(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 329 819**

(51) Int. Cl.:
H04L 9/06 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Número de solicitud europea: **06010077 .3**
(96) Fecha de presentación : **08.03.2001**
(97) Número de publicación de la solicitud: **1689114**
(97) Fecha de publicación de la solicitud: **09.08.2006**

(54) Título: **Aparato de cifrado de bloques que usa información auxiliar.**

(30) Prioridad: **09.03.2000 JP 2000-64614**

(45) Fecha de publicación de la mención BOPI:
01.12.2009

(45) Fecha de la publicación del folleto de la patente:
01.12.2009

(73) Titular/es: **MITSUBISHI DENKI KABUSHIKI KAISHA**
7-3, Marunouchi 2-chome
Chiyoda-ku, Tokyo 100-8310, JP
Nippon Telegraph & Telephone Corporation

(72) Inventor/es: **Matsui, Mitsuru;**
Tokita, Toshio;
Nakajima, Junko;
Kanda, Masayuki;
Moriai, Shiho y
Aoki, Kazumaro

(74) Agente: **Carpintero López, Mario**

DESCRIPCIÓN

Aparato de cifrado de bloques que usa información auxiliar.

Campo técnico

La presente invención se refiere a un aparato de transformación de datos, a unos procedimientos de transformación de datos, y a unos medios de almacenaje los cuales son registrados los procedimientos de transformación de datos, para su encriptación, desencriptado, y difusión de los datos con el fin de proteger una información digital existente en las comunicaciones de información.

Técnica antecedente

La Fig. 25 representa una función de encriptación que se utiliza en el DES descrito en "Gendai Ango Riron (Teoría de Cifrado Moderno)" "(Modern Cipher Theory)"] (Instituto de Ingenieros de Electrónica, Información y Comunicación ["The Institute of Electronics, Information and Communication Engineers"] publicado el 15 de Noviembre de 1997, página 46).

Como se muestra en la Fig. 25, se utilizan ocho cajas S. Estas ocho cajas S son tablas diferentes entre sí. Cada tabla emite de salida unos datos de 4 bits a partir de unos datos de entrada de 6 bits.

La Fig. 26 muestra una función de transformación no lineal que se describe en "Especificación de E2 - un Cifrado de Bloque de 128 bits" ["Specification of E2 - a 128-bit Block Cipher"] (Nippon Telegraph and Telephone Corporation" publicado el 14 de Junio de 1998, página 10), como también en Kanda M. *et al*.: "E2-a Nuevo cifrado de bloque de 128 bits" IEICE Transacciones en fundamentos electrónicos, Ciencias de comunicación e informática, ingeniería de ciencias sociales, Tokyo, JP, vol. E83-A, no. 1, enero de 2000 (2000-01), páginas 48-49, XP00237858 ISSN: 0916-8508.

Como se muestra en la Fig. 26, cada unidad de función S se compone de ocho cajas S.

Los dispositivos de encriptación convencionales utilizan múltiples cajas S. Dado que algunos cifrados están equipados con tablas diferentes entre sí, el uso de la memoria se incrementa en comparación con los equipados con una caja S. Dado que, por otro lado, otros cifrados utilizan solo una caja S, la seguridad del cifrado se reduce.

Como se muestra en la Fig. 7, cuando una unidad de transformación normal de datos (FL) 250 es insertada en la unidad de encriptación, se requiere suministrar una unidad de transformación de datos inversa (FL^{-1}) 270 en una unidad de desencriptado para descriptar los textos cifrados. Dado que, por regla general, la unidad de transformación normal de datos (FL) 250 y la unidad de transformación de datos inversa (FL^{-1}) 270 son circuitos diferentes entre sí, ello ocasiona un problema en el sentido de que la unidad de encriptación y la unidad de desencriptación no pueden proporcionar la misma configuración.

Por otro lado, en la generación de claves de extensión, se requieren operaciones complejas con el fin de generar claves de extensión que ofrezcan una mayor seguridad. Hay otro problema en el caso de la generación de claves de extensión en el sentido de que el número de bits de los datos de clave que deben ser introducidos como valor inicial debe ser fijo.

La presente invención tiene como objetivo proporcionar unos sistemas en los cuales los circuitos de encriptación y desencriptado sean los mismos, y en los cuales el área de los circuitos, el tamaño de los programas y el uso de las memorias que se utilizan para una conmutación de transformación no lineal puedan reducirse, y así mismo, puedan generarse unas claves de extensión utilizando una configuración más sencilla.

Divulgación de la invención

Este objeto es resuelto por un aparato transformación de datos de acuerdo con la reivindicación 1, el procedimiento de transformación de datos de acuerdo la reivindicación 3, el programa informático de acuerdo con la reivindicación 4, y el medio de almacenamiento de acuerdo con la reivindicación 5. Más mejoras en la transformación de datos del aparato se facilitan en la reivindicación 2.

Un aparato de transformación de datos de la presente invención se caracteriza porque en el aparato de transformación de datos que tiene una unidad de procesado de dato para introducir datos de clave y realizar al menos un encriptado de datos y desencriptado de datos,

el procesamiento de datos incluye hasta:

una unidad de transformación de subcampo para que los datos introducidos sean transformados, asumir los datos como elementos de un campo, transformar los datos mediante un circuito de elementos inverso que utilice un subcampo del campo, y una salida de datos transformados; y

un proceso de transformación afín por un vector de espacio GF (2^n) en GF (2), siempre que sean transformados por lo menos una de las primeras vueltas y una última vuelta de la unidad de transformación del subcampo, para los datos asumidos en GF (2^n) como elementos en GF(2^n) el cual corresponde de manera natural.

La anterior unidad de transformación de subcampos incluye solo unidades plurales de operaciones N/2-bit para dividir por igual los datos X que tienen bits N (N: incluido números) dentro de los datos superiores X_1 de N/2-bit y datos inferiores X_0 de N/2-bit para que sea $X = X_0 + \beta X_1$ (X_0, X_1; elementos del subcampos, β: un elemento del campo), y obtener los datos Y mediante la operación respectiva de los datos superiores Y_1 de N/2-bit y los datos inferiores Y_0 de N/2-bit para que resulte $Y = Y_0 + \beta Y_1 = 1/(X_0 + \beta X_1)$ (donde Y = 0, cuando X = 0).

El procedimiento de la transformación de datos de la presente invención se caracteriza por la cual en la que el procedimiento de transformación de datos ejecuta un proceso de procesamiento de datos para introducir datos de clave y realizar al menos un encriptado de datos y desencriptado de datos. El proceso de procesamiento de datos incluye:

una unidad de transformación de subcampo para que los datos introducidos sean transformados, asumir los datos como elementos de un campo, transformar los datos mediante un circuito de elementos inverso que utilice un subcampo del campo, y una salida de datos transformados; y

un proceso de transformación afín por un vector de espacio GF (2^n) en GF (2), siempre que sean transformados por lo menos una de las primeras vueltas y una última vuelta de la unidad de transformación del subcampo, para los datos asumidos en GF (2^n) como elementos en GF(2^n) el cual corresponde de manera natural.

La presente invención se caracteriza por un medio de almacenaje legible por computadora para almacenar un programa para que una computadora lleve a cabo el procedimiento de transformación de datos expuesto.

La presente invención se caracteriza por un programa para que una computadora lleve a cabo el procedimiento de transformación de datos expuesto.

Breve explicación de los dibujos

La Fig. 1 muestra una unidad de transformación de datos para una encriptación 100 y una unidad de transformación de datos para una desencriptado 400.

La Fig. 2 muestra unas anotaciones.

La Fig. 3 muestra una configuración de una unidad de encriptación 200 o de una unidad de desencriptado 500.

La Fig. 4 muestra otra configuración de la unidad de encriptación 200 o de la unidad de desencriptado 500.

La Fig. 5 muestra una configuración de una unidad de transformación normal de datos (FL) 251.

La Fig. 6 muestra una configuración de una unidad de transformación inversa de datos (FL^{-1}) 271.

La Fig. 7 muestra una parte de una unidad de encriptación convencional y una unidad de desencriptado convencional.

La Fig. 8 muestra una parte de la unidad de encriptación 200 y de la unidad de desencriptado 500.

La Fig. 9 muestra la unidad de transformación normal de datos (FL) 251 y la unidad de transformación inversa de datos (FL^{-1}) 271 que están situadas en simetría de puntos.

La Fig. 10 muestra la relación entre la unidad de transformación normal de datos (FL) 251 y la unidad de transformación inversa de datos (FL^{-1}) 271 que están situadas en simetría de puntos.

La Fig. 11 muestra una unidad de función no lineal F.

La Fig. 12 muestra una configuración de una primera unidad 13 de transformación de una caja S y una segunda unidad 14 de transformación de una caja S.

La Fig. 13 muestra una configuración de una unidad de transformación 21 de una caja S.

La Fig. 14 muestra una configuración de una unidad de transformación lineal 85.

La Fig. 15 muestra una configuración de una unidad de transformación lineal 87.

La Fig. 16 muestra una configuración de una unidad de generación de claves 300 o de una unidad de generación de claves 600.

La Fig. 17 explica las operaciones de una unidad de transformación 310 de la longitud de bits.

La Fig. 18 muestra una configuración de un registro de desplazamiento A 341.

La Fig. 19 muestra una configuración de una tabla de control de una unidad de control de desplazamiento 345.

La Fig. 20 muestra las operaciones del registro de desplazamiento A 341 y de un registro de desplazamiento B 342.

La Fig. 21 muestra la correspondencia entre el registro de desplazamiento A 341, el registro de desplazamiento B 342 y las claves de extensión.

La Fig. 22 muestra las operaciones de los registros de desplazamiento A 341 a D 344.

La Fig. 23 muestra la correspondencia entre los registros de desplazamiento A 341 a D 344 y las claves de extensión.

La Fig. 24 muestra una computadora equipada con la unidad de transformación de datos para la encriptación 100 y la unidad de transformación de datos para la desencriptado 400.

La Fig. 25 muestra una configuración de la función de encriptación por DES.

La Fig. 26 muestra una configuración de una función no lineal del cifrado de bloques E2 de 128 bits.

La Fig. 27 muestra otro ejemplo de unidades de transformación de cajas S.

La Fig. 28 muestra una unidad de función no lineal F que está equipada con las primera a cuarta unidades de transformación de cajas S.

La Fig. 29 muestra otra unidad de función no lineal F en la cual está desplazada la unidad de función de claves 25.

La Fig. 30 muestra otra unidad de función no lineal F en la cual está desplazado un emplazamiento de la unidad de función de clave 25.

La Fig. 31 muestra otra configuración de la unidad de función P 30.

La Fig. 32 muestra otra configuración de la unidad de función P 30.

La Fig. 33 muestra las configuraciones y operaciones de S1 a S4 de la Fig. 31.

La Fig. 34 muestra una prueba de la inexistencia de claves equivalentes.

La Fig. 35 muestra una prueba de la no existencia de claves equivalentes.

La Fig. 36 muestra otra configuración de la unidad de encriptación 200 o de la unidad de desencriptado 500.

La Fig. 37 muestra otra configuración de la unidad de encriptación 200 o de la unidad de desencriptado 500.

La Fig. 38 muestra otra configuración de la unidad de encriptación 200 o de la unidad de desencriptado 500.

La Fig. 39 muestra otra configuración de la unidad de encriptación 200 o de la unidad de desencriptado 500.

La Fig. 40 muestra otra configuración de la unidad de encriptación 200 o de la unidad de desencriptado 500.

La Fig. 41 muestra otra configuración de la unidad de encriptación 200 o de la unidad de desencriptado 500.

La Fig. 42 muestra una configuración en la cual las unidades de la Fig. 39 y de la Fig. 40 están combinadas.

La Fig. 43 muestra una configuración de la unidad de encriptación 200 o de la unidad de desencriptado 500, la cual se muestra en la Fig. 3, que utiliza la unidad de función no lineal F mostrada en la Fig. 28.

La Fig. 44 muestra una configuración modificada de la Fig. 43 mediante la utilización de una unidad de función no lineal F' en la cual la unidad de función de clave 25 de la unidad de función no lineal F está suprimida.

La Fig. 45 muestra una configuración modificada de la Fig. 44 mediante la fusión de las claves de extensión de blanqueo con las claves de extensión.

La Fig. 46 muestra una configuración modificada en la cual la unidad de función de clave 25 está suprimida de la unidad de función no lineal F y en la cual una clave de extensión k es suministrada a un circuito XOR 298, cuando la unidad de función no lineal F está configurada como se muestra en la Fig. 29.

La Fig. 47 muestra una configuración modificada en la cual la unidad de función de clave 25 está suprimida de la unidad de función no lineal F y en la cual una clave de extensión transformada de forma no lineal k' es suministrada al circuito XOR 298, cuando la unidad de función no lineal F está configurada como se muestra en la Fig. 30.

Mejor modo de llevar a cabo la invención

Realización 1

La Fig. 1 muestra una unidad de transformación de datos de encriptación 100 y una unidad de transformación de datos de descriptado 400 en esta forma de realización.

La unidad de transformación de datos de encriptación 100 es, por ejemplo, un dispositivo de encriptación que emite de salida unos textos cifrados de 128 bits a partir de unos textos no cifrados de entrada de 128 bits. La unidad de transformación de datos de desencriptado 400 es un dispositivo de desencriptado que emite de salida unos textos no cifrados de 128 bits a partir de unos textos cifrados de entrada de 128 bits. La unidad de transformación de datos de encriptación 100 se compone de una unidad de encriptación 200 y de una unidad de generación de claves 300. La unidad de encriptación 200 es una unidad de procesamiento de datos para la encriptación de textos no cifrados. La unidad de generación de claves 300 genera múltiples claves de extensión (n) de 64 bits o de 128 bits utilizando unas constantes V_i a partir de unos datos con clave de entrada de 128 bits, 192 bits o 256 bits, y para suministrarlos a la unidad de encriptación 200. La unidad de transformación de datos de desencriptado 400 se compone de una unidad de desencriptado 500 y de una unidad de generación de claves 600. La unidad de desencriptado 500 es una unidad de procesamiento de datos para descriptar textos cifrados. La unidad de generación de claves 600 es la misma o similar a la unidad de generación de claves expuesta 300. Así mismo, dado que la unidad de encriptación 200 y la unidad de desencriptado 500 pueden ejecutar el mismo procedimiento, pueden compartir un circuito o un programa, aunque la unidad de encriptación 200 y la unidad de desencriptado 500 se ilustran separadamente en las figuras. De modo similar, las unidades de generación de claves 300 y 600 no pueden compartir un circuito o un programa. Esto es, un circuito o un programa puede ser compartido por la unidad de transformación de datos de encriptación 100 y por la unidad de transformación de datos de desencriptado 400.

La Fig. 2 muestra los significados de las anotaciones utilizadas para las figuras o descripciones siguientes.

En la Fig. 3 y en las figuras subsecuentes, una mitad izquierda de datos se llama "datos de la izquierda L" y una mitad derecha de datos se llama "datos de la derecha R". Así mismo, los datos introducidos en las unidades de transformación no lineal de datos 210, 220, 230 y 240 son llamados "datos de entrada", los datos internos de las unidades de transformación no lineal de datos 210, 220, 230, y 240 se llaman "datos intermedios", y los datos emitidos de salida desde las unidades de transformación no lineal de datos 210, 220, 230, y 240 se llaman "datos de salida".

La Fig. 3 muestra un ejemplo de la unidad de encriptación 200 o de la unidad de desencriptado 500.

La Fig. 3 muestra una configuración en la cual están conectadas en cascada la unidad de transformación no lineal de datos 210 de 6 rondas, la unidad de transformación no lineal de datos 220 de 6 rondas, y la unidad de transformación no lineal de datos 230 de 6 rondas. La unidad de transformación normal de datos (FL) 251 y la unidad de transformación inversa de datos (FL^{-1}) 271 están insertadas entre la unidad de transformación no lineal de datos 210 de 6 rondas y la unidad de transformación no lineal de datos 220 de 6 rondas. Así mismo, la unidad de transformación normal de datos (FL) 253 y la unidad de transformación inversa de datos (FL^{-1}) 273 están insertadas entre la unidad de transformación no lineal de datos 220 de 6 rondas y la unidad de transformación no lineal de datos 230 de 6 rondas. Dentro de la unidad de transformación no lineal de datos 210 de 6 rondas, se disponen 6 rondas de unidades de transformación no lineal de datos. Por ejemplo, una unidad de transformación no lineal de datos 280 está compuesta por una unidad de función no lineal F y por un circuito XOR (OR exclusivo) 290. De esta forma, en el caso de la Fig. 3, en total se suministran 18 rondas de unidades de transformación no lineal de datos.

La unidad de transformación no lineal de datos 210 está equipada con una primera unidad de transformación no lineal de datos 280 y con una segunda unidad de transformación no lineal de datos 281. Para dos elementos de datos de entrada arbitrarios, los datos de entrada de la derecha R_o y los datos de entrada de la izquierda L_o, el primero lleva a cabo la primera transformación no lineal sobre los datos de entrada de la izquierda L_o utilizando una primera extensión de clave k_1, emite de salida un resultado operado con la función XOR de los datos de salida de la primera transformación no lineal y los datos de entrada de la derecha R_o como primeros datos intermedios de la izquierda L_1 y emite de salida los datos de entrada de la izquierda L_o como primeros datos intermedios de la derecha R_1. El último lleva a cabo una segunda transformación no lineal sobre los primeros datos intermedios de la izquierda R_1 utilizando una segunda clave de extensión k_2, emite de salida un resultado operado con la función XOR de los datos de salida de la segunda transformación no lineal y los primeros datos intermedios de la derecha R_1 como segundos datos

intermedios de la izquierda L_2, y emite de salida los primeros datos intermedios de la izquierda L_1 como segundos datos intermedios de la derecha R_2. La unidad de transformación no lineal de datos 210, en la cual están conectados en cascada de la primera unidad de transformación no lineal de datos 280 a la sexta unidad de transformación no lineal de datos 285, emite de salida los datos intermedios finales de la derecha R_6 y los datos intermedios de la izquierda L_6 como datos de salida después de la transformación.

La Fig. 4 muestra una configuración en la cual una unidad de transformación normal de datos (FL) 255, una unidad de transformación inversa de datos (FL^{-1}) 275, y una unidad de transformación no lineal de datos 240 de 6 rondas se añaden a la unidad de encriptación 200 como se muestra en la Fig. 3. En total, la transformación de datos se lleva a cabo mediante 24 rondas de unidades de transformación no lineal de datos.

La Fig. 5 muestra la unidad de transformación normal de datos (FL) 251.

La Fig. 5 muestra que la unidad de transformación normal de datos (FL) 251 divide los datos de entrada en dos elementos de datos, los datos de entrada 51 de la izquierda y los datos de entrada 52 de la derecha, lleva a cabo las operaciones lógicas para ambos elementos de datos, y genera los datos de salida a partir de los datos de entrada 60 de la izquierda y de los datos de entrada 61 de la derecha. Los datos de entrada 51 de la izquierda son operados con la función AND con una clave de extensión 53 en un circuito AND 54, y a continuación, los datos operados con la función AND son desplazados rotacionalmente hacia la izquierda (también llamado "desplazamiento circular") en 1 bit en una unidad de desplazamiento rotacional a la izquierda 55 de un 1 bit. Los datos desplazados son operados con la función XOR con los datos de entrada 52 de la derecha en un circuito XOR 56. La salida procedente del circuito XOR 56 se convierte en los datos de salida 61 de la derecha, y son operados con la función OR con una clave de extensión 57 en un circuito OR 58. A continuación, el resultado operado con la función OR es operado con la función XOR con los datos de entrada 51 de la izquierda en un circuito XOR 59 para generar los datos de salida 60 de la izquierda.

La Fig. 6 muestra la unidad de transformación inversa de datos (FL^{-1}) 271.

La Fig. 6 muestra que la unidad de transformación inversa de datos (FL^{-1}) 271 divide los datos de entrada en dos elementos de datos, los datos de entrada 71 de la izquierda y los datos de entrada 72 de la derecha, lleva a cabo las operaciones lógicas para ambos elementos de datos, y genera los datos de salida procedentes de los datos de salida 80 de la izquierda y los datos de salida 81 de la derecha.

Los datos de entrada 72 de la derecha son operados con la función OR con una clave de extensión 73 en un circuito OR 74, y a continuación, los datos operados con la función OR son operados con la función XOR con los datos de entrada 71 de la izquierda en un circuito XOR 75. A continuación, la salida procedente del circuito XOR 75 se convierte en los datos de salida 80 de la izquierda y son operados con la función AND con una clave de extensión 76 en un circuito AND 77. Después de ello, el resultado operado con la función AND es desplazado rotacionalmente hacia la izquierda en 1 bit en una unidad de desplazamiento rotacional 78 a la izquierda de 1 bit, y los datos desplazados son operados con la función XOR con los datos de entrada 72 de la derecha en un circuito XOR 79. La salida procedente del circuito XOR 79 se convierte en los datos de salida 81 de la derecha.

La unidad de transformación normal de datos (FL) 251 mostrada en la Fig. 5 y la unidad de transformación inversa de datos (FL^{-1}) 271 mostrada en la Fig. 6 llevan a cabo operaciones opuestas entre sí. De acuerdo con ello, utilizando la misma extensión de clave, los datos de entrada X de la Fig. 5 pueden obtenerse como datos de salida X de la Fig. 6 haciendo que los datos de salida Y de la Fig. 5 sean los datos de entrada Y de la Fig. 6.

La relación en la cual los datos de entrada en una unidad que pueden obtenerse como datos de salida procedentes de la otra unidad haciendo que los datos de salida procedentes de otra unidad sean los datos de entrada en la otra unidad es denominada una relación entre las transformaciones normal e inversa. La unidad de transformación normal de datos (FL) 251 y la unidad de transformación inversa de datos (FL^{-1}) 271 son circuitos que realizan dicha relación entre las transformaciones normal e inversa.

Tanto la unidad de desplazamiento rotacional 55 hacia la izquierda de 1 bit de la Fig. 5 como la unidad de desplazamiento rotacional 78 hacia la izquierda de 1 bit de la Fig. 6 llevan a cabo el desplazamiento hacia la izquierda, pero, ambas pueden ejecutar el desplazamiento a la derecha. Así mismo, la unidad de transformación normal de datos (FL) 251 y la unidad de transformación inversa de datos (FL^{-1}) 271 pueden ser cualquiera de una u otra configuraciones, en tanto en cuanto mantengan la relación entre las transformaciones normal e inversa. Por ejemplo, el número de desplazamientos puede modificarse. Además, pueden añadirse un circuito AND con una operación "not", un circuito OR con una operación "not", y/o un circuito XOR con una operación "not". Es decir, lo que sigue son las definiciones mostradas del circuito AND con una operación "not", el circuito OR con una operación "not", y el circuito XOR con una operación "not", representados por "andn", "orn", y "xorn", respectivamente.

x andn y : (not x) e y

x orn y : (not x) o y

x xorn y : (not x) e y

Algunos modernos CPUs están provistos de unos comandos “and”, “or”, y “xor” incluyendo “not”. Estos comandos pueden ejecutarse con el mismo coste que los comandos “and”, “or”, y “xor”.

La Fig. 7 muestra una unidad de encriptación convencional 201 y una unidad de desencriptado convencional 501.

La unidad de encriptación convencional 201 está equipada con dos unidades normales de transformación de datos FL. Así, la unidad de desencriptado debe estar equipada con dos unidades de transformación inversas de datos FL^{-1} con el fin de llevar a cabo las operaciones inversas. Por consiguiente, dado que la unidad de encriptación por regla general tiene una configuración diferente que la unidad de desencriptado, la unidad de encriptación y la unidad de desencriptado no pueden compartir el mismo circuito.

Por otro lado, como se muestra en la Fig. 8, en la presente forma de realización, la unidad de transformación normal de datos (FL) 251 y la unidad de transformación inversa de datos (FL^{-1}) 271 están situadas lado con lado en la unidad de encriptación 200, de forma que la unidad de desencriptado con la misma configuración pueda llevar a cabo la desencriptado. Por ejemplo, los datos de la derecha R son transformados por la unidad de transformación normal de datos (FL) 251 para obtener los datos de la izquierda L’, y los datos de la izquierda L son transformados por la unidad de transformación inversa de datos (FL^{-1}) 271 para obtener los datos de la derecha R’. En este caso, los datos de la derecha R pueden obtenerse introduciendo los datos de la izquierda L’ en la unidad de transformación inversa de datos (FL^{-1}) 271, y los datos de la izquierda L pueden obtenerse introduciendo los datos de la derecha R’ en la unidad normal de transformación de datos (FL) 251.

Como se describió anteriormente, la unidad de encriptación 200 y la unidad de desencriptado 500 pueden ser implementadas mediante la misma configuración, y la unidad de encriptación 200 y la unidad de desencriptado 500 pueden compartir el circuito.

La Fig. 9 muestra una configuración en la cual la unidad de transformación normal de datos (FL) 251 y la unidad de transformación inversa de datos (FL^{-1}) 271 están situadas en simetría de puntos sobre la unidad de transformación no lineal de datos 280.

De esta forma, cuando la unidad de transformación normal de datos (FL) 251 y la unidad de transformación inversa de datos (FL^{-1}) 271 están situadas en simetría de puntos sobre la unidad de transformación no lineal de datos 280, la encriptación y la desencriptado pueden llevarse a cabo utilizando la misma configuración.

La Fig. 10 muestra la correspondencia entre la unidad de transformación de datos (FL) y la unidad de transformación inversa de datos (FL^{-1}) situadas en simetría de puntos.

Como se muestra en la Fig. 10, en el caso de la Fig. 3, la unidad de transformación normal de datos (FL) 251 y la unidad de transformación inversa de datos (FL^{-1}) 271 están situadas en simetría de puntos sobre la unidad de transformación no lineal de datos 220 de 6 rondas.

En las Figs. 3, 4, 8, y 9, la unidad de transformación de datos (FL) y la unidad de transformación inversa de datos (FL^{-1}) pueden ser sustituidas entre sí. Además, en las Figs. 3, 4, 8, y 9, los datos de la derecha R y los datos de la izquierda L pueden ser sustituidos entre sí.

La Fig. 36 muestra una configuración en la cual la unidad de encriptación 200 se compone de la unidad de transformación no lineal de datos 210 de 6 rondas, la unidad de transformación no lineal de datos 220 de 6 rondas, y la unidad de transformación no lineal de datos 230 de 6 rondas.

La unidad de transformación no lineal de datos 210 de 6 rondas, la unidad de transformación no lineal de datos 220 de 6 rondas, y la unidad de transformación no lineal de datos 230 de 6 rondas son circuitos que pueden utilizarse para la encriptación y la desencriptado.

Aquí, una unidad de transformación normal/inversa de datos 211 está compuesta por la unidad de transformación no lineal de datos 210 de 6 rondas, y la unidad de transformación normal de datos (FL) 250, y la unidad de transformación inversa de datos (FL^{-1}) 271. La unidad de transformación normal/inversa de datos es un circuito que puede utilizarse tanto para la encriptación como para la desencriptado. Es decir, la unidad de transformación normal/inversa de datos es un circuito normal/inverso de transformación en el cual los datos de entrada en la unidad pueden obtenerse como datos de salida procedentes de la otra unidad haciendo que los datos de salida procedentes de la unidad sean los datos de entrada en la otra unidad.

Una unidad de transformación normal/inversa de datos 221 se compone también de la unidad de transformación no lineal de datos 220 de 6 rondas, y la unidad de transformación normal de datos (FL) 251, y la unidad de transformación inversa de datos (FL^{-1}) 273.

Así mismo, una unidad de transformación normal/inversa de datos 231 está compuesta por la unidad de transformación no lineal de datos 230 de 6 rondas, de la unidad de transformación normal de datos (FL) 253, y de la unidad de transformación inversa de datos (FL^{-1}) 275.

La unidad de encriptación 200 está configurada mediante la conexión en cascada de estas unidades normal/inversa de transformación de datos 211, 221, y 231. Y esta unidad de encriptación 200 puede también ser utilizada como unidad de desencriptado 500.

Así mismo, si un conjunto de la unidad de transformación no lineal de datos 210 de 6 rondas, de la unidad de transformación no lineal de datos 220 de 6 rondas, de la unidad de transformación normal de datos (FL) 251, y de la unidad de transformación inversa de datos (FL^{-1}) 271 se supone que sea una unidad de transformación no lineal de datos 1210, la unidad de transformación no lineal de datos 1210 es un circuito que puede utilizarse para la encriptación y la desencriptado. Aquí, una unidad de transformación normal/inversa de datos 1211 se compone de la unidad de transformación no lineal de datos 1210, de la unidad de transformación normal de datos (FL) 250, y de la unidad de transformación inversa de datos (FL^{-1}) 273.

Así mismo, si un conjunto de la unidad de transformación no lineal de datos 220 de 6 rondas, de la unidad de transformación no lineal de datos 230 de 6 rondas, de la unidad de transformación normal de datos (FL) 253, y de la unidad de transformación inversa de datos (FL^{-1}) 273 se supone que sea una unidad de transformación no lineal de datos 1220, una unidad de transformación normal/inversa de datos 1221 se compone de la unidad de transformación no lineal de datos 1220, de la unidad de transformación normal de datos (FL) 251, y de la unidad de transformación inversa de datos (FL^{-1}) 275.

Las unidades de transformación normal/inversa de datos 1211 y 1221 pueden utilizarse con destino a la unidad de desencriptado.

Así mismo, si un conjunto de las unidades 210 a 230 de transformación no lineal de datos de 6 rondas se supone que sean una unidad de transformación no lineal de datos 2210, la unidad de transformación no lineal de datos 2210 es un circuito que puede utilizarse tanto para la encriptación como para la desencriptado.

Aquí, la unidad de transformación no lineal de datos 2210, la unidad de transformación normal de datos (FL) 250, y la unidad de transformación inversa de datos (FL^{-1}) 275 constituyen una unidad de transformación normal/inversa de datos 2211.

La unidad de transformación normal/inversa de datos 2211 puede ser utilizada con destino a la unidad de desencriptado.

De acuerdo con lo anteriormente descrito, la unidad de encriptación 200 o la unidad de desencriptado 500 pueden ser configuradas mediante la conexión en cascada de múltiples unidades de transformación normales/inversas de datos.

Así mismo, en la unidad de encriptación 200 o en la unidad de desencriptado 500, la unidad de transformación normal/inversa de datos puede constituirse jerárquicamente alojando la unidad de transformación normal/inversa de datos dentro de la unidad de transformación normal/inversa de datos.

La Fig. 37 muestra un caso en el que la unidad de encriptación 200 y la unidad de desencriptado tienen la misma configuración que incluye la unidad de transformación no lineal de datos 210 de 6 rondas.

En la Fig. 37, la unidad de transformación no lineal de datos 210 de 6 rondas incluye unas rondas pares de unidades de transformación no lineal de datos 280 como se muestra en las Figs. 3 y 4. Los datos A son transformados en datos A' mediante una primera unidad de entrada de transformación normal de datos 256, los datos A' son introducidos en un primer puerto de entrada 261, los datos A' introducidos desde el primer puerto de entrada 261 son emitidos de salida desde un primer puerto de salida 263 como datos A_1'. Así mismo, la entrada de los datos B desde un segundo puerto de entrada 262 son emitidos de salida desde un segundo puerto de salida 264 como datos B_1. Los datos B_1 emitidos de salida desde el segundo puerto de salida 264 son transformados en datos B_1' por una segunda unidad de salida de transformación inversa de datos 279.

Los datos A_1' emitidos de salida desde el primer puerto de salida 263 de la unidad de encriptación 200 son introducidos en el segundo puerto de entrada 262 de la unidad de desencriptado 500 como datos A_1'. Los datos B_1' emitidos de salida desde la segunda unidad de salida de transformación inversa de datos 279 son introducidos en la primera unidad de entrada de transformación normal de datos 256 como datos B_1', y emitidos de salida como datos B_1.

La unidad de transformación no lineal de datos 210 introduce los datos B_1 y emite de salida los datos B. Así mismo, la unidad de transformación no lineal 210 introduce los datos A_1' y emite de salida los datos A'. La segunda unidad de salida de transformación inversa de datos 279 introduce los datos A' y emite de salida los datos A.

En la Fig. 38, la unidad de transformación no lineal de datos 219 de ronda impar incluye unas rondas impares de unidades de transformación no lineal de datos 280. De acuerdo con ello, la entrada de los datos A' procedente del primer puerto de entrada 261 son emitidos de salida desde el segundo puerto de salida 264 como datos A_1'. A continuación los datos A_1' son transformados por la segunda unidad de transferencia inversa de datos de salida 279, y emitidos de salida como datos A_1". Así mismo, los datos B introducidos en el segundo puerto de entrada 262 son emitidos de salida desde el primer puerto de salida 263 como datos B_1.

Los datos B_1 emitidos de salida desde el primer puerto de salida 262 de la unidad de encriptación 200 son introducidos en el segundo puerto de entrada 262 de la unidad de desencriptado 500 como datos B_1. Los datos A_1" emitidos de salida desde la segunda unidad de transformación inversa de datos de salida 279 de la unidad de encriptación 200 son introducidos en la unidad de desencriptado 500 como datos A_1" e introducidos en la primera unidad 256 de transformación normal de datos de entrada.

En los casos de las Figs. 37 y 38, la unidad de encriptación 200 y la unidad de desencriptado 500 tienen la misma configuración, llevando a cabo la encriptación y la desencriptado.

La Fig. 39 muestra un caso en el que la segunda unidad de transformación normal de datos de entrada 257 está dispuesta en el segundo puerto de entrada 262, y la primera unidad de transformación inversa de datos de salida 278 está dispuesta en el primer puerto de salida 263.

La Fig. 40 muestra un caso en el que la primera unidad de transformación inversa de datos de entrada 276 está dispuesta en el primer puerto de entrada 261, y la segunda unidad de transformación normal de datos de salida 259 está dispuesta en el segundo puerto de salida 264.

La Fig. 41 muestra un caso en el que las unidades de transformación normal/inversa de datos 256/258 están dispuestas en los puertos de entrada/salida 261, 263 de la izquierda y las unidades de transformación inversa de datos 277, 279 están dispuestas en los puertos de entrada/salida 262, 264 de la derecha.

La Fig. 42 muestra un caso en el que las Figs. 39 y 40 están combinadas.

Puede implementarse otro supuesto mediante la combinación de las Figs. 37 y 39, caso que no se muestra en la figura. Así mismo, las Figs. 38 y 39 pueden combinarse. Así mismo, la unidad de transformación no lineal de datos 210 de 6 rondas (rondas par) puede sustituirse por la unidad de transformación no lineal de datos 219 de rondas impar en las Figs. 37, 39 a 42, que no se muestran en las figuras. En los casos de las Figs. 39 a 42, la unidad de encriptación y la unidad de desencriptado pueden ser implementadas mediante la misma configuración.

Realización 2

La Fig. 11 muestra una configuración de una unidad de función no lineal F de la unidad de transformación no lineal de datos 280.

Una unidad de función no lineal F introduce unos datos de entrada 10 de la función F, lleva a cabo la transformación no lineal, y emite de salida unos datos de salida 40 de la función F. Los datos de entrada 10 de la función F de 64 bits son divididos en ocho elementos de datos, y procesados en la unidad de 8 bits. Cada uno de los datos de 8 bits es introducido en cada uno de los ocho circuitos XOR 12 de una unidad 25 de función de clave, operados con la función XOR con una clave de extensión 11, y se lleva a cabo una transformación no lineal utilizando una sustitución en una unidad de función S 20. A continuación, en una unidad de función P 30, dos elementos de los datos de 8 bits son operados con la función XOR mediante dieciséis circuitos XOR 815, y son emitidos de salida los datos de salida 40 de la función F de 64 bits. En la unidad 20 de función S, se disponen cuatro primeras unidades de transformación de cajas S 13 y cuatro segundas unidades de transformación de cajas S 14.

La Fig. 12 muestra un ejemplo de implementación de la primera unidad de transformación 13 de las cajas S y de la segunda unidad de transformación de caja S 14.

Dentro de la primera unidad de transformación de cajas S 13, se dispone una tabla de transformación T. La tabla de transformación T almacena previamente los valores de 0 a 255 de manera arbitraria (al azar) correspondientes a los valores de 0 a 255. La tabla de transformación T introduce los valores de 0 a 255 y emite de salida el valor (valor de 0 a 255) correspondiente a cada valor. Por ejemplo, cuando se introduce 1, la tabla de transformación T emite de salida 7. La tabla de transformación T lleva a cabo una transformación no lineal determinada con arreglo a consideraciones de seguridad, por ejemplo, verificando si la función es biyectiva o no, si la probabilidad diferencial es suficientemente pequeña o no, etc.

La segunda unidad de transformación de caja S 14 incluye la primera unidad de transformación de caja S 13 y una unidad de desplazamiento rotacional 22 a la izquierda de 1 bit (en la figura, "<<<" de "<<<1" muestra el desplazamiento rotacional a la izquierda y "1" muestra 1 bit). La unidad de desplazamiento rotacional 22 a la izquierda de 1 bit lleva a cabo el desplazamiento rotacional a la izquierda en 1 bit hasta una salida procedente de la primera unidad de transformación de caja S 13. Por ejemplo, cuando se introduce 1, la primera unidad de transformación de caja S 13 emite de salida 7, y la unidad de desplazamiento rotacional 22 de la izquierda de 1 bit emite de salida 14.

Si la primera unidad de transformación de caja S 13 y la segunda unidad de transformación de caja S 14 están configuradas como se muestra en la Fig. 12, se puede obtener un efecto similar al caso en el que se proporcionan dos tipos de tablas de transformación T, aunque no se requiere que existan dos tipos de tablas de transformación T. Mediante la inclusión de una sola tabla de transformación T, el uso de la memoria requerido para el almacenaje de la tabla de transformación T puede reducirse, y puede también reducirse la escala del circuito.

Así mismo, como se muestra en la Fig. 27, mediante la provisión de una unidad de desplazamiento rotacional a la derecha de 1 bit (">>>1" de la tercera unidad de transformación de caja S 15 de la Fig. 27) así como, o, en lugar de la unidad de desplazamiento rotacional 22 a la izquierda de 1 bit, puede obtenerse un efecto similar en un caso en el que se proporcione también una tabla de transformación diferente T. De otra forma, también es posible transformar los datos de entrada y utilizando la tabla de transformación de datos T después del desplazamiento de los datos de entrada y por la unidad de desplazamiento rotacional a la izquierda de 1 bit ("<<<1" de la cuarta unidad de transformación de caja S 16 de la Fig. 27) dispuesta para los datos de entrada y. La Fig. 27 muestra los casos de s(y), s(y)<<<1, s(y)>>>1, s(y<<<1), pero también son aplicables los casos de s(y>>>1), s(y<<<1)<<<1, s(y<<<1)>>>1, s(y>>>1)<<<1, s (y>>>1)>>>1. Haciendo que la cantidad desplazada sea 1 bit, resulta posible algunas veces funcionar más rápido que en casos de desplazamiento de 3 bits o 5 bits en el caso de que las CPUs, etc. tengan un comando de desplazamiento de 1 solo bit. Así mismo, cuando este proceso de desplazamiento se lleve a cabo mediante un software que lleve a cabo un desplazamiento de 1 solo bit, a veces es posible funcionar más rápido. Así mismo, el desplazamiento no está limitado a llevarse a cabo en 1 bit, sino que puede utilizarse un número arbitrario de bits, como por ejemplo 2 bits o 3 bits. Mediante el desplazamiento en un número arbitrario de bits, a veces resulta posible obtener un efecto similar al de la provisión de tipos diferentes de tablas.

La Fig. 28 muestra una unidad de función S 20 que utiliza las primera a cuarta unidades 13, 14, 15, 16 de transformación mostradas en la Fig. 27 de cuatro cajas S.

Otra configuración de la unidad de función P 30 se muestra en la Fig. 31.

A partir de los datos de entrada de 8 bits y_1, y_2, y_2, y_4, se obtienen unos datos Z_1,Z_2, Z_3, Z_4, de 32 bits mediante referencia a S1, S2, S3, S4, respectivamente, y son operados con la función XOR en un circuito 913. A partir de los datos de entrada de 8 bits y_5, y_6, y_7, y_8, se obtienen los datos Z_5, Z_6, Z_7, Z_8, de 32 bits mediante referencia a S2, S3, S4, S1, respctivamente, y son operados con la función XOR en un circuito 916. Este resultado U_2 operado con la función XOR y el primer resultado U_1 operado con la función XOR son operados con la función XOR en un circuito 917 para emitir de salida z_1', z_2', z_3', z_4'. A continuación, el resultado U_1, operado con la función XOR desde el circuito 913 es desplazado a la izquierda en un byte (en la Fig. 31, "<<<1" representa un desplazamiento rotacional de 1 byte, no un desplazamiento rotacional de 1 bit) en un circuito 918. El resultado desplazado es operado con la función XOR con la salida procedente del circuito 917 para emitir de salida z_5', z_6', z_7', z_8'.

Como se muestra en (a) a (d) de la Fig. 33, S1 se configura utilizando la primera unidad de transmisión de caja S 13, S2, se configura utilizando la segunda unidad de transformación de caja S 14, S3 se configura utilizando la tercera unidad de transformación de caja S 15, S4 se configura utilizando la cuarta unidad de transformación de caja S 16. Los datos de salida 8 bits procedentes de cada unidad de transformación son copiados cuatro veces para obtener datos de 32 bits, y así mismo, los datos de 32 bits son enmascarados para emitir de salida solo tres elementos de los datos (24 bits).

El desplazamiento rotacional de 1 byte del circuito 918 es un desplazamiento cíclico por unidad de longitud de bits (8 bits = 1 byte) el cual es procesado por la caja S.

La Fig. 32 muestra la unidad de función P cuya configuración es equivalente a la Fig. 31, pero la implementación es diferente.

A partir de los datos de entrada de 8 bits y_1, y_2, y_3, y_4, se obtienen los datos de 32 bits z_1, z_2, z_3, z_4, mediante referencia a S5, S6, S7, S8, y son operados con la función XOR en un circuito 933 para emitir de salida un resultado operativo A. A partir de los datos de entrada de 8 bits y_5, y_6, y_7, y_8, se obtienen los datos de 32 bits Z_5, Z_6, Z_7, Z_8 mediante referencia a S9, SA, SB, SC, y son operados con la función XOR en un circuito 936 para emitir de salida un resultado operativo B. El resultado operativo B es desplazado rotacionalmente a la derecha en 1 byte (en la Fig. 32, de modo similar a la Fig. 31, el desplazamiento se lleva a cabo por una unidad de longitud de bits (8 bits = 1 byte) que es procesado por la caja S, no 1 bit) en un circuito 937 y el resultado operativo B y el resultado operativo A son operados con la función XOR en un circuito 938. Este resultado operativo C es desplazado rotacionalmente hacia la parte superior (izquierda) en un byte en un circuito 939, y el resultado operativo C es también operado con la función XOR con el resultado operativo A en un circuito 940. Este resultado operativo D es desplazado rotacionalmente hacia arriba (izquierda) en 2 byte en un circuito 941, y el resultado operativo D es también operado con la función XOR con la salida procedente del circuito 939 en un circuito 942. Este resultado operativo E es desplazado rotacionalmente (a la derecha) en 1 byte en un circuito 943, y el resultado operativo E es también operado con la función XOR con la salida procedente del circuito 941 en un circuito 944. La salida F procedente del circuito 944 es emitido de salida como z_1', z_2', z_3', z_4', y la salida procedente del circuito 943 es emitida de salida como $z_{5'}$, $z_{6'}$, z_7', z_8'.

S5 y SC se configuran utilizando la primera unidad de transformación de caja S 13 y un desplazamiento lógico, S6 y S9 se configuran utilizando la segunda unidad de transformación de caja S 14 y un desplazamiento lógico, S7 y SA se configuran utilizando la tercera unidad de transformación de caja S 15 y un desplazamiento lógico, S8 y SB se configuran utilizando la cuarta unidad de transformación de caja S 16 y un desplazamiento lógico. El desplazamiento lógico se utiliza para emitir de salida unos datos de salida de 8 bits procedentes de cada unidad de transformación hasta un emplazamiento predeterminado situado dentro de los datos de salida de 32 bits. El desplazamiento lógico se fija para desplazarse hacia la izquierda en 0 byte en S5 y SA, en 1 byte en S6 y SB, en 2 bytes en S7 y SC, en 3 bytes en S8 y S9. Es decir, suponiendo una salida de 8 bits a partir de la unidad de transformación como z, puede representarse

una salida de 32 bits, como [0, 0, 0, z] (0 muestra que cada uno de los ocho bits es 0) en S5 y SA, [0, 0, z, 0] en S6 y SB, [0, z, 0, 0] en S7 y SC, [z, 0, 0, 0] en S8 y S9.

La implementación es posible utilizando unas tablas de sustitución cuya entrada sea de 8 bits y cuya salida sea de 32 bits, lo cual se calcula para producir directamente una salida predeterminada.

En los casos de las Figs. 31 y 32, puede disponerse el aparato que lleve a cabo una transformación a una velocidad más alta que la transformación empleada para el cifrado E2 convencional mostrado en la Fig. 26, y sobre el cual resulte posible también una implementación flexible.

En la Fig. 11, cuando las cajas S de la unidad de función S 20 están configuradas respectivamente mediante tipos diferentes de cajas S, se requieren ocho tablas de transformación T. Por otro lado, cuando las cajas S están configuradas como se muestra en la Fig. 12, el uso de la memoria requerida para el almacenaje de las tablas de transformación T puede reducirse a al menos la mitad.

Así mismo, ocho elementos de datos de 8 bits son introducidos por división de tiempo en la primera unidad de transformación de caja S 13 y en la segunda unidad de transformación de caja S 14 mostrada en la Fig. 12, de forma que las respectivas ocho cajas S convencionales pueden ser sustituidas por la primera unidad de transformación de caja S 13 y por la segunda unidad de transformación de caja S 14.

La Fig. 13 muestra otro ejemplo de la caja S de la unidad de función S 20.

La configuración concreta se explica con detalle en Matui, Sakurai, “circuito de división y circuito compartido del Campo Galois de multiplicación y división” [“Galois Field division circuit and shared circuit for multiplication and division”] (Registro de Patente japonesa No. 2641285 [2 de mayo de 1997]).

Los datos de 8 bits son introducidos en la unidad de transformación de caja S 21 y los datos de 8 bits son emitidos de salida. La unidad de transformación de caja S 21 se configura mediante una unidad de transformación lineal 17 de N bits (aquí, N = 8), una unidad de transformación de subcampo 18, y una unidad de transformación lineal 19 de N bits. La unidad de transformación lineal 17 de N bits lleva a cabo operaciones de datos de 8 bits. La unidad de transformación de subcampo 18 lleva a cabo operaciones de datos de solo 4 bits que son elementos del Campo Galois, GF (2^4). La unidad de transformación lineal 19 de N bits lleva a cabo una operación de datos de 8 bits. Una unidad de transformación lineal 85 de la unidad de transformación lineal 17 de N bits es un circuito que lleva a cabo la transformación lineal mostrada en la Fig. 14. Una unidad de transformación lineal 87 es un circuito que lleva a cabo la transformación lineal mostrada en la Fig. 15.

La unidad de transformación lineal 85 puede ser sustituida por un circuito que lleve a cabo una transformación afín (una transformación lineal puede considerarse como un tipo de transformación afín). De modo similar, la unidad de transformación lineal 87 puede ser sustituida por un circuito que lleve a cabo otra transformación afín. La unidad de transformación afín 85 transforma los datos de 8 bits (X) en datos de 8 bits (X’). Los datos de 8 bits obtenidos (X’) se supone que son elementos del Campo Galois (2^8). Los datos de 4 bits superiores y los datos de 4 bits inferiores (X_1 y X_0) de los datos X’ se supone que son, respectivamente, elementos del Campo Galois de subcampo (2^4) y son emitidos de salida hasta la unidad de transformación de subcampo 18. Aquí, por ejemplo, suponiendo que un elemento β del CG (2^8) sea un elemento que satisface el polinomio irreductible $X^8 + X^6 + X^5 + X^3 + 1 = 0$, y $\alpha = \beta^{238}$, una base del CG (2^4) de subcampo puede ser representada como [1, α, α^2, α^3]. Si los elementos del CG (2^4), X_0, X_1, son representados utilizando esto, la relación subsecuente puede establecerse como $X' = X_O + \beta X_1$. (Para los detalles, consúltese Matui, Sakurai, “circuito de división y circuito compartido del Campo Galois de multiplicación y división” [“Galois Field division circuit and shared circuit for multiplication and division”] (Registro de Patente japonesa No. 2641285 [2 de Mayo de 1997])). La unidad de transformación de subcampo 18 se configura solo por las unidades operativas cada una de las cuales lleva a cabo operaciones de datos de 4 bits.

Aquí, como ejemplo de “subcampo” de extracción, el subcampo CG (2^m) donde n = 2m puede tomarse en consideración para un CG (2^n) determinado. En este ejemplo, n = 8, m = 4.

La unidad de transformación de subcampo 18 es un circuito de elementos inversos que utiliza el subcampo construido por el circuito mostrado en el documento ““circuito de división y circuito compartido del Campo Galois de multiplicación y división” [“Galois Field division circuit and shared circuit for multiplication and division”] (Registro de Patente japonesa No. 2641285 [2 de Mayo de 1997]). Como resultado operativo de este circuito de elementos inversos, los datos de 4 bits superiores y los datos de 4 bits inferiores (Y_1 e Y_0), cada uno de los cuales puede considerarse como un elemento del CG (2^4), son emitidos de salida hasta la unidad de transformación lineal 87 como datos de 8 bits Y los cuales pueden considerarse como un elemento del CG (2^8), donde $Y = Y_0 + \beta Y_1$. Como se expuso anteriormente, este circuito de elementos inversos es un circuito de computación $Y = Y_0 + \beta Y_1 = 1/(X_o + \beta X_1)$. Así mismo, hay algunas formas de adoptar una “base”, como por ejemplo una “base polinómica” y una “base normal”, en la representación del elemento de “campo finito” (cómo adoptar una base) en el circuito de elementos inversos.

Una primera característica de la unidad de transformación de caja S 21 mostrada en la Fig. 13 es computar los datos con una anchura de bits (4 bits) que es la mitad de la anchura de bits (8 bits) de la entrada de datos de la transformación no lineal. Es decir, el circuito de elementos inversos se caracteriza por llevar a cabo operaciones de datos de solo 4 bits.

Sin embargo, la velocidad de computación puede reducirse llevando a cabo operaciones de solo 4 bits. Este supuesto ofrece la ventaja de que la escala de un circuito completo puede ser mucho menor que en un supuesto en el que se lleven a cabo operaciones de datos de 8 bits.

Así mismo, una segunda característica de la unidad de transformación de caja S 21 consiste en que la unidad de transformación lineal 17 de N bits y la unidad de transformación lineal 19 de N bits, donde N = 8, se disponen a ambos lados de la unidad de transformación de subcampo 18. Cuando la unidad de transformación de caja S 21 es implementada utilizando la unidad de transformación de subcampo 18, existe la ventaja de que puede reducirse una escala del entero circuito y la configuración resulta más sencilla en comparación con el supuesto que emplea una tabla de transformación T que almacena valores aleatorios, mientras que por el contrario, la seguridad puede reducirse. De acuerdo con ello, las transformaciones lineales o las transformaciones afines se llevan a cabo a ambos lados de la unidad de transformación de subcampo 18, de forma que pueda recuperarse la reducción del nivel de seguridad debido a la implementación que utiliza la unidad de transformación de subcampo 18.

En la Fig. 13, las transformaciones lineales se llevan a cabo a ambos lados de la unidad de transformación de subcampo 18; sin embargo, la transformación lineal puede llevarse a cabo solo en un lado. En otra variante, la transformación lineal puede llevarse a cabo en un lado, y la transformación afín puede llevarse a cabo en el otro lado.

La Fig. 29 muestra un supuesto en el que la unidad de función de clave 25 se muestra en la Fig. 11, esto es, la unidad de función de clave 25 situada antes de la unidad de función S 20 y de la unidad de función P 30, ahora se sitúa después de la unidad de función S 20 y de la unidad de función P 30.

La Fig. 30 muestra un supuesto en el que la unidad de función de clave 25 está situada entre la unidad de función S y la unidad de función 30.

Mediante el empleo de la configuración mostrada en la Fig. 29 o la Fig. 30, se puede tener un efecto de que una implementación proporciona una operación de velocidad más alta de lo que lo hace la configuración mostrada en la Fig. 11. Así mismo, mediante la modificación de la generación de las claves de extensión, puede obtenerse la misma salida utilizando la configuración mostrada en la Fig. 29 o en la Fig. 30 a partir de la misma entrada que la de la configuración de la Fig. 11. En la unidad de función F convencional mostrada en la Fig. 26, se ofrecen dos funciones S, en cada una de las cuales se lleva a cabo en primer término una operación con la clave de extensión y a continuación se lleva a cabo una operación de función S. Por el contrario, en el caso mostrado en la Fig. 29, una unidad de función de clave 25 está situada en la etapa final de la función F. En el caso mostrado en la Fig. 30, la unidad de función de clave 25 está situada entre la unidad de función S 20 y la unidad de función P 30.

La Fig. 43 muestra un caso en el que la unidad de transformación no lineal F mostrada en la Fig. 28 se emplea en la unidad de encriptación 200 o en la unidad de desencriptado 500 mostradas en la Fig. 3.

Los datos de la izquierda son introducidos en la unidad de transformación no lineal F como datos de entrada 10 de la función F, y los datos de salida 40 de la función F son emitidos de salida. Los datos de salida 40 de la función F son operados con la función XOR con los datos de la derecha, y el resultado operado con la función XOR se convierte en los datos de la izquierda de la siguiente ronda. Cuando los datos de la izquierda son introducidos en la unidad de transformación lineal F como datos de entrada 10 de la función F, al mismo tiempo, los datos de la izquierda son utilizados como datos de la derecha de la siguiente ronda. En la configuración mostrada en la Fig. 43, las operaciones de la unidad de la función de clave 25, de la unidad de función S 20, y de la unidad de función P 30 se llevan a cabo en la unidad de transformación no lineal F, de forma que la carga de las operaciones resulta considerable dentro de la unidad de transformación no lineal F. Un caso ejemplar en el que puede conseguirse una velocidad de procesamiento más alta mediante la distribución de la carga operativa de la unidad de transformación no lineal F, se expondrá a continuación con referencia a las figuras.

La Fig. 44 muestra un supuesto en el que se utiliza la unidad de transformación no lineal F'. La unidad de transformación no lineal F' es una unidad en la que la unidad de la función de claves 25 es suprimida de la unidad de transformación no lineal F mostrada en la Fig. 43. La clave de extensión k_1 es operada con la función XOR con los datos de la izquierda L_0 en un circuito XOR 891. Así mismo, la clave de extensión k_2 es operada con la función XOR con los datos de la derecha R_0 en un circuito XOR 297. Los datos de la izquierda son introducidos en la unidad de transformación no lineal F' como datos de entrada 10 de la función F, y transformados por la unidad de función F 20 y por la unidad de función P 30. La salida procedente del circuito XOR 297 y los datos de salida 40 de la función F son operados con la función XOR en un circuito XOR 290 para emitir de salida los datos de la izquierda L_1.

Por otro lado, las unidades de generación de claves 300, 600 llevan a cabo una operación XOR de las claves de extensión k_1 y k_2 y emiten de salida la clave de extensión modificada $k_1 + k_3$. La salida R_1 del circuito XOR 891 y la clave de extensión $k_1 + k_3$ son operadas con la función XOR en un circuito XOR 298 para emitir de salida los datos de la derecha. Las unidades de generación de claves 300, 600 modifican las claves de extensión para generar y emitir de salida $k_1 + k_3$, $k_2 + k_4$, $k_3 + k_5$, ..., $k_{16} + k_{18}$. Las unidades de generación de claves 300, 600 suministran las claves de extensión modificadas a procesos distintos del proceso de la función no lineal (F) para operar con los datos. Como resultado de ello, los datos de la izquierda L_{18} y los datos de la derecha R_{18} resultan ser los mismos que los datos de la izquierda L_{18} y que los datos de la derecha R_{18} en el caso de la Fig. 43.

Las claves de extensión modificadas son suministradas a procesos distintos del proceso de la función no lineal (F) y operados con los datos, y en consecuencia, las operaciones con los datos con clave pueden ser llevadas a cabo fuera de la unidad de función no lineal F', a saber, en los circuitos XOR 297 y 298, mientras que las operaciones de la unidad de función S 20 y de la unidad de función P 30 son llevadas a cabo en la unidad de función no lineal F'. Por consiguiente, las operaciones de la unidad de función de clave 25 son eliminadas de la unidad de función no lineal F, y la carga de la unidad de función no lineal F es distribuida, lo cual permite una implementación de alta velocidad.

La Fig. 45 muestra un caso en el que las operaciones de la clave de extensión de blanqueo kw_1, se llevan a cabo también como operaciones de las otras claves de extensión en la configuración mostrada en la Fig. 44. La Fig. 45 muestra un caso en el que la unidad de generación de claves previamente lleva a cabo una operación XOR de una parte de la clave de extensión de blanqueo, kw_{1alta} y de la primera clave de extensión k_1 (es decir, la unidad de generación de claves modifica la clave de extensión) y suministra el resultado de la operación al circuito XOR 891.

La figura muestra también un caso en el que la unidad de generación de claves lleva a cabo previamente una operación XOR como parte de la clave de extensión de blanqueo kw_{1baja} y de la segunda clave de extensión k_2 (es decir, la unidad de generación de claves modifica la clave de extensión) y suministra el resultado de la operación al circuito XOR 297.

De esta forma, puede ser eliminada la operación en el circuito XOR 293 mostrado en la Fig. 44. Así mismo, en un caso mostrado en la Fig. 45, la unidad de generación de claves lleva a cabo una operación XOR de una parte de la clave de extensión de blanqueo kw_{2baja} y de la clave de extensión k_{17} (es decir, la unidad de generación de claves modifica la clave de extensión) y suministra el resultado de la operación al circuito XOR 299. Así mismo también, la unidad de generación de claves lleva a cabo una operación XOR de la otra parte de la clave de extensión de blanqueo kw_{2alta} y de la clave de extensión k_{18} (es decir, la unidad de generación de claves modifica la clave de extensión) y suministra el resultado de la operación al circuito XOR 892.

De esta forma, se elimina la operación del circuito XOR 296 mostrada en la Fig. 44.

La Fig. 46 muestra un caso en el que la unidad de función de clave 25 es suprimida de la unidad de función no lineal F, y en su lugar, la unidad de generación de claves suministra la clave de extensión k al circuito XOR 298 cuando la unidad de función no lineal F está configurada como se muestra en la Fig. 29.

La Fig. 47 muestra un caso en el que la unidad de función de claves 25 es suprimida de la unidad de función no lineal F, y en su lugar, la unidad de generación de claves suministra la clave de extensión transformada de manera no lineal k' = P(k) al circuito XOR 298 cuando la unidad de función no lineal F está configurada como se muestra en la Fig. 30. En el caso de la Fig. 47, se lleva a cabo la misma operación efectuada por el proceso de función P sobre los datos con clave para generar datos con clave transformados de manera no lineal, y los datos con clave transformados de manera no lineal son suministrados a procesos distintos del proceso de función no lineal (F) para procesar los datos que van a ser operados con los datos como datos con clave destinados a los datos de procesamiento. En ambos casos, de las Figs. 46 y 47, debido a que la unidad de función de clave 25 es eliminada de la unidad de función no lineal F, la carga operativa de la unidad de función no lineal F se reduce, y la operación del circuito XOR 298 situado fuera de la unidad de función no lineal F puede llevarse a cabo en paralelo con las operaciones llevadas a cabo por la unida de función no lineal F, lo que posibilita un procesamiento de alta velocidad.

Realización 3

La Fig. 16 muestra una configuración de la unidad de generación de claves 300 (o de la unidad de generación de claves 600) mostrada en la Fig. 1.

La unidad de generación de claves 300 incluye una unidad de transformación 310 de la longitud de los bits, una primera unidad de transformación de claves 320 de G bits, una segunda unidad de transformación de claves 330 de G bits, y una unidad de desplazamiento de claves 340. A partir de los datos con clave de entrada con 128 bits, 192 bits, o 256 bits, la unidad de generación de claves 300 genera unos datos con clave K_1 de 128 bits y unos datos con clave K_2 de 128 bits, y emite de salida varias claves de extensión de 64 bits. La unidad de transformación 310 de la longitud de los bits convierte la longitud de los bits de los datos con clave que van a ser emitidos de salida de forma que la longitud de bits de los datos con clave de salida fijados incluso si son introducidos los datos con clave con diferente número de bits. En otras palabras, la unidad de transformación 310 de longitud de los bits genera unos datos con clave SK_{altos} de 128 bits superiores y unos datos con clave SK_{bajos} de 128 bits inferiores y emite de salida los primeros hasta la unidad de transformación de claves 320 de G bits y hasta la unidad de desplazamiento de claves 340. Así mismo, los últimos son emitidos de salida hasta la segunda unidad de transformación de claves 330 de G bits y hasta la unidad de desplazamiento de claves 340. Así mismo, los datos con clave de 128 bits que son un resultado operado con la función XOR de los primeros y de los últimos son emitidos de salida hasta la primera unidad de transformación de claves 320 de G bits.

La Fig. 17 muestra las operaciones internas de la unidad de transformación 310 de la longitud de los bits.

Cuando los datos con clave de 128 bits son introducidos en la unidad de transformación 310 de la longitud de los bits, los datos con clave introducidos son emitidos de salida como datos con clave SK_{altos} de los 128 bits superiores sin cambio alguno. Así mismo, los datos con clave SK_{bajos} de los 128 bits inferiores son puestos a 0 y emitidos de salida.

Cuando los datos con clave de 192 bits son introducidos en la unidad de transformación 310 de la longitud de los bits, los datos superiores 128 bits de los datos con clave de entrada son emitidos de salida como datos con clave superiores de 128 bits SK_{altos} sin cambio alguno. Así mismo, los datos con clave inferiores de 128 bits SK_{bajos} son generados mediante la combinación de los 64 bits inferiores de los datos con clave de 192 bits introducidos y de los datos inversos de 64 bits, los cuales son generados mediante la inversión de los datos inferiores de 64 bits de los datos con clave de 192 bits introducidos, y emitidos de salida.

Cuando los datos con clave de 256 bits son introducidos, los datos superiores de 128 bits de los datos con clave introducidos son emitidos de salida como SK_{altos}, y los datos inferiores de 128 bits son emitidos de salida como SK_{bajos}.

Unos datos XOR de los datos con clave de 128 bits SK_{altos} y SK_{bajos} son introducidos en la primera unidad de transformación de claves 320 de G bits desde la unidad de transformación 310 de la longitud de los bits, operados mediante transformaciones no lineales de dos rondas, operados con la función XOR con los datos con clave superiores de 128 bits SK_{altos}, adicionalmente operados por unas transformaciones lineales de dos rondas, y unos datos con clave de 128 bits K_1 son emitidos de salida.

Cuando la longitud de los datos con clave introducidos en la unidad de transformación 310 de la longitud de los bits es de 128 bits, la unidad de desplazamiento de claves 340 genera la clave de extensión utilizando los datos con clave de 128 bits emitidos de salida desde la unidad de transformación de claves 320 de G bits y los datos con clave originalmente introducidos. Cuando la longitud de los datos con clave introducidos en la unidad de transformación 310 de la longitud de los bits es de 192 bits o de 256 bits, los datos con clave de 128 bits emitidos de salida desde la primera unidad de transformación de claves 320 de G bits son también introducidos en la segunda unidad de transformación 340 de G bits, operados con la función XOR con los datos con clave inferiores de 128 bits SK_{bajos}, operados por dos transformaciones no lineales de dos rondas, y unos datos con clave de 128 bits K_2 son emitidos de salida. Dos elementos de los datos con clave de 128 bits, procedentes de la unidad de transformación de claves 320 de G bits y de la segunda unidad de transformación de claves 330 de G bits, son emitidos de salida hasta la unidad de desplazamiento de claves 340. La unidad de desplazamiento de claves 340 genera la clave de extensión utilizando los dos elementos de datos con clave de 128 bits y los datos con clave originariamente introducidos.

La unidad de desplazamiento de claves 340 incluye un registro de desplazamiento A 341, un registro de desplazamiento B 342, un registro de desplazamiento C 343, un registro de desplazamiento D 344 y una unidad de control de desplazamiento 345. La unidad de control de desplazamiento 345 emite de salida una señal de selección 346 hasta cada uno de los registros de desplazamiento para controlar las operaciones de los registros de desplazamiento.

La Fig. 18 muestra una configuración del registro de desplazamiento A 341.

El registro de desplazamiento A 341 incluye un selector A 347 que incorpora un grupo de conmutadores para 128 bits y un registro A 348 de 128 bits. Una señal de selección 346 incluye una señal de conmutación para indicar la conexión con todos los conmutadores del selector A 347 al mismo tiempo sobre el lado A y sobre el lado B. La figura muestra un caso en el que el grupo de conmutadores del selector A 347 ha seleccionado A en base a la señal de selección 346, y en este caso, el registro A 348 lleva a cabo un desplazamiento rotacional hacia la izquierda en 17 bits. Así mismo, cuando el grupo de conmutadores está conectado a B, el registro A lleva a cabo el desplazamiento rotacional a la izquierda en 15 bits. El desplazamiento en 15 bits o el desplazamiento en 17 bits se lleva a cabo mediante un ciclo cronometrado.

El número de bits de desplazamiento (15, 17) es uno entre muchos ejemplos, y puede aplicarse otro número de bits de desplazamiento.

La Fig. 19 muestra una parte de una tabla de control almacenada en la unidad de control de desplazamiento 345.

La tabla de control es una tabla que almacena el número de bits que el registro desplaza en cada reloj. Por ejemplo, en la tabla de control del registro A, en el primer reloj, se especifica un desplazamiento de 15 bits. Y, en el segundo reloj, se especifica un desplazamiento de 15 bits adicionales. De modo similar, en cada uno de los tercero y cuarto relojes, se especifica un desplazamiento de 15 bits. En cada uno de los quinto a octavo relojes se especifica un desplazamiento de 17 bits.

La Fig. 20 muestra un resultado de control con arreglo al cual la unidad de control de desplazamiento 345 controla cada registro de desplazamiento utilizando la tabla mostrada en la Fig. 19 en el caso de generar la clave de extensión a partir de los datos con clave de 128 bits.

Los datos con clave superiores de 128 bits SK_{altos} introducidos desde la unidad de transformación 310 de la longitud de los bits son insertados en el registro de desplazamiento A 341. Los datos con clave de 128 bits K_1 emitidos de salida desde la unidad de transformación de claves 320 de G bits son insertados en el registro de desplazamiento B 342. En esta situación, el registro de desplazamiento A 341 y el registro de desplazamiento B 342 operan en base a la tabla de

control mostrada en la Fig. 19. En la Fig. 20, los datos de una columna inclinada muestran que deben ser ignorados y no deben ser emitidos de salida. Los datos de las otras columnas son emitidos de salida como claves de extensión como se muestra en la Fig. 21.

La Fig. 21 muestra una correspondencia entre el valor de los registros y la clave de extensión.

La Fig. 20 muestra un caso en el que se llevan a cabo cuatro desplazamientos de 15 bits en cada reloj, y a partir del quinto reloj, los desplazamientos se llevan a cabo en 17 bits en cada reloj. La decisión para emitir o no de salida los 64 bits superiores y los 64 bits inferiores desde el registro de desplazamiento A 341 y el registro de desplazamiento B 342 como clave de extensión y su orden de emisión de salida se especifica en la tabla de control, que no se muestra en la figura. Y de acuerdo con la tabla de control, mediante la emisión de salida de la señal de selección 346 que incluye una señal de instrucción de salida al registro de desplazamiento, la clave de extensión es emitida de salida desde cada registro de desplazamiento en 64 bits.

La Fig. 22 muestra un caso en el que la clave de extensión es generada desde unos datos con clave de 192 bits o de 256 bits.

Es decir, los datos con clave superiores de 128 bits SK_{altos} introducidos desde la unidad de transformación 310 de la longitud de los bits son insertados en el registro de desplazamiento A 341, los datos con clave inferiores de 128 bits SK_{bajos} se insertan en el registro de desplazamiento B 342, los datos con clave de 128 bits K_1 emitidos de salida desde la primera unidad de transformación de claves 320 de G bits se insertan en el registro de desplazamiento C 343, y los datos con clave de 128 bits K_2 emitidos de salida desde la segunda unidad de transformación de claves 330 de G bits se insertan en el registro de desplazamiento D 344.

Los datos de una columna que tienen una inclinación muestran las claves no utilizadas para las claves de extensión.

La Fig. 23 muestra una correspondencia entre el valor del registro y la clave de extensión.

Las claves no utilizadas para las claves de extensión y la correspondencia entre el valor del registro y la clave de extensión mostrada en la Fig. 23 son almacenadas en la tabla de control situada en el controlador.

Como se muestra en la Fig. 19, la unidad de control de desplazamiento 345 almacena el número de bits de desplazamiento de los datos con clave fijados en el registro de desplazamiento A 341. Es decir, las claves de extensión son generadas de forma secuencial mediante el desplazamiento de los datos con clave insertados en el registro de desplazamiento A 341 mediante $Z_0 = 0$ bits, $Z_1 = 15$ bits, $Z_2 = 45$ bits, $Z_3 = 60$ bits, $Z_4 = 77$ bits, $Z_5 = 94$ bits, $Z_6 = 111$ bits, $Z_7 = 128$ bits como se muestra en la tabla de control del registro de desplazamiento A.

La suma del número de bits de desplazamiento resulta ser 15 + 15 + 15 +15 + 17 +17 + 17 +17 = 128, de forma que el registro de 128 bits lleva a cabo el desplazamiento rotacional de 128 bits y el registro retorna a su estado inicial.

La razón por la cual la suma del número de bits de desplazamiento alcanza 128 bits (el número de bits del registro) para retornar al estado inicial es que el procesamiento siguiente puede iniciarse de inmediato si el procesamiento siguiente es asignado al registro del estado inicial. Así mismo, en el caso de que se lleve a cabo una transformación inversa (desencriptado), el proceso de generación de la clave de extensión se inicia desde el estado inicial, y de acuerdo con ello, tanto la transformación (encriptación) como la transformación inversa (desencriptado) pueden llevarse a cabo mediante la fijación del estado inicial. Así mismo, la razón por la cual la suma del número de bits de desplazamiento no se fija en más de 128 bits (el número de bits del registro) es para impedir la generación de valores idénticos a los del estatus existente en el mismo registro de desplazamiento debido a la ejecución del desplazamiento en más de un ciclo (mayor de 128 bits de desplazamiento). Esto es porque, por ejemplo, la realización del desplazamiento rotacional en 2 bits que es menos de 128 bits (el número de bits del registro) y la realización del desplazamiento rotacional de 130 bits, que es más de 128 bits (el número de bits del registro), producen el valor idéntico. Es deseable fijar dichos valores en la tabla de control del registro A porque, al realizar los desplazamientos del registro en un ciclo, el número de bits de desplazamiento varía de forma irregular a lo largo del único ciclo. Sin embargo, con el fin de facilitar la configuración del registro del desplazamiento, se desea un desplazamiento en un número de bits fijo. Por consiguiente, un registro está configurado para llevar a cabo dos tipos de desplazamientos en 15 bits y en 17 bits (en un reloj), y la operación de desplazamiento mediante números de bits diferentes puede implementarse utilizando los dos tipos de desplazamientos, de acuerdo con el siguiente procedimiento.

Establecida la relación para que $Z_1 - Z_0 = 15$ (aquí, $Z_1 - Z_0 = B_1$), $Z_2 - Z_1 = 30$ (es decir, $Z_2 - Z_1 = 2B_1$), por consiguiente, $Z_2 - Z_1 = 2(Z_1 - Z_0)$. Así mismo, como se muestra en la tabla de control del registro de desplazamiento B, establecida la relación para que $Z_5 - Z_4 = 34$ (aquí, $Z_5 - Z_4 = 2B_2$), $Z_6 - Z_5 = 17$ (es decir, $Z_6 - Z_5 = B_2$), por consiguiente, $Z_5 - Z_4 = 2(Z_6 - Z_5)$. Es decir, la diferencia entre los números de los bits de desplazamiento nos da 15 bits y 30 bits, o 17 bits y 34 bits, y el número de bits de desplazamiento (30 bits o 34 bits) se establece en un múltiplo integral (2 veces = I veces) del número de bits (15 bits y 17 bits) para un desplazamiento de una sola vez.

De esta forma, cuando las diferencias del número de bits de desplazamiento se establecen o bien respecto del número de bits de desplazamiento para una sola vez, o bien para el múltiplo del número entero mayor de dos (I veces, I es un número entero mayor de 2) y el número de bits de desplazamiento para una sola vez, mediante la operación

del registro de desplazamiento A 341 de una o dos veces (I veces), es posible implementar fácilmente operaciones de desplazamiento cuyo número de bits de desplazamiento esté almacenado en la tabla de control. Operar dos veces (I veces) significa que la operación de desplazamiento termina con dos relojes (I relojes) del reloj operativo suministrado para operar el registro de desplazamiento A 341.

Aquí, al desplazarse I veces (dos veces), tanto los datos más altos como los datos más bajos de los datos desplazados hasta I - 1 veces (2 - 1 = 1 vez) son ignorados y no se utilizan para la clave de extensión. Por ejemplo, en el caso de desplazamiento de Z_1 = 15 a Z_2 = 45, I = (Z_2 - Z_1) / (el número de bits de desplazamiento de una sola vez) = (45 - 15)/15 = 2, y tanto los datos más altos como los datos más bajos de los datos desplazados después del desplazamiento I - 1 veces (2 - 1 = 1 vez) son ignorados y no se utilizan para la clave de extensión. Esto puede apreciarse en la Fig. 20, en la cual las columnas de la clave [8] y la clave [9] presentan inclinaciones, que muestran que estas claves no se utilizan para las claves de extensión. Y uno u otro o ambos datos más altos y datos más bajos de los datos desplazados después del desplazamiento de I veces (2 veces) es o son utilizados como clave de extensión. Esto puede apreciarse en la Fig. 20, la cual muestra que la clave [12] y la clave [13] son emitidas de salida como claves de extensión.

Las razones por las cuales las operaciones de desplazamiento en base a un múltiplo de un entero mayor de dos empleado de acuerdo con lo anteriormente descrito, se deben a la posibilidad de realizar el desplazamiento de no solo 15 bits o de 17 bits, sino también de 30 (= 15 x 2) bits, 34 (= 17 x 2) bits, (o 45 (= 15 x 3) bits o 51 (= 17 x 3) bits, etc.), lo que modifica el número de desplazamientos y mejora en mayor medida la seguridad. Y, la razón por la que son suministrados casos en los que los datos desplazados no se utilizan para la clave de extensión es también para mejorar la seguridad.

Se desea generar los datos que no son utilizados para la clave de extensión (en las Figs. 20 y 22, las claves de las columnas con inclinaciones, las cuales no se utilizan para las claves de extensión) cuando, por ejemplo, el procesamiento del hardware o el procesamiento del programa no se lleva a cabo de forma consecutiva. Como ejemplos concretos, en la Fig. 3 se desea generar dichos datos cuando se llevan a cabo las operaciones de la unidad de transformación normal de datos (FL) y de la unidad de transformación inversa de datos (FL^{-1}), o antes o después de dichas operaciones o en momentos de inactividad de procesos o momentos de conmutación de procesos, como por ejemplo una llamada a función por un programa, una llamada a subrutina, o un proceso de tratamiento de interrupciones.

La característica de la tabla de control mostrada en la Fig. 19 es que la tabla de control especifica el número de bits de desplazamiento de B_1 = 8 x 2 - 1 = 15 (B_1 = 8 x J_1 - 1, donde J_1 es un número entero mayor de 1) y el número de bits de desplazamiento B_2 = 8 x 2 + 1 = 17 (B_2 = 8 x J_2 + 1, donde J_2 es un número entero mayor de 0, J_1 = J_2 o $J_1 \neq J2$). Establecer la cantidad de desplazamiento hasta un ± 1 del múltiplo integral de ocho es para llevar a cabo el desplazamiento por bits impares, lo que mejora la seguridad en comparación con la realización de desplazamiento sólo por números pares, dado que la operación de la clave de extensión de la unidad de procesamiento de datos se lleva a cabo por una unidad de 8 bits, esto es, una unidad de bits pares. Y dado que la cantidad de desplazamiento puede establecerse mediante la adición/sustracción de un bit a / del múltiplo de ocho, por ejemplo, en alguna CPU que tenga un comando de desplazamiento de 1 solo bit, la operación de desplazamiento del tipo expuesto lleva a cabo un procesamiento de alta velocidad en comparación con el desplazamiento en 3 bits o 5 bits. Y así mismo, en el caso de que esta operación de desplazamiento que utiliza el hardware que puede desplazar solamente 1 bit, hay casos en el que es posible llevar a cabo un procesamiento de alta velocidad.

En la descripción anterior de la unidad 310 de longitud de bits, tres tipos de anchuras de bits de datos con clave son introducidos. Incluso cuando los datos con clave con una longitud de Q bits, en los cuales Q oscila entre 128 bits (G bits) y 256 bits (2G bits) (G < Q < 2G), la unidad de transformación 310 de longitud de los bits puede extender los datos con clave al mismo tamaño de los datos con clave cuando son introducidos los datos con clave de 256 bits, utilizando algún tipo de algoritmo. Es decir, cuando son introducidos los datos con clave con una longitud de Q, la cual oscila entre G bits y 2G bits, la unidad de transformación 310 de longitud de bits puede convertir los datos con clave de Q bits en datos con clave de 2G bits.

A continuación, se expondrá la prueba de la no existencia de una clave equivalente con referencia a la Fig. 34.

En la explicación que sigue de la Fig. 34, "+" indica una operación XOR. Aquí, se supone que se introducen dos datos con clave de 128 bits SK1 y SK2 (SK1 ≠ SK2) y que la unidad de transformación 310 de la longitud de los bits emite de salida $SK1_{altos}$ = SK1 = (SKH1 | SKL1) a partir de SK1 y $SK2_{altos}$ = SK2 = (SKH2 | SKL2) a partir de SK2. Aquí, SKHi (i = 1,2) significa los datos superiores de 64 bits de SKi y SKLi (i = 1,2) significa los datos inferiores de 64 bits de SKi.

Suponiendo que los datos XOR de SKH1 y SKH2 son ΔA y que los datos XOR de SKL1 y SKL2 son ΔB, puede decirse "al menos ΔA ≠ 0 o ΔB ≠ 0", dado que SK1 ≠ SK2.

Como se muestra en la Fig. 34, estos ΔA y ΔB se convierten en ΔA + ΔD, ΔB + ΔC, respectivamente, mediante la recepción de las dos rondas de transformaciones no lineales. Esto significa que los datos XOR (ΔA | ΔB) de $SK1_{altos}$ $SK2_{altos}$ se convierten en datos XOR (ΔA + ΔD | ΔB + ΔC) después de llevar a cabo las dos rondas de transformaciones no lineales en $SK1_{altos}$ y los datos transformados después de llevar a cabo las dos rondas de transformaciones no lineales en $SK2_{altos}$. De acuerdo con ello, cuando estos elementos de datos después de llevar a cabo las dos rondas de transformaciones no lineales son operados con la función XOR con $SK1_{altos}$ y $SK2_{altos}$, respectivamente, en un circuito

XOR 999, los resultados operados con la función XOR de dos elementos de datos se convierten en ($\Delta D \mid \Delta C$). Si la transformación no lineal es una función biyectiva, la introducción de $\Delta X \neq 0$ determina siempre la emisión de salida de $\Delta Y \neq 0$, de forma que, cuando "al menos $\Delta A \neq 0$ o $\Delta B \neq 0$", puede decirse que "al menos $\Delta C \neq 0$ o que $\Delta D \neq 0$". Por consiguiente, dado que es imposible emitir de salida los mismos datos a partir de $SK1_{altos}$ y $SK2_{altos}$ mediante las dos rondas de transformaciones no lineales, se demuestra la no existencia de la clave equivalente.

Por otro lado, como se muestra en la Fig. 35, se tomará en consideración otro caso, en el cual se llevan a cabo tres rondas de transformaciones no lineales en lugar de dos rondas de transformaciones no lineales. Dado que puede decirse que "al menos $\Delta A \neq 0$ o $\Delta B \neq 0$", puede haber un caso en el que o bien ΔA o ΔB puedan ser 0. Si $\Delta A = 0$, $\Delta C = 0$, y de la misma manera expuesta anteriormente, los datos XOR ($0 \mid \Delta B$) de $SK1_{altos}$ y $SK2_{altos}$ se convierten en los datos XOR ($\Delta B + \Delta E \mid \Delta D$) después de llevar a cabo las tres rondas de transformaciones no lineales en $SK1_{altos}$ y los datos transformados después de llevar a cabo las tres rondas de transformaciones no lineales en $SK2_{altos}$. De acuerdo con ello, cuando estos elementos de datos después de recibir las tres rondas de transformaciones no lineales son operados con la función XOR con $SK1_{altos}$ y $SK2_{altos}$, respectivamente, en el circuito XOR 999, los resultados operados con la función XOR de dos elementos de datos se convierten en ($\Delta B + \Delta E \mid \Delta B + \Delta D$). Aquí, cuando se supone que $\Delta B = \Delta D = \Delta E \neq 0$, lo siguiente es cierto: ($\Delta B + \Delta E \mid \Delta B + \Delta D$) = ($0 \mid 0$). Esto es, cuando estos elementos de datos después de llevar a cabo las tres rondas de transformaciones no lineales son operados con la función XOR con $SK1_{altos}$ y $SK2_{altos}$, respectivamente, los resultados de la operación son los mismos. Es decir $SK1_{altos}$ y $SK2_{altos}$ emiten de salida los mismos datos, de forma que existen las claves equivalentes, lo que es preocupante con respecto a la seguridad.

No solo el caso expuesto de transformación lineal de tres rondas, sino que una transformación no lineal general puede emitir de salida el equivalente K_1 a partir de SK1 y SK2 diferentes, lo que significa que puede existir una clave equivalente. Sin embargo, es posible demostrar la no existencia de la clave equivalente cuando se emplea la transformación lineal de dos rondas de acuerdo con la presente forma de realización.

Así mismo, puede haber otro caso en el que la no existencia de la clave equivalente se demuestre distinto del de la transformación no lineal de dos rondas de acuerdo con la presente forma de realización, sin embargo, es preferente utilizar la transformación no lineal de dos rondas debido a su configuración sencilla además de la demostrada no existencia de la clave equivalente.

La Fig. 24 muestra una computadora para la instalación de la unidad de transformación de datos para la encriptación 100 o la unidad de transformación de datos para la desencriptado 400.

La unidad de transformación de datos para la encriptación 100 y/o la unidad de transformación de datos para la desencriptado 400 están conectadas al bus como tarjeta de circuito impreso. Esta tarjeta de circuito impreso está provista de una CPU, una memoria, y un elemento de circuito lógico, y encripta los textos no cifrados suministrados a partir de la CPU convirtiéndolos en textos cifrados utilizando la operación anteriormente referida, y devuelve los datos a la CPU. O descripta los textos cifrados suministrados a partir de la CPU y devuelve los textos no cifrados a la CPU.

De esta forma, la unidad de transformación de datos para la encriptación 100 o la unidad de transformación de datos para la desencriptado 400 pueden ser implementadas por hardware. Así mismo, la unidad de transformación de datos para la encriptación 100 o la unidad de transformación de datos para la desencriptado 400 pueden ser también implementadas mediante software como procedimiento de transformación de datos. Es decir, la operación expuesta puede llevarse a cabo utilizando el programa almacenado en una unidad de disco magnético o en una unidad de disco flexible. Como una alternativa, la operación expuesta puede ser implementada mediante la combinación de hardware y software, aunque esto no se muestra en la figura. Así mismo, no se requiere implementar toda la operación referida utilizando una computadora, sino que es posible implementar la operación referida mediante un sistema distribuido, como por ejemplo un servidor y un cliente, o una computadora central y una computadora terminal, aunque esto no se muestra en la figura.

En las Figs. ilustradas 1 a 47, una flecha muestra una dirección del flujo operativo, y las figuras que tienen la flecha son diagramas de bloque de la unidad de transformación de datos y diagramas de flujo. "... unidad" mostrado en el diagrama de bloques expuestos puede ser sustituido por "... etapa" o "... proceso", de forma que los diagramas pueden ser considerados como diagramas de flujo operativos o diagramas de flujo de programa que muestran el procedimiento de transformación de datos.

En las formas de realización anteriores, se ha expuesto un caso en el que se utilizan textos no cifrados y textos cifrados de 128 bits, pero los datos pueden ser textos no cifrados y textos cifrados de 256 bits, o textos no cifrados y textos cifrados con otro número de bits.

Así mismo, en las formas de realización anteriores, se ha expuesto un caso en el que se han utilizado unos datos con clave de 128 bits, 192 bits, 256 bits y unas claves de extensión de 64 bits, pero los datos con clave pueden tener otro número de bits.

Si la longitud de bits de los textos no cifrados y de los textos cifrados, los datos con clave y la clave de extensión se modifican, por supuesto, la longitud de bits que va a ser procesada por cada unidad, cada etapa, o cada proceso, se modifica de acuerdo con la longitud de los bits.

Aplicabilidad industrial

Además, de acuerdo con una realización de la presente invención, la unidad de transformación de subcampo 18 se usa, lo que hace que la configuración sea más sencilla, y se proporcionan la unidad de transformación lineal 85 y la unidad de transformación lineal 87 de forma que se mejora incluso la seguridad si se usa la unidad de transformación de subcampo 18.

REIVINDICACIONES

1. Un aparato de transformación de datos que tiene una unidad (200, 500) de procesado de datos para realizar al menos una de la encriptación y desencriptación de datos basados en la introducción de datos clave a esto en la que la unidad de procesado de datos (200, 500) comprende:

una unidad de transformación de subcampos (18) para transformar los datos incluidos con la finalidad de ser transformados por un circuito de elementos inverso que utiliza un subcampo del campo mediante el tratamiento de los datos incluidos como un elemento en el campo; y la salida de datos transformados; y

una unidad de transformación afín (85) que opera en el vector de espacio GF $(2)^n$ en GF (2), para los datos transformados en GF (2^n) sean transformados incluidos así en los datos en GF $(2)^n$ mediante el tratamiento de datos en GF (2^n) como un elemento de GF $(2)^n$, la unidad de transformación afín (85) que está conectada en serie a la unidad de transformación de subcampos (18).

2. El aparato de transformación de datos de la reivindicación 1, en el que la unidad de transformación de subcampo incluye solamente unidades de operación de N/2 bits plurales para dividir por igual X datos que tengan N (: número par) de bits dentro de los X_1 datos de 2/n bits superiores y X_0 datos de N/2 bits inferiores de forma que $X = X_0 + \beta X_1$, (X_0, X_1: elementos del subcampo, β: un elemento del campo), y obtener datos Y respectivamente por medio de la operación de los datos Y_1 superiores de N/2 de bits y los datos Y_0 inferiores de N/2 de bits, Y_0 será $Y = Y_0 \beta Y_1 = 1/(X_0 + \beta X_1)$ (donde Y=0 cuando X=0).

3. Un procedimiento de transformación de datos para ejecutar un proceso de procesado de datos para realizar al menos un encriptado de datos y un desencriptado de datos, en el que el proceso de procesado de datos comprende:

Un proceso de transformación de subcampos para que la introducción de datos sean transformados incluidos así por un circuito de elemento inverso que utiliza un subcampos del campo mediante el tratamiento de la introducción de datos como un elemento del campo; y la salida de los datos transformados; y

Un proceso de transformación afín que opera en el vector de espacio GF $(2)^n$ en GF (2), para los datos transformados en GF (2^n) sean transformados incluidos así en los datos en GF $(2)^n$ mediante el tratamiento de datos en GF (2^n) como un elemento de GF $(2)^n$, el proceso de transformación afín que está conectado en series al proceso de transformación del subcampos.

4. Un programa que está constituido por una programación de medios de códigos adaptados para llevar a cabo los pasos del procedimiento de la reivindicación 3 cuando dicho programa continúe en un ordenador.

5. Un medio informático portador legible que conlleve el programa informático de realización de la reivindicación 4.

Fig. 1

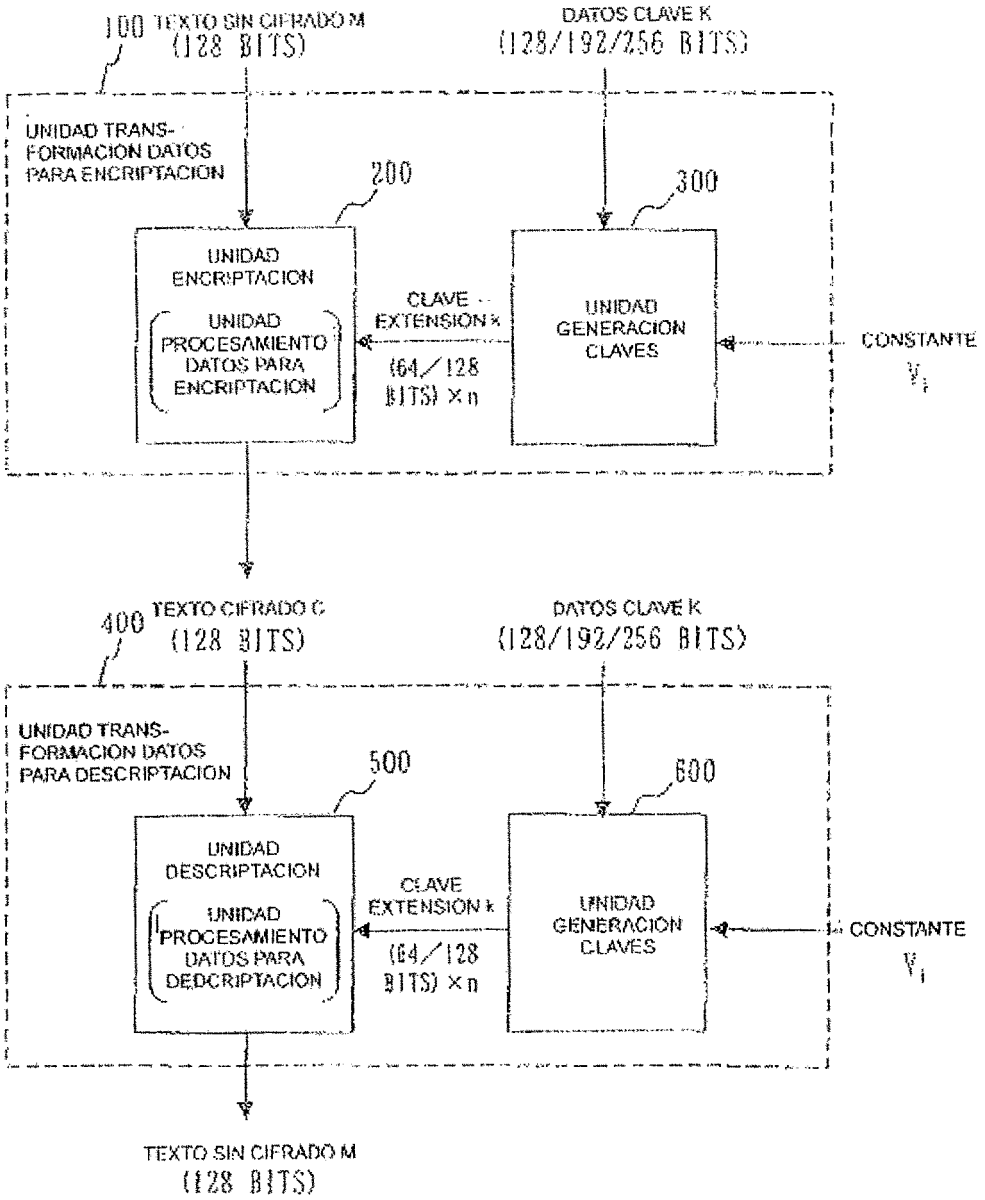

100
TEXTO SIN CIFRADO M
(128 BITS)
DATOS CLAVE K
(128/192/256 BITS)
UNIDAD TRANS-FORMACION DATOS PARA ENCRIPTACION
200
300
UNIDAD ENCRIPTACION
UNIDAD PROCESAMIENTO DATOS PARA ENCRIPTACION
CLAVE EXTENSION k
(64/128 BITS) x n
UNIDAD GENERACION CLAVES
CONSTANTE
V_i
400
TEXTO CIFRADO C
(128 BITS)
DATOS CLAVE K
(128/192/256 BITS)
UNIDAD TRANS-FORMACION DATOS PARA DESCRIPTACION
500
600
UNIDAD DESCRIPTACION
UNIDAD PROCESAMIENTO DATOS PARA DECRIPTACION
CLAVE EXTENSION k
(64/128 BITS) x n
UNIDAD GENERACION CLAVES
CONSTANTE
V_i
TEXTO SIN CIFRADO M
(128 BITS)

Fig. 2

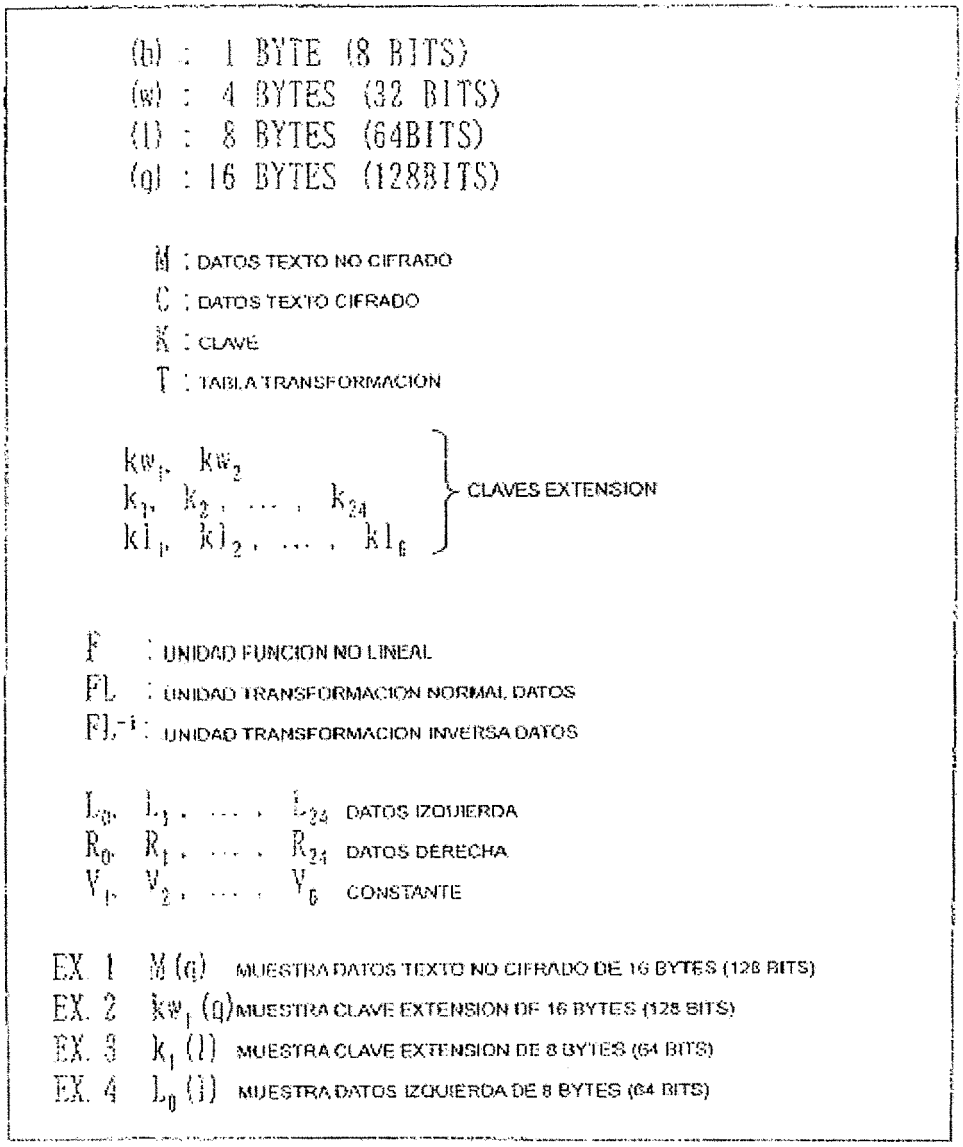

(b) : 1 BYTE (8 BITS)
(w) : 4 BYTES (32 BITS)
(l) : 8 BYTES (64BITS)
(q) : 16 BYTES (128BITS)
M : DATOS TEXTO NO CIFRADO
C : DATOS TEXTO CIFRADO
K : CLAVE
T : TABLA TRANSFORMACION
kw1, kw2
k1, k2, ... , k24
kl1, kl2, ... , kl6
CLAVES EXTENSION
F : UNIDAD FUNCION NO LINEAL
FL : UNIDAD TRANSFORMACION NORMAL DATOS
FL-1 : UNIDAD TRANSFORMACION INVERSA DATOS
L0, L1, ... , L24 DATOS IZQUIERDA
R0, R1, ... , R24 DATOS DERECHA
V1, V2, ... , V6 CONSTANTE
EX. 1 M(q) MUESTRA DATOS TEXTO NO CIFRADO DE 16 BYTES (128 BITS)
EX. 2 kw1(q) MUESTRA CLAVE EXTENSION DE 16 BYTES (128 BITS)
EX. 3 k1(l) MUESTRA CLAVE EXTENSION DE 8 BYTES (64 BITS)
EX. 4 L0(l) MUESTRA DATOS IZQUIERDA DE 8 BYTES (64 BITS)

Fig. 3

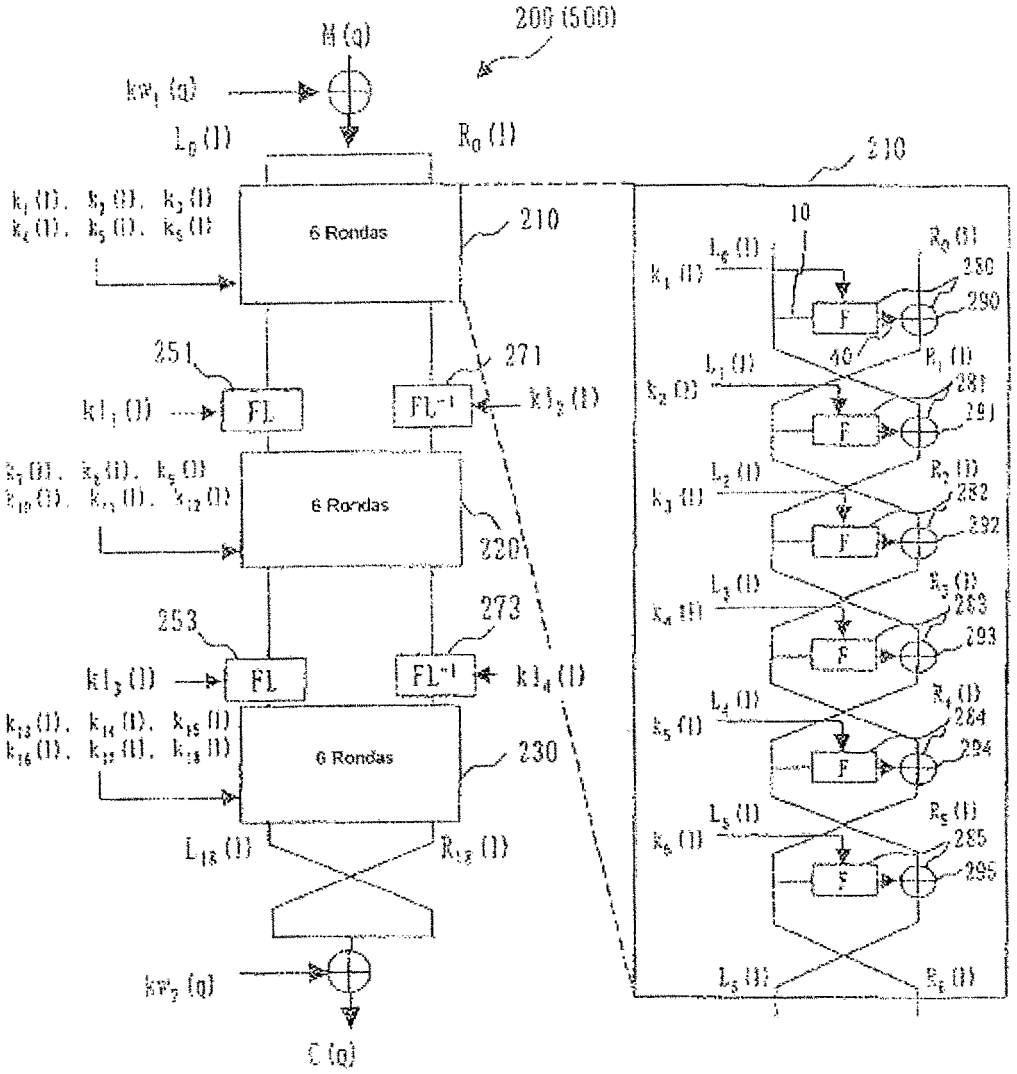

M (q)
200 (500)
kw1 (q)
L0 (1)
R0 (1)
k1 (1), k2 (1), k3 (1)
k4 (1), k5 (1), k6 (1)
6 Rondas
210
251
271
kl1 (1)
FL
FL-1
kl2 (1)
k7 (1), k8 (1), k9 (1)
k10 (1), k11 (1), k12 (1)
6 Rondas
220
253
273
kl3 (1)
FL
FL-1
kl4 (1)
k13 (1), k14 (1), k15 (1)
k16 (1), k17 (1), k18 (1)
6 Rondas
230
L18 (1)
R18 (1)
kw2 (q)
C (q)
210
10
L0 (1)
R0 (1)
k1 (1)
250
290
F
40
L1 (1)
R1 (1)
k2 (1)
281
291
L2 (1)
R2 (1)
k3 (1)
282
292
L3 (1)
R3 (1)
k4 (1)
283
293
L4 (1)
R4 (1)
k5 (1)
284
294
L5 (1)
R5 (1)
k6 (1)
285
295
L6 (1)
R6 (1)

Fig. 4

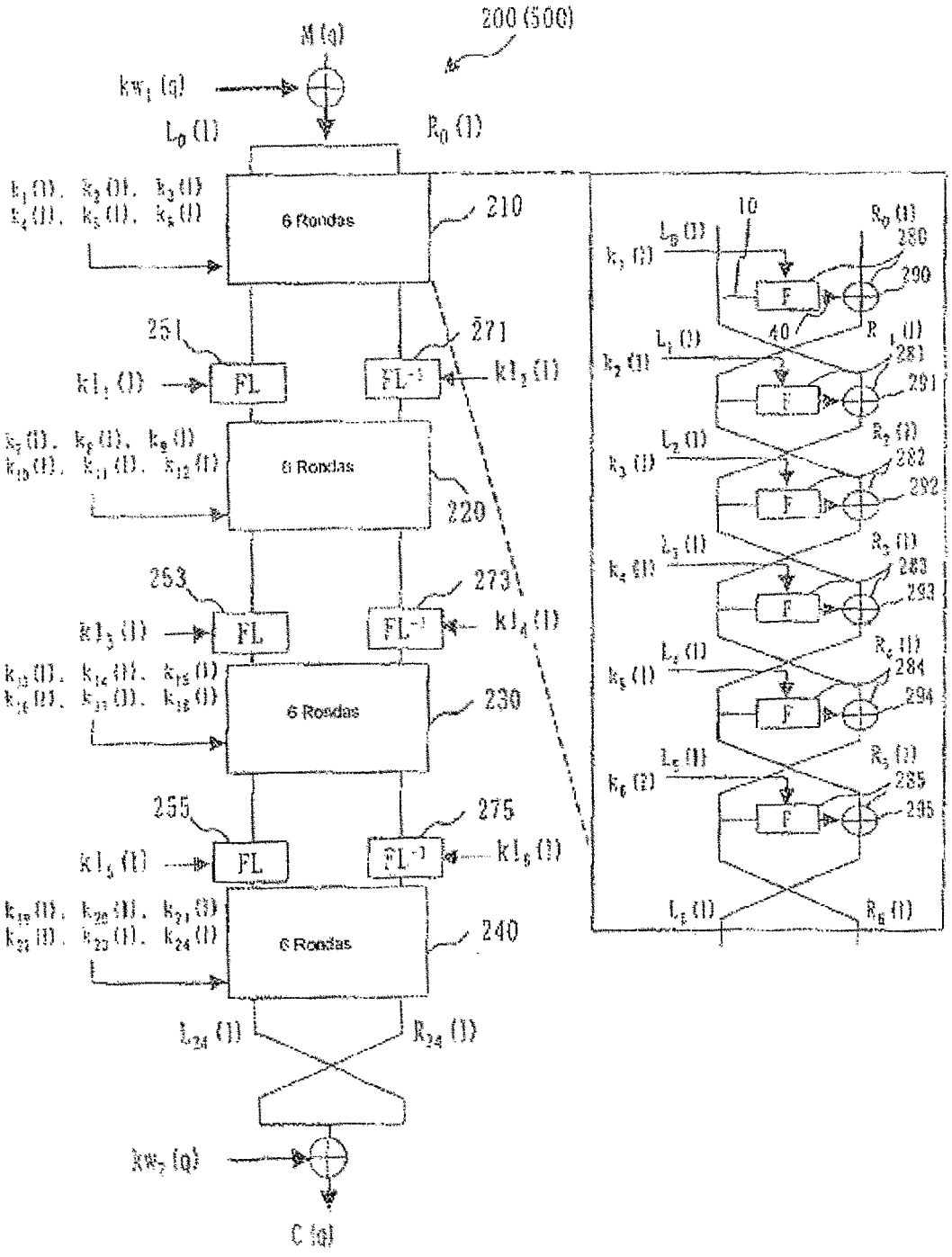

M (q)
200 (500)
kw1 (q)
L0 (1)
R0 (1)
6 Rondas
210
FL
FL-1
251
271
6 Rondas
220
253
273
6 Rondas
230
255
275
6 Rondas
240
L24 (1)
R24 (1)
kw2 (q)
C (q)
10
40
F

Fig. 5

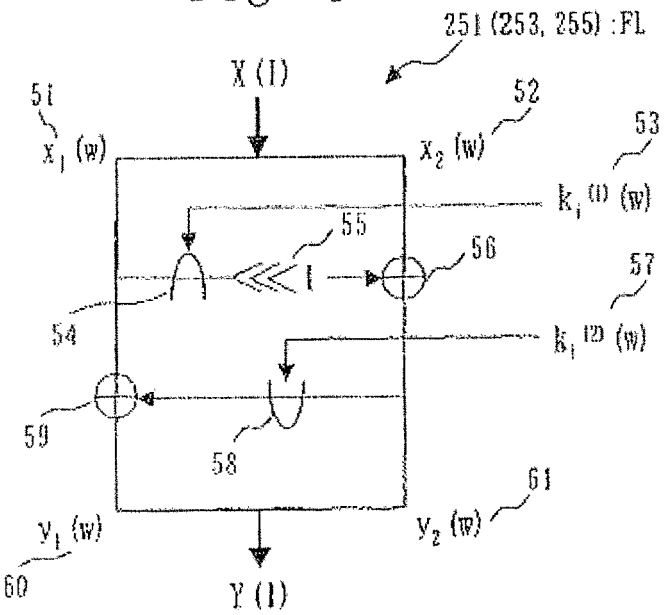

251 (253, 255) :FL
X (I)
51
x_1 (w)
52
x_2 (w)
53
$k_1^{(1)}$ (w)
55
56
57
54
$k_1^{(2)}$ (w)
59
58
61
y_1 (w)
y_2 (w)
60
Y (I)

Fig. 6

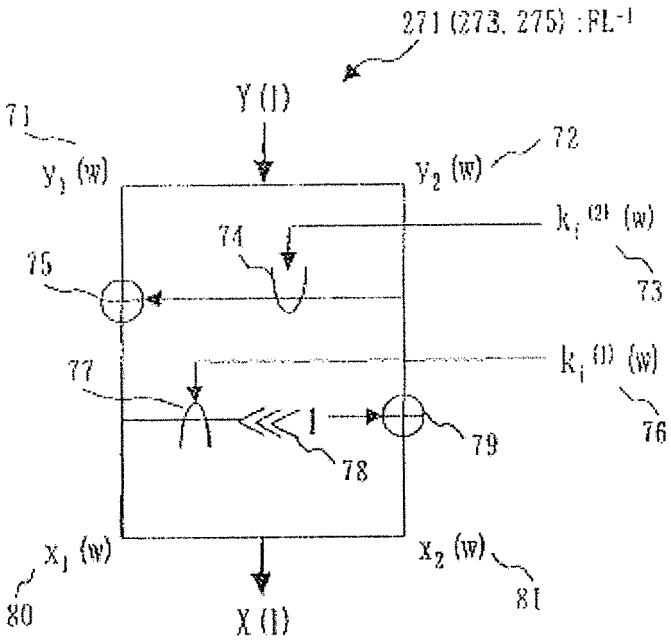

271 (273, 275) :FL-1
Y (1)
71
y_1 (w)
y_2 (w)
72
$k_i^{(2)}$ (w)
74
75
73
$k_i^{(1)}$ (w)
77
1
79
76
78
x_1 (w)
x_2 (w)
80
X (1)
81

Fig. 7

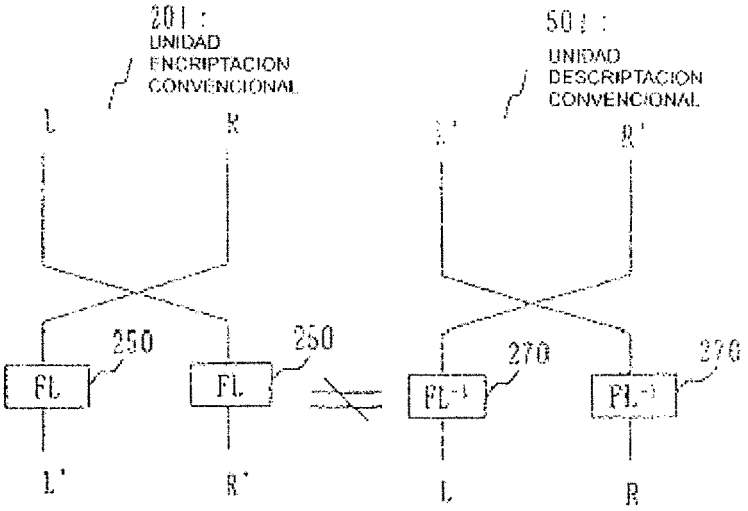

201 :
UNIDAD
ENCRIPTACION
CONVENCIONAL
501 :
UNIDAD
DESCRIPTACION
CONVENCIONAL
L
R
L'
R'
FL
250
270
L'
R'
L
R

Fig. 8

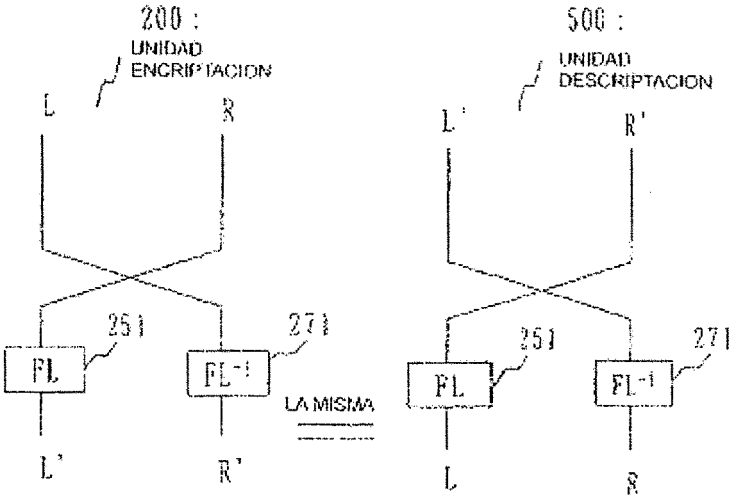

200 :
UNIDAD
ENCRIPTACION
L
R
251
271
FL
FL-1
LA MISMA
L'
R'
500 :
UNIDAD
DESCRIPTACION
L'
R'
251
271
FL
FL-1
L
R

Fig. 9

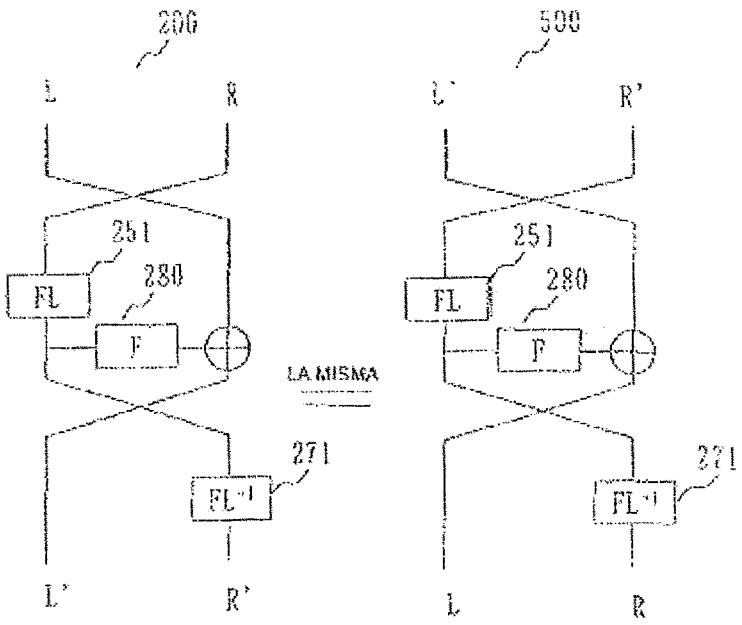


Fig. 10

Fig 3	FL251⇔FL^{-1} 273 FL253⇔FL^{-1} 271
Fig 4	FL251⇔FL^{-1} 275 FL253⇔FL^{-1} 273 FL255⇔FL^{-1} 271

Fig. 11

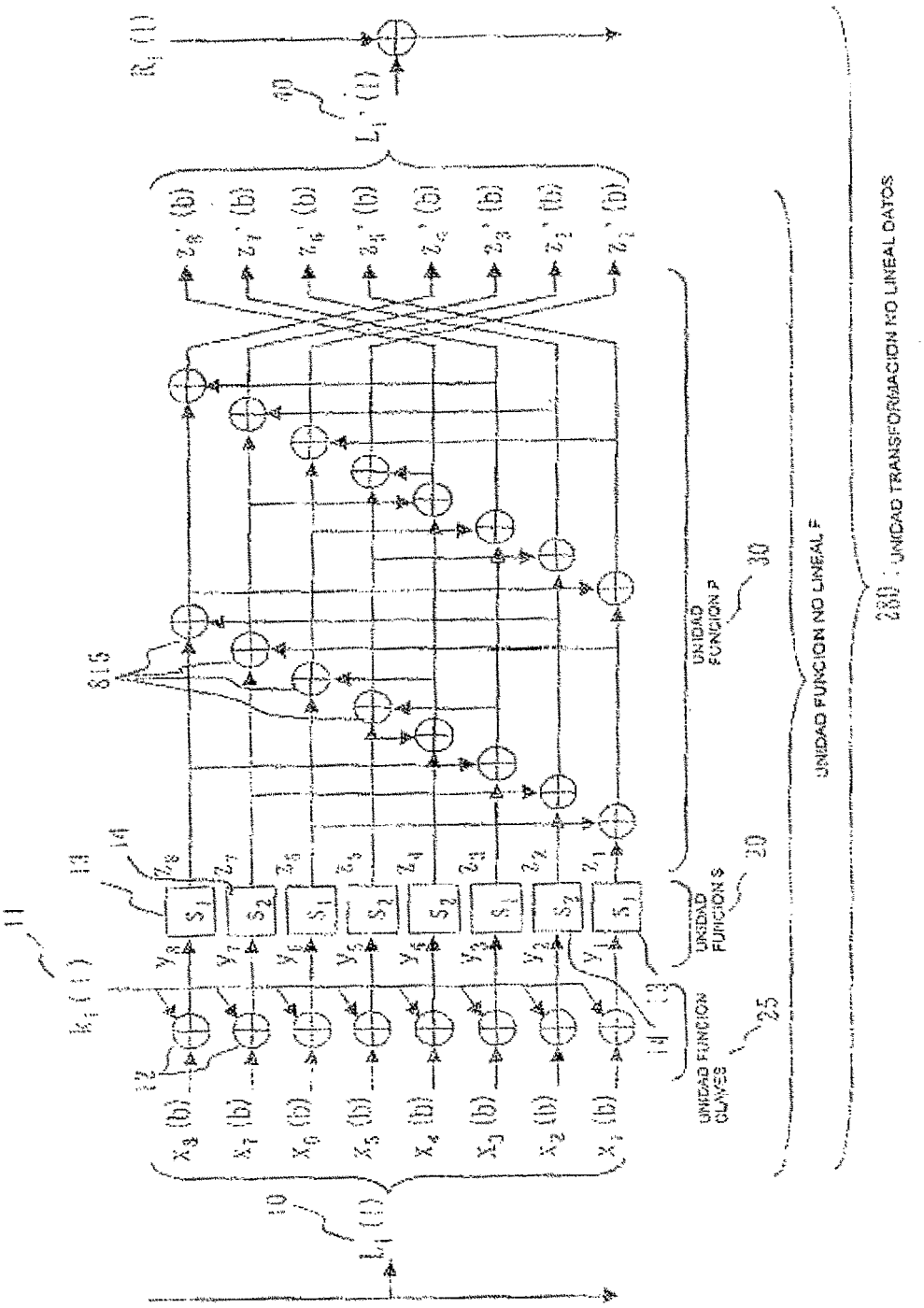

UNIDAD FUNCION S
UNIDAD FUNCION F
UNIDAD FUNCION CLAVES
UNIDAD FUNCION NO LINEAL F
UNIDAD TRANSFORMACION NO LINEAL DATOS

Fig.12

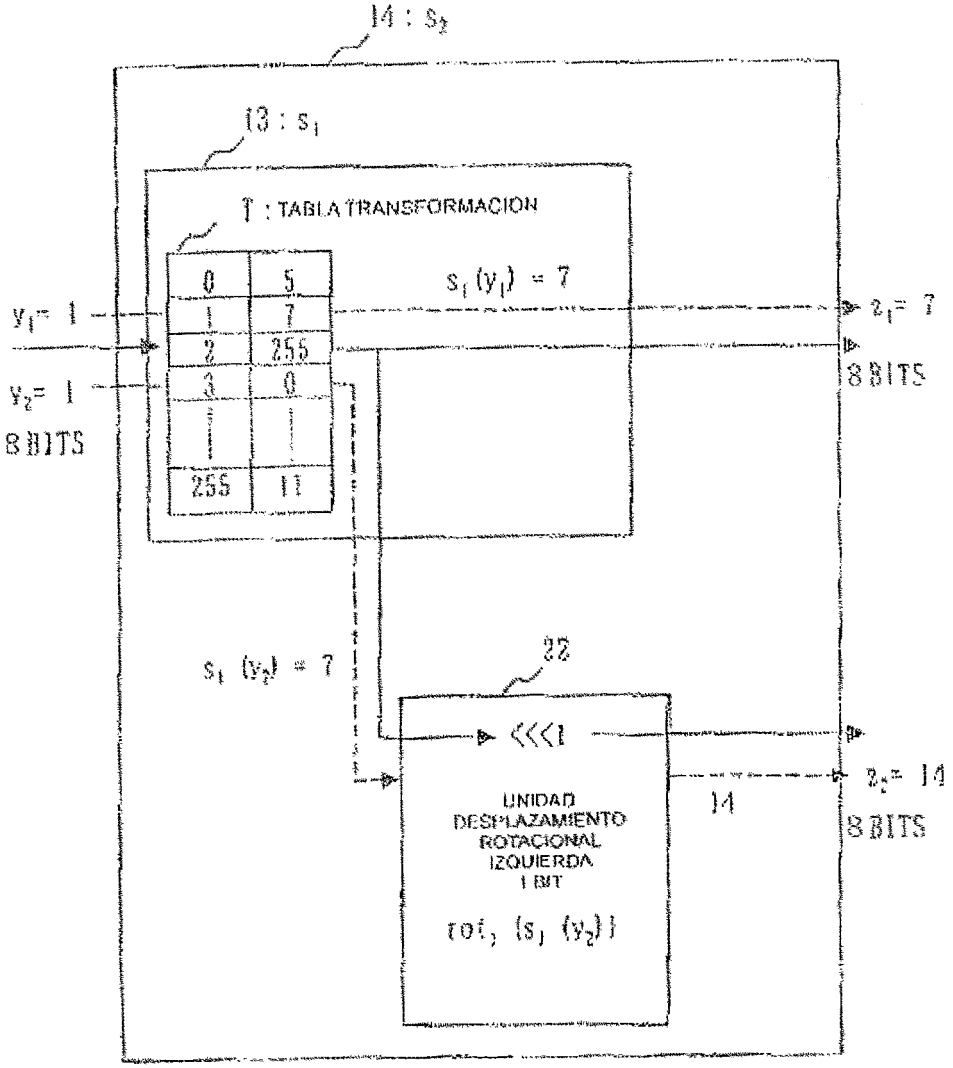

14 : s_2
13 : s_1
T : TABLA TRANSFORMACION
0 5
1 7
2 255
3 0
255 11
$s_1(y_1) = 7$
$y_1 = 1$
$y_2 = 1$
8 BITS
$z_1 = 7$
8 BITS
$s_1(y_2) = 7$
22
<<<1
UNIDAD DESPLAZAMIENTO ROTACIONAL IZQUIERDA 1 BIT
$rot_1(s_1(y_2))$
14
$z_2 = 14$
8 BITS

Fig. 13

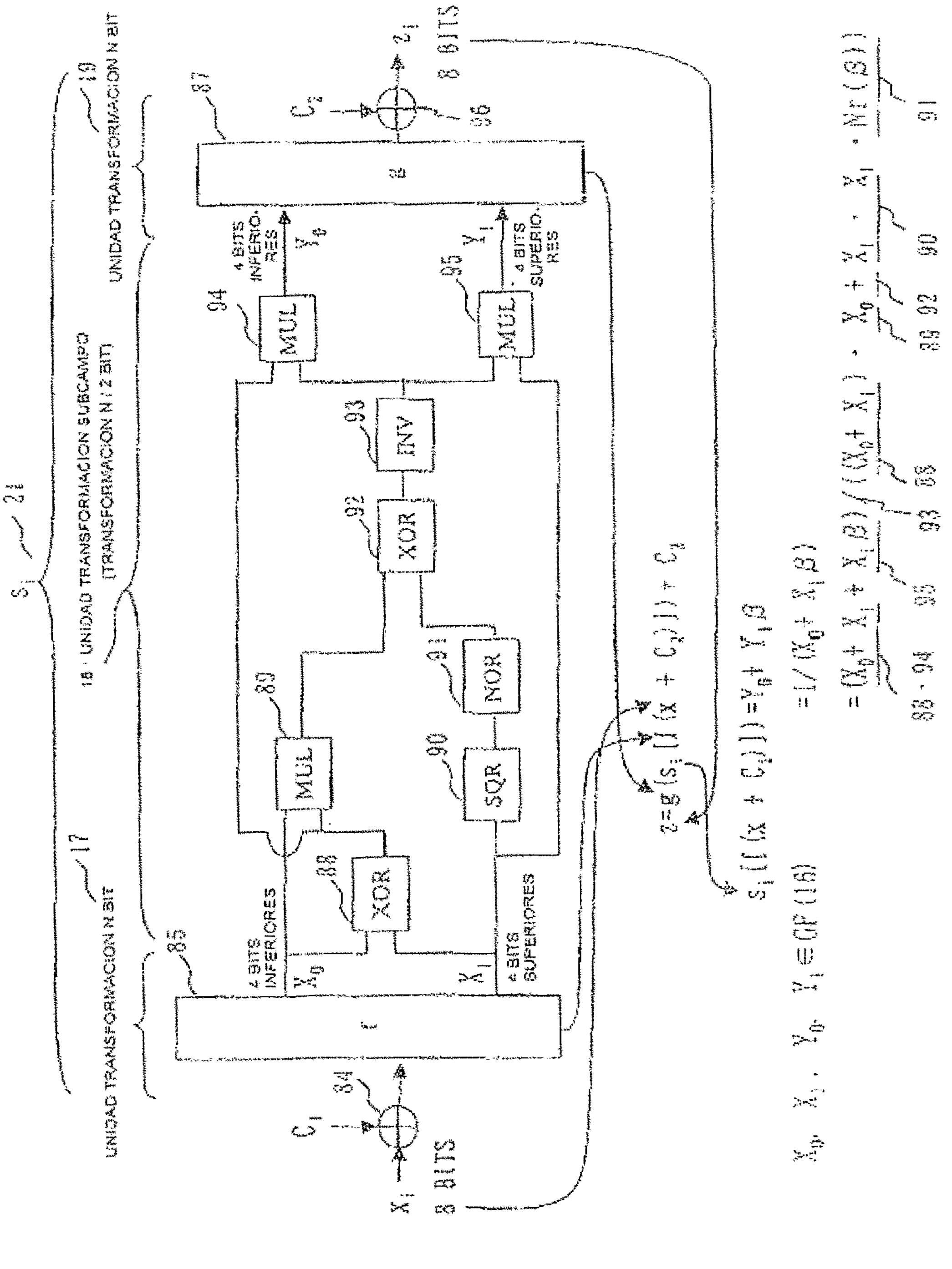

UNIDAD TRANSFORMACION N BIT
18 - UNIDAD TRANSFORMACION SUBCAMPO (TRANSFORMACION N / 2 BIT)
8 BITS
4 BITS INFERIORES
4 BITS SUPERIORES
4 BITS INFERIO-RES
4 BITS SUPERIO-RES
MUL
XOR
SQR
NOR
INV

Fig.14

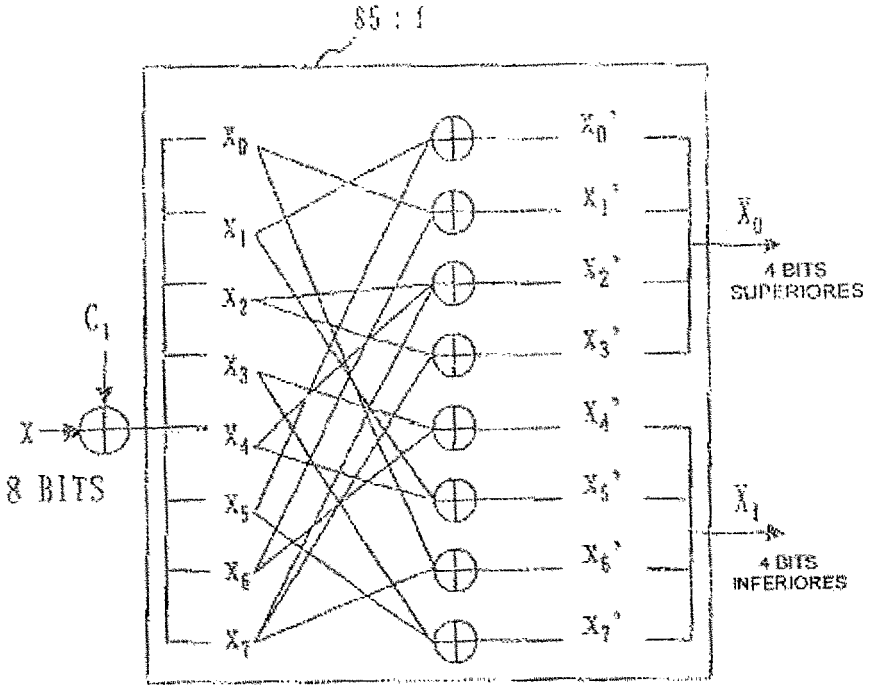


Fig.15

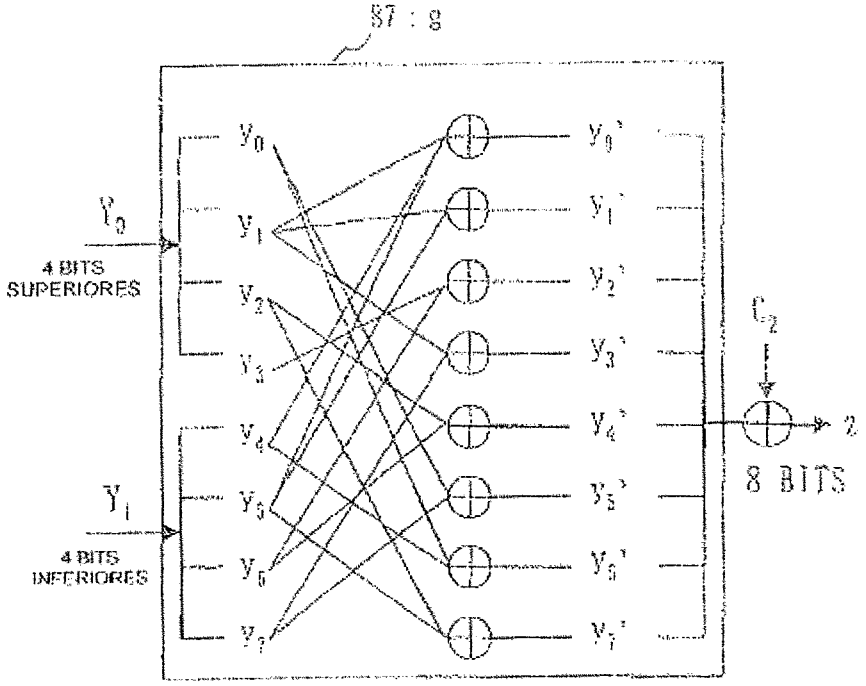

87 : 8
Y_0
4 BITS SUPERIORES
Y_1
4 BITS INFERIORES
y_0
y_1
y_2
y_3
y_4
y_5
y_6
y_7
y_0'
y_1'
y_2'
y_3'
y_4'
y_5'
y_6'
y_7'
C_2
2
8 BITS

Fig. 16

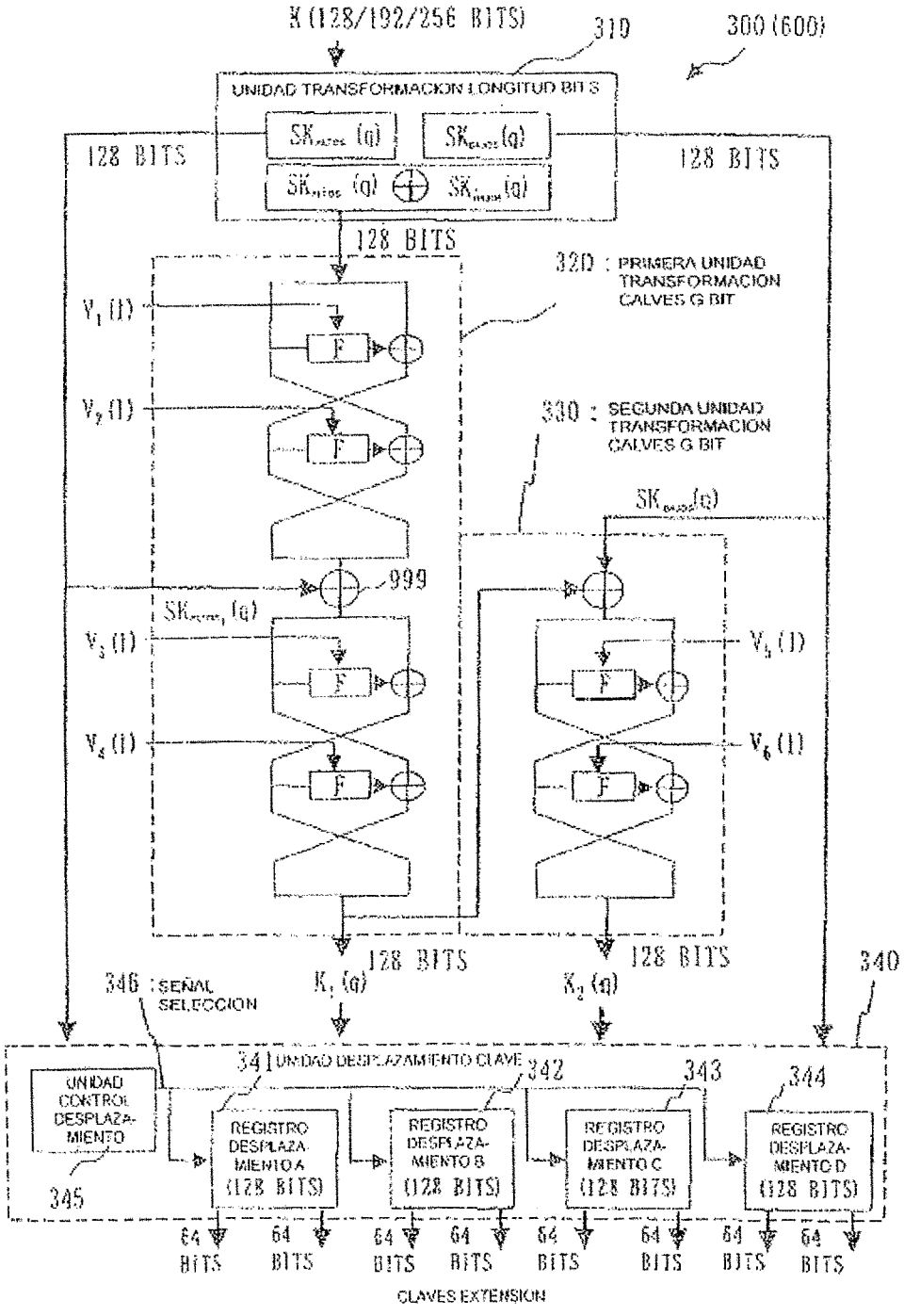

K (128/192/256 BITS)
310
300 (600)
UNIDAD TRANSFORMACION LONGITUD BITS
128 BITS
128 BITS
128 BITS
320 : PRIMERA UNIDAD TRANSFORMACION CALVES G BIT
330 : SEGUNDA UNIDAD TRANSFORMACION CALVES G BIT
999
128 BITS
128 BITS
340
346 : SEÑAL SELECCION
341 UNIDAD DESPLAZAMIENTO CLAVE
342
343
344
UNIDAD CONTROL DESPLAZA-MIENTO
345
REGISTRO DESPLAZA-MIENTO A (128 BITS)
REGISTRO DESPLAZA-MIENTO B (128 BITS)
REGISTRO DESPLAZA-MIENTO C (128 BITS)
REGISTRO DESPLAZA-MIENTO D (128 BITS)
64 BITS
CLAVES EXTENSION

Fig. 17

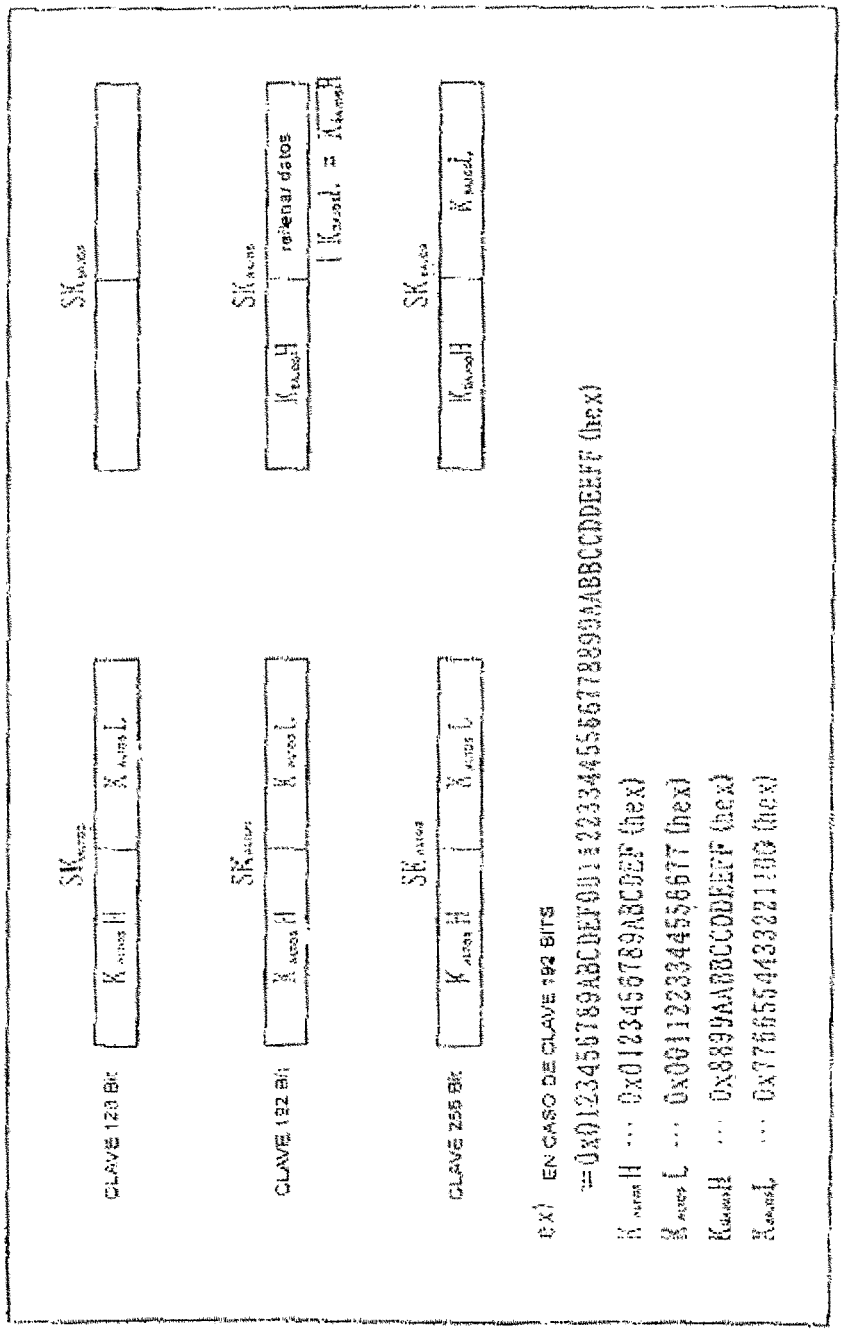

CLAVE 128 Bit
CLAVE 192 Bit
CLAVE 256 Bit
rellenar datos
EN CASO DE CLAVE 192 BITS
=0x0123456789ABCDEF0011223344556677899AABBCCDDEEFF (hex)
0x0123456789ABCDEF (hex)
0x0011223344556677 (hex)
0x8899AABBCCDDEEFF (hex)
0x7766554433221100 (hex)

Fig.18

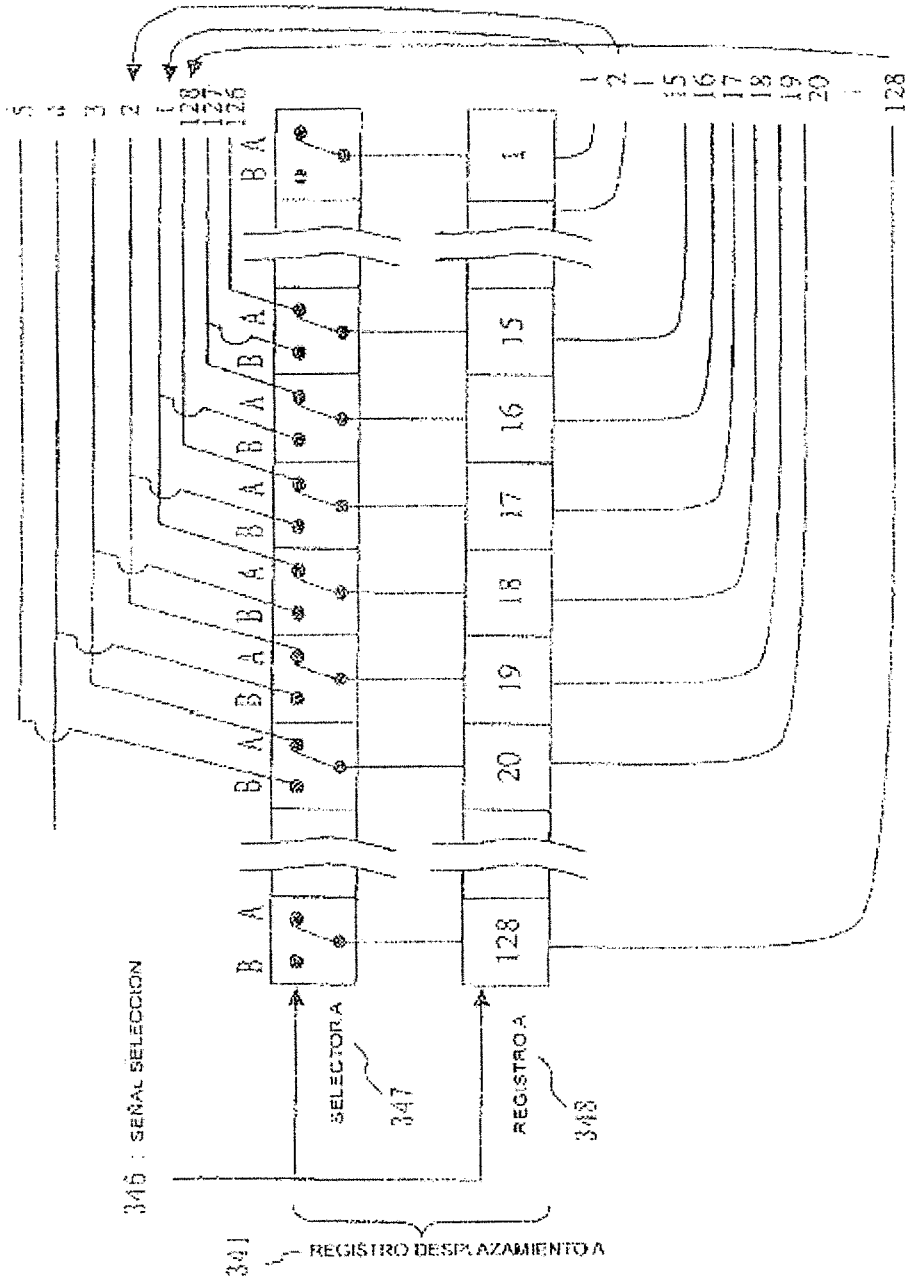

SEÑAL SELECCION
SELECTOR A
REGISTRO A
REGISTRO DESPLAZAMIENTO A

Fig.19

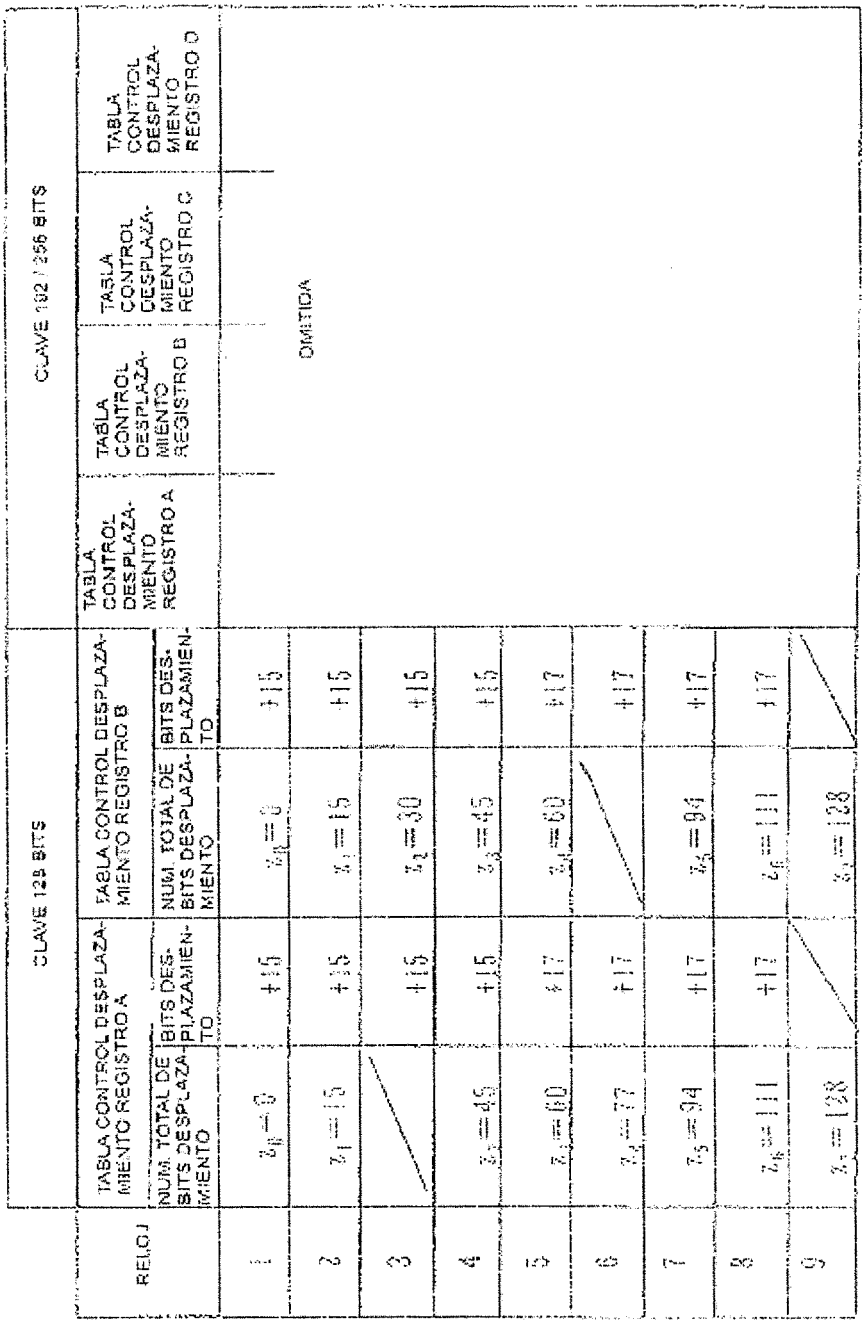

CLAVE 128 BITS
CLAVE 192 / 256 BITS
RELOJ
TABLA CONTROL DESPLAZAMIENTO REGISTRO A
NUM. TOTAL DE BITS DESPLAZAMIENTO
BITS DESPLAZAMIENTO
TABLA CONTROL DESPLAZAMIENTO REGISTRO B
NUM. TOTAL DE BITS DESPLAZAMIENTO
BITS DESPLAZAMIENTO
TABLA CONTROL DESPLAZAMIENTO REGISTRO A
TABLA CONTROL DESPLAZAMIENTO REGISTRO B
TABLA CONTROL DESPLAZAMIENTO REGISTRO C
TABLA CONTROL DESPLAZAMIENTO REGISTRO D
OMITIDA
1
2
3
4
5
6
7
8
9
+15
+17

Fig. 20

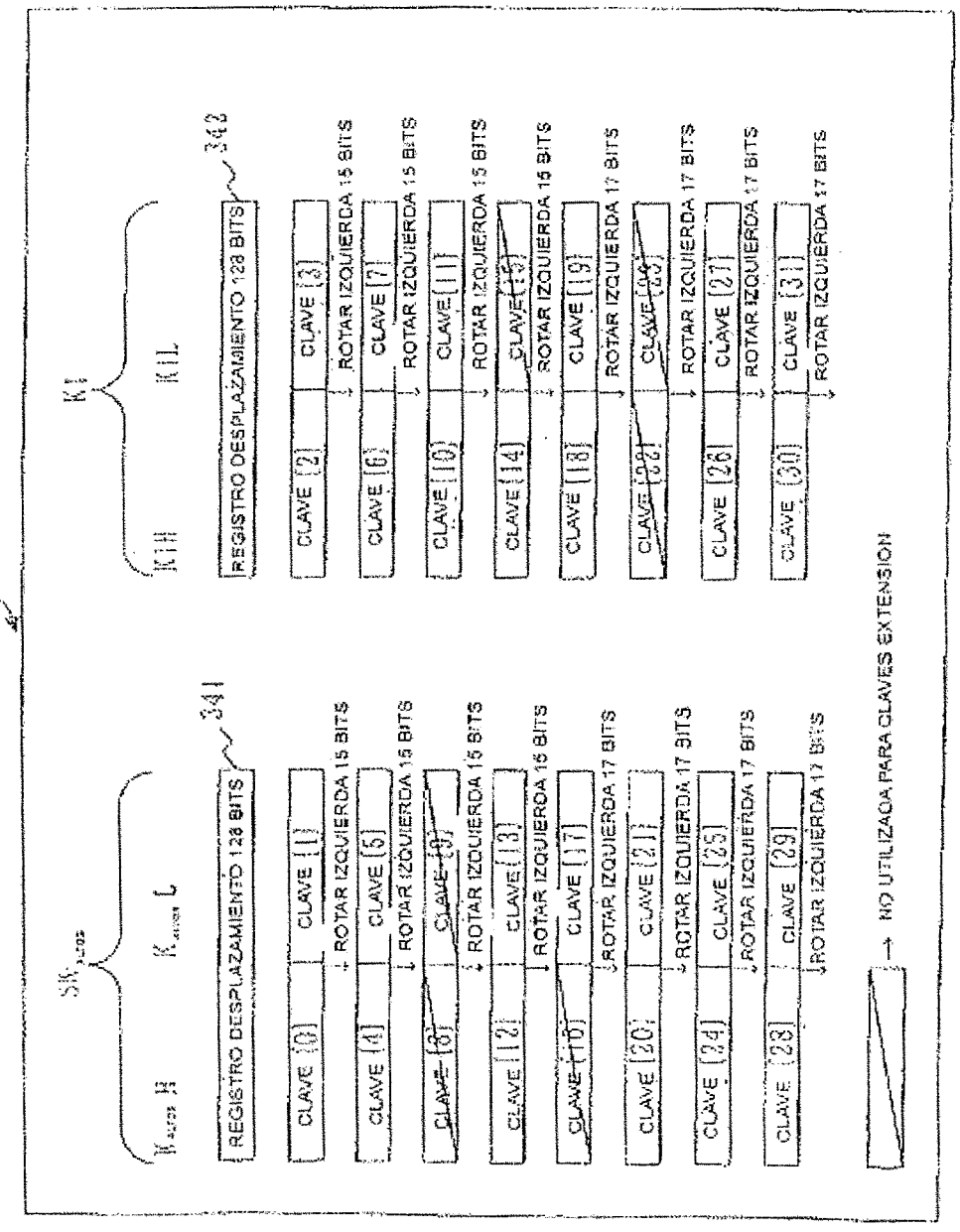

EN CASO DE CLAVE 128 BITS
REGISTRO DESPLAZAMIENTO 128 BITS
341
CLAVE [0]
CLAVE [1]
ROTAR IZQUIERDA 15 BITS
CLAVE [4]
CLAVE [5]
ROTAR IZQUIERDA 15 BITS
CLAVE [8]
CLAVE [9]
ROTAR IZQUIERDA 15 BITS
CLAVE [12]
CLAVE [13]
ROTAR IZQUIERDA 15 BITS
CLAVE [16]
CLAVE [17]
ROTAR IZQUIERDA 17 BITS
CLAVE [20]
CLAVE [21]
ROTAR IZQUIERDA 17 BITS
CLAVE [24]
CLAVE [25]
ROTAR IZQUIERDA 17 BITS
CLAVE [28]
CLAVE [29]
ROTAR IZQUIERDA 17 BITS
NO UTILIZADA PARA CLAVES EXTENSION
REGISTRO DESPLAZAMIENTO 128 BITS
342
CLAVE [2]
CLAVE [3]
ROTAR IZQUIERDA 15 BITS
CLAVE [6]
CLAVE [7]
ROTAR IZQUIERDA 15 BITS
CLAVE [10]
CLAVE [11]
ROTAR IZQUIERDA 15 BITS
CLAVE [14]
CLAVE [15]
ROTAR IZQUIERDA 15 BITS
CLAVE [18]
CLAVE [19]
ROTAR IZQUIERDA 17 BITS
CLAVE [22]
CLAVE [23]
ROTAR IZQUIERDA 17 BITS
CLAVE [26]
CLAVE [27]
ROTAR IZQUIERDA 17 BITS
CLAVE [30]
CLAVE [31]
ROTAR IZQUIERDA 17 BITS

Fig. 21

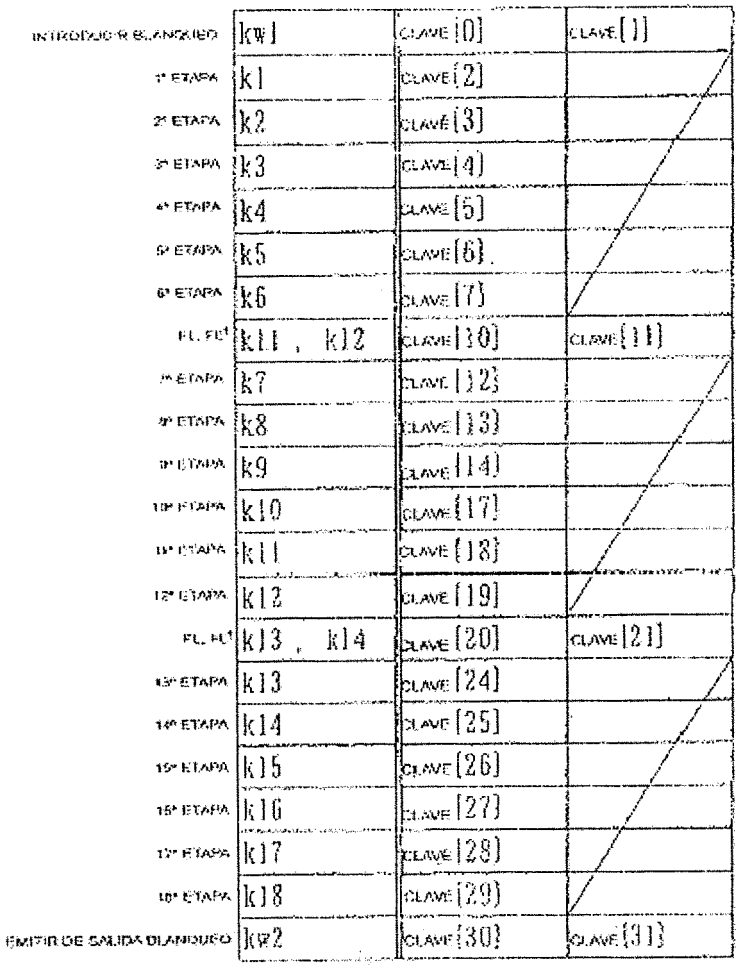

INTRODUCIR BLANQUEO
1ª ETAPA
2ª ETAPA
3ª ETAPA
4ª ETAPA
5ª ETAPA
6ª ETAPA
FL, FL'
7ª ETAPA
8ª ETAPA
9ª ETAPA
10ª ETAPA
11ª ETAPA
12ª ETAPA
FL, FL'
13ª ETAPA
14ª ETAPA
15ª ETAPA
16ª ETAPA
17ª ETAPA
18ª ETAPA
EMITIR DE SALIDA BLANQUEO
kw1
k1
k2
k3
k4
k5
k6
k11 , k12
k7
k8
k9
k10
k11
k12
k13 , k14
k13
k14
k15
k16
k17
k18
kw2
CLAVE [0]
CLAVE [2]
CLAVE [3]
CLAVE [4]
CLAVE [5]
CLAVE [6]
CLAVE [7]
CLAVE [10]
CLAVE [12]
CLAVE [13]
CLAVE [14]
CLAVE [17]
CLAVE [18]
CLAVE [19]
CLAVE [20]
CLAVE [24]
CLAVE [25]
CLAVE [26]
CLAVE [27]
CLAVE [28]
CLAVE [29]
CLAVE [30]
CLAVE [1]
CLAVE [11]
CLAVE [21]
CLAVE [31]

Fig. 22

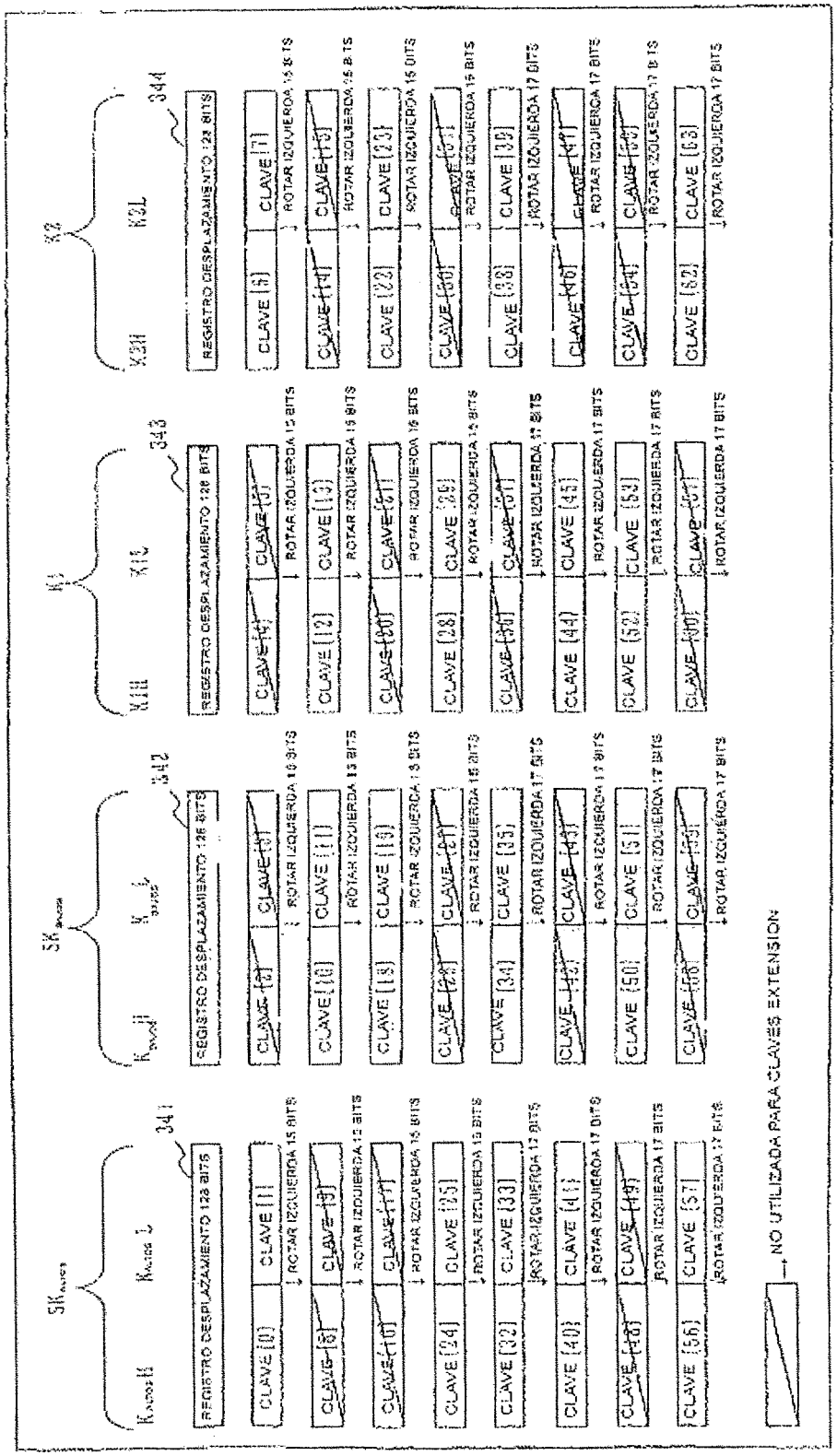

EN CASOS DE CLAVE DE 192 BITS Y 256 BITS
REGISTRO DESPLAZAMIENTO 128 BITS
CLAVE [0]
CLAVE [1]
ROTAR IZQUIERDA 15 BITS
CLAVE [32]
CLAVE [33]
ROTAR IZQUIERDA 17 BITS
CLAVE [64]
CLAVE [65]
ROTAR IZQUIERDA 15 BITS
CLAVE [96]
CLAVE [97]
ROTAR IZQUIERDA 17 BITS
CLAVE [128]
CLAVE [129]
ROTAR IZQUIERDA 15 BITS
CLAVE [160]
CLAVE [161]
ROTAR IZQUIERDA 17 BITS
CLAVE [192]
CLAVE [193]
ROTAR IZQUIERDA 15 BITS
CLAVE [224]
CLAVE [225]
ROTAR IZQUIERDA 17 BITS
NO UTILIZADA PARA CLAVES EXTENSION

Fig. 23

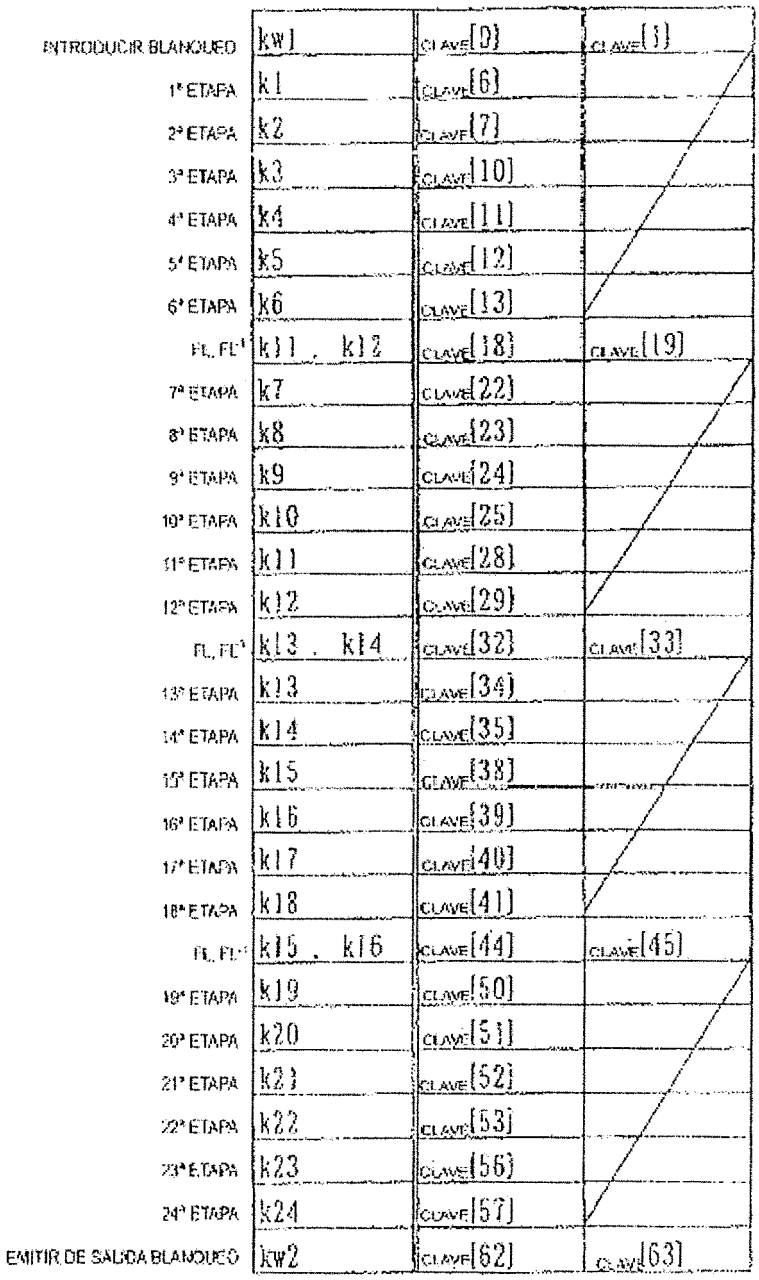

INTRODUCIR BLANQUEO
1ª ETAPA
2ª ETAPA
3ª ETAPA
4ª ETAPA
5ª ETAPA
6ª ETAPA
FL, FL'
7ª ETAPA
8ª ETAPA
9ª ETAPA
10ª ETAPA
11ª ETAPA
12ª ETAPA
FL, FL'
13ª ETAPA
14ª ETAPA
15ª ETAPA
16ª ETAPA
17ª ETAPA
18ª ETAPA
FL, FL'
19ª ETAPA
20ª ETAPA
21ª ETAPA
22ª ETAPA
23ª ETAPA
24ª ETAPA
EMITIR DE SALIDA BLANQUEO
kw1
k1
k2
k3
k4
k5
k6
k11 , k12
k7
k8
k9
k10
k11
k12
k13 , k14
k13
k14
k15
k16
k17
k18
k15 , k16
k19
k20
k21
k22
k23
k24
kw2
CLAVE[0]
CLAVE[6]
CLAVE[7]
CLAVE[10]
CLAVE[11]
CLAVE[12]
CLAVE[13]
CLAVE[18]
CLAVE[22]
CLAVE[23]
CLAVE[24]
CLAVE[25]
CLAVE[28]
CLAVE[29]
CLAVE[32]
CLAVE[34]
CLAVE[35]
CLAVE[38]
CLAVE[39]
CLAVE[40]
CLAVE[41]
CLAVE[44]
CLAVE[50]
CLAVE[51]
CLAVE[52]
CLAVE[53]
CLAVE[56]
CLAVE[57]
CLAVE[62]
CLAVE[1]
CLAVE[19]
CLAVE[33]
CLAVE[45]
CLAVE[63]

Fig. 24

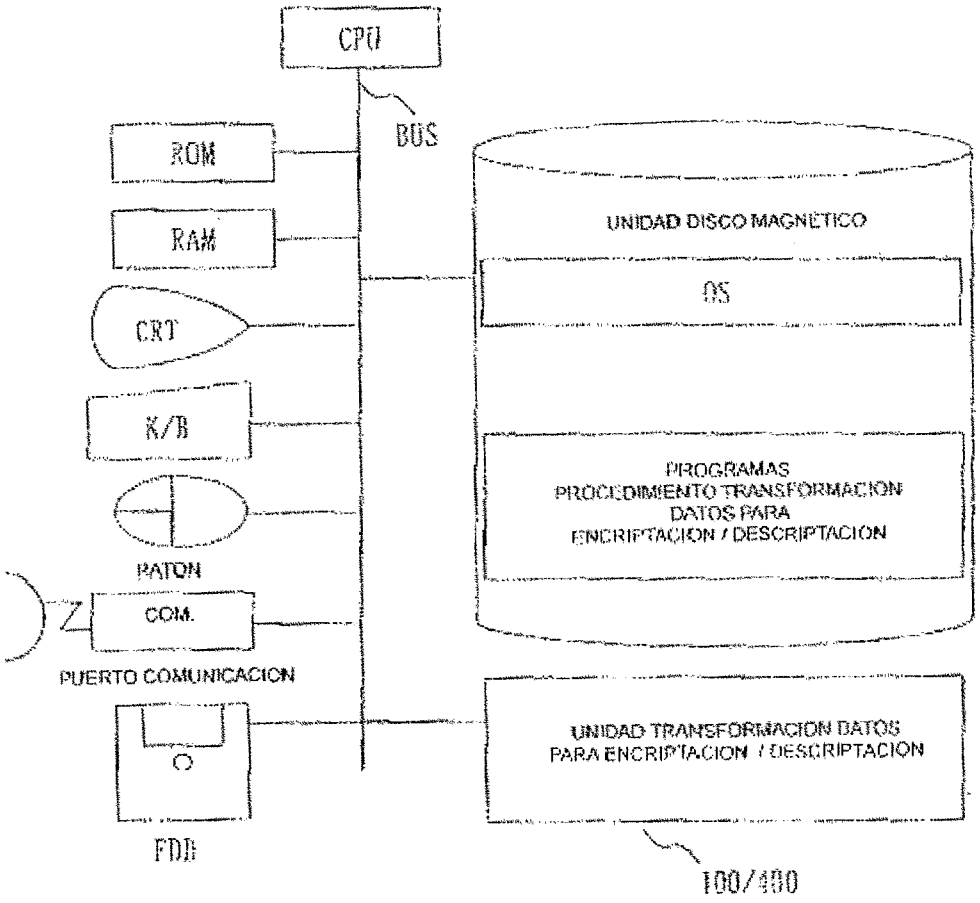

CPU
BUS
ROM
RAM
CRT
K/B
RATON
COM.
PUERTO COMUNICACION
FDD
UNIDAD DISCO MAGNETICO
OS
PROGRAMAS
PROCEDIMIENTO TRANSFORMACION
DATOS PARA
ENCRIPTACION / DESCRIPTACION
UNIDAD TRANSFORMACION DATOS
PARA ENCRIPTACION / DESCRIPTACION
100/400

Fig. 25

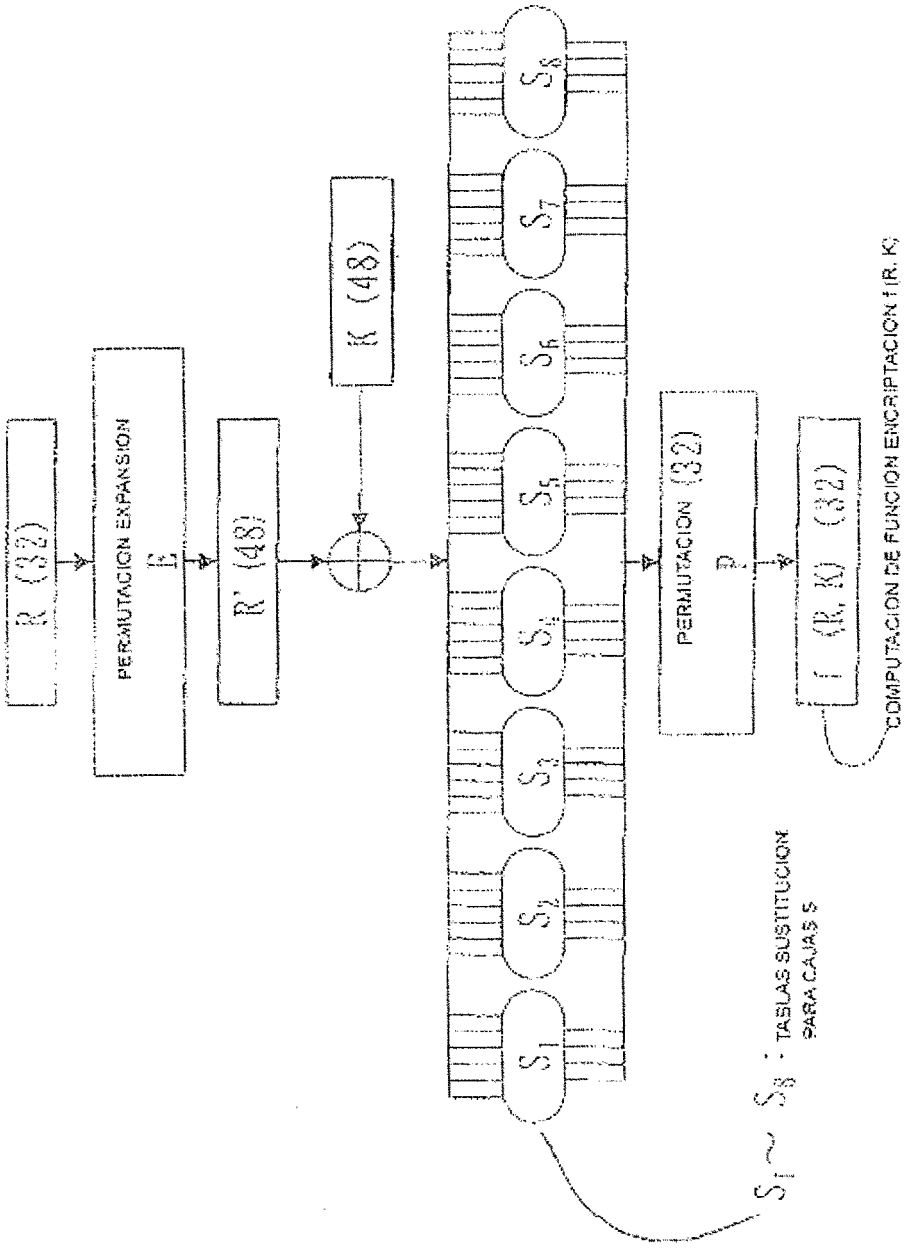

R (32)
PERMUTACION EXPANSION
E
R' (48)
K (48)
S1
S2
S3
S4
S5
S6
S7
S8
PERMUTACION (32)
P
f (R, K) (32)
COMPUTACION DE FUNCION ENCRIPTACION f (R, K)
S1 ~ S8 : TABLAS SUSTITUCION PARA CAJAS S

Fig. 26

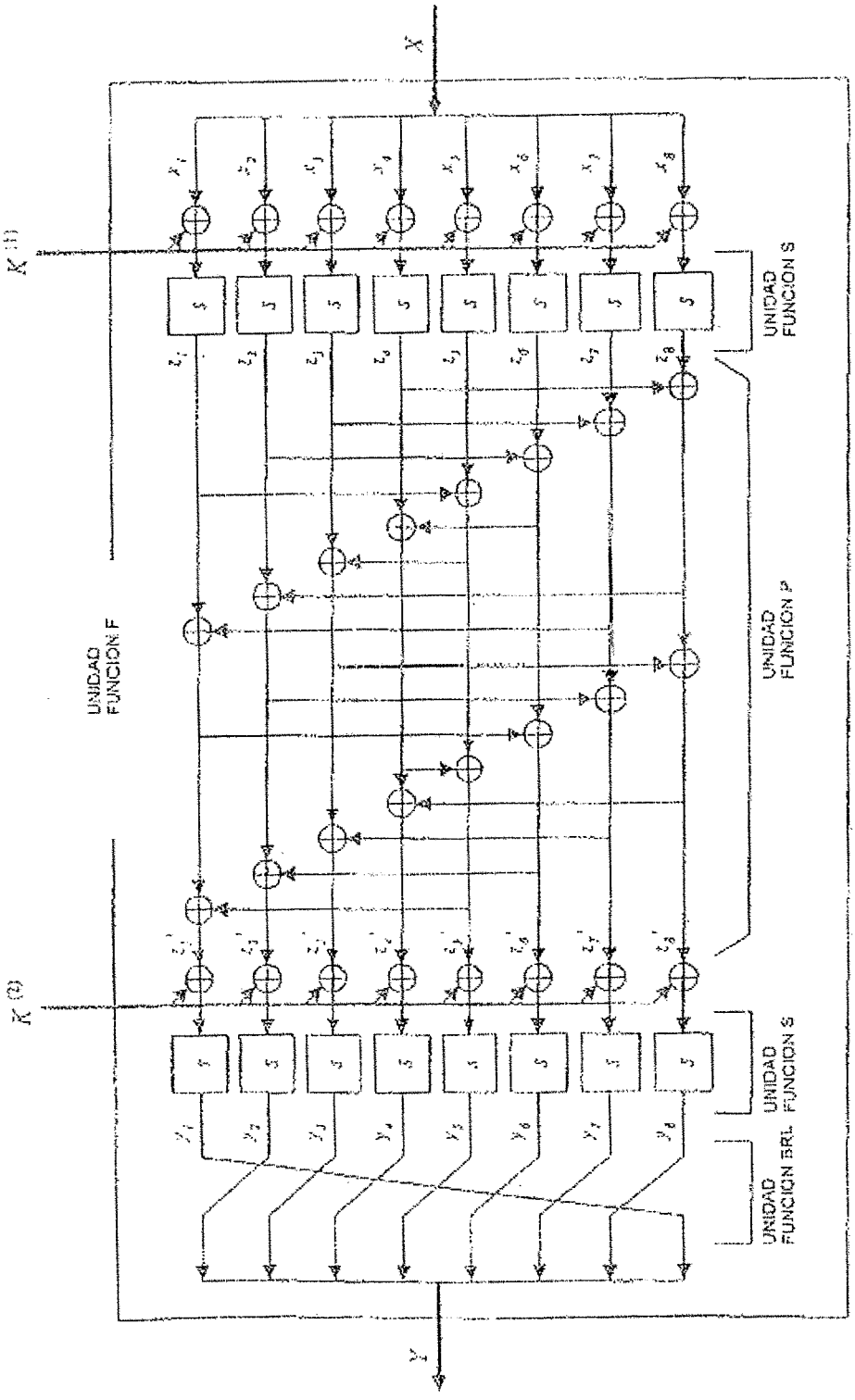

X
K(1)
K(2)
UNIDAD FUNCION F
UNIDAD FUNCION S
UNIDAD FUNCION P
UNIDAD FUNCION S
UNIDAD FUNCION SRL
Y

Fig. 27

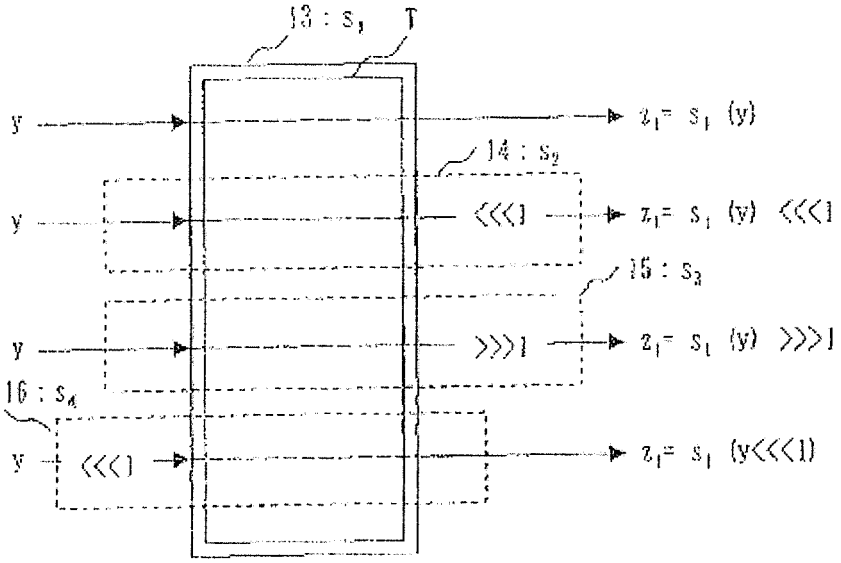


Fig. 28

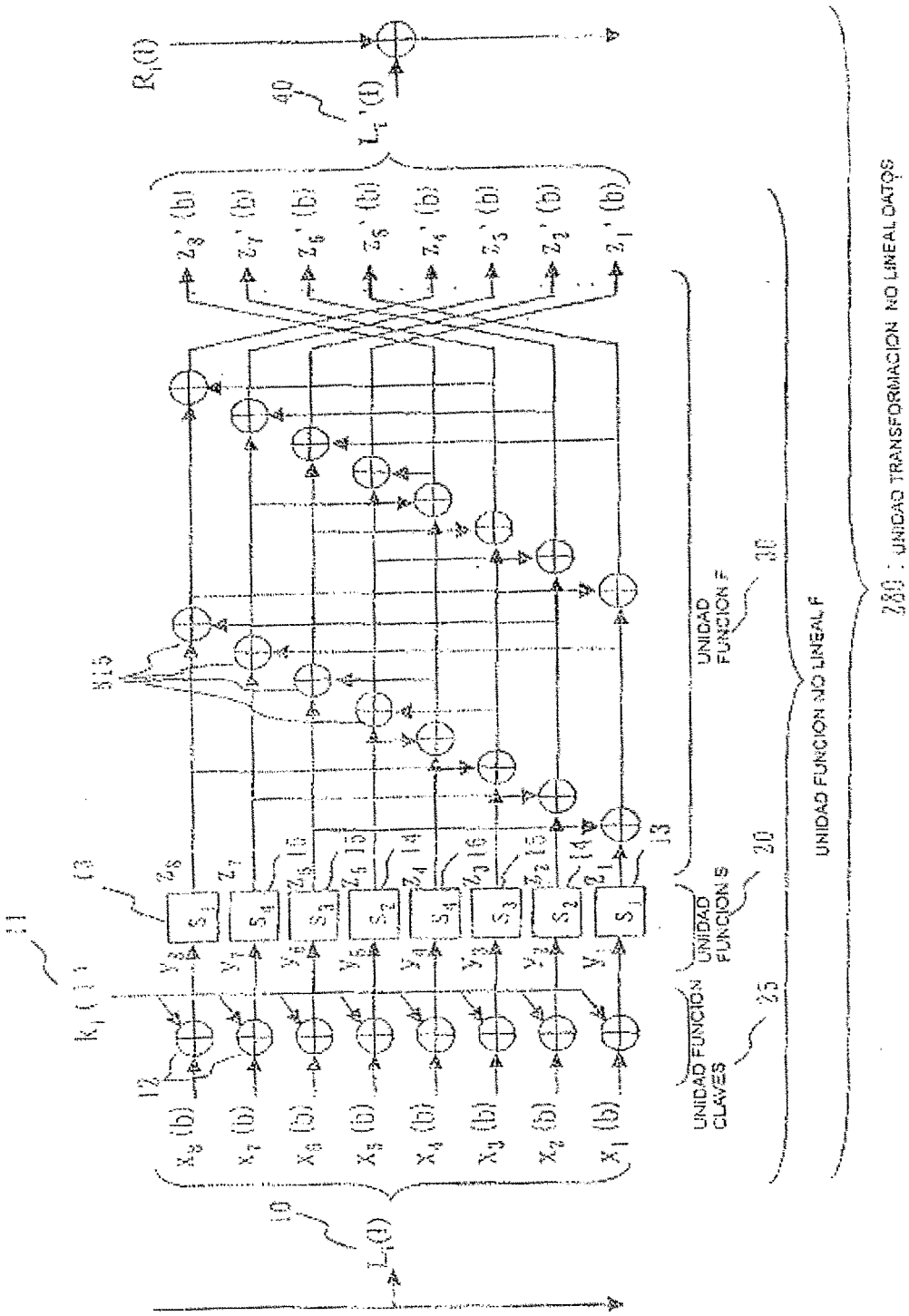

UNIDAD FUNCION F
UNIDAD FUNCION S
UNIDAD FUNCION CLAVES
UNIDAD FUNCION NO LINEAL F
UNIDAD TRANSFORMACION NO LINEAL DATOS

Fig. 29

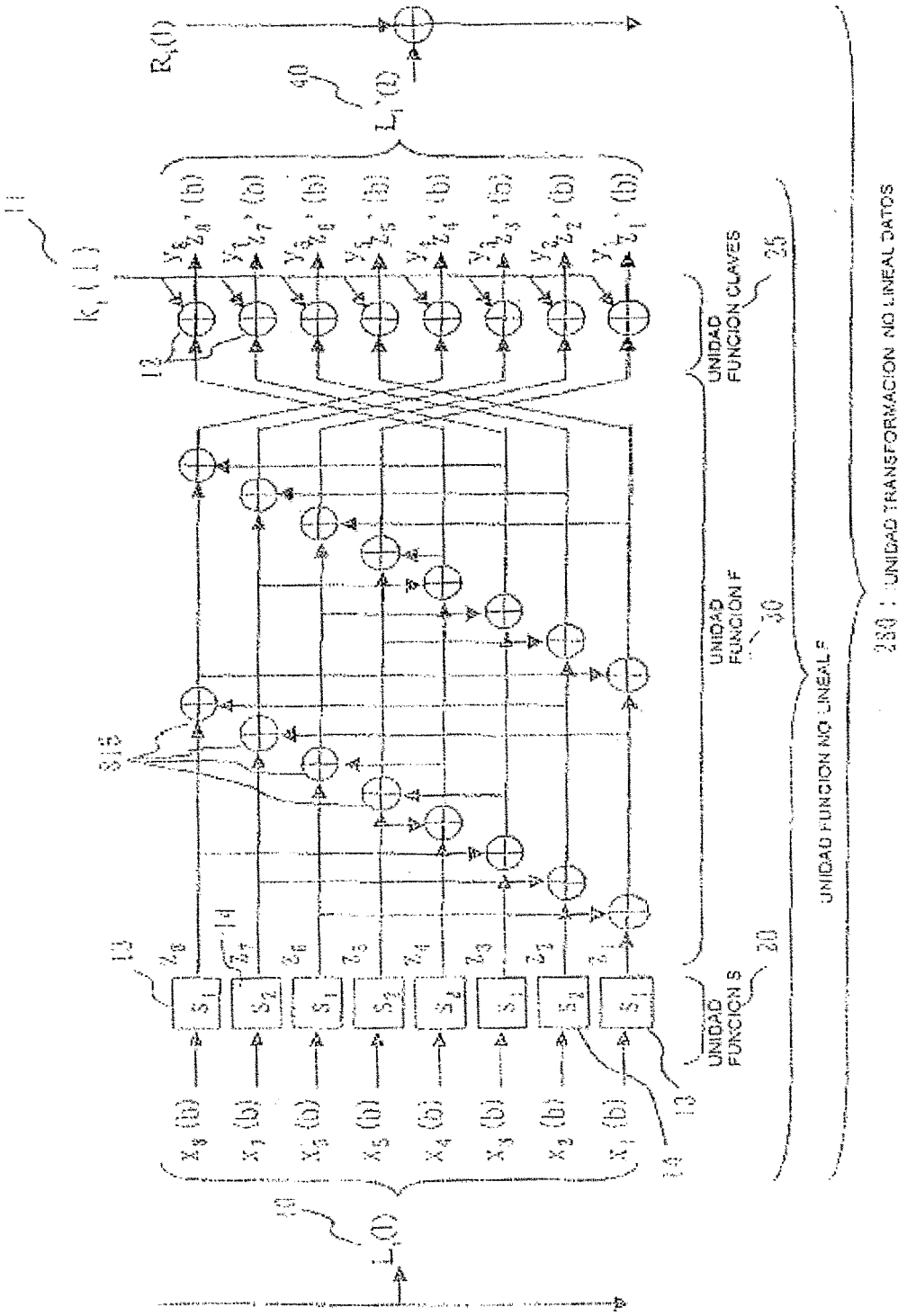

UNIDAD FUNCION CLAVES
UNIDAD FUNCION F
UNIDAD FUNCION S
UNIDAD FUNCION NO LINEAL F
UNIDAD TRANSFORMACION NO LINEAL DATOS

Fig. 30

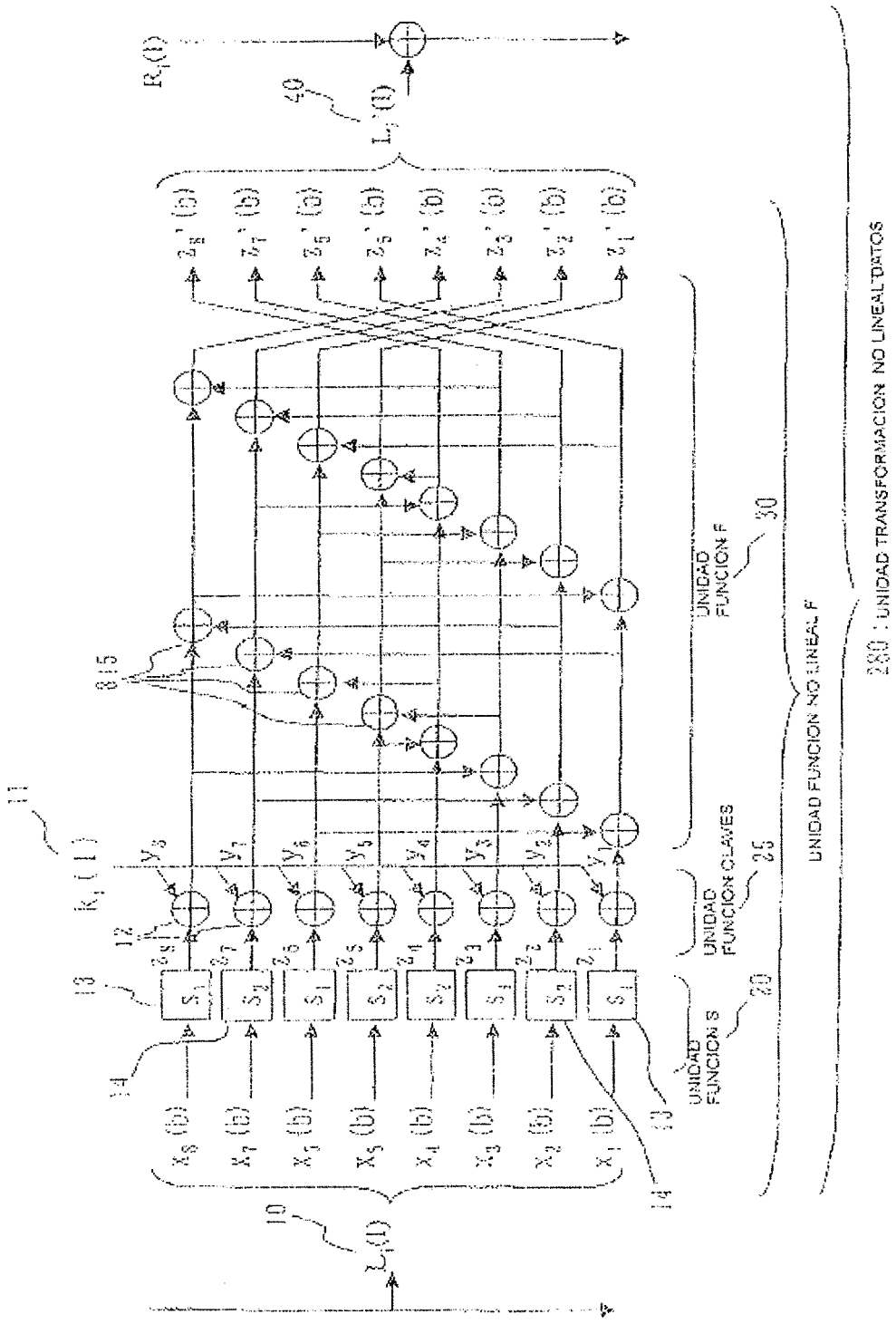

UNIDAD FUNCION S
UNIDAD FUNCION CLAVES
UNIDAD FUNCION F
UNIDAD FUNCION NO LINEAL F
280 : UNIDAD TRANSFORMACION NO LINEAL DATOS

Fig. 31

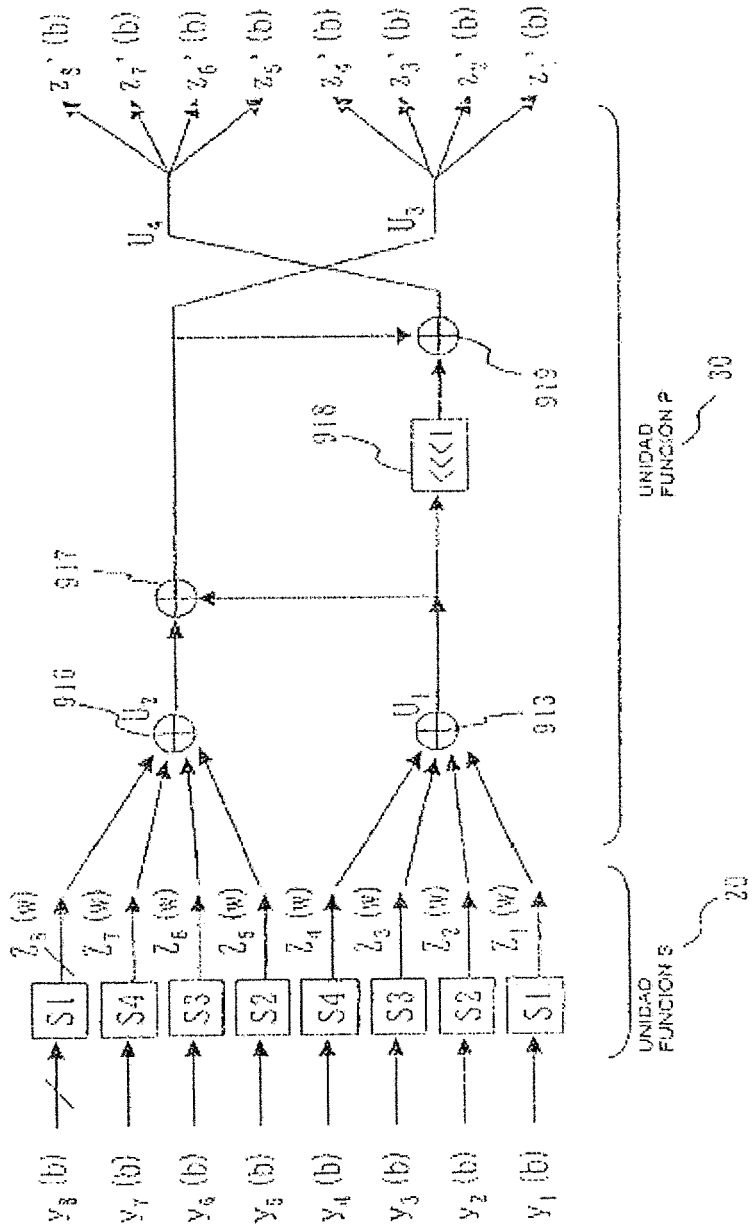

UNIDAD FUNCIÓN P
30
UNIDAD FUNCIÓN S
20
S1
S4
S3
S2
S4
S3
S2
S1
<<<1
913
916
917
918
919

Fig. 32

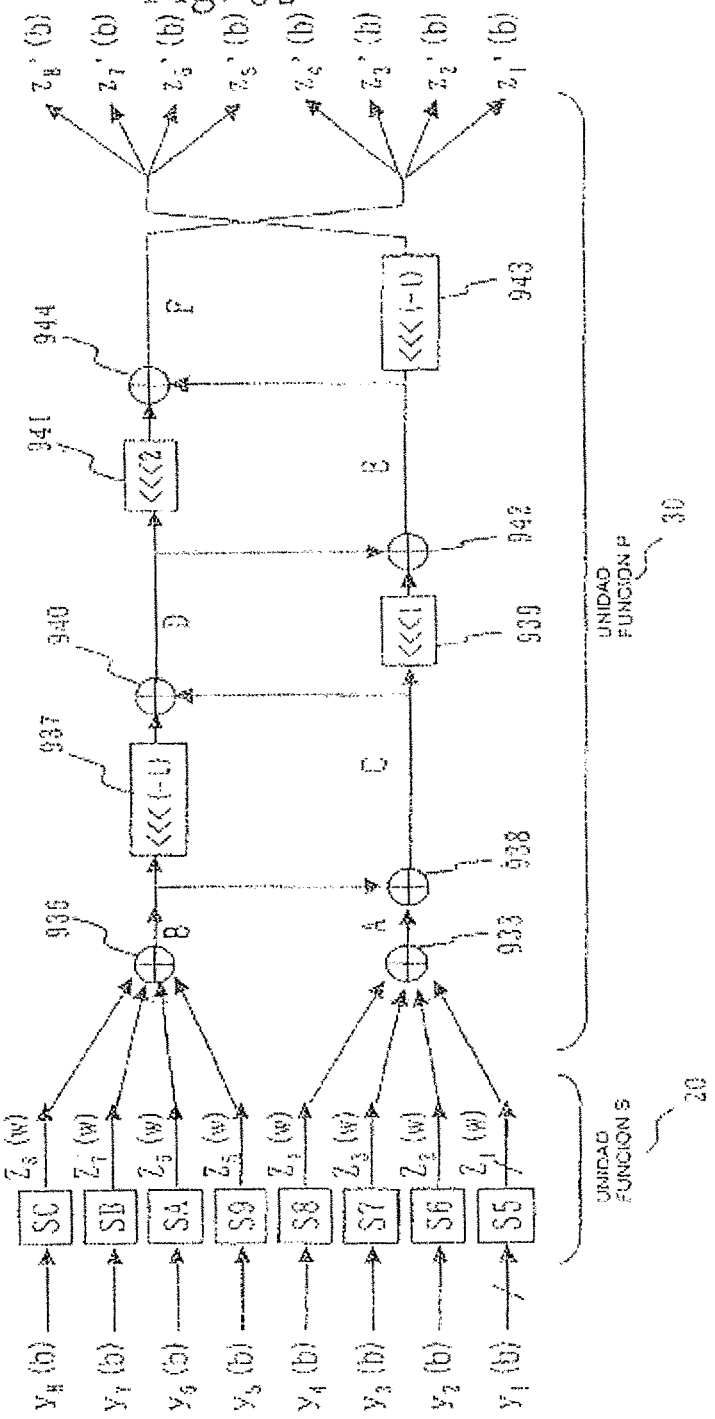

UNIDAD FUNCION S
20
UNIDAD FUNCION P
30
SC
SB
SA
S9
S8
S7
S6
S5
936
937
938
939
940
941
942
943
944
A
B
C
D
E
F
<<<(i-1)
<<<1
<<<2

Fig. 33

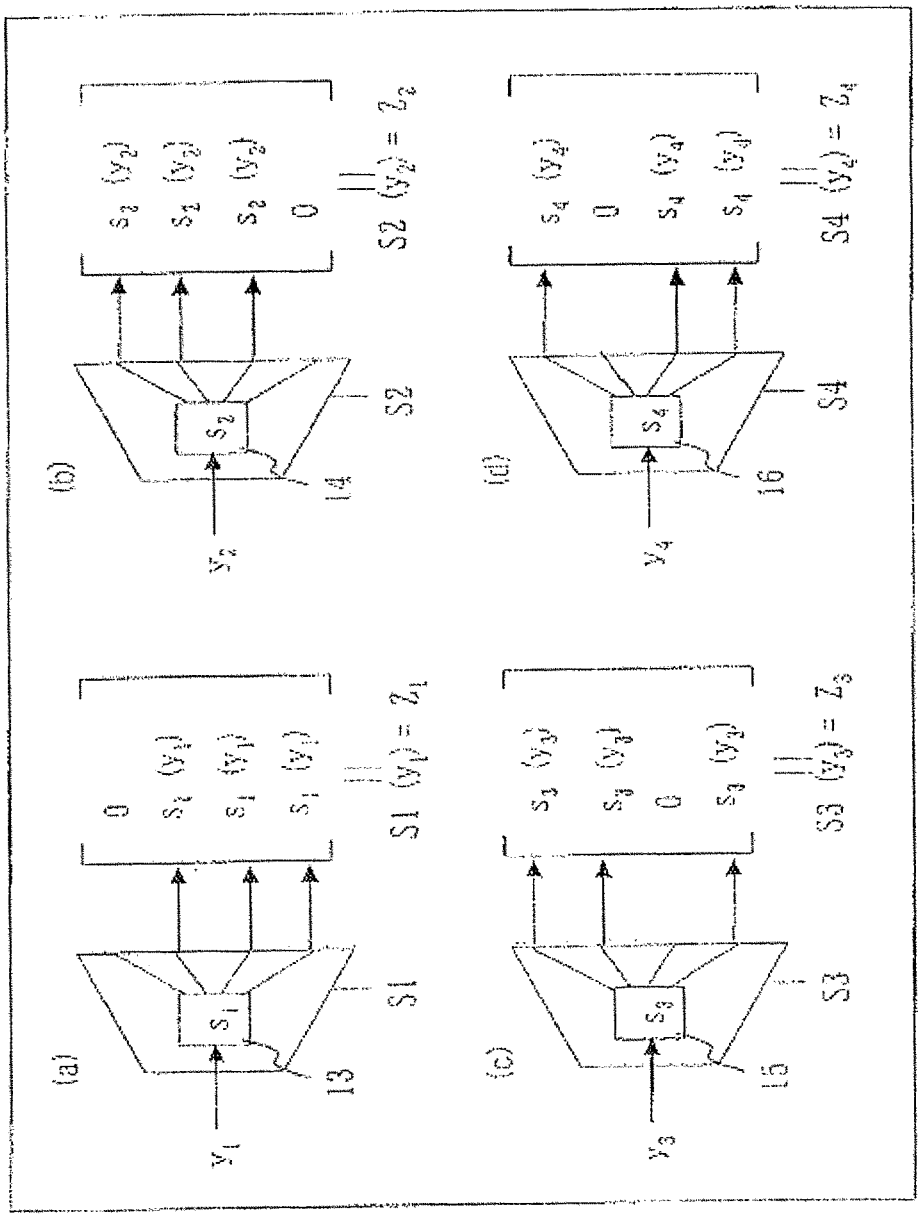

(a)
S1
13
y_1
s_1
0 $s_1(y_1)$ $s_1(y_1)$ $s_1(y_1)$
$S1(y_1) = z_1$
(b)
S2
14
y_2
s_2
$s_2(y_2)$ $s_2(y_2)$ $s_2(y_2)$ 0
$S2(y_2) = z_2$
(c)
S3
15
y_3
s_3
$s_3(y_3)$ $s_3(y_3)$ 0 $s_3(y_3)$
$S3(y_3) = z_3$
(d)
S4
16
y_4
s_4
$s_4(y_4)$ 0 $s_4(y_4)$ $s_4(y_4)$
$S4(y_4) = z_4$

Fig.34

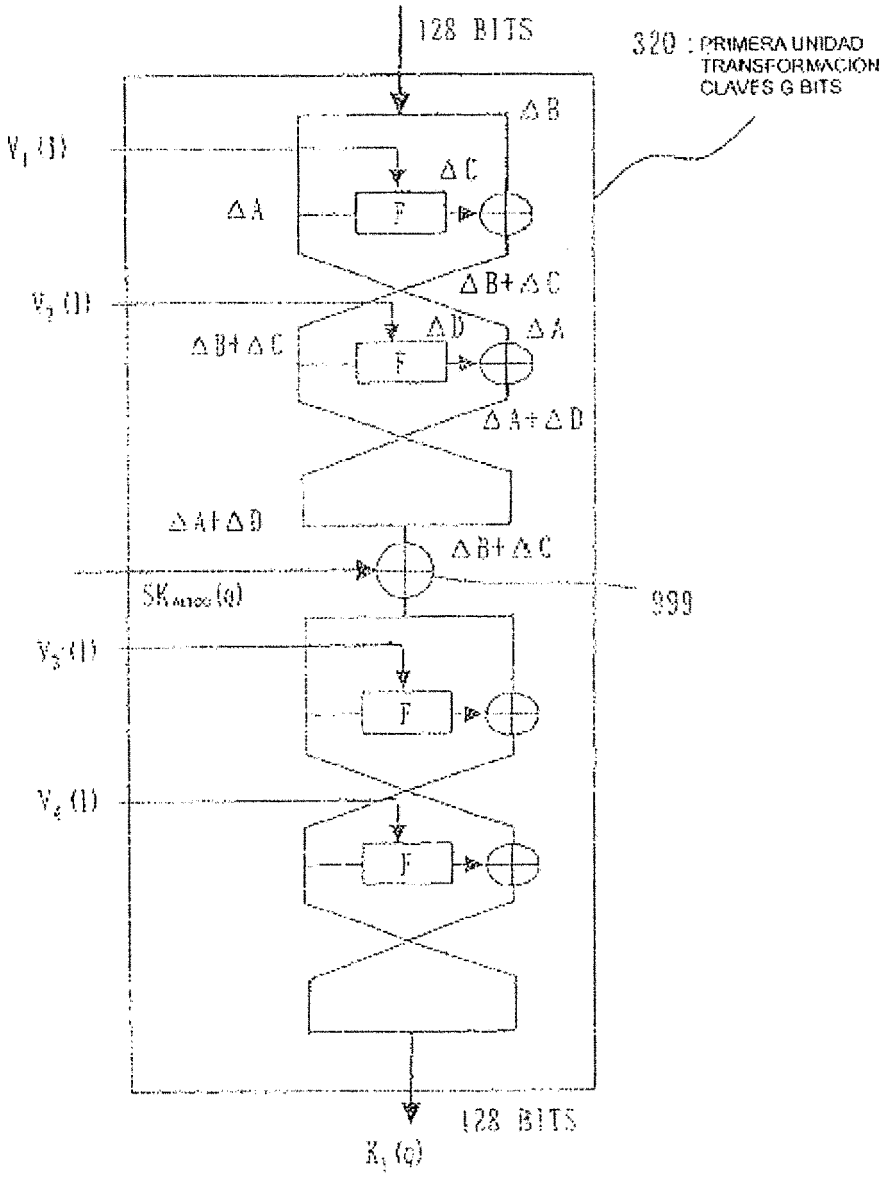

128 BITS
320 : PRIMERA UNIDAD TRANSFORMACION CLAVES G BITS
ΔB
ΔC
ΔA
F
ΔB+ΔC
ΔD
ΔA+ΔD
999
128 BITS

Fig.35

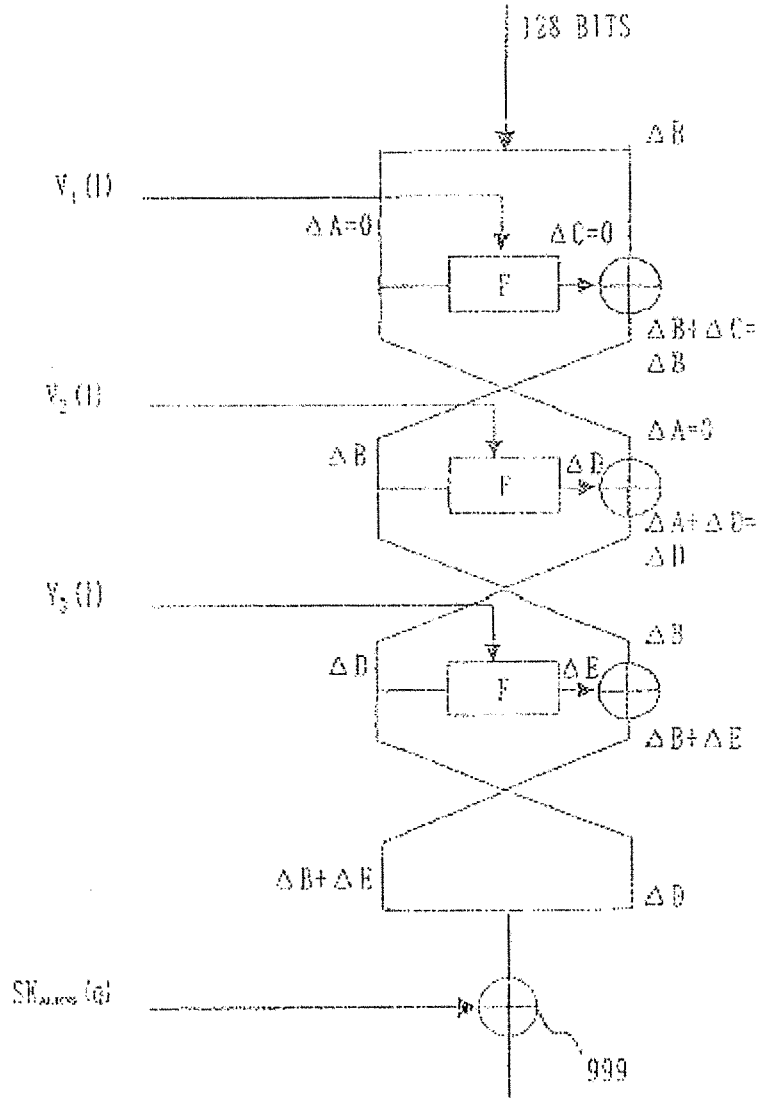

128 BITS
ΔB
V_1 (1)
ΔA=0
ΔC=0
F
ΔB+ΔC=
ΔB
V_2 (1)
ΔA=0
ΔB
ΔD
F
ΔA+ΔD=
ΔD
V_3 (1)
ΔB
ΔD
ΔE
F
ΔB+ΔE
ΔB+ΔE
ΔD
SK_{ALICE} (G)
999

Fig.36

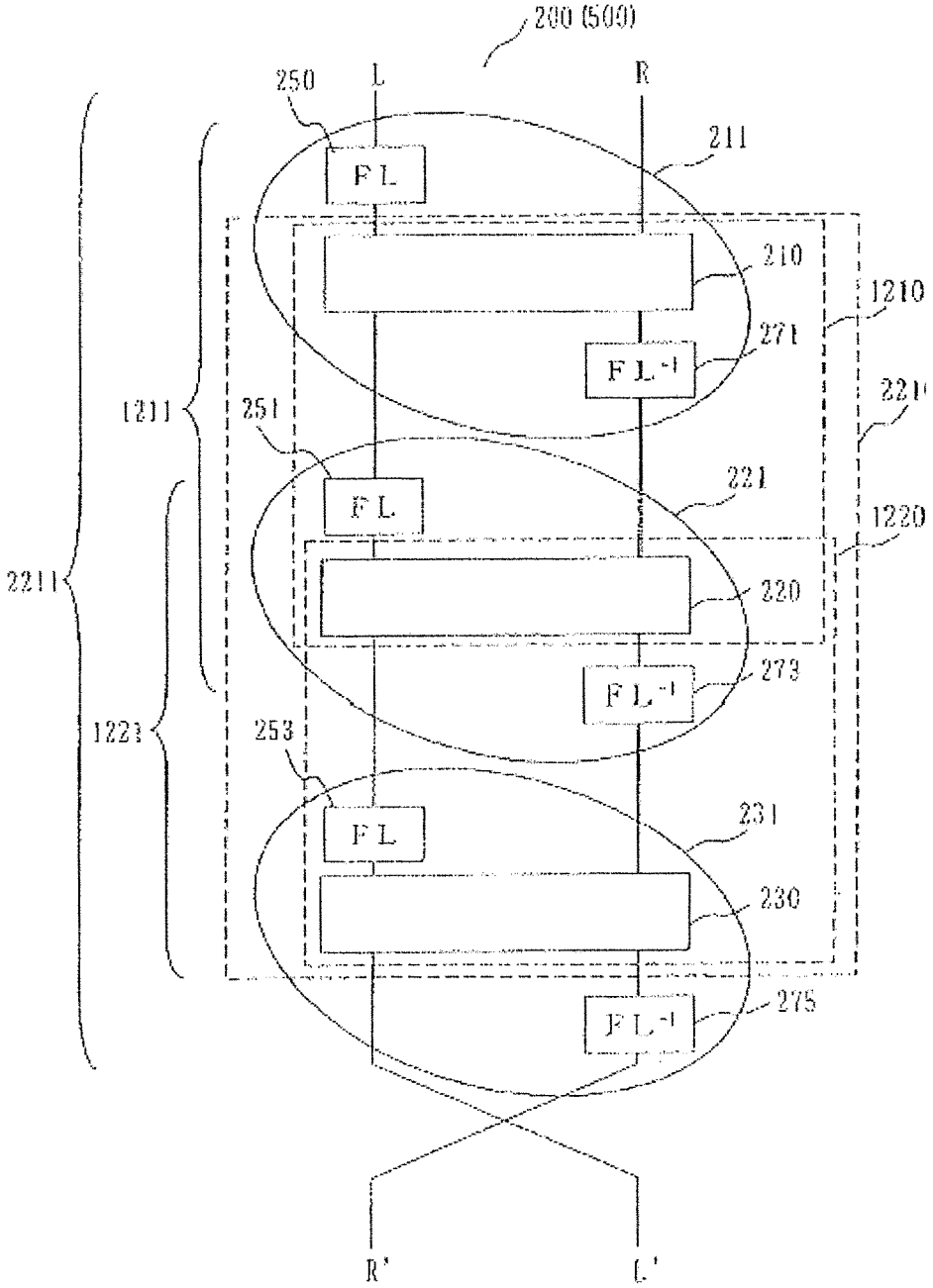

200 (500)
250
L
R
211
FL
210
1210
271
FL-1
1211
251
221
FL
1220
220
2211
273
FL-1
1223
253
231
FL
230
275
FL-1
R'
L'

Fig. 37

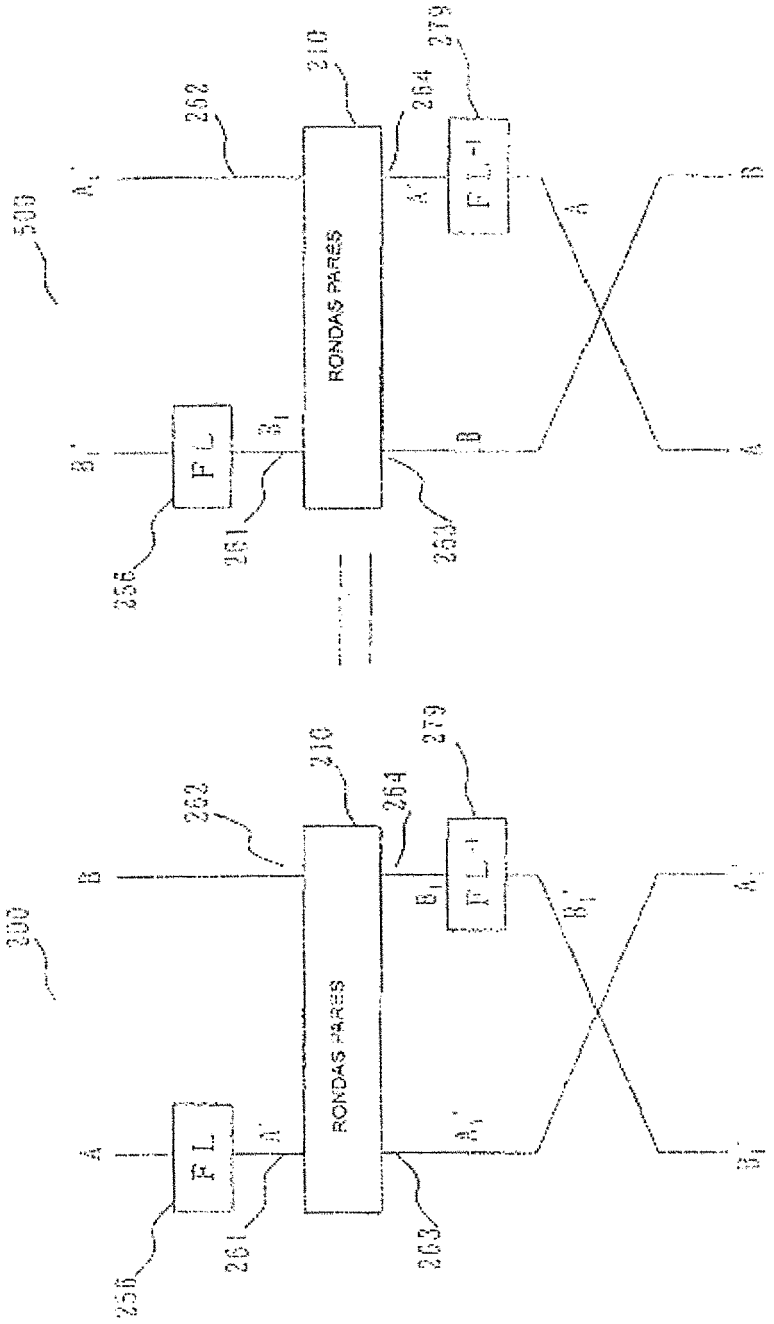


Fig. 38

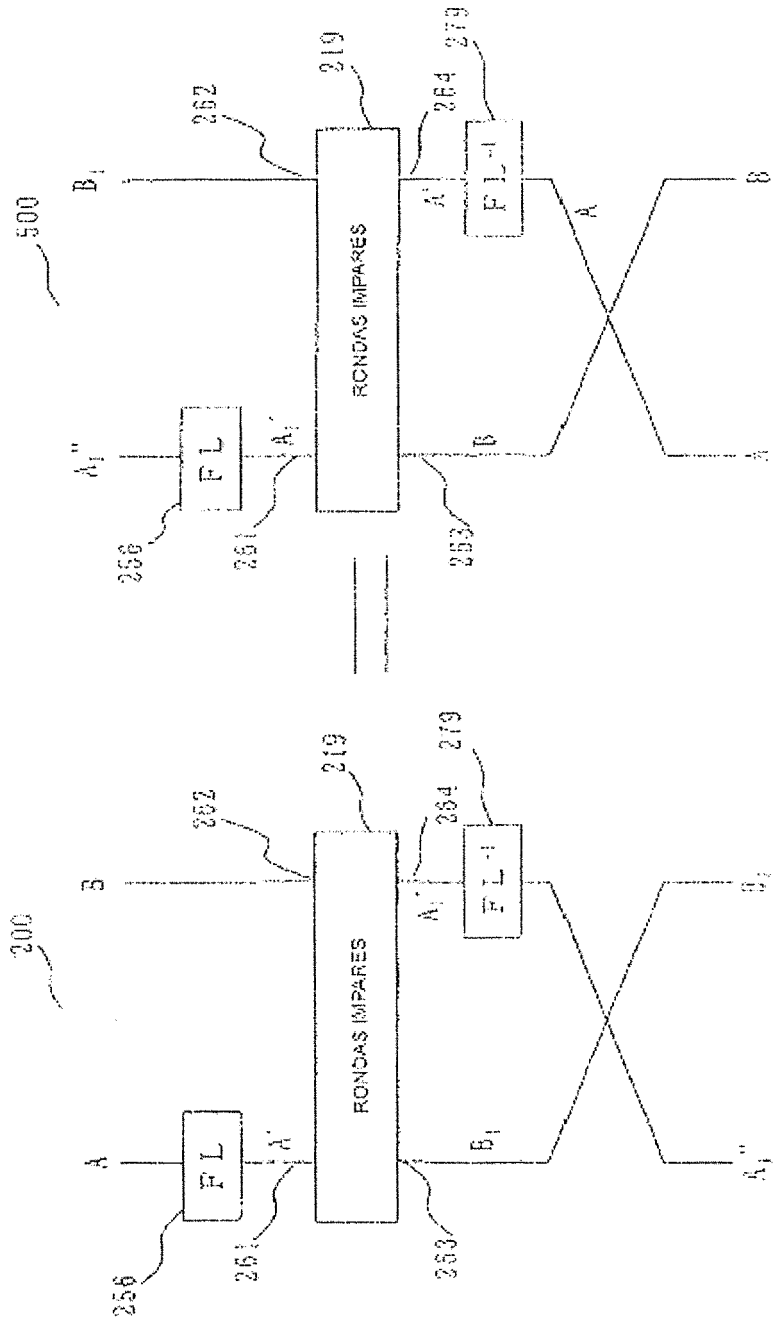

500
200
RONDAS IMPARES
RONDAS IMPARES
FL
FL
FL-1
FL-1
219
219
252
252
253
253
256
256
261
261
263
263
264
264
279
279

Fig. 39

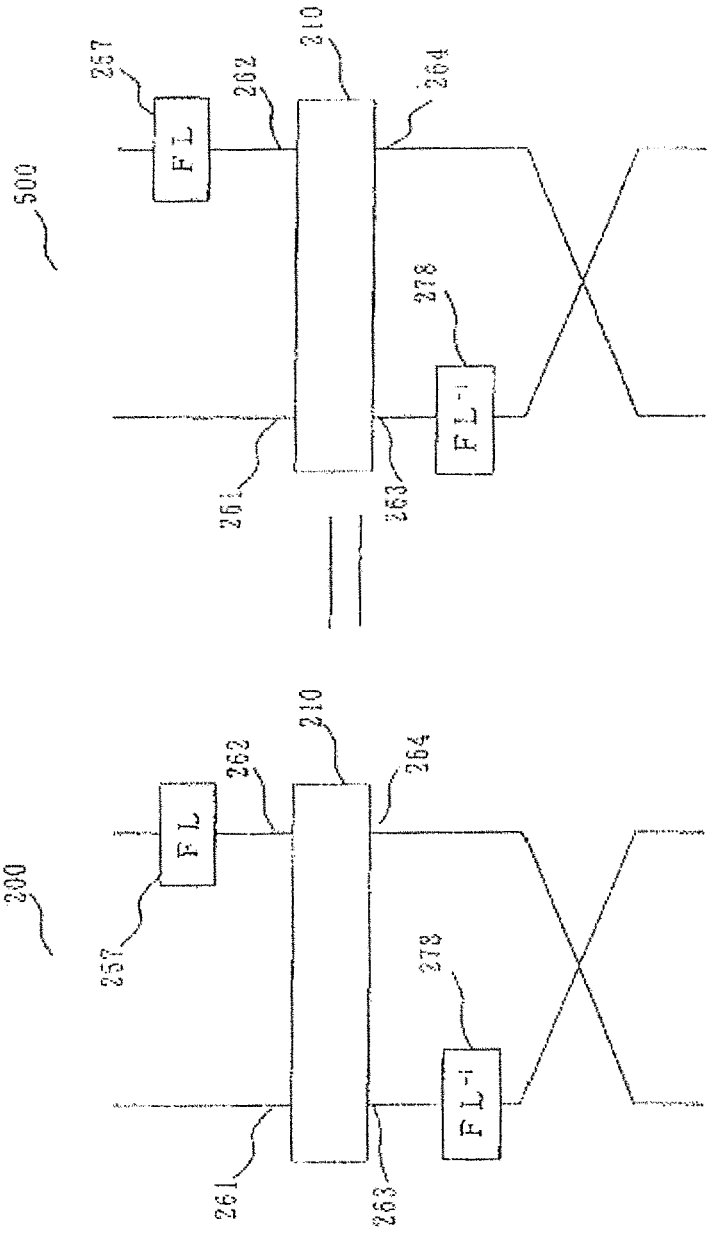

500
257
FL
202
210
264
278
FL-1
261
263
200
257
FL
262
210
264
278
FL-1
261
263

Fig. 40

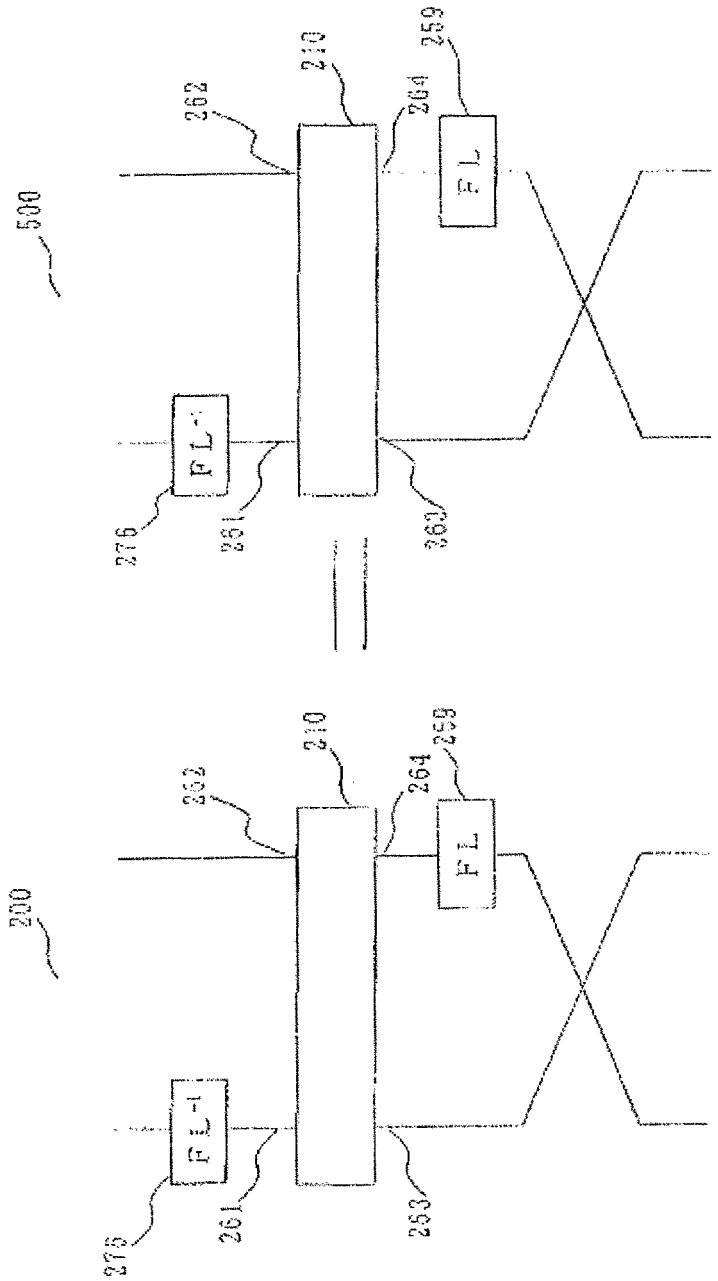

500
262
210
264
259
FL
FL-1
276
261
260
200
252
210
264
259
FL
FL-1
276
261
253

Fig. 41

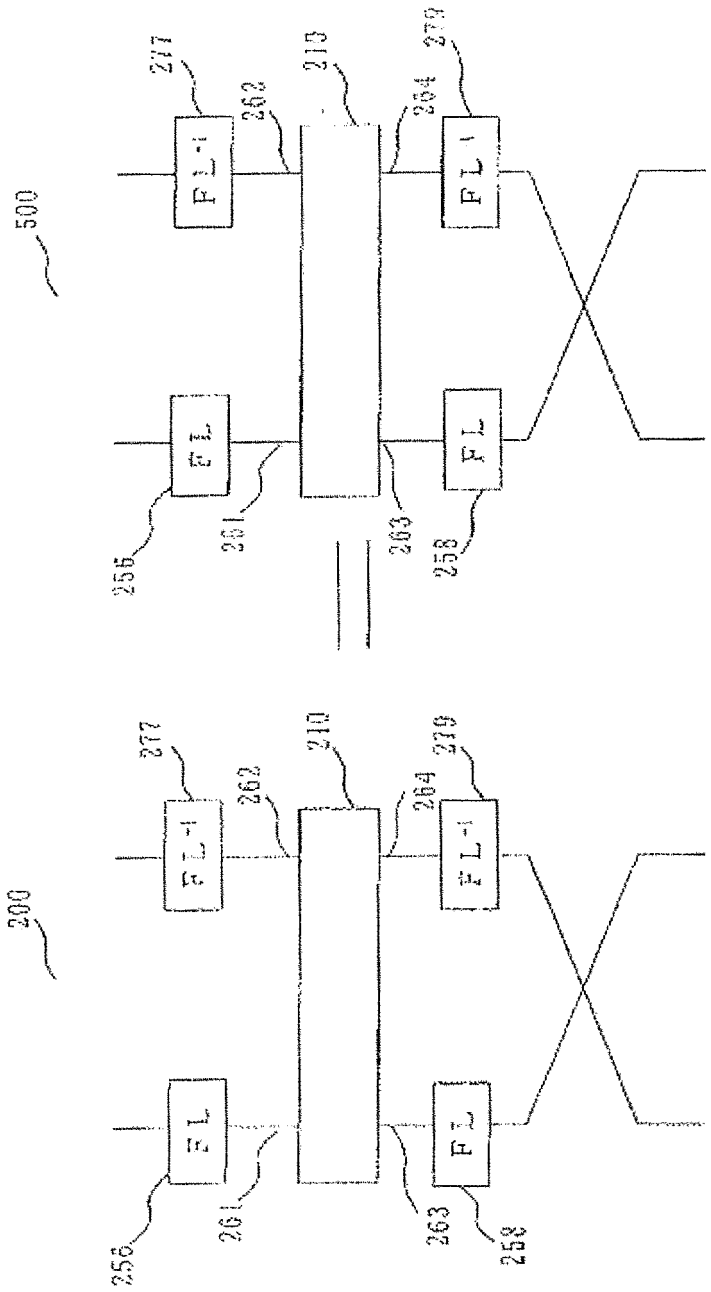

500
277
262
213
264
279
FL-1
FL
256
261
263
258
200
277
262
210
264
279
FL-1
FL
256
261
263
258

Fig. 42

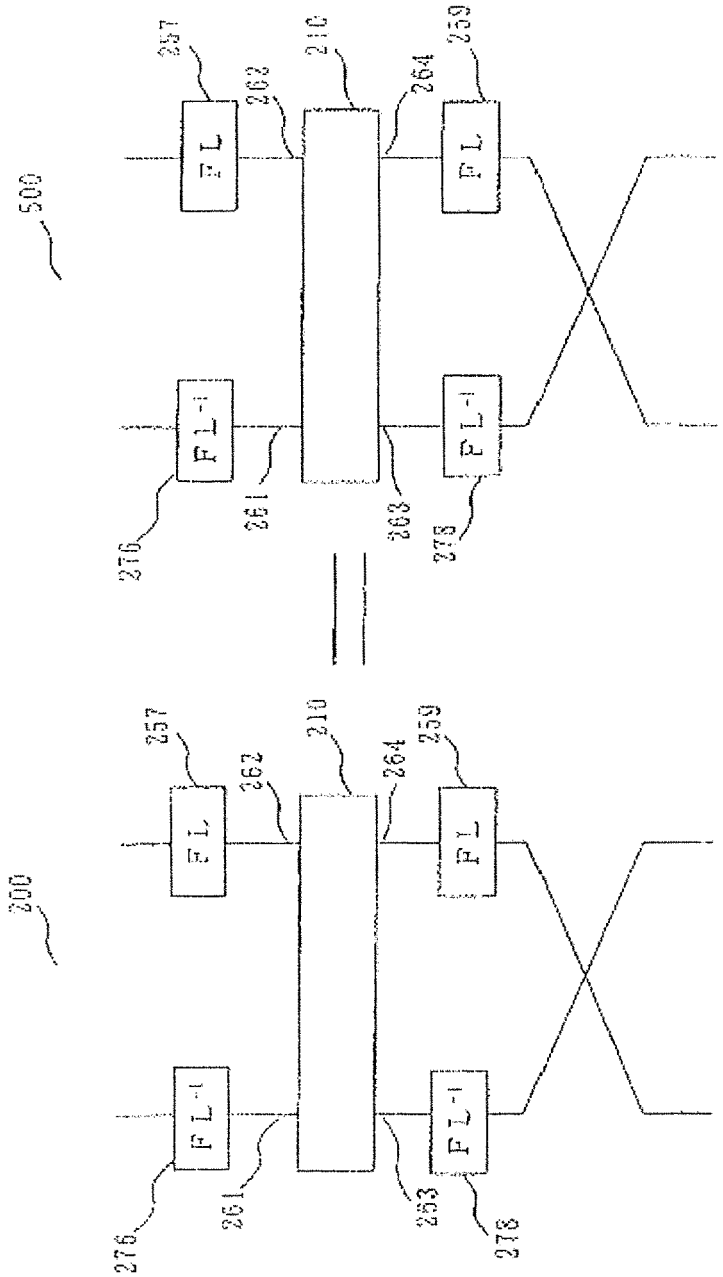

500
257
262
210
264
259
FL
FL
FL-1
FL-1
276
261
268
278
200
257
262
210
264
259
FL
FL
FL-1
FL-1
276
261
253
278

Fig. 43

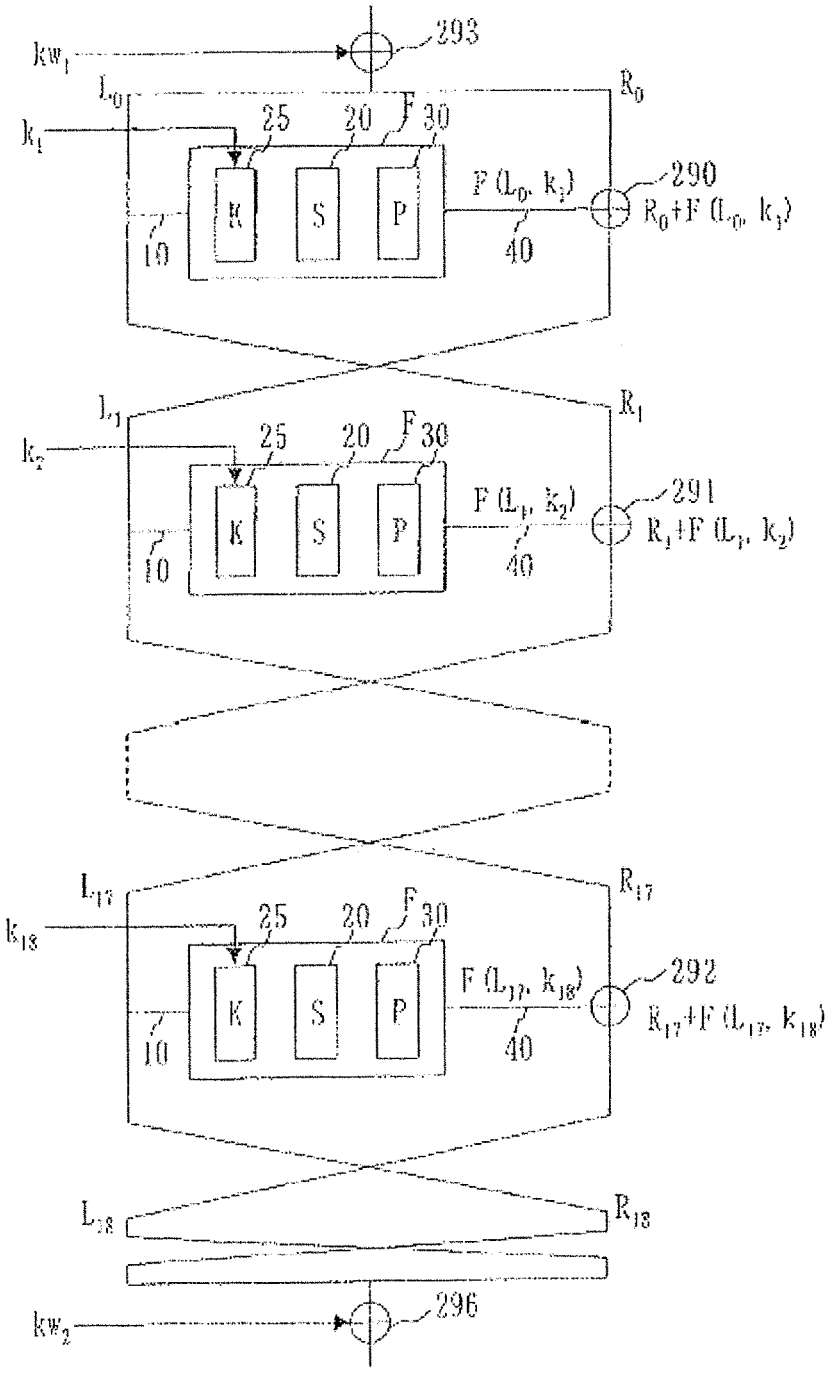

293
kw_1
L_0
R_0
k_1
25
20
F
30
K
S
P
10
40
$F(L_0, k_1)$
290
$R_0+F(L_0, k_1)$
L_1
R_1
k_2
$F(L_1, k_2)$
291
$R_1+F(L_1, k_2)$
L_{17}
R_{17}
k_{18}
$F(L_{17}, k_{18})$
292
$R_{17}+F(L_{17}, k_{18})$
L_{18}
R_{18}
kw_2
296

Fig. 44

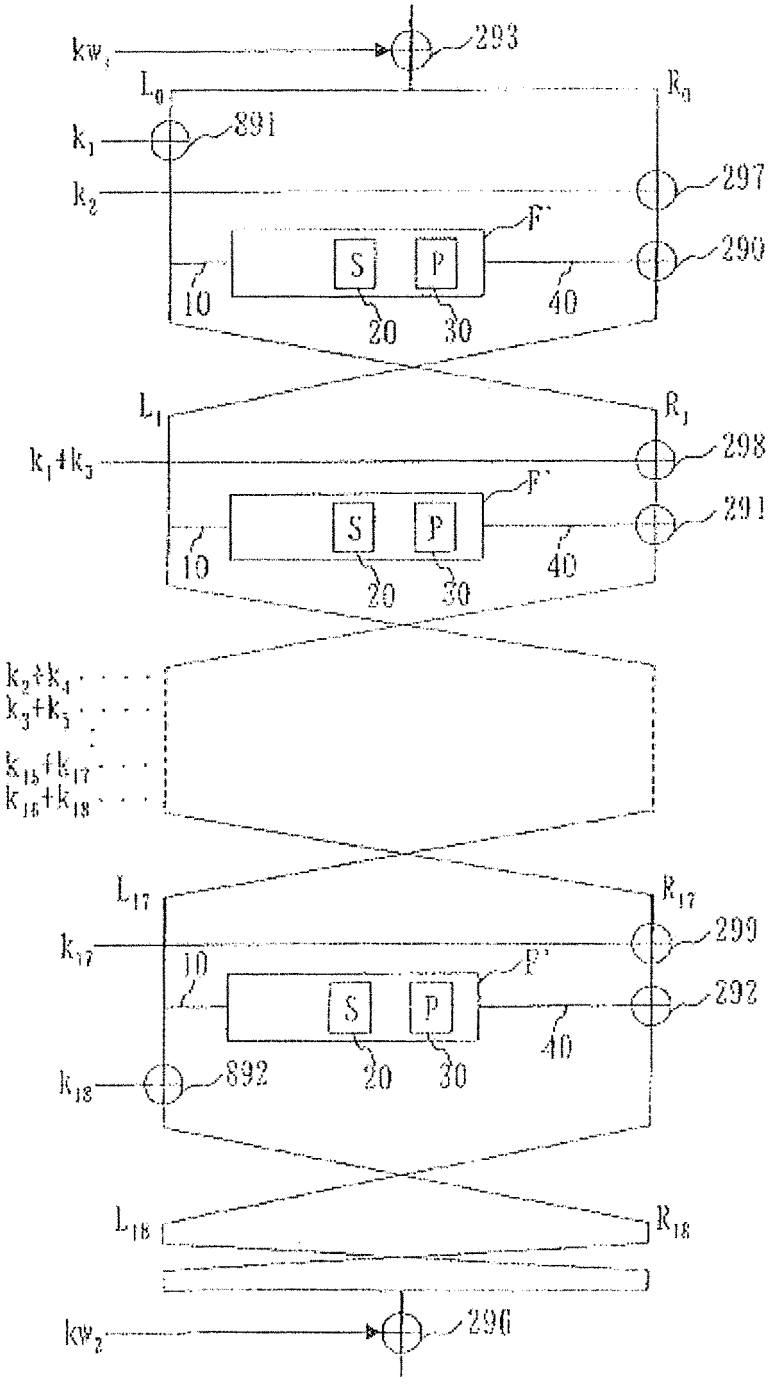

kw1
293
L0
R0
k1
891
k2
297
F
S
P
290
10
20
30
40
L1
R1
k1+k3
298
291
k2+k4
k3+k5
k15+k17
k16+k18
L17
R17
k17
299
292
k18
892
L18
R18
kw2
296

Fig. 45

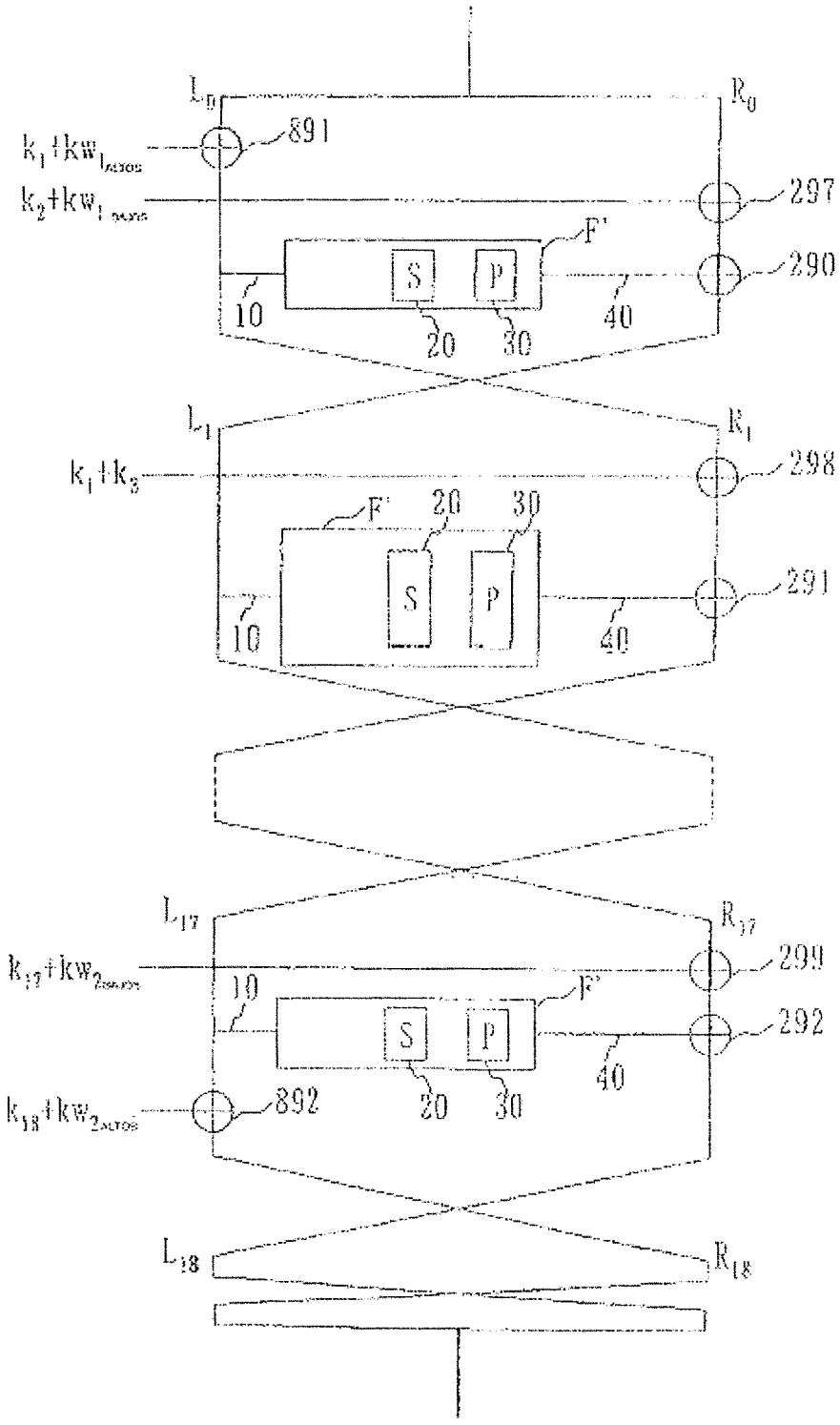

L_0
R_0
891
$k_1+kw_{1\ even}$
$k_2+kw_{1\ even}$
297
F'
S
P
290
10
20
30
40
L_1
R_1
k_1+k_3
298
F'
20
30
S
P
291
10
40
L_{17}
R_{17}
$k_{17}+kw_{2\ even}$
299
10
F'
S
P
292
40
$k_{18}+kw_{2\ even}$
892
20
30
L_{18}
R_{18}

Fig. 46

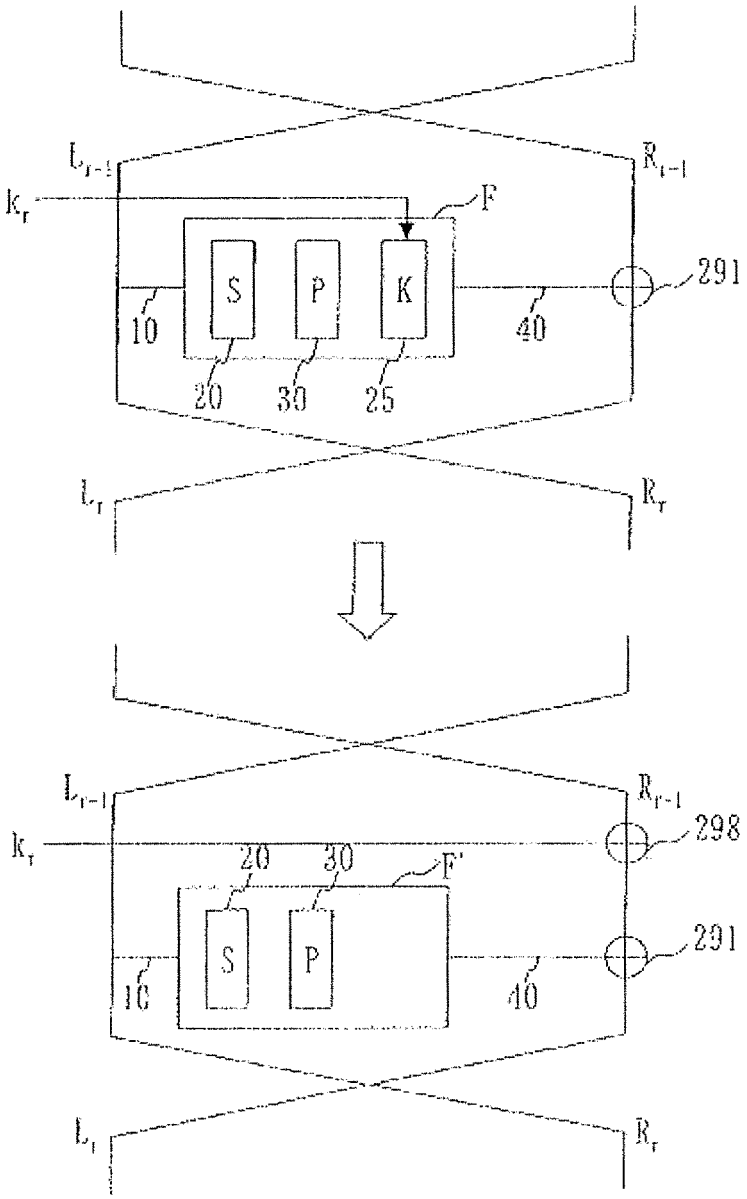
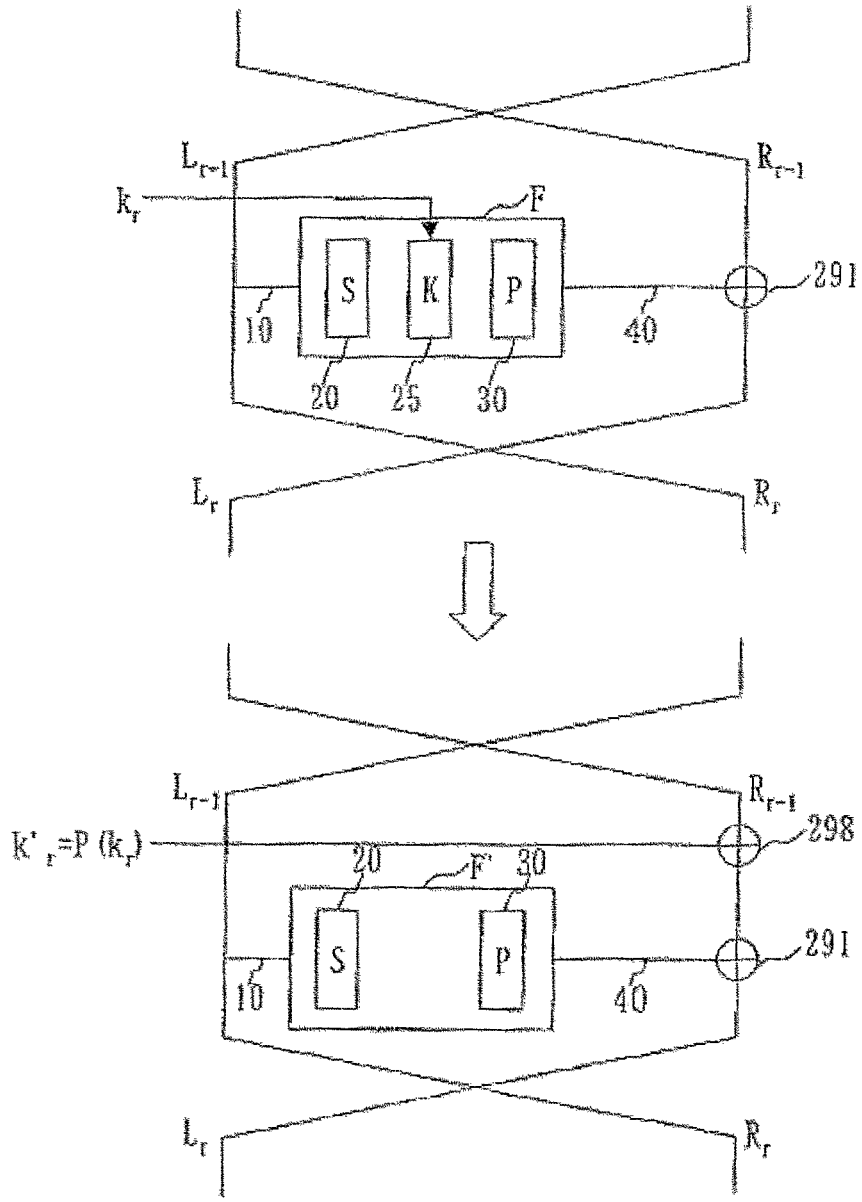

L_{r-1}
R_{r-1}
k_r
F
S
P
K
291
10
40
20
30
25
L_r
R_r
L_{r-1}
R_{r-1}
k_r
298
20
30
F'
S
P
291
10
40
L_r
R_r

Fig. 47


L_{r-1}
R_{r-1}
k_r
F
S
K
P
291
10
40
20
25
30
L_r
R_r
L_{r-1}
R_{r-1}
$K'_r=P(k_r)$
298
20
F'
30
S
P
291
10
40
L_r
R_r