en esta función radica toda la seguridad y complejidad del cifrado. El algoritmo que sigue a continuación muestra cómo obtener a partir una clave de 64 bits (compuesta por cualquier de los 64 caracteres alfanuméricos), 8 claves diferentes de 48 bits, cada una de ellas utilizadas en el algoritmo DES:



- En primera instancia, se eliminan los bits de paridad (octavo bit) de cada byte de la clave, para obtener una clave que posea una longitud de 56 bits.

- Se aplica luego una primera permutación llamada Permutación 1 (**PC-1)**, según la siguiente tabla:

PC-1														
	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

- Dividir esta matriz en dos matrices Li y $\mathbf{R}_i$ (izquierda y derecha), cada una de 28 bits:

$\mathbf{L}_i$							
	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

$\mathbf{R}_i$							
	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

- Rotar los dos bloques una posición hacia la izquierda de manera circular, luego los dos bloques de 28 bits se agrupan en un bloque de 56 bits.
- Este bloque pasa por una Permutación 2 (**PC-2)** conformando la clave $\mathbf{K}_i$ de 48 bits

pc-2												
	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

- Iterando el algoritmo se obtienen las 16 claves $\mathbf{K}_1$ a $\mathbf{K}_{16}$ utilizadas en el algoritmo DES.

Una vez obtenido el bloque de 48 bits, se efectúa la operación *or-exclusivo* entre $E(R_{i-1})$ y la clave *i-ésima Ki* y el resultado es dividido en ocho bloques B_j de seis bits.

$$E(R_{i-1}) \oplus K_i = B_1.B_2.B_3.B_4.B_5.B_6.B_7.B_8$$

Cada uno de los bloques B_j se usa como entrada para cada una de las tabla de selección-sustitución S_j (cajas S), dichas cajas realizan una transformación no lineal que da como salida una secuencia de 4 bits, $S_j(B_j)$. Sobre el funcionamiento de cada una de las cajas *S*, que generan $S_j(B_j)$. cada una de estas 8 cajas tiene asociada una matriz de selección definida, por ejemplo, se muestra la matriz de selección S_1 que será aplicada al bloque B_1, en una transformación no lineal:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
10	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
11	15	12	8	2	4	9	**1**	7	5	11	3	14	10	0	6	13

Sea por ejemplo B_1=101101 se obtendrá $S_1(B_1)$=0001, siguiendo el siguiente procedimiento:

- Con los bits b1=1 y b6=1 se define la fila correspondiente 11.
- Con los bits b2=0, b3=1, b4=1 y b5=0 se define la columna correspondiente 0110=6.
- El elemento seleccionado es el 1, que expresado en binario en 4 bits es $S_1(B_1)$=0001.

Es decir, los 6 bits $B_j = b_1b_2b_3b_4b_5b_6$ se convierten en 4 bits que es la expresión binaria en 4 bits del valor decimal del elemento de la matriz S_j cuya fila, viene determinada por el valor de los bits b_1b_6 y cuya columna está determinada por el número decimal correspondiente al

binario $b_2b_3b_4b_5$ de los bits centrales de B_j. A continuación, se muestran las otras siete matrices de selección:

S_2

	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

S_3

	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

S_4

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

S_6

	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

S_7

	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

S_8

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

Las salidas $S_j(B_j)$, de cada caja, se concatenan para generar 32 bits que van a ser sometidos a la permutación P

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

La salida será de la forma

$$P(S_1(B_1).\ S_2(B_2).\ S_3(B_3).\ S_4(B_4).\ S_5(B_5).\ S_6(B_6).\ S_7(B_7).\ S_8(B_8))$$

Hay que notar que esto se itera quince veces, difiriendo en que el resultado de la última ronda se toma como entrada para la permutación inversa.

- Reconexión de bloques izquierdo y derecho o permutación inicial inversa: al final de las iteraciones, los dos bloques $\mathbf{L}_{16}$ y $\mathbf{R}_{16}$ se vuelven a conectar y se someten a una permutación llamada permutación inicial inversa, con la que se obtiene un texto cifrado de 64 bits

IP-1								
	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

b) Triple-DES Se encripta tres veces una clave DES. Esto se puede hacer de varias maneras:
- DES-EEE3: Tres encriptaciones DES con tres claves distintas.
- DES-EDE3: Tres operaciones DES con la secuencia encriptar-desencriptar-encriptar con tres claves diferentes.
- DES-EEE2 y DES-EDE2: Igual que los anteriores pero la primera y tercera operación emplean la misma clave.
- Dependiendo del método elegido, el grado de seguridad varía; el método más seguro es el DES-EEE3.

c) AES (Advanced Encription Standard) es un algoritmo de cifrado por bloques que se ha convertido en el estándar.

d) RC2 (Ron's Code o Rivest's Cipher) es un algoritmo de cifrado por bloques de clave de tamaño variable, trabaja con bloques de 64 bits y es tres veces más rápido que el DES en software. Si se elige claves de mayor tamaño es más seguro que el DES ante ataques de fuerza bruta.

e) RC4 es un algoritmo de tamaño de clave variable con operaciones a nivel de byte. Se basa en el uso de una permutación aleatoria y tiene un periodo estimado de más de 10100. Además, es un algoritmo de ejecución rápida en software. Se emplea para encriptación de ficheros y para encriptar la comunicación en protocolos como el SSL (TLS).

f) RC5 es un algoritmo parametrizable con tamaño de bloque variable, tamaño de clave variable y número de rotaciones variable. Los valores más comunes de los parámetros son 64 o 128 bits para el tamaño de bloque, de 0 a 255 rotaciones y claves de 0 a 2048 bits. Fue diseñado en 1994 por Ron Rivest. El RC5 tiene 3 rutinas: expansión de la clave, encriptación y desencriptación. En la primera rutina la clave proporcionada por el usuario se expande para llenar una tabla de claves cuyo tamaño depende del número de rotaciones. La tabla se emplea en la encriptación y desencriptación. Para la encriptación sólo se emplean tres operaciones: suma de enteros, o-exclusiva de bits y rotación de variables. La mezcla de rotaciones dependientes de los datos y de distintas operaciones lo hace resistente al criptoanálisis lineal y diferencial. El algoritmo RC5 es fácil de implementar y analizar y, de momento, se considera que es seguro.

g) IDEA (International Data Encription Algorithm) es un algoritmo de cifrado por bloques de 64 bits iterativo, la clave es de 128 bits, la encriptación precisa 8 rotaciones complejas.

El algoritmo funciona de la misma forma para encriptar que para desencriptar (excepto en el cálculo de las subclaves). El algoritmo es fácilmente implementable en hardware y software, aunque algunas de las operaciones que realiza no son eficientes en software, por lo que su eficiencia es similar a la del DES. El algoritmo es considerado inmune al criptoanálisis diferencial y no se conocen ataques por criptoanálisis lineal ni debilidades algebraicas. La única debilidad conocida es un conjunto de 251 claves débiles, pero dado que el algoritmo tiene 2128 claves posibles no es un problema serio.

h) SAFER (Secure And Fast Encription Routine) es un algoritmo de cifrado por bloques no propietario. Está orientado a bytes y emplea un tamaño de bloque de 64 bits y claves de 64 (SAFER K-64) o 128 bits (SAFER K-128). Tiene un número variable de rotaciones, pero se recomienda un mínimo de seis. El algoritmo original fue considerado inmune al criptoanálisis lineal y diferencial, pero Knudsen descubrió una debilidad en el generador de claves y el algoritmo fue modificado (SAFER SK-64 y SAFER SK-128).

i) Blowfish es un algoritmo de cifrado por bloques de 64 bits desarrollado por Scheiner. Es un algoritmo de tipo Feistel y cada rotación consiste en una permutación que depende de la clave y una sustitución que depende de la clave y los datos. Todas las operaciones se basan en o-exclusivas sobre palabras de 32 bits. La clave tiene tamaño variable (con un máximo de 448 bits) y se emplea para generar varios vectores de subclaves. Este algoritmo está diseñado para 32 bits y es mucho más rápido que el DES. El algoritmo es considerado seguro, aunque se han descubierto algunas claves débiles, un ataque contra una versión del algoritmo con tres rotaciones y un ataque diferencial contra una variante del algoritmo.