(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 405 942**

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

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **08.03.2001** **E 06010076 (5)**

(97) Fecha y número de publicación de la concesión europea: **06.03.2013** **EP 1689113**

(54) Título: **Aparato de cifrado en bloques que utiliza una transformación auxiliar**

(30) Prioridad:

09.03.2000 JP 2000064614

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
04.06.2013

(73) Titular/es:

MITSUBISHI DENKI KABUSHIKI KAISHA (50.0%)
7-3, MARUNOUCHI 2-CHOME
CHIYODA-KU TOKYO 100-8310, JP y
NIPPON TELEGRAPH & TELEPHONE CORPORATION (50.0%)

(72) Inventor/es:

MATSUI, MITSURU;
TOKITA, TOSHIO;
NAKAJIMA, JUNKO;
KANDA, MASAYUKI ;
MORIAI, SHIHO y
AOKI, KAZUMARO

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 405 942 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Aparato de cifrado en bloques que utiliza una transformación 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 soportes informáticos en los cuales son registrados los procedimientos de transformación de datos, para su cifrado, descifrado y difusión de los datos con el fin de proteger información digital existente en las comunicaciones de información.

Técnica anterior

La figura 25 representa una función de cifrado que se utiliza en el DES (Estándar de Cifrado de Datos) descrito en "Gendai Ango Riron" ("Modern Cipher Theor") (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).

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

En Schneier B: “Applied Cryptography, Description of DES”, 1996, “Applied Cryptography. Protocols, Algorithms, and Source Code in C, Nueva York, John Wiley & Sons, US, páginas 270-277, XP002237575, ISBN: 0-471-11709-9, se describe el estándar de cifrado de datos en general.

La figura 26 muestra una función de transformación no lineal que se describe en "Specification of E2 - a 128-bit Block Cipher" (Nippon Telegraph and Telephone Corporation" publicado el 14 de Junio de 1998, página 10).

Tal como se muestra en la figura 26, cada unidad de función S se compone de ocho cajas S.

Los dispositivos de cifrado 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.

Tal como se muestra en la figura 7, cuando una unidad de transformación normal de datos (FL) 250 es insertada en la unidad de cifrado, se requiere suministrar una unidad de transformación inversa de datos (FL^{-1}) 270 en una unidad de descifrado para descifrar los textos cifrados. Dado que, por regla general, la unidad de transformación normal de datos (FL) 250 y la unidad de transformación inversa de datos (FL^{-1}) 270 son circuitos diferentes entre sí, ello ocasiona un problema en el sentido de que la unidad de cifrado y la unidad de descifrado 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.

En Daemen J. et al.: “AES Proposal: Rijndael” AES Proposal, 3 de septiembre de 1999 (1999-09.03), páginas 1-45, XP001060386, se describe el algoritmo de cifrado por bloques de Rijndael. El Rijndael es un cifrado por bloques iterados con una longitud de bloque variable y una longitud de clave variable. La longitud del bloque y la longitud de la clave se pueden especificar de manera independiente en 128, 192 ó 256 bits.

Anderson R. et al.: “Serpent: A proposal for the advanced encryption standard”, páginas 1-23, describe un cifrado por bloques como un candidato para el estándar de cifrado avanzado. Utiliza cajas S similares a las del DES en una nueva estructura. Tiene un tamaño de bloque de 128 bits y una clave de 256 bits.

La presente invención tiene como objetivo proporcionar unos sistemas en los cuales los circuitos de cifrado y descifrado son 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 el cálculo de una transformación no lineal pueden reducirse, y además, pueden generarse las claves de extensión utilizando una configuración más sencilla.

Divulgación de la invención

Este objetivo se soluciona mediante el aparato de transformación de datos de acuerdo con la reivindicación 1, el procedimiento de transformación de datos de acuerdo con la reivindicación 6, el programa informático de acuerdo con la reivindicación 8 y el soporte informático de acuerdo con la reivindicación 9. Otras mejoras del aparato y del procedimiento se proporcionan en las respectivas reivindicaciones dependientes.

Un aparato de transformación de datos de la presente invención se caracteriza porque tiene una unidad de procesamiento de datos para introducir datos de clave y para llevar a cabo al menos uno entre el cifrado de datos y

el descifrado de datos, la unidad de procesamiento de datos incluye una unidad de transformación no lineal para realizar una transformación no lineal de los datos,

la unidad de transformación no lineal incluye:

una primera unidad (s_1) de transformación para introducir una parte de los datos a transformar como unos primeros datos parciales, que transforma los primeros datos parciales usando una tabla T de transformación, que introduce los datos, transforma un valor de los datos en otro valor, y que emite los datos, y emitir los datos transformados, y
una segunda unidad (s_2) de transformación para introducir al menos otra parte de los datos a transformar como segundos datos parciales, transformando los segundos datos parciales mediante una transformación usando dicha tabla T de transformación y una operación para la segunda parte, y emitiendo los datos transformados.

La primera unidad de transformación (s_1) anterior introduce los datos y_1 en la tabla T de transformación para emitir datos $s_1(y_1)$ y emite los datos $s_1(y_1)$ como datos $z_1 = s_1(y_1)$, y la segunda unidad (s_2) de transformación introduce datos y_2 a la tabla T de transformación para emitir datos $s_1(y_2)$, realiza el cambio rotacional en $s_1(y_2)$ para emitir (rot ($s_1(y_2)$)), y emite los datos (rot ($s_1(y_2)$)) como datos z_2 = rot ($s_1(y_2)$).

La unidad de procesamiento de datos anterior también incluye una tercera unidad (s_3) de transformación y una cuarta unidad (s_4) de transformación para introducir respectivamente datos parciales que son diferentes de los primeros datos parciales y los segundos datos parciales como terceros datos parciales e introducir datos parciales que son diferentes de los primeros datos parciales, los segundos datos parciales, y los terceros datos parciales como cuartos datos parciales, transformando los terceros datos parciales y los cuartos datos parciales mediante la transformación usando la tabla T de transformación y una operación para la tercera parte y una operación para la cuarta parte, las cuales son ambas diferentes de la operación para la segunda parte de la segunda unidad (s_2) de transformación, y emitiendo los datos transformados.

Un procedimiento de transformación de datos de la presente invención se caracteriza porque en un procedimiento de transformación de datos para ejecutar un proceso de procesamiento de datos para realizar al menos uno de cifrado de datos y descifrado de datos en base a datos de clave de entrada,

el proceso de procesamiento de datos incluye un proceso de transformación no lineal para realizar una transformación no lineal de los datos,

el proceso de transformación no lineal incluye:

un primer proceso (s_1) de transformación para introducir una parte de los datos a transformar como unos primeros datos parciales, transformando los primeros datos parciales usando una tabla T de transformación que introduce datos, transforma un valor de los datos en otro valor, y emite los datos, y emitiendo los datos transformados, y
un segundo proceso (s_2) de transformación para introducir al menos otra parte de los datos a transformar como segundos datos parciales, transformando los segundos datos parciales mediante transformación usando la tabla T de transformación y una operación para la segunda parte, y emitiendo los datos transformados.

La presente invención se caracteriza por un soporte informático legible por ordenador para almacenar un programa para tener un ordenador que realice el procedimiento de transformación de datos anterior.

La presente invención se caracteriza por un programa que tiene un ordenador que realiza el procedimiento de transformación de datos anterior.

Breve explicación de los dibujos

La figura 1 muestra una unidad de transformación de datos para el cifrado 100 y una unidad de transformación de datos para el descifrado 400.
La figura 2 muestra notaciones.
La figura 3 muestra una configuración de una unidad 200 de cifrado o de una unidad 500 de descifrado.
La figura 4 muestra otra configuración de la unidad 200 de cifrado o de la unidad 500 de descifrado.
La figura 5 muestra una configuración de una unidad de transformación normal de datos (FL) 251.
La figura 6 muestra una configuración de una unidad de transformación inversa de datos (FL^{-1}) 271.
La figura 7 muestra una parte de una unidad de cifrado convencional y una unidad de descifrado convencional.
La figura 8 muestra una parte de la unidad 200 de cifrado y de la unidad 500 de descifrado.
La figura 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 puntual.
La figura 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 puntual.
La figura 11 muestra una unidad F de función no lineal.
La figura 12 muestra una configuración de una primera unidad 13 de transformación de caja S y una segunda unidad 14 de transformación de caja S.

La figura 13 muestra una configuración de una unidad 21 de transformación de caja S.
La figura 14 muestra una configuración de una unidad 85 de transformación lineal.
La figura 15 muestra una configuración de una unidad 87 de transformación lineal.
La figura 16 muestra una configuración de una unidad 300 de generación de claves o de una unidad 600 de generación de claves.
La figura 17 explica las operaciones de una unidad 310 de transformación de longitud de bits.
La figura 18 muestra una configuración de un registro A 341 de desplazamiento.
La figura 19 muestra una configuración de una tabla de control de una unidad 345 de control de desplazamiento.
La figura 20 muestra las operaciones del registro A 341 de desplazamiento y de un registro B 342 de desplazamiento.
La figura 21 muestra la correspondencia entre el registro A 341 de desplazamiento, el registro B 342 de desplazamiento y las claves de extensión.
La figura 22 muestra las operaciones de los registros A 341 a D 344 de desplazamiento.
La figura 23 muestra la correspondencia entre los registros A 341 a D 344 de desplazamiento y las claves de extensión.
La figura 24 muestra un ordenador equipado con la unidad de transformación de datos para el cifrado 100 y la unidad de transformación de datos para el descifrado 400.
La figura 25 muestra una configuración de la función de cifrado del DES.
La figura 26 muestra una configuración de la función no lineal del cifrado de bloques E2 de 128 bits.
La figura 27 muestra otro ejemplo de unidades de transformación de cajas S.
La figura 28 muestra una unidad F de función no lineal que está equipada con las unidades primera a cuarta de transformación de cajas S.
La figura 29 muestra otra unidad F de función no lineal en la cual se desplaza una ubicación de la unidad 25 de función de claves.
La figura 30 muestra otra unidad F de función no lineal en la cual se desplaza una ubicación de la unidad 25 de función de clave.
La figura 31 muestra otra configuración de una unidad 30 de función P.
La figura 32 muestra otra configuración de la unidad 30 de función P.
La figura 33 muestra configuraciones y operaciones de S1 a S4 de la figura 31.
La figura 34 muestra una prueba de inexistencia de claves equivalentes.
La figura 35 muestra una prueba de la no existencia de claves equivalentes.
La figura 36 muestra otra configuración de la unidad 200 de cifrado o de la unidad 500 de descifrado.
La figura 37 muestra otra configuración de la unidad 200 de cifrado o de la unidad 500 de descifrado.
La figura 38 muestra otra configuración de la unidad 200 de cifrado o de la unidad 500 de descifrado.
La figura 39 muestra otra configuración de la unidad 200 de cifrado o de la unidad 500 de descifrado.
La figura 40 muestra otra configuración de la unidad 200 de cifrado o de la unidad 500 de descifrado.
La figura 41 muestra otra configuración de la unidad 200 de cifrado o de la unidad 500 de descifrado.
La figura 42 muestra una configuración en la cual las unidades de la figura 39 y de la figura 40 están combinadas.
La figura 43 muestra una configuración de la unidad 200 de cifrado o de la unidad 500 de descifrado, la cual se muestra en la figura 3, que utiliza la unidad F de función no lineal mostrada en la figura 28.
La figura 44 muestra una configuración modificada de la figura 43 mediante la utilización de una unidad F de función no 20 lineal en la cual la unidad 25 de función de clave de la unidad F de función no lineal está suprimida.
La figura 45 muestra una configuración modificada de la figura 44 mediante la fusión de las claves de extensión de blanqueo con las claves de extensión.
La figura 46 muestra una configuración modificada en la cual la unidad 25 de función de clave está suprimida de la unidad F de función no lineal y en la cual una clave de extensión k es suministrada a un circuito XOR 298, cuando la unidad F de función no lineal está configurada tal como se muestra en la figura 29.
La figura 47 muestra una configuración modificada en la cual la unidad 25 de función de clave está suprimida de la unidad F de función no lineal y en la cual una clave k' de extensión transformada de forma lineal es suministrada al circuito XOR 298, cuando la unidad F de función no lineal está configurada como se muestra en la figura 30.

Mejor modo de llevar a cabo la invención

Realización 1.

La figura 1 muestra una unidad de transformación de datos para el cifrado 100 y una unidad de transformación de datos para el descifrado 400 en esta realización.

La unidad de transformación de datos para el cifrado 100 es, por ejemplo, un dispositivo de cifrado que emite de salida textos cifrados de 128 bits a partir de textos no cifrados de entrada de 128 bits. La unidad de transformación de datos para el descifrado 400 es un dispositivo de descifrado que emite de salida textos no cifrados de 128 bits a partir de textos cifrados de entrada de 128 bits. La unidad de transformación de datos para el cifrado 100 cosiste en una unidad 200 de cifrado y en una unidad 300 de generación de claves. La unidad 200 de cifrado es una unidad de procesamiento de datos para el cifrado de textos no cifrados. La unidad 300 de generación de claves genera múltiples (n) claves de extensión de 64 bits o de 128 bits utilizando constantes Vi a partir de datos de clave 40 de

entrada de 128 bits, 192 bits o 256 bits, y los suministra a la unidad 200 de cifrado. La unidad 400 de transformación de datos para el descifrado se compone de una unidad 500 de descifrado y de una unidad 600 de generación de claves. La unidad 500 de descifrado es una unidad de procesamiento de datos para descifrar textos cifrados. La unidad 600 de generación de claves es la misma o similar a la anterior unidad 300 de generación de claves. Así mismo, dado que la unidad 200 de cifrado y la unidad 500 de descifrado pueden ejecutar el mismo procedimiento, pueden compartir un circuito o un programa, aunque la unidad 200 de cifrado y la unidad 500 de descifrado se ilustran separadamente en las figuras. De modo similar, las unidades 300 y 600 de generación de claves pueden compartir un circuito o un programa. Esto es, un circuito, o un programa, puede compartirse mediante la unidad de transformación de datos para el cifrado 100 y por la unidad de transformación de datos para el descifrado 400.

La figura 2 muestra los significados de las notaciones utilizadas para las siguientes figuras o descripciones.

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

La figura 3 muestra un ejemplo de la unidad 200 de cifrado o de la unidad 500 de descifrado.

La figura 3 muestra una configuración en la cual están conectadas en cascada la unidad 210 de transformación no lineal de datos de 6 rondas, la unidad 220 de transformación no lineal de datos de 6 rondas y la unidad 230 de transformación no lineal de datos 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 210 de transformación no lineal de datos de 6 rondas y la unidad 220 de transformación no lineal de datos de 6 rondas. Además, 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 220 de transformación no lineal de datos de 6 rondas y la unidad 230 de transformación no lineal de datos de 6 rondas. Dentro de la unidad 210 de transformación no lineal de datos de 6 rondas, se disponen 6 rondas de unidades de transformación no lineal de datos. Por ejemplo, una unidad 280 de transformación no lineal de datos está compuesta por una unidad F de función no lineal y por un circuito XOR (O exclusivo) 290. De esta forma, en el caso de la figura 3, en total se suministran 18 rondas de unidades de transformación no lineal de datos.

La unidad 210 de transformación no lineal de datos está equipada con una primera unidad 280 de transformación no lineal de datos y con una segunda unidad 281 de transformación no lineal de datos. 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 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 los datos de entrada de la izquierda L_o como primeros datos intermedios de la derecha R_1. El último lleva a cabo la segunda transformación no lineal sobre los primeros datos intermedios de la izquierda L_1 utilizando una segunda clave de extensión k_2, emite 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 los primeros datos intermedios de la izquierda L_1 como segundos datos intermedios de la derecha R_2. La unidad 210 de transformación no lineal de datos, en la cual están conectadas en cascada desde la primera unidad 280 de transformación no lineal de datos a la sexta unidad 285 de transformación no lineal de datos, emite 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 figura 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 240 de transformación no lineal de datos de 6 rondas se añaden a la unidad 200 de cifrado mostrada en la figura 3. En total, la transformación de datos se lleva a cabo mediante 24 rondas de unidades de transformación no lineal de datos.

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

La figura 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 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 55 de desplazamiento rotacional a la izquierda 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 57 de extensión 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 figura 6 muestra la unidad de transformación inversa de datos (FL^{-1}) 271.

La figura 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 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 78 de desplazamiento rotacional 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 figura 5 y la unidad de transformación inversa de datos (FL^{-1}) 271 mostrada en la figura 6 llevan a cabo operaciones opuestas entre sí. Por consiguiente, utilizando la misma clave de extensión, los datos de entrada X de la figura 5 pueden obtenerse como datos de salida X de la figura 6 haciendo que los datos de salida Y de la figura 5 sean los datos de entrada Y de la figura 6.

La relación en la cual los datos de entrada a una unidad pueden obtenerse como datos de salida procedentes de la otra unidad, haciendo que los datos de salida procedentes de la primera unidad sean los datos de entrada a 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 55 de desplazamiento rotacional hacia la izquierda de 1 bit de la figura 5 como la unidad 78 de desplazamiento rotacional hacia la izquierda de 1 bit de la figura 6 llevan a cabo el desplazamiento hacia la izquierda, pero ambas pueden ejecutar el desplazamiento a la derecha. Además, la unidad de transformación normal de datos (FL) 251 y la unidad de transformación inversa de datos (FL^{-1}) 271 pueden ser una entre otras 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 "no", un circuito OR con una operación "no", y/o un circuito XOR con una operación "no". A saber, lo que sigue son las definiciones mostradas del circuito AND con una operación "no", el circuito OR con una operación "no", y el circuito XOR con una operación "no", representados por "andn", "orn" y "xorn", respectivamente.

x andn y : (no x) e y
x orn y : (no x) o y
x xorn y : (no x) e y

Algunas CPUs modernas están provistas de comandos "and", "or" y "xor" que incluyen "no". Estos comandos pueden ejecutarse con el mismo coste que los comandos "and", "or" y "xor".

La figura 7 muestra una unidad 201 de cifrado convencional y una unidad 501 de descifrado convencional.

La unidad 201 de cifrado convencional está equipada con dos unidades normales de transformación de datos FL. Así, la unidad de descifrado 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 cifrado por regla general tiene una configuración distinta que la unidad de descifrado, la unidad de cifrado y la unidad de descifrado no pueden compartir el mismo circuito.

Por otro lado, como se muestra en la figura 8, en la presente 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 200 de cifrado, de forma que la unidad de descifrado con la misma configuración pueda llevar a cabo el descifrado. 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.

Tal como se describió anteriormente, la unidad 200 de cifrado y la unidad 500 de descifrado pueden ser implementadas mediante la misma configuración, y la unidad 200 de cifrado y la unidad 500 de descifrado pueden compartir el circuito.

La figura 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 puntual sobre la unidad 280 de transformación no lineal de datos.

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 puntual sobre la unidad 280 de transformación no lineal de datos, el cifrado y el descifrado pueden llevarse a cabo utilizando la misma configuración.

La figura 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 puntual.

Tal como se muestra en la figura 10, en el caso de la figura 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 puntual sobre la unidad 220 de transformación no lineal de datos de 6 rondas.

En las figuras 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 figuras 3, 4, 8 y 9, los datos de la derecha R y los datos de la izquierda L pueden ser sustituidos entre sí.

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

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

En este caso, una unidad 211 de transformación normal/inversa de datos está compuesta por la unidad 210 de transformación no lineal de datos de 6 rondas, 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 el cifrado como para el descifrado. A saber, 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 los datos de salida procedentes de la otra unidad haciendo que los datos de salida procedentes de la unidad sean los datos de entrada a la otra unidad.

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

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

La unidad 200 de cifrado está configurada mediante la conexión en cascada de estas unidades 221, 221 y 231 normales/inversas de transformación de datos. Y esta unidad 200 de cifrado puede también ser utilizada como unidad 500 de descifrado.

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

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

Las unidades 1211 y 1221 de transformación normal/inversa de datos pueden utilizarse para la unidad de descifrado.

Además, si se supone que un conjunto de las unidades 210 a 230 de transformación no lineal de datos de 6 rondas sea una unidad 2210 de transformación no lineal de datos, la unidad 2210 de transformación no lineal de datos es un circuito que puede utilizarse tanto para el cifrado como para el descifrado.

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

La unidad 2211 de transformación normal/inversa de datos puede ser utilizada para la unidad de descifrado.

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

Además, en la unidad 200 de cifrado o en la unidad 500 de descifrado, 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 figura 37 muestra un caso en el que la unidad 200 de cifrado y la unidad de descifrado tienen la misma configuración, que incluye la unidad 210 de transformación no lineal de datos de 6 rondas.

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

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

La unidad 210 de transformación no lineal de datos introduce los datos B1 y emite los datos B. Además, la unidad 210 de transformación no lineal introduce los datos A_1' y emite los datos A'. La segunda unidad 279 de salida de transformación inversa de datos introduce los datos A' y emite los datos A.

En la figura 38, la unidad 219 de transformación no lineal de datos de ronda impar incluye rondas impares de unidades 280 de transformación no lineal de datos. En consecuencia, los datos A' ingresados desde el primer puerto 261 de entrada son emitidos desde el segundo puerto 264 de salida como datos A_1'. A continuación, los datos A_1' son transformados por la segunda unidad 279 de transferencia inversa de datos de salida y emitidos como datos A_1''. Además, los datos B introducidos en el segundo puerto 262 de entrada son emitidos desde el primer puerto 263 de salida como datos B_1.

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

En los casos de las figuras 37 y 38, la unidad 200 de cifrado y la unidad 500 de descifrado tienen la misma configuración, llevando a cabo el cifrado y el descifrado.

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

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

La figura 41 muestra un caso en el que las unidades 256, 258 de transformación normal/inversa de datos se proporcionan en los puertos 261, 263 de entrada/salida de la izquierda y las unidades 277, 278 de transformación inversa de datos se proporcionan en los puertos 262, 264 de entrada/salida de la derecha.

La figura 42 muestra un caso en el que las figuras 39 y 40 están combinadas.

Puede implementarse otro caso mediante la combinación de las figuras 37 y 39, que no se muestra en la figura. Además, las figuras 38 y 39 pueden combinarse. Además, la unidad 210 de transformación no lineal de datos de 6 rondas (rondas pares) puede sustituirse por la unidad 219 de transformación no lineal de datos de rondas impares en las figuras 37 y 39 a 42, que no se muestran en las figuras. En los casos de las figuras 39 a 42, la unidad de cifrado y la unidad de descifrado pueden ser implementadas mediante la misma configuración.

Realización 2.

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

La unidad F de función no lineal introduce datos 10 de entrada de la función F, lleva a cabo la transformación no lineal y emite datos 40 de salida de la función F. Los datos 10 de entrada de la función F de 64 bits se dividen en ocho elementos de datos, y se procesan 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 11 de extensión, y se lleva a cabo una transformación no lineal utilizando una sustitución en una unidad 20 de función S. A continuación, en una unidad 30 de función P, dos elementos de los datos de 8 bits son operados con la función XOR mediante dieciséis circuitos XOR 815, y son emitidos los datos 40 de salida de la función F de 64 bits. En la unidad 20 de función S, se proporcionan cuatro primeras unidades 13 de transformación de cajas S y cuatro segundas unidades 14 de transformación de cajas S.

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

Dentro de la primera unidad 13 de transformación de cajas S, se proporciona una tabla de transformación T. La tabla de transformación T almacena previamente valores de 0 a 255 de manera arbitraria (al azar) correspondientes a valores de 0 a 255. La tabla de transformación T introduce los valores de 0 a 255 y emite el valor (valor de 0 a 255) correspondiente a cada valor. Por ejemplo, cuando se introduce 1, la tabla de transformación T emite 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 máxima es suficientemente pequeña o no, etc.

La segunda unidad 14 de transformación de cajas S incluye la primera unidad 13 de transformación de cajas S y una unidad 22 de desplazamiento rotacional 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 22 de desplazamiento rotacional a la izquierda de 1 bit lleva a cabo el desplazamiento rotacional a la izquierda en 1 bit para una salida procedente de la primera unidad 13 de transformación de cajas S. Por ejemplo, cuando se introduce 1, la primera unidad 13 de transformación de cajas S emite 7 y la unidad 22 de desplazamiento rotacional a la izquierda de 1 bit emite 14.

Si la primera unidad 13 de transformación de cajas S y la segunda unidad 14 de transformación de cajas S están configuradas como se muestra en la figura 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.

Además, tal como se muestra en la figura 27, mediante la provisión de una unidad 15 de desplazamiento rotacional a la derecha de 1 bit (">>>1" de la tercera unidad 15 de transformación de cajas S de la figura 27) así como, o en lugar de, la unidad 22 de desplazamiento rotacional a la izquierda de 1 bit, puede obtenerse un efecto similar en un caso en el que se proporcione también una tabla T de transformación diferente. De otra forma, también es posible transformar los datos y de entrada utilizando la tabla de transformación de datos T después del desplazamiento de los datos y de entrada por la unidad de desplazamiento rotacional a la izquierda de 1 bit ("<<<1" de la cuarta unidad 16 de transformación de cajas S de la figura 27) proporcionada para los datos y de entrada. La figura 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 y 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 solamente un comando de desplazamiento de 1 bit. Además, cuando este proceso de desplazamiento se lleve a cabo mediante hardware que lleve a cabo un desplazamiento de 1 solo bit, a veces es posible funcionar más rápido. Además, el desplazamiento no está limitado a llevarse a cabo a razón de 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 figura 28 muestra una unidad 20 de función S que utiliza las unidades 13, 14, 15, 16 primera a cuarta de transformación de cuatro cajas S, mostradas en la figura 27.

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

A partir de los datos de entrada de 8 bits y_1, y_2, y_3, y_4, se obtienen datos Z_1, Z_2, Z_3, Z_4, de 32 bits mediante referencia a S_1, S_2, S_3, S_4, 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 S_2, S_3, S_4, S_1, respectivamente, y son operados con la función XOR en un circuito 916. Este resultado U_2 operado con la función XOR y el anterior resultado U_1 operado con la función XOR son operados con la función XOR en un circuito 917 para emitir 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 octeto (en la figura 31, "<<<1" representa un desplazamiento rotacional de 1 octeto, 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 Z_5', Z_6', Z_7', Z_8'.

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

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

La figura 32 muestra la unidad de función P cuya configuración es equivalente a la figura 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 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 y SC, y son operados con la función XOR en un circuito 936 para emitir un resultado operativo B. El resultado operativo B es desplazado rotacionalmente a la derecha en 1 octeto (en la figura 32, de manera similar a la figura 31, el desplazamiento es llevado a cabo en una unidad de longitud de bits (8 bits = 1 octeto) que es procesada 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 octeto 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 la parte superior (izquierda) en 2 octetos 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 octeto 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 emitida como z_1', z_2', z_3', z_4', y la salida procedente del circuito 943 es emitida como z_5', z_6', z_7', z_8'.

S5 y SC se configuran utilizando la primera unidad 13 de transformación de caja S y un desplazamiento lógico, S6 y S9 se configuran utilizando la segunda unidad 14 de transformación de caja S y un desplazamiento lógico, S7 y SA se configuran utilizando la tercera unidad 15 de transformación de caja S y un desplazamiento lógico, S8 y SB se configuran utilizando la cuarta unidad 16 de transformación de caja S y un desplazamiento lógico. El desplazamiento lógico se utiliza para emitir datos de salida de 8 bits procedentes de cada unidad de transformación hasta una ubicación predeterminada situada dentro de los datos de salida de 32 bits. El desplazamiento lógico se fija para desplazar hacia la izquierda en 0 octetos en S5 y SA, en 1 octeto en S6 y SB, en 2 octetos en S7 y SC y en 3 octetos en S8 y S9. A saber, 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, y [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 figuras 31 y 32, puede proporcionarse 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 figura 26, y además, sobre el cual resulte posible una implementación flexible.

En la figura 11, cuando las cajas S de la unidad 20 de función S 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 figura 12, el uso de la memoria requerida para el almacenaje de las tablas de transformación T puede reducirse a al menos la mitad.

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

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

La configuración concreta se explica con detalle en Matui y Sakurai, "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 21 de transformación de caja S y se emiten los datos de 8 bits. La unidad 21 de transformación de caja S se configura mediante una unidad 17 de transformación lineal de N bits (aquí, N = 8), una unidad 18 de transformación de subcampo y una unidad 19 de transformación lineal de N bits. La unidad 17 de transformación lineal de N bits lleva a cabo operaciones de datos de 8 bits. La unidad 18 de transformación de subcampo lleva a cabo operaciones de datos de solo 4 bits que son elementos del Campo de Galois, GF (24). La unidad 19 de transformación lineal de N bits lleva a cabo una operación de datos de 8 bits. Una unidad 85 de transformación lineal de la unidad 17 de transformación lineal de N bits es un circuito que lleva a cabo la transformación lineal mostrada en la figura 14. Una unidad 87 de transformación lineal es un circuito que lleva a cabo la transformación lineal mostrada en la figura 15.

La unidad 85 de transformación lineal 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 87 de transformación lineal puede ser sustituida por un circuito que lleve a cabo otra transformación afín. La unidad 85 de transformación afín transforma los datos de 8 bits (X) en datos de 8 bits (X'). Se supone que los datos de 8 bits obtenidos (X') son elementos del Campo de Galois (28). Se supone que los datos de los 4 bits superiores y los datos de los 4 bits inferiores (X_1 y X_0) de los datos X' son, respectivamente, elementos de Campo de Galois de subcampo (2^4) y son emitidos hacia la unidad 18 de transformación de subcampo. Aquí, por ejemplo, sea un elemento β del CG (2^8) un elemento que satisface el polinomio irreductible $X^8 + X^6 + X^5 + X^3 + 1 = 0$, y $a = \beta^{238}$, una base del CG (2^4) de subcampo, que 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 siguiente puede establecerse como $X' = X_O + \beta X_1$. (Para los detalles, consúltese Matui y Sakurai, ["Galois Field division circuit and shared circuit for multiplication and division"] (Registro de Patente japonesa Nº 2641285 [2 de Mayo de 1997])). La unidad 18 de transformación de subcampo se configura solo por unidades operativas, cada una de las cuales lleva a cabo operaciones de datos de 4 bits.

En este caso, como ejemplo de extracción de "subcampo", 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 18 de transformación de subcampo es un circuito de elementos inversos que utiliza el subcampo construido por el circuito mostrado en el documento "Galois Field division circuit and shared circuit for multiplication and division" (Registro de Patente Nº 2641285 [2 de Mayo de 1997]). Como resultado operativo de este circuito de elementos inversos, los datos de los 4 bits superiores y los datos de los 4 bits inferiores (Y_1 e Y_0), cada uno de los cuales puede considerarse como un elemento del CG (2^4), son emitidos hacia la unidad 87 de transformación lineal como datos Y de 8 bits, los cuales pueden considerarse como un elemento del CG (2^8), donde $Y = Y_0 + \beta Y_1$. Tal como se expuso anteriormente, este circuito de elementos inversos es un circuito para calcular $Y = Y_0 + \beta Y_1 = 1/(X_o + \beta X_1)$. Además, 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 21 de transformación de caja S mostrada en la figura 13 es calcular datos con un ancho de bits (4 bits) que es la mitad del ancho de bits (8 bits) de la entrada de datos para la transformación no lineal. A saber, el circuito de elementos inversos se caracteriza por llevar a cabo operaciones de datos de solo 4 bits.

Sin embargo, la velocidad de cálculo puede reducirse llevando a cabo operaciones de solo 4 bits. Este supuesto ofrece la ventaja de que una 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.

Además, una segunda característica de la unidad 21 de transformación de caja S consiste en que la unidad 17 de transformación lineal de N bits y la unidad 19 de transformación lineal de N bits, donde N = 8, se proporcionan a ambos lados de la unidad 18 de transformación de subcampo. Cuando la unidad 21 de transformación de caja S es implementada utilizando la unidad 18 de transformación de subcampo, existe la ventaja de que puede reducirse una escala del circuito entero, 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. En consecuencia, las transformaciones lineales o las transformaciones afines se llevan a cabo a ambos lados de la unidad 18 de transformación de subcampo, de forma que pueda recuperarse la reducción del nivel de seguridad debido a la implementación que utiliza la unidad 18 de transformación de subcampo.

En la figura 13, las transformaciones lineales se llevan a cabo a ambos lados de la unidad 18 de transformación de subcampo; 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 figura 29 muestra un supuesto en el que la unidad 25 de función de clave se muestra en la figura 11, esto es, la unidad 25 de función de clave situada antes de la unidad 20 de función S y de la unidad 30 de función P, ahora se sitúa después de la unidad 20 de función S y de la unidad 30 de función P.

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

Mediante el empleo de la configuración mostrada en la figura 29 o la figura 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 figura 11. Además, mediante la modificación de la generación de las claves de extensión, puede obtenerse la misma salida utilizando la configuración mostrada en la figura 29 o en la figura 30 a partir de la misma entrada que la de la configuración de la figura 11. En la unidad de función F convencional mostrada en la figura 26, se proporcionan 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 figura 29, una unidad 25 de función de clave está situada en la etapa final de la función F. En el caso mostrado en la figura 30, la unidad 25 de función de clave está situada entre la unidad 20 de función S y la unidad 30 de función P.

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

Los datos de la izquierda son introducidos en la unidad F de transformación no lineal como datos 10 de entrada de la función F, y se emiten los datos 40 de salida de la función F. Los datos 40 de salida 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 F de transformación lineal como los datos 10 de entrada 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 figura 43, las operaciones de la unidad 25 de la función de clave, de la unidad 20 de función S y de la unidad 30 de función P se llevan a cabo en la unidad F de transformación no lineal, de forma que la carga de las operaciones resulta considerable dentro de la unidad F de transformación no lineal. Un caso ejemplar en el que puede conseguirse una velocidad de procesamiento más alta mediante la distribución de la carga operativa de la unidad F de transformación no lineal, se expondrá a continuación con referencia a las figuras.

La figura 44 muestra un supuesto en el que se utiliza la unidad F" de transformación no lineal. La unidad F" de transformación no lineal es una unidad en la que la unidad 25 de la función de claves es suprimida de la unidad F de transformación no lineal mostrada en la figura 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. Además, 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 F" de transformación no lineal como los datos 10 de entrada de la función F, y transformados por la unidad 20 de función S y por la unidad 30 de función P. La salida procedente del circuito XOR 297 y los datos 40 de salida de la función F son operados con la función XOR en un circuito XOR 290 para emitir los datos de la izquierda L_1.

Por otro lado, las unidades 300, 600 de generación de claves llevan a cabo una operación XOR de las claves de extensión k_1 y k_2 y emiten la clave de extensión modificada $k_1 + k_9$. 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 los datos de la derecha. Las unidades 300, 600 de generación de claves modifican las claves de extensión para generar y emitir $k_1 + k_3$, $k_2 + k_4$, $k_3 + k_5$, …, $k_{16} + k_{18}$. Las unidades 300, 600 de generación de claves suministran las claves de extensión modificadas a los procesos distintos del proceso (F) de función no lineal 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 figura 43.

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

La figura 45 muestra un caso en el que las operaciones de la clave de extensión de blanqueo kw_1, se llevan a cabo asimismo como operaciones de las otras claves de extensión en la configuración mostrada en la figura 44. La figura 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, kw1superior y de la primera clave de extensión k_1 (a saber, 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 $kw_{1inferior}$ y de la segunda clave de extensión k_2 (a

saber, 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 figura 44. Además, en un caso mostrado en la figura 45, la unidad de generación de claves lleva a cabo una operación XOR de una parte de la clave de extensión de blanqueo $kw_{2inferior}$ y de la clave de extensión k_{17} (a saber, la unidad de generación de claves modifica la clave de extensión) y suministra el resultado de la operación al circuito XOR 299. Y además, la unidad de generación de claves lleva a cabo una operación XOR de la otra parte de la clave de extensión de blanqueo $kw_{2superior}$ y de la clave de extensión k_{18} (a saber, 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 figura 44.

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

La figura 47 muestra un caso en el que la unidad 25 de función de claves es suprimida de la unidad F de función no lineal 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 F de función no lineal está configurada como se muestra en la figura 30. En el caso de la figura 47, se lleva a cabo la misma operación efectuada por el proceso de función P sobre los datos de clave para generar datos de clave transformados de manera no lineal, y los datos de 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 de clave para datos de procesamiento. En ambos casos de las figuras 46 y 47, debido a que la unidad 25 de función de clave es eliminada de la unidad F de función no lineal, la carga operativa de la unidad F de función no lineal se reduce, y la operación del circuito XOR 298 situado fuera de la unidad F de función no lineal puede llevarse a cabo en paralelo con las operaciones llevadas a cabo por la unidad F de función no lineal, lo que posibilita un procesamiento de alta velocidad.

Realización 3.

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

La unidad 300 de generación de claves incluye una unidad 310 de transformación de la longitud de los bits, una primera unidad 320 de transformación de claves de G bits, una segunda unidad 330 de transformación de claves de G bits y una unidad 340 de desplazamiento de claves. A partir de los datos con clave de entrada con 128 bits, 192 bits o 256 bits, la unidad 300 de generación de claves genera datos de clave k_1 de 128 bits y datos de clave k_2 de 128 bits, y emite varias claves de extensión de 64 bits. La unidad 310 de transformación de la longitud de los bits convierte la longitud de los bits de los datos de clave que van a ser emitidos de forma que la longitud de bits de los datos de clave de salida quede fijada incluso si son introducidos datos de clave con diferente número de bits. En otras palabras, la unidad 310 de transformación de longitud de los bits genera datos de clave $SK_{superior}$ de los 128 bits superiores y datos de clave $SK_{inferior}$ de los 128 bits inferiores, y emite los primeros hasta la primera unidad 320 de transformación de claves de G bits y hasta la unidad 340 de desplazamiento de claves. Además, los últimos son emitidos hacia la segunda unidad 330 de transformación de claves de G bits y hacia la unidad 340 de desplazamiento de claves. Además, los datos de clave de 128 bits que son un resultado operado con la función XOR de los primeros y de los últimos son emitidos hacia la primera unidad 320 de transformación de claves de G bits.

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

Cuando los datos de clave de 128 bits son introducidos en la unidad 310 de transformación de la longitud de los bits, los datos de clave introducidos son emitidos como datos de clave $SK_{superior}$ de los 128 bits superiores sin cambio alguno. Además, los datos de clave $SK_{inferior}$ de los 128 bits inferiores son fijados en 0 y emitidos.

Cuando los datos de clave de 192 bits son introducidos en la unidad 310 de transformación de la longitud de los bits, los datos superiores de 128 bits de los datos de clave de entrada son emitidos como los datos de clave superiores de 128 bits $SK_{superior}$ sin cambio alguno. Además, los datos de clave inferiores de 128 bits $SK_{inferior}$ son generados mediante la combinación de los 64 bits inferiores de los datos de clave de 192 bits de entrada 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 de clave de 192 bits de entrada, y emitidos.

Cuando los datos de clave de 256 bits son introducidos, los datos superiores de 128 bits de los datos de clave introducidos son emitidos como $SK_{superior}$, y los datos inferiores de 128 bits son emitidos como $SK_{inferior}$.

Unos datos XOR de los datos de clave de 128 bits $SK_{superior}$ y $SK_{inferior}$ son introducidos en la primera unidad 320 de transformación de claves de G bits desde la unidad 310 de transformación de la longitud de los bits, operados mediante transformaciones no lineales de dos rondas, operados con la función XOR con los datos de clave superiores de 128 bits $SK_{superior}$, adicionalmente operados por unas transformaciones no lineales de dos rondas, y se

emiten los datos de clave de 128 bits K_1.

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

La unidad 340 de desplazamiento de claves 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 345 de control de desplazamiento. La unidad 345 de control de desplazamiento emite una señal 346 de selección hacia cada uno de los registros de desplazamiento para controlar las operaciones de los registros de desplazamiento.

La figura 18 muestra una configuración del registro de desplazamiento A341.

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 346 de selección incluye una señal de conmutación para indicar la conexión de 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 346 de selección y, en este caso, el registro A 348 lleva a cabo un desplazamiento rotacional hacia la izquierda de 17 bits. Además, cuando el grupo de conmutadores está conectado con B, el registro A lleva a cabo el desplazamiento rotacional a la izquierda de 15 bits. El desplazamiento de 15 bits o el desplazamiento de 17 bits se llevan a cabo mediante un ciclo de reloj.

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

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

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 relojes tercero y cuarto, se especifica un desplazamiento de 15 bits. En cada uno de los relojes quinto a octavo se especifica un desplazamiento de 17 bits.

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

Los datos de clave superiores de 128 bits $SK_{superior}$ introducidos desde la unidad 310 de transformación de la longitud de los bits son insertados en el registro de desplazamiento A 341. Los datos de clave de 128 bits K_1 emitidos desde la primera unidad 320 de transformación de claves de G bits son insertados en el registro de desplazamiento B 342. Con esta condición, el registro de desplazamiento A 341 y el registro de desplazamiento B 342 operan en base a la tabla de control mostrada en la figura 19. En la figura 20, los datos de una columna inclinada muestran que deben ser ignorados y no deben ser emitidos. Los datos en las otras columnas son emitidos como claves de extensión, como se muestra en la figura 21.

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

La figura 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 de emitir o no los 64 bits superiores y los 64 bits inferiores desde el registro de desplazamiento A 341 y el registro de desplazamiento B 342 como la clave de extensión, y su orden de emisión, se especifican en la tabla de control, que no se muestra en la figura. Y según la tabla de control, mediante la emisión de la señal 346 de selección que incluye una señal de instrucción de salida para el registro de desplazamiento, la clave de extensión es emitida desde cada registro de desplazamiento en 64 bits.

La figura 22 muestra un caso en el que la clave de extensión es generada a partir de unos datos de clave de 192 bits o de 256 bits.

A saber, los datos de clave superiores de 128 bits $SK_{superior}$ introducidos desde la unidad 310 de transformación de la longitud de los bits son insertados en el registro de desplazamiento A 341, los datos de clave inferiores de 128 bits $SK_{inferior}$ se insertan en el registro de desplazamiento B 342, los datos de clave de 128 bits K_1 emitidos desde la primera unidad 320 de transformación de claves de G bits se insertan en el registro de desplazamiento C 343, y los datos de clave de 128 bits K2 emitidos desde la segunda unidad 330 de transformación de claves de G bits se insertan en el registro de desplazamiento D 344.

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

La figura 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 figura 23 son almacenadas en la tabla de control situada en el controlador.

Como se muestra en la figura 19, la unidad 345 de control de desplazamiento almacena el número de bits para el desplazamiento de los datos de clave fijados en el registro de desplazamiento A 341. A saber, las claves de extensión son generadas de forma secuencial mediante el desplazamiento de los datos de 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 y Z_7 = 128 bits, tal 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. Además, en el caso de que se lleve a cabo una transformación inversa (descifrado), el proceso para generar la clave de extensión se inicia desde el estado inicial y, en consecuencia, tanto la transformación (cifrado) como la transformación inversa (descifrado) pueden llevarse a cabo mediante la fijación del estado inicial. Además, 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 impedir la generación de valores idénticos a los del estatus dentro del mismo registro de desplazamiento, debido a la ejecución del desplazamiento en más de un ciclo (más 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 de modo tal que, al realizar los desplazamientos del registro en un ciclo, el número de bits de desplazamiento varíe 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.

Establézcase la relación para que $Z_1 - Z_0 = 15$ (aquí, $Z_1 - Z_0 = B_1$), $Z_2 - Z_1 = 30$ (a saber, $Z_2 - Z_1 = 2B_1$), por consiguiente, $Z_2 - Z_1 = 2 (Z_1 - Z_0)$. Además, como se muestra en la tabla de control del registro de desplazamiento B, establézcase la relación para que $Z_5 - Z_4 = 34$ (aquí, $Z_5 - Z_4 = 2B_2$), $Z_6 - Z_5 = 17$ (a saber, $Z_6 - Z_5 = B_2$), por consiguiente, $Z_5 - Z_4 = 2 (Z_6 - Z_5)$. A saber, las diferencias entre los números de los bits de desplazamiento nos dan 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 entero (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, bien respecto del número de bits de desplazamiento para una sola vez, o bien para el múltiplo del número entero mayor que 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 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.

En este caso, al desplazar I veces (dos veces) tanto los datos superiores como los datos inferiores 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 superiores como los datos inferiores 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 figura 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, de los datos superiores y los datos inferiores 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 figura 20, la cual muestra que la clave [12] y la clave [13] son emitidas como las claves de extensión.

Las razones por las cuales las operaciones de desplazamiento en base a un múltiplo de un entero mayor de dos se emplean de acuerdo con lo anteriormente descrito, son la habilitación para realizar el desplazamiento de no solo 15 bits o 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 adicionalmente 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 mejorar la seguridad.

Se desea generar los datos que no son utilizados para la clave de extensión (en las figuras 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 figura 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, tales como, por ejemplo, una llamada de función por un programa, una llamada de subrutina, o un proceso de tratamiento de interrupciones.

La característica de la tabla de control mostrada en la figura 19 es que la tabla de control especifica el número de bits de desplazamiento de $B_1 = 8 \times 2 - 1 = 15$ ($B_1 = 8 \times J_1 - 1$, donde J_1 es un número entero mayor que 1) y el número de bits de desplazamiento de $B_2 = 8 \times 2 + 1 = 17$ ($B_2 = 8 \times J_2 + 1$, donde J_2 es un número entero mayor que 0, $J_1 = J_2$ o $J_1 \neq J_2$). Establecer la magnitud del desplazamiento hasta un ± 1 del múltiplo entero de ocho es llevar a cabo el desplazamiento por los bits impares, lo que mejora la seguridad en comparación con la realización del desplazamiento sólo por los bits pares, dado que la operación de la clave de extensión en 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 magnitud del desplazamiento puede establecerse mediante la suma/resta de 1 bit a/del múltiplo de ocho, por ejemplo, en alguna CPU que tenga solamente un comando de desplazamiento de 1 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 además, en el caso de que esta operación de desplazamiento que utiliza el hardware que puede desplazar solamente 1 bit, hay casos en que es posible llevar a cabo un procesamiento de alta velocidad.

En la descripción anterior de la unidad 310 de longitud de bits, ingresan tres tipos de anchos de bits de datos de clave. Incluso cuando los datos de 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 310 de transformación de longitud de bits puede extender los datos de clave al mismo tamaño de los datos de clave cuando son introducidos los datos de clave de 256 bits, utilizando algún tipo de algoritmo. A saber, cuando son introducidos los datos de clave con una longitud de Q, la cual oscila entre G bits y 2G bits, la unidad 310 de transformación de la longitud en bits puede convertir los datos de clave de Q bits en datos de clave de 2G bits.

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

En la explicación que sigue de la figura 34, "+" indica una operación XOR. Aquí, se supone que se introducen dos datos de clave de 128 bits SK1 y SK2 (SK1 f SK2) y que la unidad 310 de transformación de la longitud de los bits emite $SK1_{superior}$ = SK1 = (SKH1 | SKL1) a partir de SK1 y $SK2_{superior}$ = SK2 = (SKH2 | SKL2) a partir de SK2. Aquí, SKHi (i = 1, 2) indica los datos superiores de 64 bits de SKi y SKLi (i = 1, 2) indica 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 $\Delta A \neq 0$ o $\Delta B \neq 0$", dado que SK1 ≠ SK2.

Como se muestra en la figura 34, estos ΔA y ΔB se convierten en $\Delta A + \Delta D$, $\Delta B + \Delta C$, respectivamente, mediante la recepción de las dos rondas de transformaciones no lineales. Esto significa que los datos XOR ($\Delta A \mid \Delta B$) de $SK1_{superior}$ y $SK2_{superior}$ se convierten en datos XOR ($\Delta A + \Delta D \mid \Delta B + \Delta C$) después de llevar a cabo las dos rondas de transformaciones no lineales en $SK1_{superior}$ y los datos transformados después de llevar a cabo las dos rondas de transformaciones no lineales en $SK2_{superior}$. Por consiguiente, cuando estos elementos de datos, después de llevar a cabo las dos rondas de transformaciones no lineales, son operados con la función XOR con $SK1_{superior}$ y $SK2_{superior}$, 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 $\Delta Y \neq 0$, 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 los mismos datos a partir de $SK1_{superior}$ y $SK2_{superior}$ mediante las dos rondas de transformaciones no lineales, se demuestra la no existencia de la clave equivalente.

Por otro lado, tal como se muestra en la figura 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 $SK1_{superior}$ y $SK2_{superior}$ 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 $SK1_{superior}$ y los datos transformados después de llevar a cabo las tres rondas de transformaciones no

lineales en $SK2_{superior}$. En consecuencia, cuando estos elementos de datos, después de recibir las tres rondas de transformaciones no lineales, son operados con la función XOR con $SK1_{superior}$ y $SK2_{superior}$, 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 | 0). A saber, cuando estos elementos de datos, después de llevar a cabo las tres rondas de transformaciones no lineales, son operados con la función XOR con $SK1_{superior}$ y $SK2_{superior}$, respectivamente, los resultados de la operación son los mismos. A saber $SK1_{superior}$ y $SK2_{superior}$ emiten los mismos datos, de forma que existen las claves equivalentes, lo que es preocupante con respecto a la seguridad.

No solo el caso expuesto anteriormente de transformación lineal de tres rondas, sino que una transformación no lineal general puede emitir 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 a la presente realización.

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

La figura 24 muestra un ordenador para la instalación de la unidad 100 de transformación de datos para el cifrado o la unidad 400 de transformación de datos para el descifrado.

La unidad 100 de transformación de datos para el cifrado y/o la unidad 400 de transformación de datos para el descifrado están conectadas al bus como tarjeta de circuitos impresos. Esta tarjeta de circuitos impresos está provista de una CPU, una memoria y un elemento de circuito lógico, y cifra 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 bien descifra los textos cifrados suministrados a partir de la CPU y devuelve los textos no cifrados a la CPU.

De esta forma, la unidad 100 de transformación de datos para el cifrado o la unidad 400 de transformación de datos para el descifrado pueden ser implementadas por hardware. Además, la unidad 100 de transformación de datos para el cifrado o la unidad 400 de transformación de datos para el descifrado pueden ser también implementadas mediante software, como procedimiento de transformación de datos. A saber, la operación expuesta anteriormente 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 anterior puede ser implementada mediante la combinación de hardware y software, aunque esto no se muestra en la figura. Además, no se requiere implementar toda la operación anterior utilizando un ordenador, sino que es posible implementar la operación referida mediante un sistema distribuido, como por ejemplo un servidor y un cliente, o un ordenador central y un ordenador terminal, aunque esto no se muestra en la figura.

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

En las realizaciones 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 bien textos no cifrados y textos cifrados con otro número de bits.

Además, en las realizaciones anteriores, se ha expuesto un caso en el que se utilizan datos de clave de 128 bits, 192 bits, 256 bits y claves de extensión de 64 bits, pero los datos de clave pueden tener otro número de bits.

Si la longitud de bits de los textos no cifrados y de los textos cifrados, los datos de 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.

Disponibilidad Industrial

Según la realización de la presente invención, se proporcionan la unidad de transformación normal de datos (FL) 251 y la unidad de transformación inversa de datos (FL^{-1}) para implementar el cifrado y el descifrado, usando el mismo algoritmo, de modo tal que la unidad 200 de cifrado y la unidad 500 de descifrado puedan compartir el circuito.

Además, según la realización de la presente invención, la tabla T de transformación es compartida por la primera unidad 13 de transformación de caja S y la segunda unidad 14 de transformación de caja S, por lo que se simplifica la configuración.

Además, según la realización de la presente invención, se usa la unidad 18 de transformación de subcampos, lo que simplifica la configuración, y se proporcionan la unidad 85 de transformación lineal y la unidad 87 de transformación lineal, de modo tal que se mejore la seguridad incluso si se usa la unidad 18 de transformación de subcampo.

Además, según la realización de la presente invención, la unidad 345 de control de desplazamiento puede hacer que el registro de desplazamiento opere un cierto número de veces para llevar a cabo el desplazamiento de los datos de clave con el número de bits de desplazamiento (por ejemplo, 30 bits o 34 bits), que no es un número fijo de bits, tal como solamente 15 bits o 17 bits, y mejora la seguridad.

Además, según la realización de la presente invención, se proporciona un caso en el cual los datos desplazados en el registro de desplazamiento no se usan para la clave de extensión, lo cual mejora adicionalmente la seguridad.

Además, según la realización de la presente invención, incluso si se ingresan los datos de clave con un número distinto de bits, la unidad 310 de transformación de la longitud de bits cambia los datos de clave con una longitud fija, lo que permite operar una generación flexible de clave.

Además, según la realización de la presente invención, la transformación no lineal de dos rondas se usa en la primera unidad 320 de transformación de clave de G bits, por lo que puede demostrarse la inexistencia de la clave equivalente a K_1, lo que mejora la seguridad.

Además, según la realización de la presente invención, se altera la ubicación de la función 25 de clave, lo que habilita un procesamiento de alta velocidad.

REIVINDICACIONES

1. Un aparato de transformación de datos que tiene una unidad (200, 500) de procesamiento de datos para llevar a cabo al menos uno entre el cifrado de datos y el descifrado de datos, en base a datos de clave ingresados,

en el que la unidad (200, 500) de procesamiento de datos comprende una unidad de transformación no lineal para realizar una transformación no lineal de los datos, en el que la unidad de transformación no lineal incluye:

una primera unidad (s_1, 13) de transformación para transformar una parte de los datos a transformar introducidos en la unidad de transformación no lineal como unos primeros datos parciales usando una tabla T de transformación que tiene una entrada y una salida de 8 bits para realizar una transformación no lineal, que se utiliza para transformar un valor de los datos introducidos en otro valor, y emitir los datos transformados, y
una segunda unidad (s_2, 14) de transformación para transformar al menos otra parte de los datos a transformar introducidos en la unidad de transformación no lineal como segundos datos parciales mediante el uso de dicha tabla T de transformación y operación de cambio rotacional, y emitiendo los datos transformados, en el que la primera unidad de transformación (s_1) introduce los datos y_1 en la tabla T de transformación para emitir datos $s_1(y_1)$ y emite los datos $s_1(y_1)$ como datos $z_1 = s_1(y_1)$, y
la segunda unidad (s_2) de transformación introduce datos y_2 a la tabla T de transformación para emitir datos $s_1(y_2)$, realiza el cambio rotacional en $s_1(y_2)$ para emitir (rot ($s_1(y_2)$)), y emite los datos (rot ($s_1(y_2)$)) como datos z_2 = rot ($s_1(y_2)$).

2. El aparato de transformación de datos de la reivindicación 1, **caracterizado porque** la unidad de procesamiento de datos también incluye una tercera unidad (s_3) de transformación y una cuarta unidad (s_4) de transformación para introducir respectivamente datos parciales que son diferentes de los primeros datos parciales y los segundos datos parciales como terceros datos parciales e introducir datos parciales que son diferentes de los primeros datos parciales, los segundos datos parciales, y los terceros datos parciales como cuartos datos parciales, transformando los terceros datos parciales y los cuartos datos parciales mediante la transformación usando la tabla T de transformación y una operación para la tercera parte y una operación para la cuarta parte, las cuales son diferentes de la operación para la segunda parte de la segunda unidad (s_2) de transformación, y emitiendo los datos transformados.

3. El aparato de transformación de datos de la reivindicación 2, **caracterizado porque** la operación de la tercera parte y/o la operación de la cuarta parte es un cambio rotacional.

4. Un procedimiento de transformación de datos para ejecutar un proceso de procesamiento de datos para realizar al menos uno de cifrado de datos y descifrado de datos en base a datos de clave de entrada,
en el que el proceso de procesamiento de datos comprende un proceso de transformación no lineal para realizar una transformación no lineal de los datos,
en el que el proceso de transformación no lineal incluye:

un primer proceso (s_1) de transformación para transformar una parte de los datos a transformar introducidos en la unidad de transformación no lineal como unos primeros datos parciales usando una tabla T de transformación que tiene una entrada y una salida de 8 bits para realizar una transformación no lineal, que se utiliza para transformar un valor de los datos introducidos en otro valor, y emitir los datos transformados, y
un segundo proceso (s_2) de transformación para transformar al menos otra parte de los datos a transformar introducidos en la unidad de transformación no lineal como segundos datos parciales mediante el uso de dicha tabla T de transformación y operación de cambio rotacional, y emitiendo los datos transformados, en el que el primer proceso (s_1) de transformación introduce los datos y_1 en la tabla T de transformación para emitir datos $s_1(y_1)$ y emite los datos $s_1(y_1)$ como datos $z_1 = s_1(y_1)$, y
el segundo proceso (s_2) de transformación introduce datos y_2 a la tabla T de transformación para emitir datos $s_1(y_2)$, realiza el cambio rotacional en $s_1(y_2)$ para emitir (rot ($s_1(y_2)$)), y emite los datos (rot ($s_1(y_2)$)) como datos z_2 = rot ($s_1(y_2)$).

5. Un programa de ordenador que comprende medios de código de programa de ordenador adaptados para realizar las etapas del procedimiento de la reivindicación 4 cuando dicho programa es ejecutado en un ordenador.

6. Soporte informático legible por ordenador que lleva el programa de ordenador de la reivindicación 5.

Fig. 1

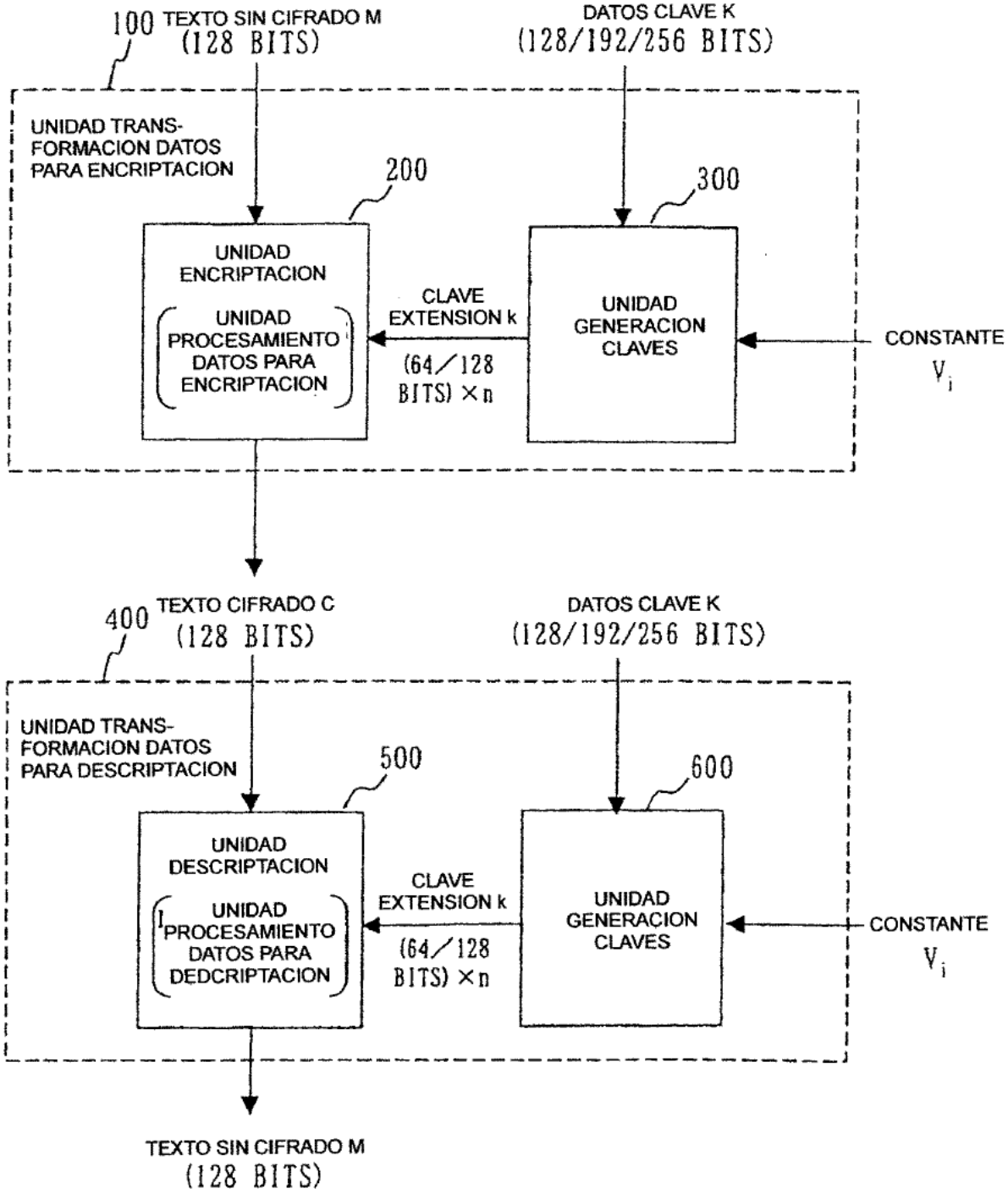

100
TEXTO SIN CIFRADO M
(128 BITS)
DATOS CLAVE K
(128/192/256 BITS)
UNIDAD TRANS-
FORMACION DATOS
PARA ENCRIPTACION
200
300
UNIDAD
ENCRIPTACION
UNIDAD
PROCESAMIENTO
DATOS PARA
ENCRIPTACION
CLAVE
EXTENSION k
(64/128
BITS) ×n
UNIDAD
GENERACION
CLAVES
CONSTANTE
Vi
400
TEXTO CIFRADO C
(128 BITS)
DATOS CLAVE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

kw_1, kw_2
k_1, k_2, ..., k_{24}
kl_1, kl_2, ..., kl_6
} CLAVES EXTENSION

F : UNIDAD FUNCION NO LINEAL
FL : UNIDAD TRANSFORMACION NORMAL DATOS
FL^{-1} : UNIDAD TRANSFORMACION INVERSA DATOS

L_0, L_1, ..., L_{24} DATOS IZQUIERDA
R_0, R_1, ..., R_{24} DATOS DERECHA
V_1, V_2, ..., V_6 CONSTANTE

EX. 1 M(q) MUESTRA DATOS TEXTO NO CIFRADO DE 16 BYTES (128 BITS)
EX. 2 kw_1(q) MUESTRA CLAVE EXTENSION DE 16 BYTES (128 BITS)
EX. 3 k_1(l) MUESTRA CLAVE EXTENSION DE 8 BYTES (64 BITS)
EX. 4 L_0(l) MUESTRA DATOS IZQUIERDA DE 8 BYTES (64 BITS)

Fig. 3

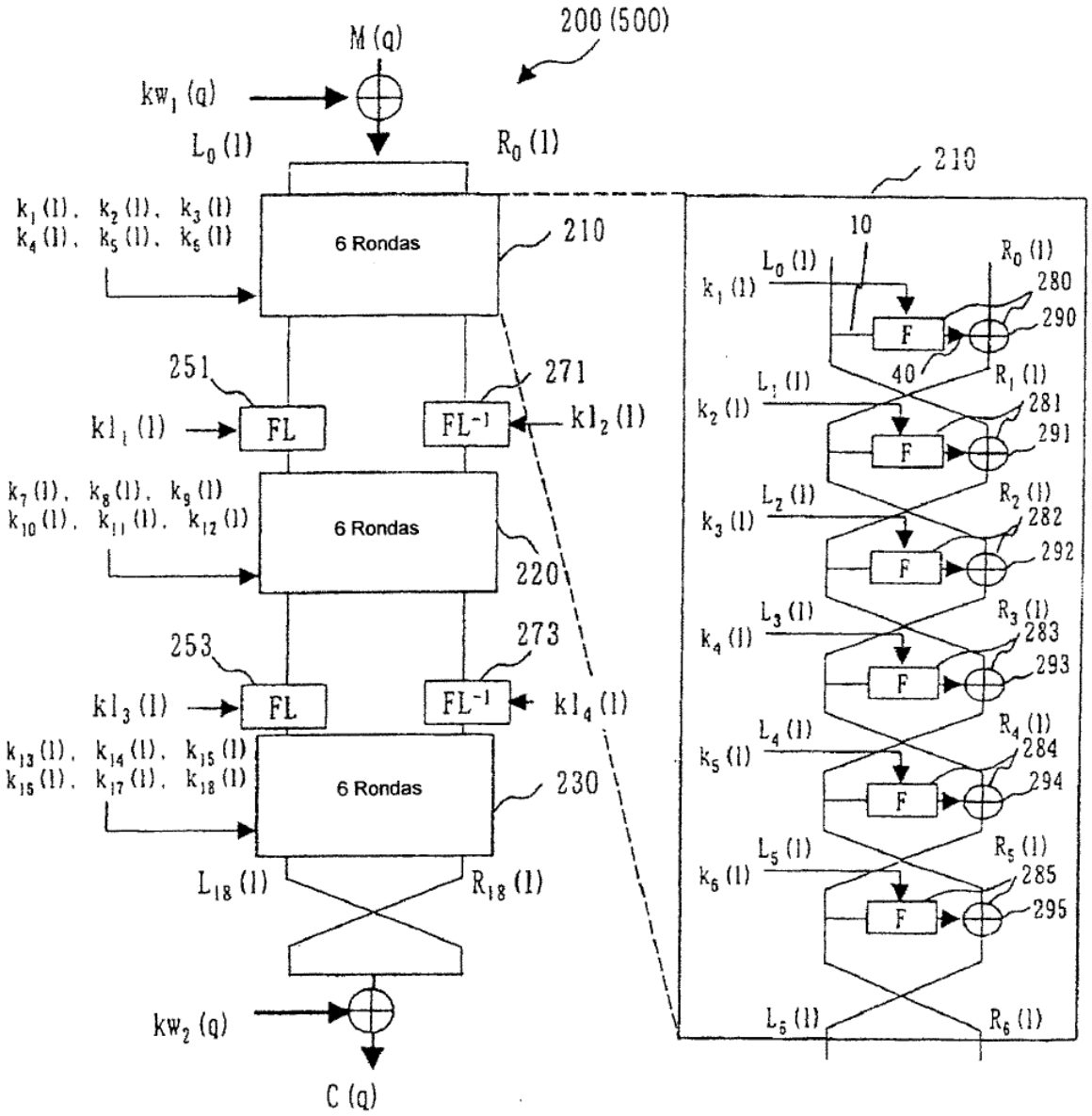

200 (500)
M (q)
kw1 (q)
L0 (1)
R0 (1)
k1 (1), k2 (1), k3 (1)
k4 (1), k5 (1), k6 (1)
6 Rondas
210
251
271
kl1 (1)
FL
FL-1
kl2 (1)
k7 (1), k8 (1), k9 (1)
k10 (1), k11 (1), k12 (1)
6 Rondas
220
253
273
kl3 (1)
FL
FL-1
kl4 (1)
k13 (1), k14 (1), k15 (1)
k16 (1), k17 (1), k18 (1)
6 Rondas
230
L18 (1)
R18 (1)
kw2 (q)
C (q)
210
10
L0 (1)
R0 (1)
k1 (1)
280
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

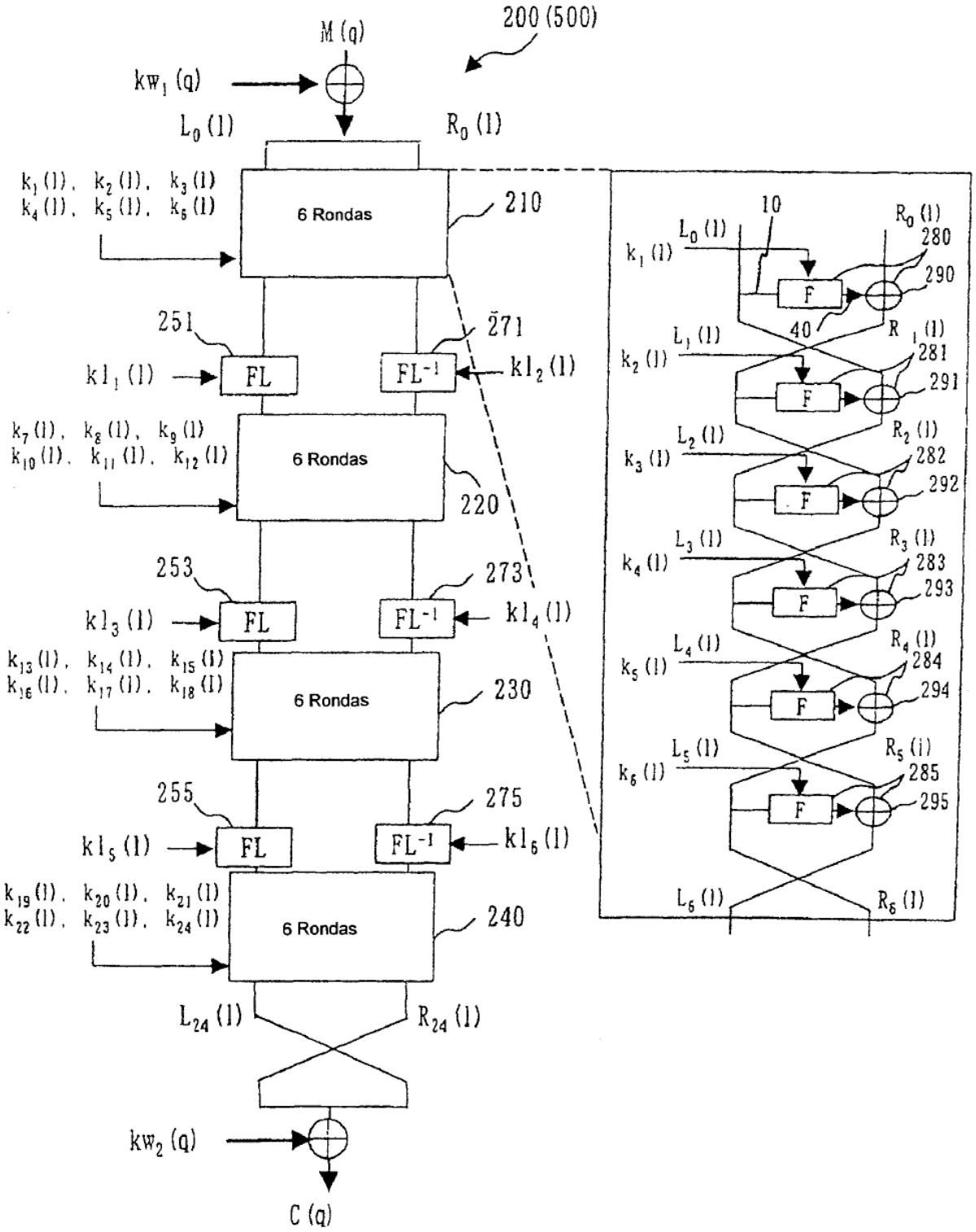

M (q)
200 (500)
kw1 (q)
L0 (l)
R0 (l)
k1 (l), k2 (l), k3 (l)
k4 (l), k5 (l), k6 (l)
6 Rondas
210
251
271
kl1 (l)
FL
FL-1
kl2 (l)
k7 (l), k8 (l), k9 (l)
k10 (l), k11 (l), k12 (l)
6 Rondas
220
253
273
kl3 (l)
FL
FL-1
kl4 (l)
k13 (l), k14 (l), k15 (l)
k16 (l), k17 (l), k18 (l)
6 Rondas
230
255
275
kl5 (l)
FL
FL-1
kl6 (l)
k19 (l), k20 (l), k21 (l)
k22 (l), k23 (l), k24 (l)
6 Rondas
240
L24 (l)
R24 (l)
kw2 (q)
C (q)
10
L0 (l)
R0 (l)
k1 (l)
280
290
F
40
L1 (l)
R1 (l)
k2 (l)
281
291
L2 (l)
R2 (l)
k3 (l)
282
292
L3 (l)
R3 (l)
k4 (l)
283
293
L4 (l)
R4 (l)
k5 (l)
284
294
L5 (l)
R5 (l)
k6 (l)
285
295
L6 (l)
R6 (l)

Fig. 5

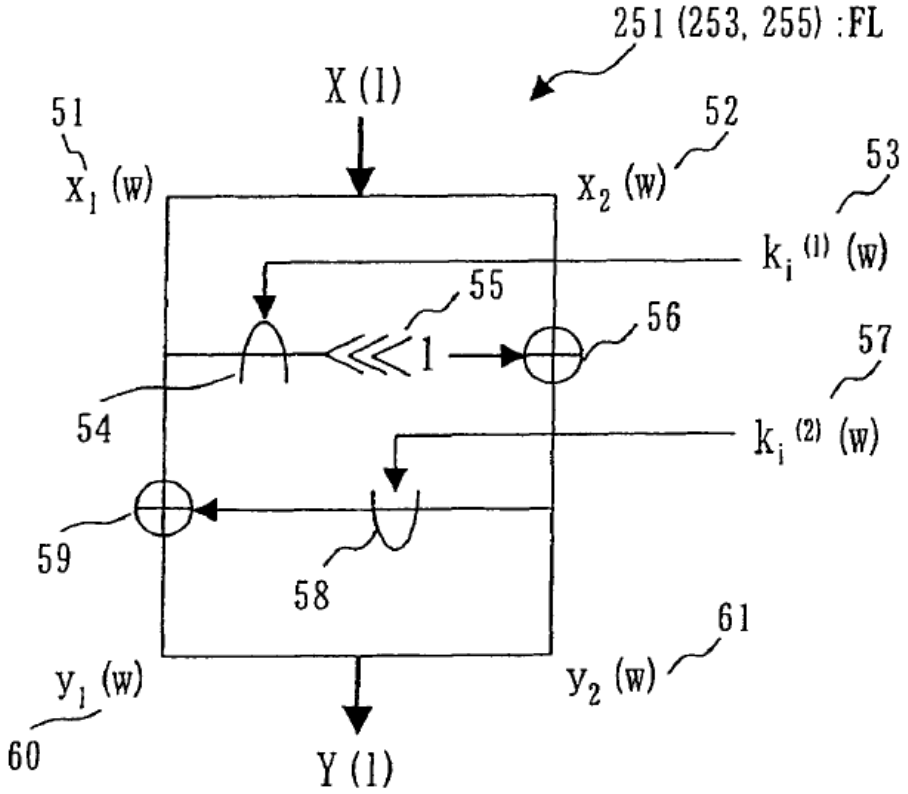

251 (253, 255) :FL
X (l)
51
x_1 (w)
52
x_2 (w)
53
$k_i^{(1)}$ (w)
55
56
1
57
54
$k_i^{(2)}$ (w)
59
58
61
y_1 (w)
y_2 (w)
60
Y (l)

Fig. 6

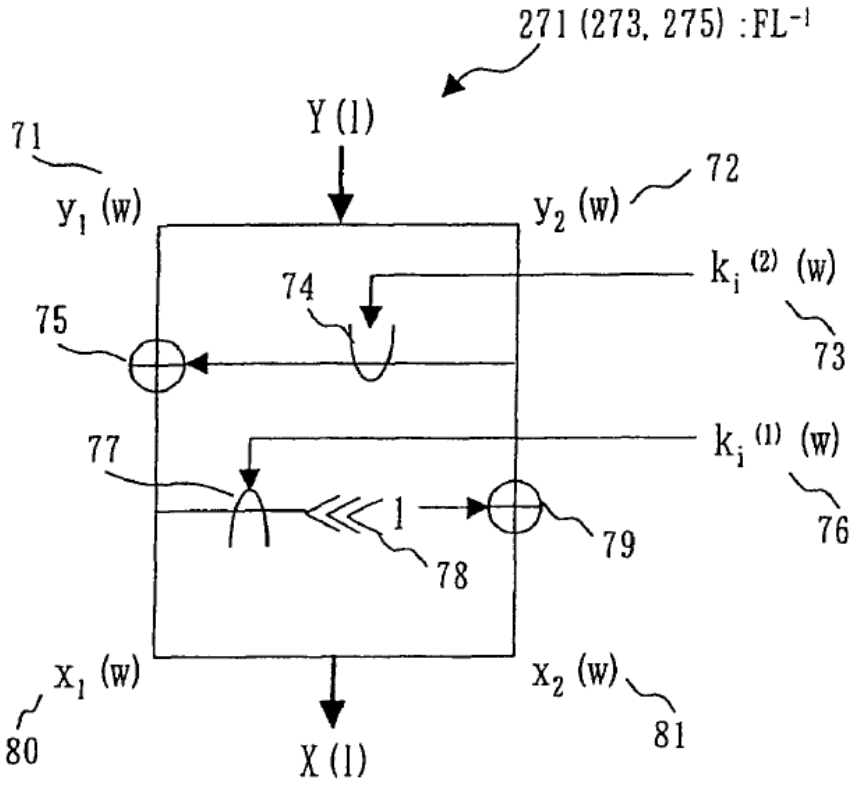

271 (273, 275) :FL⁻¹
Y (l)
71
y1 (w)
y2 (w)
72
ki (2) (w)
74
75
73
77
ki (1) (w)
l
79
76
78
x1 (w)
x2 (w)
80
X (l)
81

Fig. 7

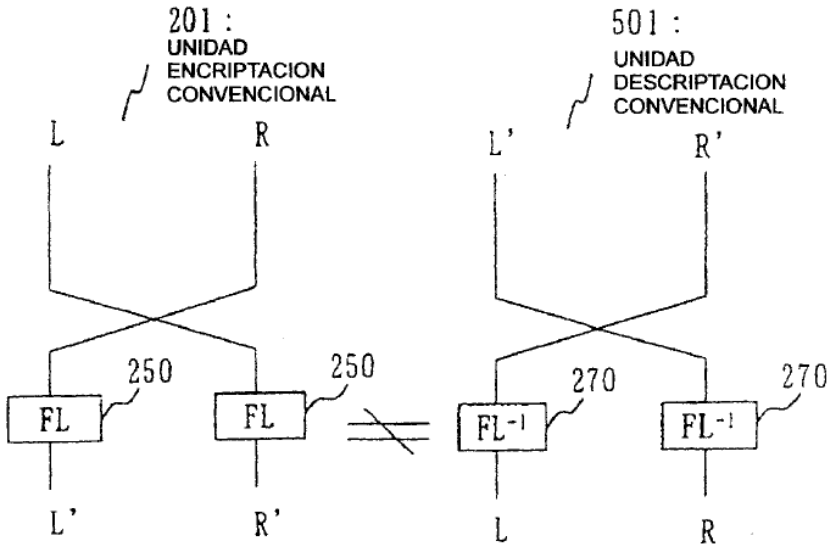

201 :
UNIDAD
ENCRIPTACION
CONVENCIONAL
501 :
UNIDAD
DESCRIPTACION
CONVENCIONAL
L
R
L'
R'
250
250
270
270
FL
FL
FL-1
FL-1
L'
R'
L
R

Fig. 8

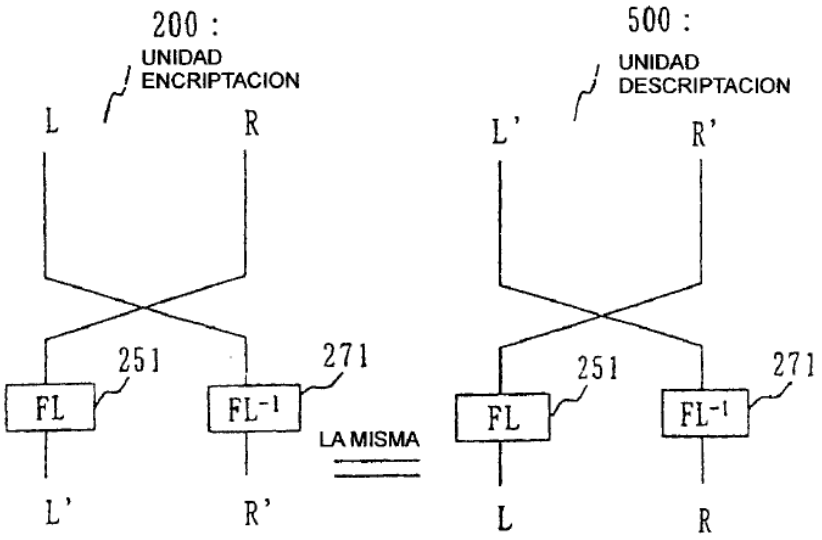

200 :
UNIDAD
ENCRIPTACION
L
R
251
271
FL
FL-1
L'
R'
LA MISMA
500 :
UNIDAD
DESCRIPTACION
L'
R'
251
271
FL
FL-1
L
R

Fig. 9

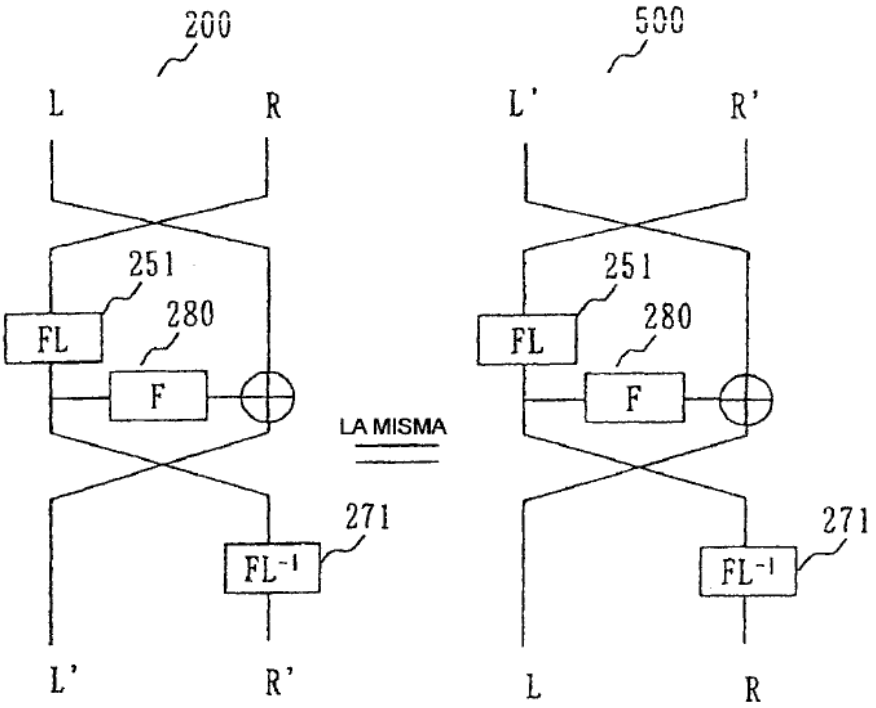

200
L
R
251
FL
280
F
271
FL⁻¹
L'
R'
LA MISMA
500
L'
R'
251
FL
280
F
271
FL⁻¹
L
R

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

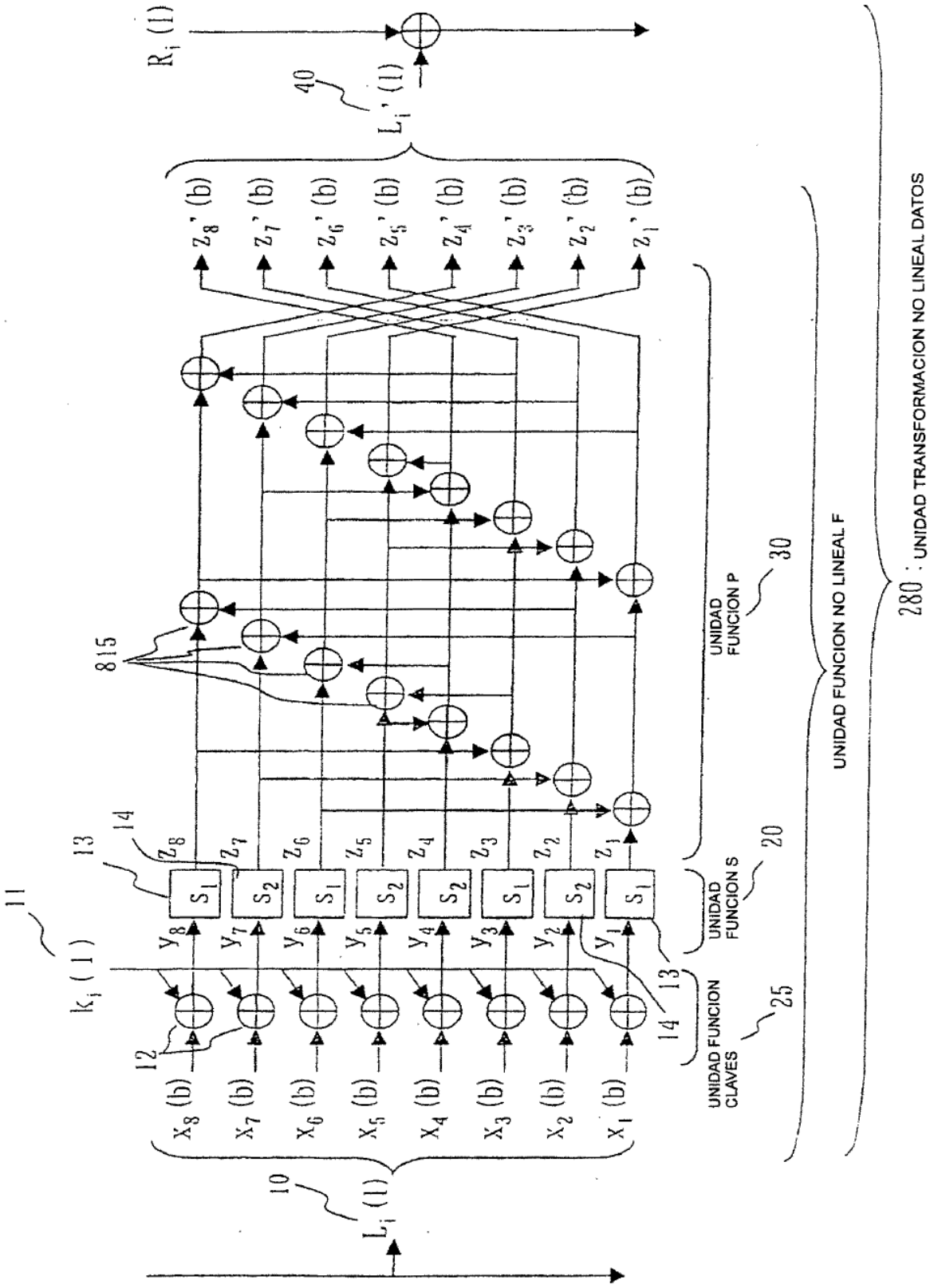

R_i (l)
40
L_i' (l)
z_8' (b)
z_7' (b)
z_6' (b)
z_5' (b)
z_4' (b)
z_3' (b)
z_2' (b)
z_1' (b)
815
UNIDAD FUNCION P
30
UNIDAD FUNCION NO LINEAL F
280 : UNIDAD TRANSFORMACION NO LINEAL DATOS
14
13
z_8
z_7
z_6
z_5
z_4
z_3
z_2
z_1
S_1
S_2
UNIDAD FUNCION S
20
11
k_i (l)
y_8
y_7
y_6
y_5
y_4
y_3
y_2
y_1
12
UNIDAD FUNCION CLAVES
25
x_8 (b)
x_7 (b)
x_6 (b)
x_5 (b)
x_4 (b)
x_3 (b)
x_2 (b)
x_1 (b)
10
L_i (l)

Fig.12

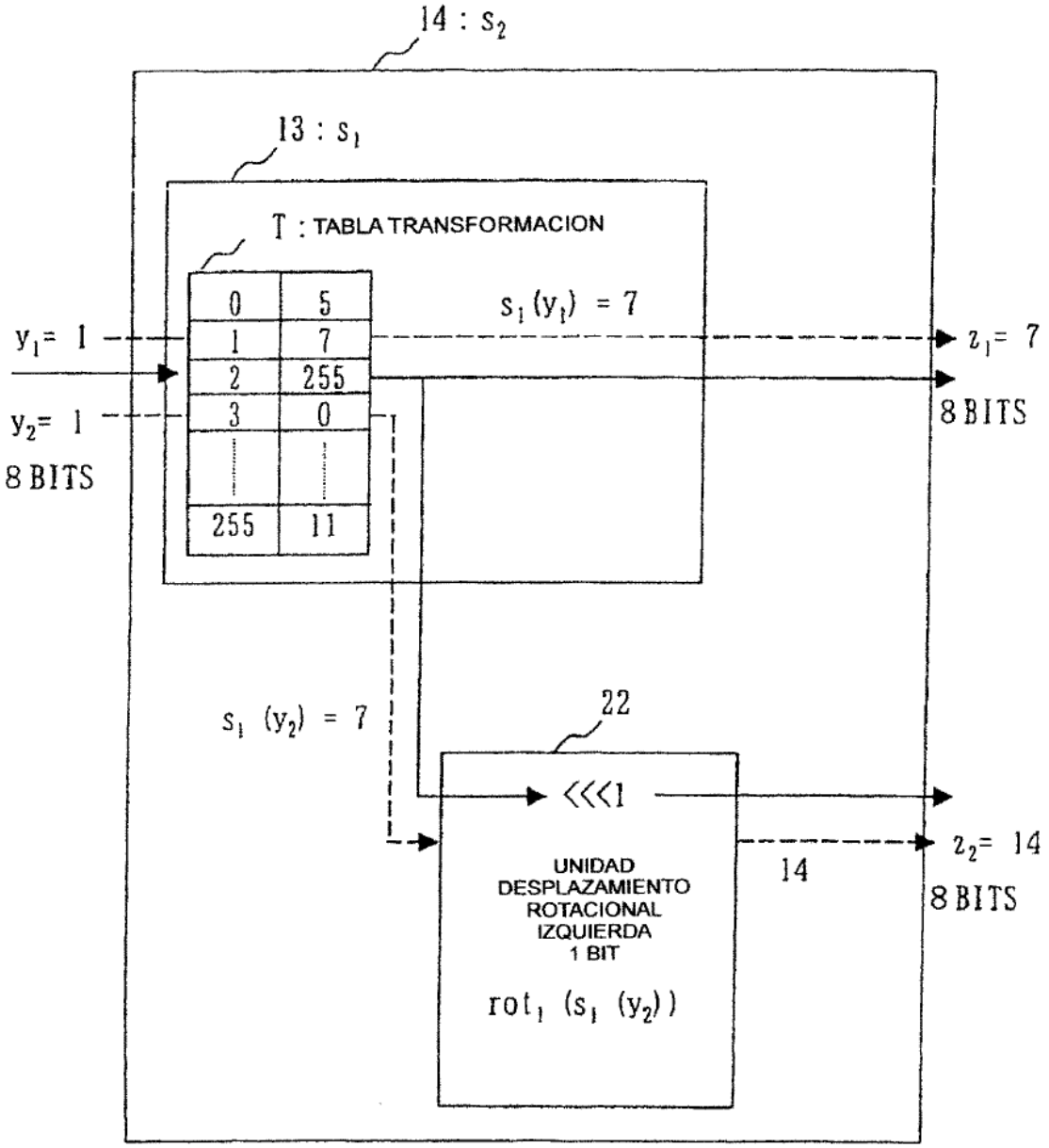

14 : s2
13 : s1
T : TABLA TRANSFORMACION
0 5
1 7
2 255
3 0
255 11
y1= 1
y2= 1
8 BITS
s1 (y1) = 7
z1= 7
8 BITS
s1 (y2) = 7
22
<<<1
UNIDAD DESPLAZAMIENTO ROTACIONAL IZQUIERDA 1 BIT
rot1 (s1 (y2))
14
z2= 14
8 BITS

Fig. 13

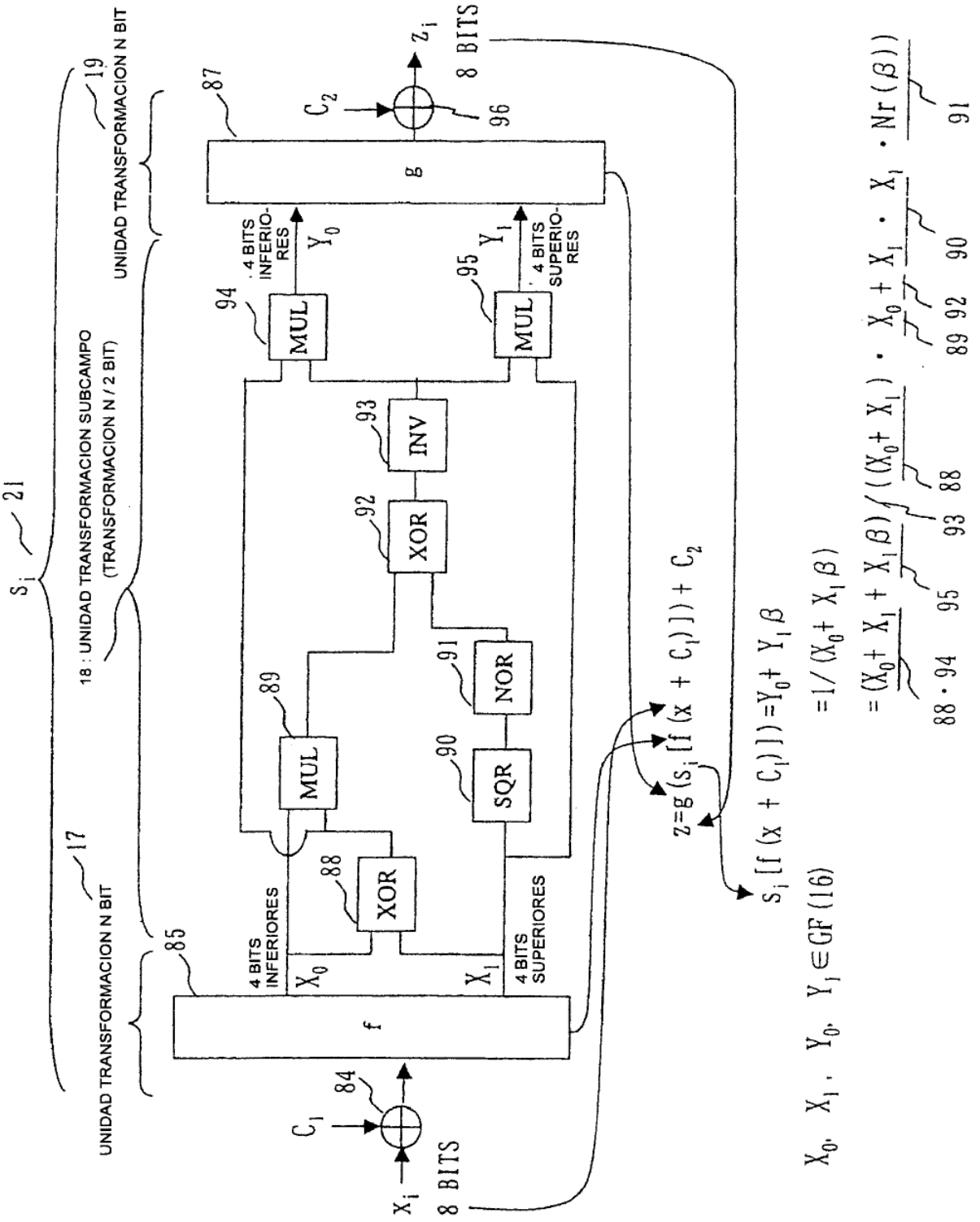

UNIDAD TRANSFORMACION N BIT
19
UNIDAD TRANSFORMACION N BIT
17
18 : UNIDAD TRANSFORMACION SUBCAMPO
(TRANSFORMACION N / 2 BIT)
21
S_i
85
f
84
C_1
X_i
8 BITS
4 BITS INFERIORES
X_0
X_1
4 BITS SUPERIORES
88
XOR
89
MUL
90
SQR
91
NOR
92
XOR
93
INV
94
MUL
95
MUL
4 BITS INFERIORES
Y_0
Y_1
4 BITS SUPERIORES
87
g
C_2
96
z_i
8 BITS
$z=g(s_i[f(x+C_1)])+C_2$
$s_i[f(x+C_1)])=Y_0+Y_1\beta$
$=1/(X_0+X_1\beta)$
$=(X_0+X_1+X_1\beta)/((X_0+X_1)\cdot X_0+X_1\cdot X_1\cdot Nr(\beta))$
88 · 94 95 93 88 89 92 90 91
$X_0,\ X_1,\ Y_0,\ Y_1 \in GF(16)$

Fig.14

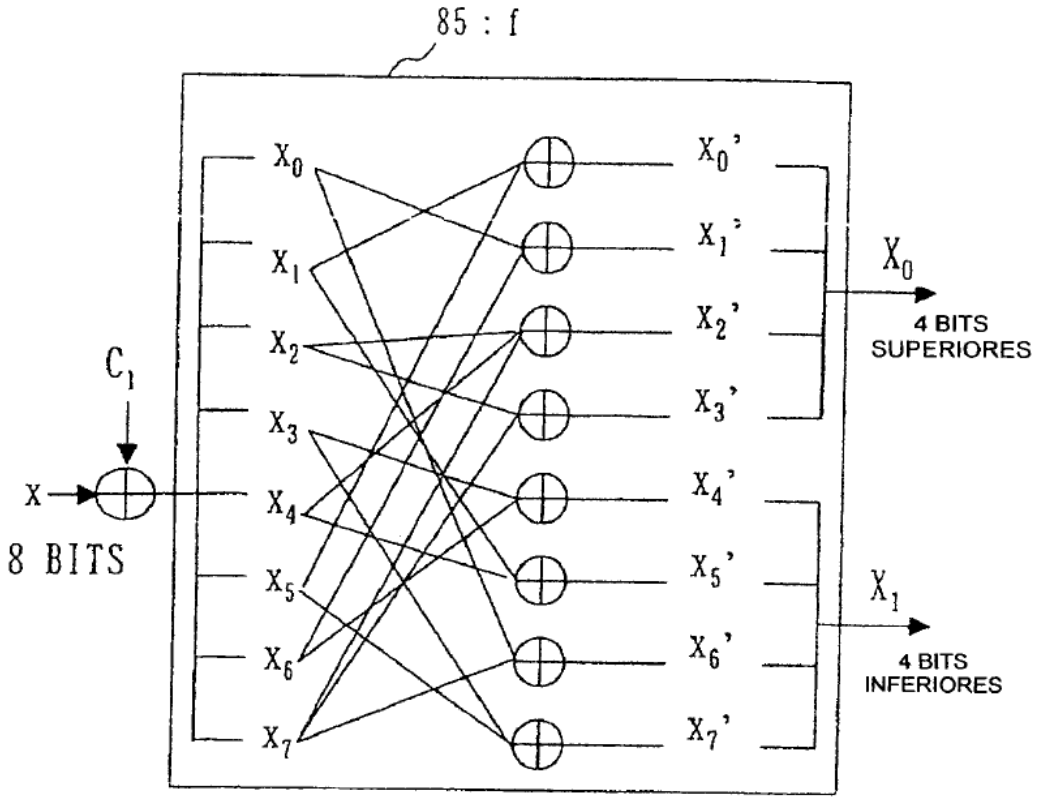

85 : f
X0
X1
X2
X3
X4
X5
X6
X7
X0'
X1'
X2'
X3'
X4'
X5'
X6'
X7'
C1
X
8 BITS
X0
4 BITS
SUPERIORES
X1
4 BITS
INFERIORES

Fig.15

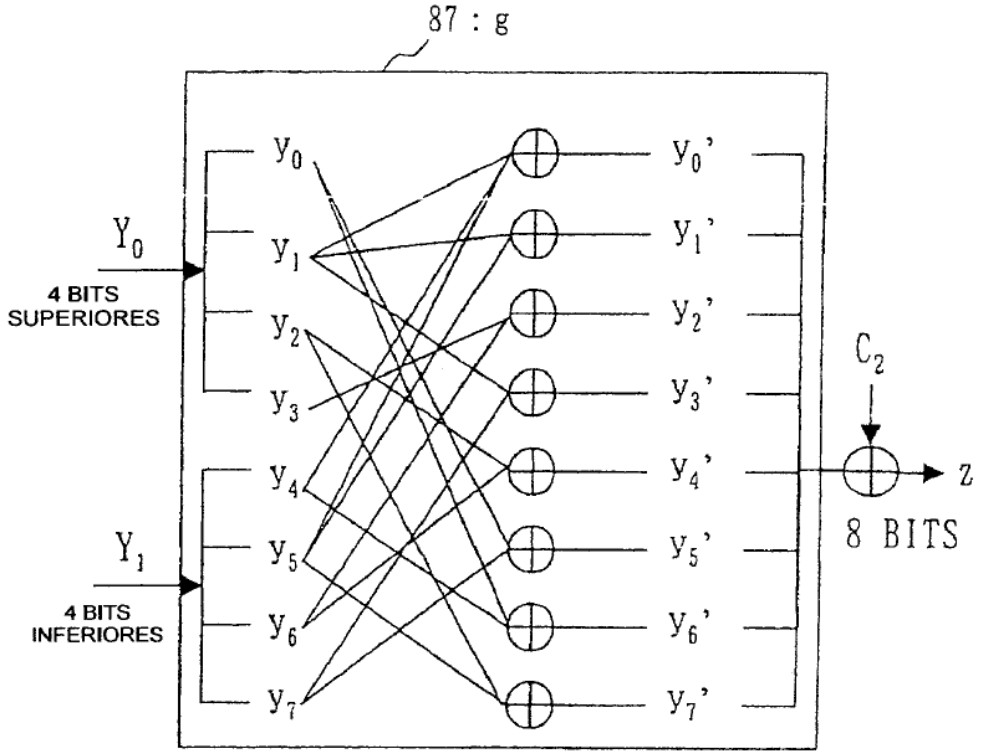

87 : g
Y_0
4 BITS SUPERIORES
Y_1
4 BITS INFERIORES
y_0
y_1
y_2
y_3
y_4
y_5
y_6
y_7
y_0'
y_1'
y_2'
y_3'
y_4'
y_5'
y_6'
y_7'
C_2
z
8 BITS

Fig. 16

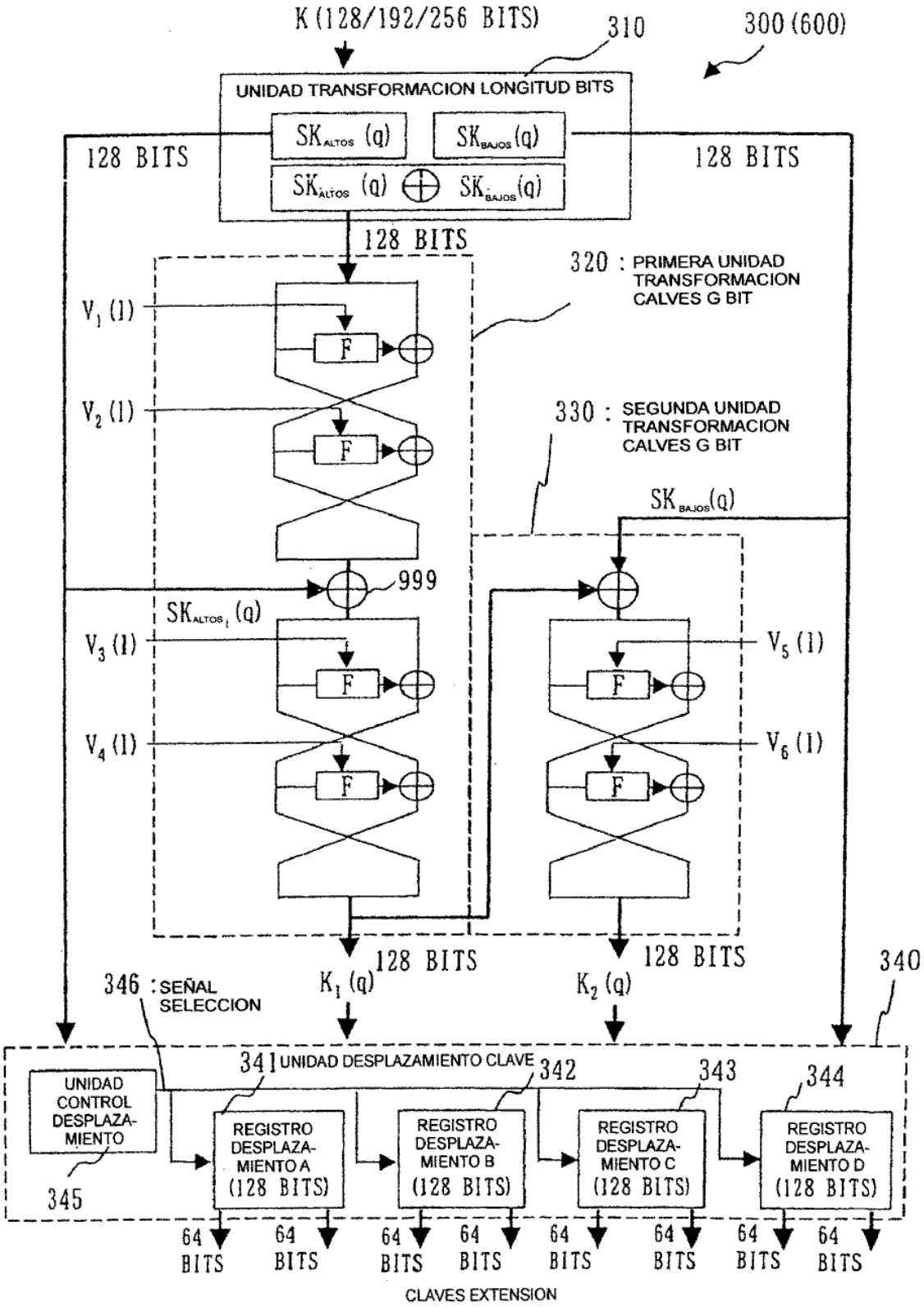

K (128/192/256 BITS)
310
300 (600)
UNIDAD TRANSFORMACION LONGITUD BITS
SK ALTOS (q)
SK BAJOS (q)
SK ALTOS (q) ⊕ SK BAJOS (q)
128 BITS
128 BITS
128 BITS
320 : PRIMERA UNIDAD TRANSFORMACION CALVES G BIT
V1 (I)
V2 (I)
F
330 : SEGUNDA UNIDAD TRANSFORMACION CALVES G BIT
SK BAJOS (q)
999
SK ALTOS i (q)
V3 (I)
V4 (I)
V5 (I)
V6 (I)
128 BITS
128 BITS
K1 (q)
K2 (q)
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

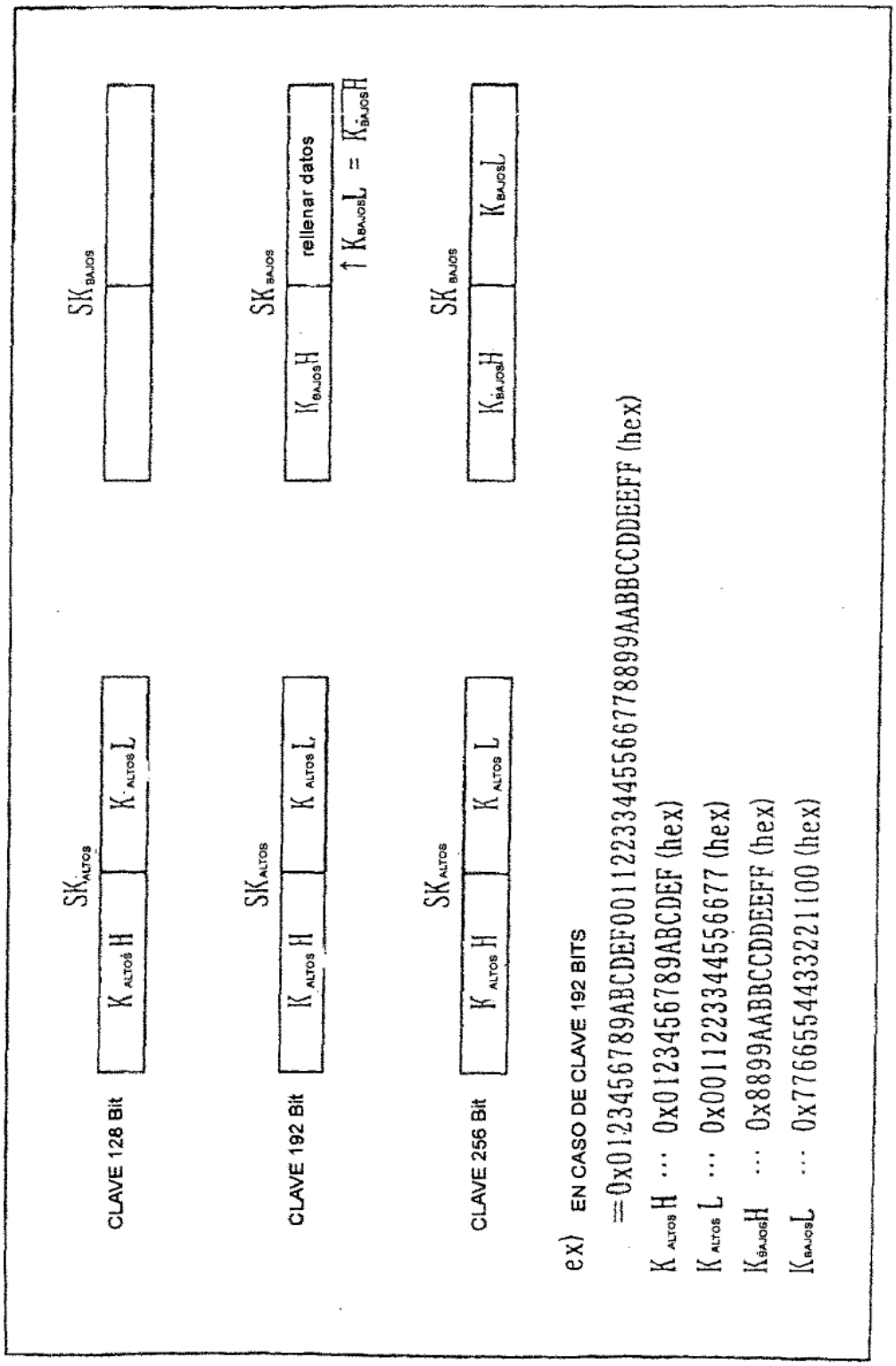

SK BAJOS
SK ALTOS
K ALTOS H
K ALTOS L
CLAVE 128 Bit
SK BAJOS
SK ALTOS
K BAJOS H
rellenar datos
↑ K BAJOS L = K BAJOS H
K ALTOS H
K ALTOS L
CLAVE 192 Bit
SK BAJOS
SK ALTOS
K BAJOS H
K BAJOS L
K ALTOS H
K ALTOS L
CLAVE 256 Bit
ex) EN CASO DE CLAVE 192 BITS
=0x0123456789ABCDEF0011223344556677889 9AABBCCDDEEFF (hex)
K ALTOS H ··· 0x0123456789ABCDEF (hex)
K ALTOS L ··· 0x0011223344556677 (hex)
K BAJOS H ··· 0x8899AABBCCDDEEFF (hex)
K BAJOS L ··· 0x7766554433221100 (hex)

Fig.18

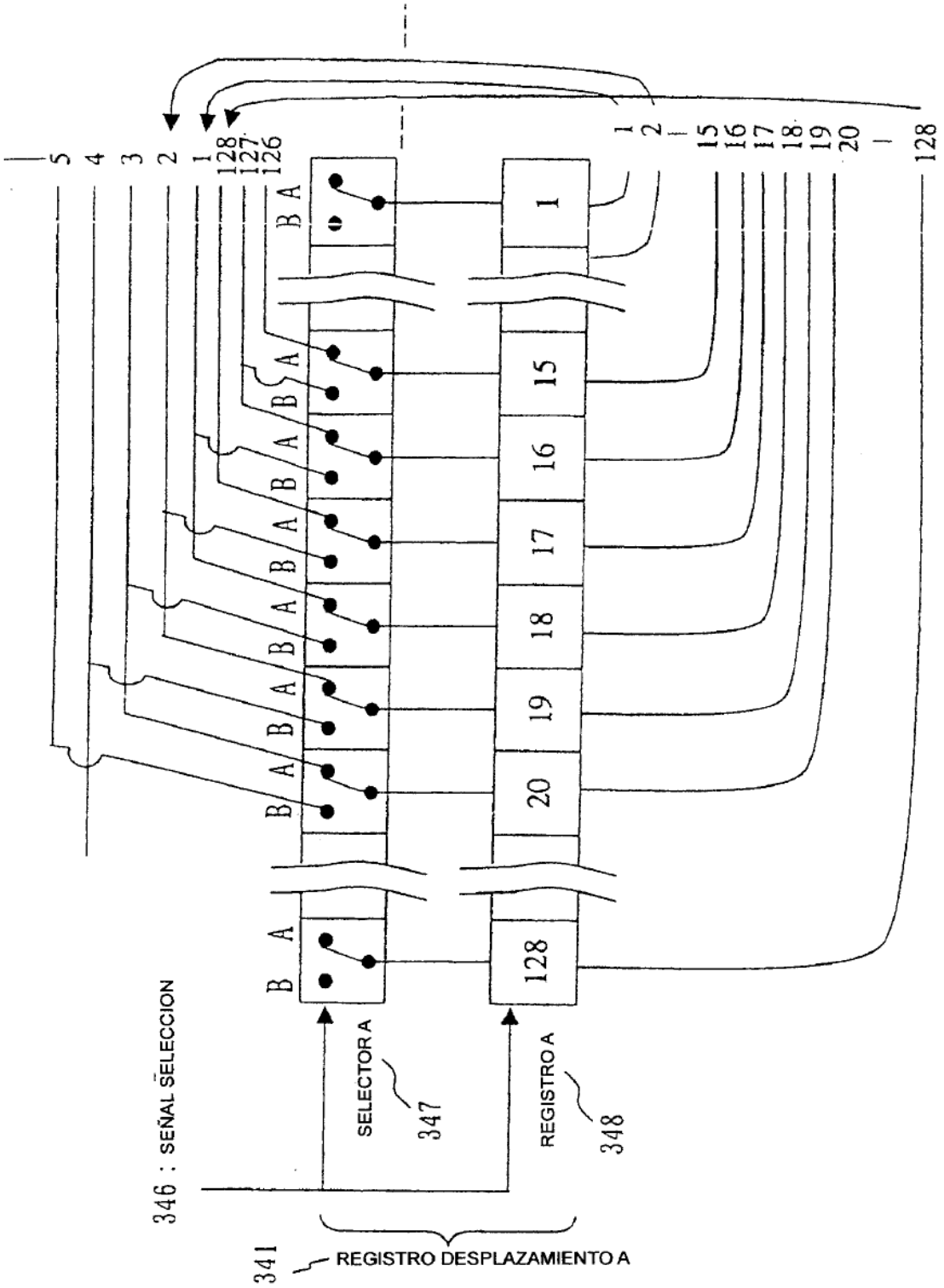

5
4
3
2
1
128
127
126
1
2
15
16
17
18
19
20
128
B A
1
15
16
17
18
19
20
128
346 : SEÑAL SELECCION
SELECTOR A
347
REGISTRO A
348
341
REGISTRO DESPLAZAMIENTO A

Fig. 19

RELOJ	CLAVE 128 BITS				CLAVE 192 / 256 BITS			
	TABLA CONTROL DESPLAZAMIENTO REGISTRO A		TABLA CONTROL DESPLAZAMIENTO REGISTRO B		TABLA CONTROL DESPLAZAMIENTO REGISTRO A	TABLA CONTROL DESPLAZAMIENTO REGISTRO B	TABLA CONTROL DESPLAZAMIENTO REGISTRO C	TABLA CONTROL DESPLAZAMIENTO REGISTRO D
	NUM. TOTAL DE BITS DESPLAZAMIENTO	BITS DESPLAZAMIENTO	NUM. TOTAL DE BITS DESPLAZAMIENTO	BITS DESPLAZAMIENTO				
1	$z_0=0$	+15	$z_0=0$	+15	OMITIDA			
2	$z_1=15$	+15	$z_1=15$	+15				
3		+15	$z_2=30$	+15				
4	$z_2=45$	+15	$z_3=45$	+15				
5	$z_3=60$	+17	$z_4=60$	+17				
6	$z_4=77$	+17		+17				
7	$z_5=94$	+17	$z_5=94$	+17				
8	$z_6=111$	+17	$z_6=111$	+17				
9	$z_7=128$		$z_7=128$					

Fig. 20

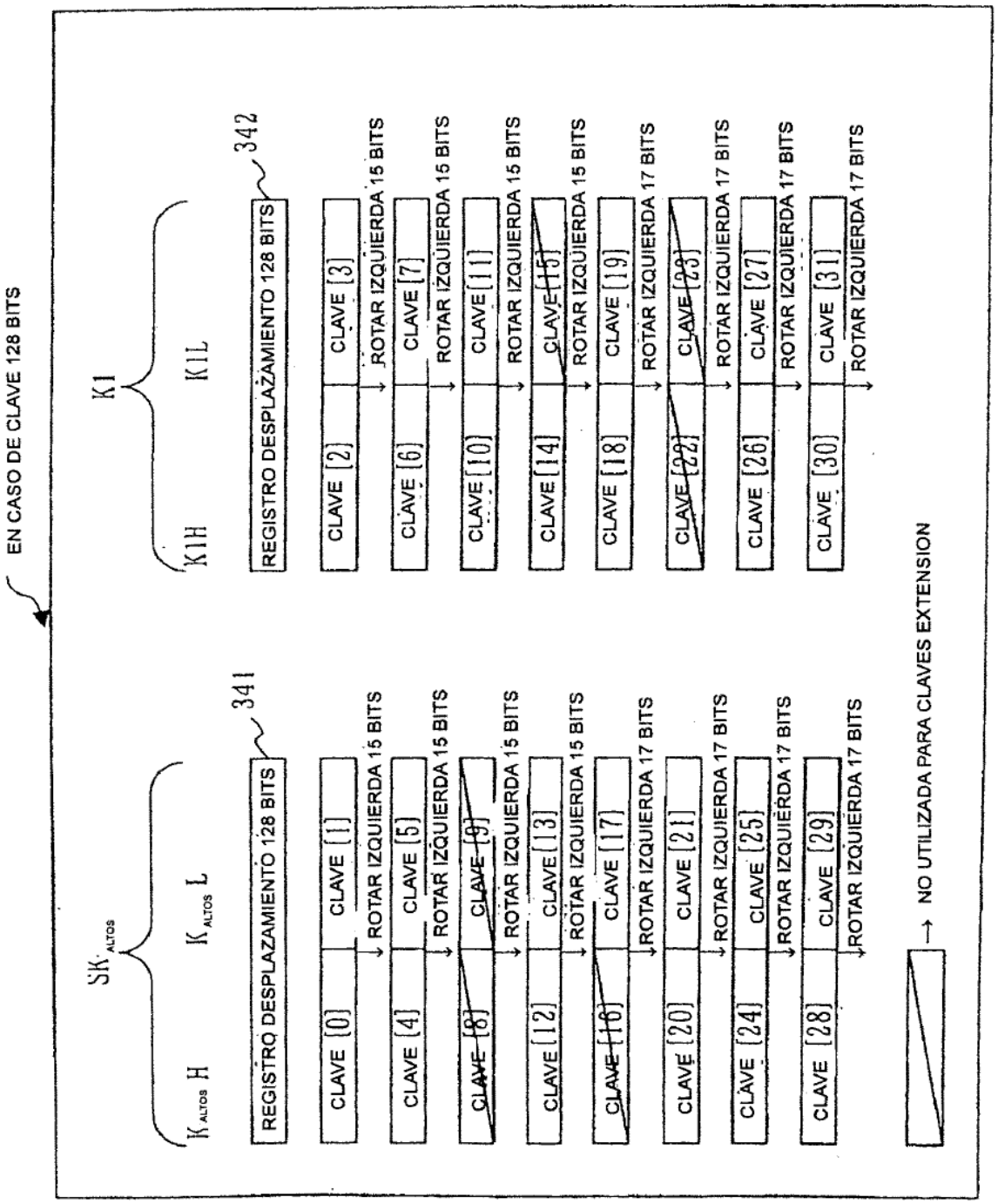

EN CASO DE CLAVE 128 BITS
SK ALTOS
K ALTOS H
K ALTOS L
341
REGISTRO DESPLAZAMIENTO 128 BITS
CLAVE [0]
CLAVE [1]
ROTAR IZQUIERDA 15 BITS
CLAVE [4]
CLAVE [5]
ROTAR IZQUIERDA 15 BITS
CLAVE [8]
CLAVE [9]
ROTAR IZQUIERDA 15 BITS
CLAVE [12]
CLAVE [13]
ROTAR IZQUIERDA 15 BITS
CLAVE [16]
CLAVE [17]
ROTAR IZQUIERDA 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
K1
K1H
K1L
342
REGISTRO DESPLAZAMIENTO 128 BITS
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
NO UTILIZADA PARA CLAVES EXTENSION

Fig. 21

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

Fig. 22

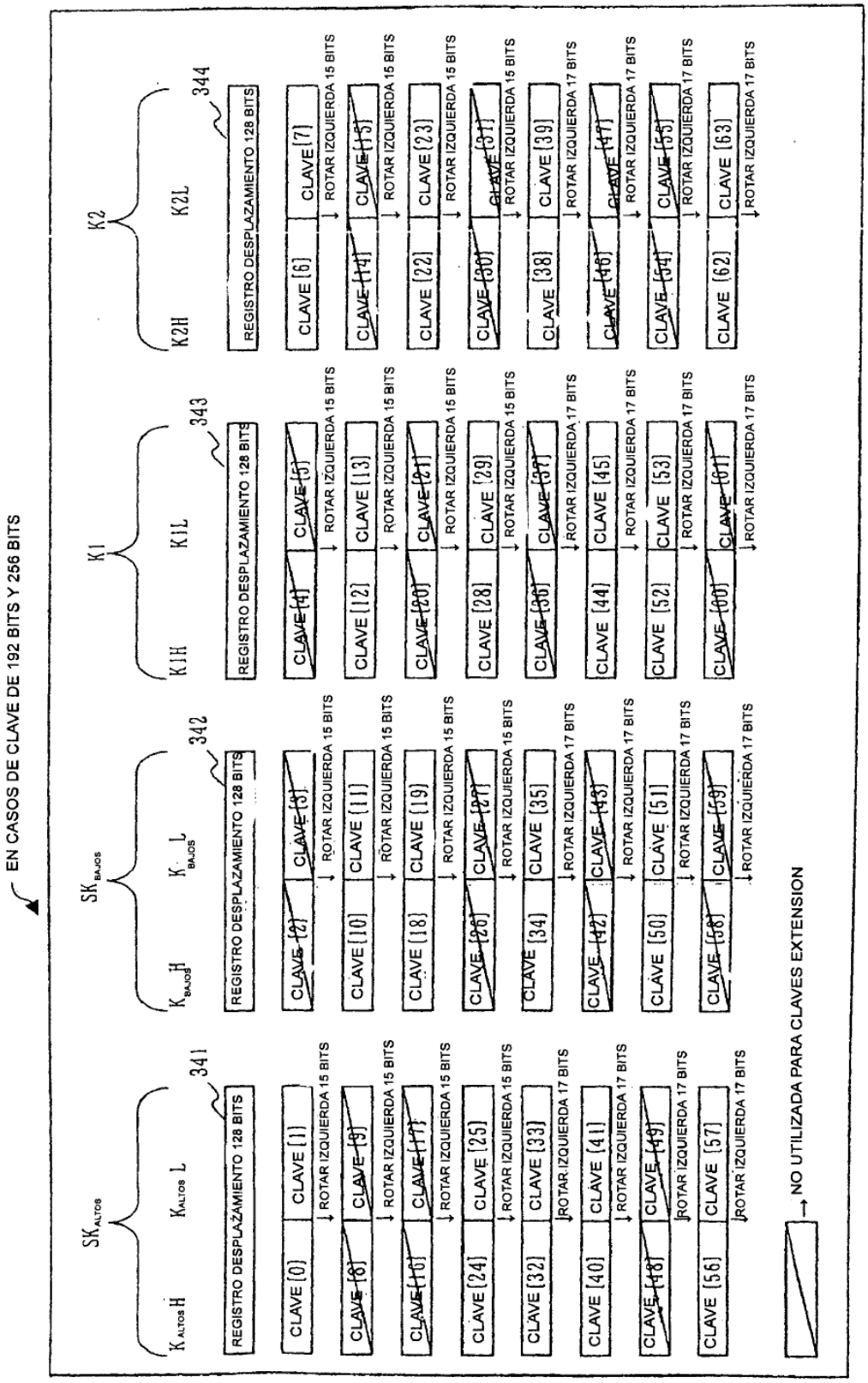

EN CASOS DE CLAVE DE 192 BITS Y 256 BITS
SK ALTOS
K ALTOS H
K ALTOS L
341
REGISTRO DESPLAZAMIENTO 128 BITS
CLAVE [0]
CLAVE [1]
ROTAR IZQUIERDA 15 BITS
CLAVE [8]
CLAVE [9]
ROTAR IZQUIERDA 15 BITS
CLAVE [16]
CLAVE [17]
ROTAR IZQUIERDA 15 BITS
CLAVE [24]
CLAVE [25]
ROTAR IZQUIERDA 15 BITS
CLAVE [32]
CLAVE [33]
ROTAR IZQUIERDA 17 BITS
CLAVE [40]
CLAVE [41]
ROTAR IZQUIERDA 17 BITS
CLAVE [48]
CLAVE [49]
ROTAR IZQUIERDA 17 BITS
CLAVE [56]
CLAVE [57]
ROTAR IZQUIERDA 17 BITS
SK BAJOS
K BAJOS H
K BAJOS L
342
REGISTRO DESPLAZAMIENTO 128 BITS
CLAVE [2]
CLAVE [3]
ROTAR IZQUIERDA 15 BITS
CLAVE [10]
CLAVE [11]
ROTAR IZQUIERDA 15 BITS
CLAVE [18]
CLAVE [19]
ROTAR IZQUIERDA 15 BITS
CLAVE [26]
CLAVE [27]
ROTAR IZQUIERDA 15 BITS
CLAVE [34]
CLAVE [35]
ROTAR IZQUIERDA 17 BITS
CLAVE [42]
CLAVE [43]
ROTAR IZQUIERDA 17 BITS
CLAVE [50]
CLAVE [51]
ROTAR IZQUIERDA 17 BITS
CLAVE [58]
CLAVE [59]
ROTAR IZQUIERDA 17 BITS
K1
K1H
K1L
343
REGISTRO DESPLAZAMIENTO 128 BITS
CLAVE [4]
CLAVE [5]
ROTAR IZQUIERDA 15 BITS
CLAVE [12]
CLAVE [13]
ROTAR IZQUIERDA 15 BITS
CLAVE [20]
CLAVE [21]
ROTAR IZQUIERDA 15 BITS
CLAVE [28]
CLAVE [29]
ROTAR IZQUIERDA 15 BITS
CLAVE [36]
CLAVE [37]
ROTAR IZQUIERDA 17 BITS
CLAVE [44]
CLAVE [45]
ROTAR IZQUIERDA 17 BITS
CLAVE [52]
CLAVE [53]
ROTAR IZQUIERDA 17 BITS
CLAVE [60]
CLAVE [61]
ROTAR IZQUIERDA 17 BITS
K2
K2H
K2L
344
REGISTRO DESPLAZAMIENTO 128 BITS
CLAVE [6]
CLAVE [7]
ROTAR IZQUIERDA 15 BITS
CLAVE [14]
CLAVE [15]
ROTAR IZQUIERDA 15 BITS
CLAVE [22]
CLAVE [23]
ROTAR IZQUIERDA 15 BITS
CLAVE [30]
CLAVE [31]
ROTAR IZQUIERDA 15 BITS
CLAVE [38]
CLAVE [39]
ROTAR IZQUIERDA 17 BITS
CLAVE [46]
CLAVE [47]
ROTAR IZQUIERDA 17 BITS
CLAVE [54]
CLAVE [55]
ROTAR IZQUIERDA 17 BITS
CLAVE [62]
CLAVE [63]
ROTAR IZQUIERDA 17 BITS
NO UTILIZADA PARA CLAVES EXTENSION

Fig. 23

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

Fig. 24

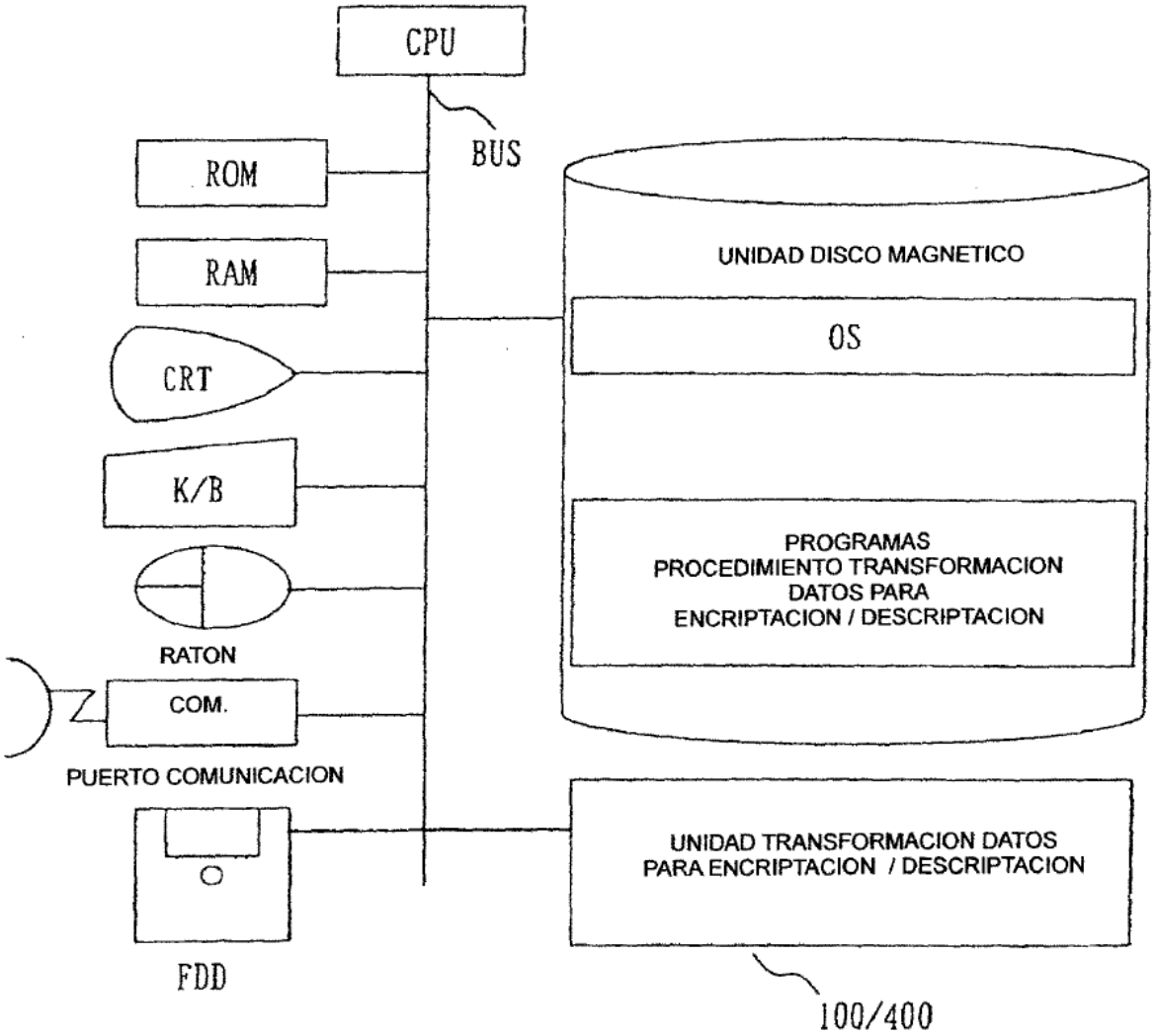

CPU
BUS
ROM
RAM
CRT
K/B
RATON
COM.
PUERTO COMUNICACION
FDD
UNIDAD DISCO MAGNETICO
OS
PROGRAMAS
PROCEDIMIENTO TRANSFORMACION
DATOS PARA
ENCRIPTACION / DESCRIPTACION
UNIDAD TRANSFORMACION DATOS
PARA ENCRIPTACION / DESCRIPTACION
100/400

Fig. 25

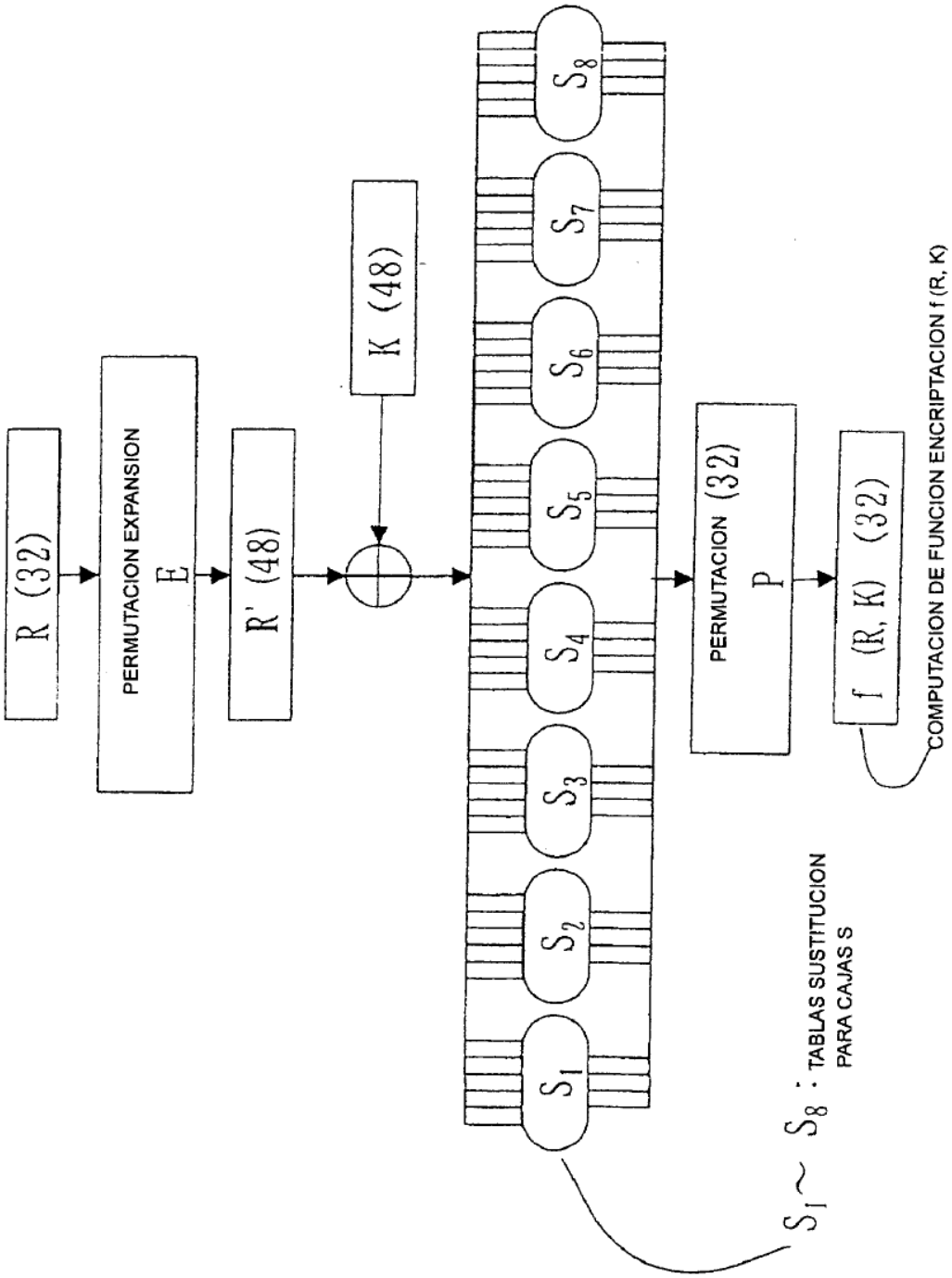

R (32)
PERMUTACION EXPANSION
E
R' (48)
K (48)
S1
S2
S3
S4
S5
S6
S7
S8
PERMUTACION (32)
P
f (R, K) (32)
COMPUTACION DE FUNCION ENCRIPTACION f (R, K)
S1 ~ S8 : TABLAS SUSTITUCION PARA CAJAS S

Fig. 26

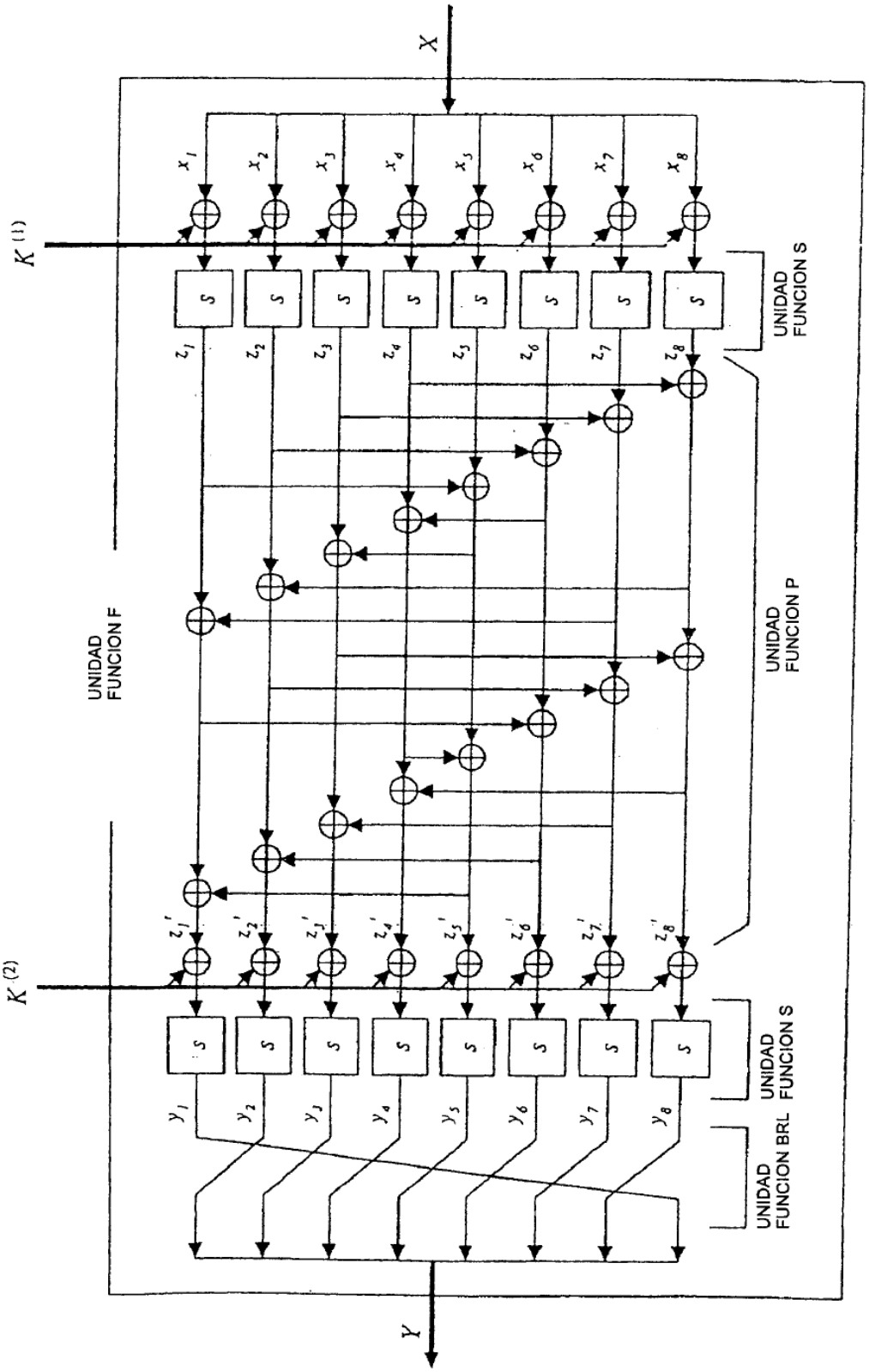

X
$K^{(1)}$
$K^{(2)}$
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8
S
z_1 z_2 z_3 z_4 z_5 z_6 z_7 z_8
z_1' z_2' z_3' z_4' z_5' z_6' z_7' z_8'
y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8
UNIDAD FUNCION S
UNIDAD FUNCION P
UNIDAD FUNCION F
UNIDAD FUNCION S
UNIDAD FUNCION BRL
Y

Fig. 27

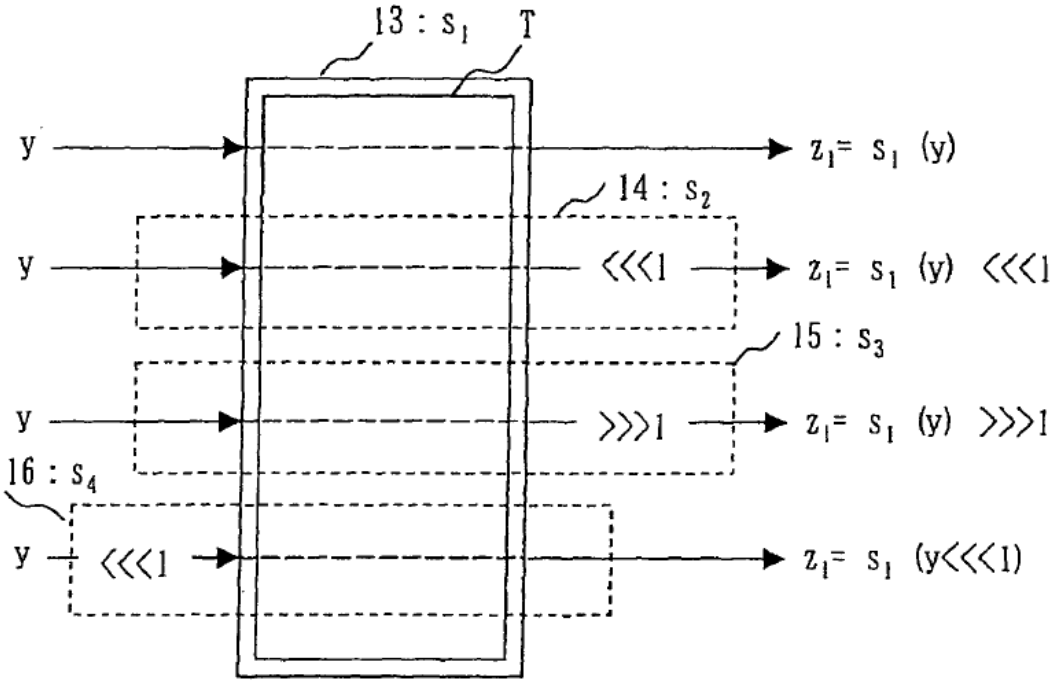


Fig. 28

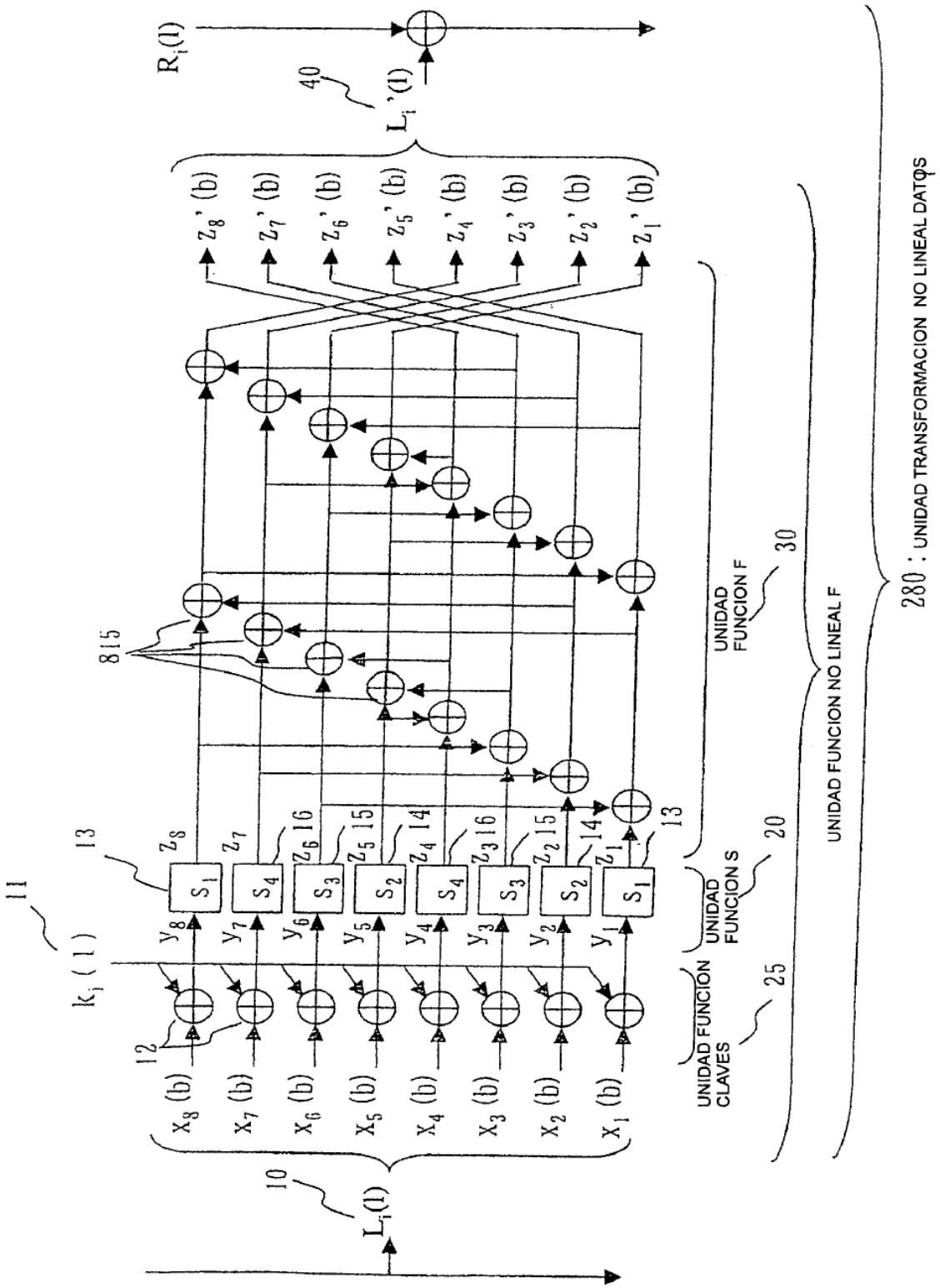
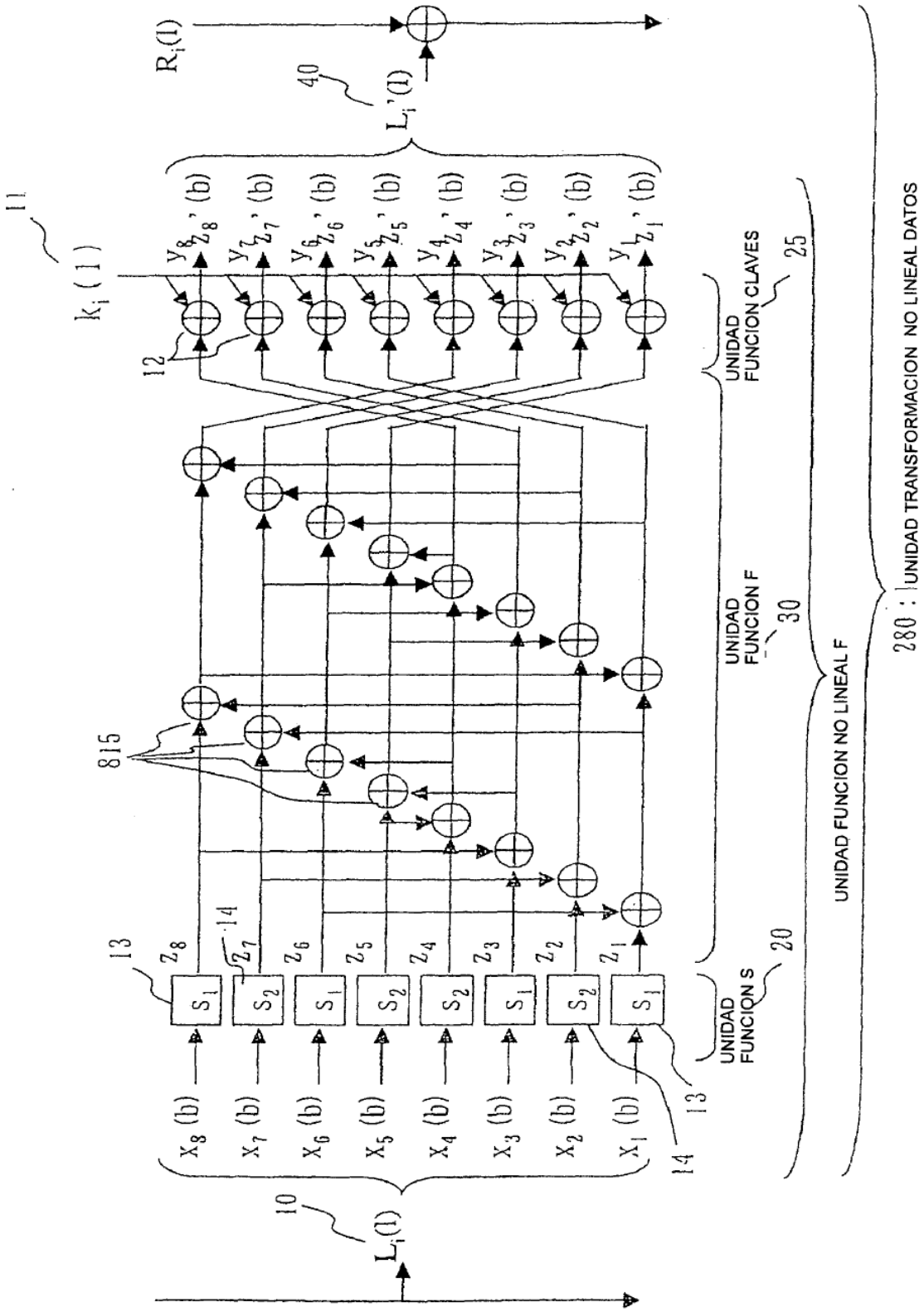

R_i(l)
40
L_i'(l)
z_8'(b)
z_7'(b)
z_6'(b)
z_5'(b)
z_4'(b)
z_3'(b)
z_2'(b)
z_1'(b)
815
13
11
k_i(l)
z_8
z_7
z_6 16
z_5 15
z_4 14
z_3 16
z_2 15
z_1 14
13
S_1
S_4
S_3
S_2
S_4
S_3
S_2
S_1
y_8
y_7
y_6
y_5
y_4
y_3
y_2
y_1
12
x_8(b)
x_7(b)
x_6(b)
x_5(b)
x_4(b)
x_3(b)
x_2(b)
x_1(b)
10
L_i(l)
UNIDAD FUNCION F
30
UNIDAD FUNCION S
20
UNIDAD FUNCION CLAVES
25
UNIDAD FUNCION NO LINEAL F
280 : UNIDAD TRANSFORMACION NO LINEAL DATOS

Fig. 29



$R_i(l)$
40
$L_i'(l)$
11
$k_i(l)$
12
y_8 z_8' (b)
y_7 z_7' (b)
y_6 z_6' (b)
y_5 z_5' (b)
y_4 z_4' (b)
y_3 z_3' (b)
y_2 z_2' (b)
y_1 z_1' (b)
UNIDAD FUNCION CLAVES
25
UNIDAD FUNCION F
30
UNIDAD FUNCION NO LINEAL F
280 : UNIDAD TRANSFORMACION NO LINEAL DATOS
815
13
14
z_8 z_7 z_6 z_5 z_4 z_3 z_2 z_1
S_1 S_2 S_1 S_2 S_2 S_1 S_2 S_1
UNIDAD FUNCION S
20
x_8 (b) x_7 (b) x_6 (b) x_5 (b) x_4 (b) x_3 (b) x_2 (b) x_1 (b)
10
$L_i(l)$

Fig. 30

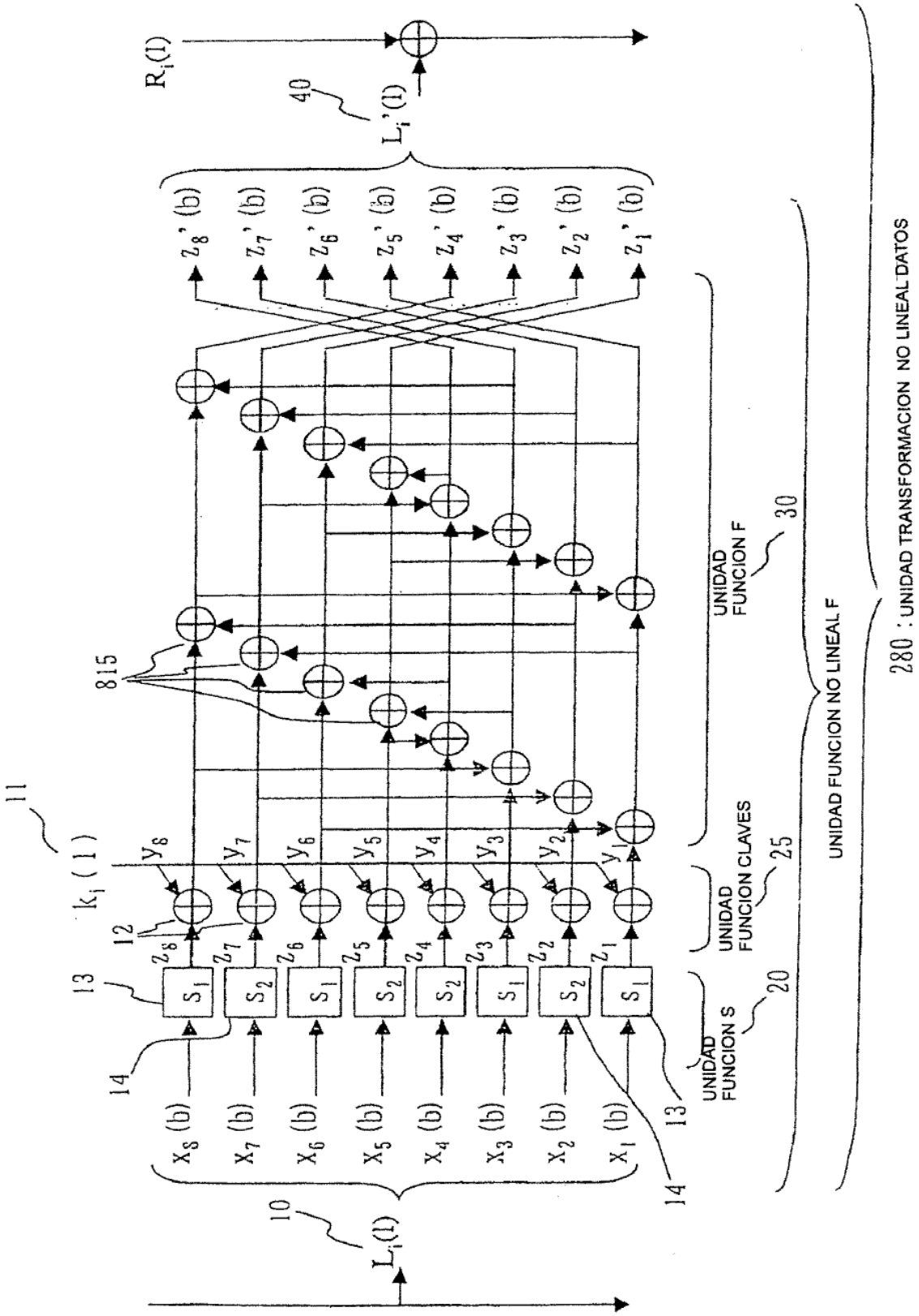

R_i(I)
40
L_i'(I)
z_8'(b)
z_7'(b)
z_6'(b)
z_5'(b)
z_4'(b)
z_3'(b)
z_2'(b)
z_1'(b)
815
11
k_i(I)
y_8
y_7
y_6
y_5
y_4
y_3
y_2
y_1
12
13
14
z_8
z_7
z_6
z_5
z_4
z_3
z_2
z_1
S_1
S_2
x_8(b)
x_7(b)
x_6(b)
x_5(b)
x_4(b)
x_3(b)
x_2(b)
x_1(b)
10
L_i(I)
UNIDAD FUNCION F
30
UNIDAD FUNCION CLAVES
25
UNIDAD FUNCION S
20
UNIDAD FUNCION NO LINEAL F
280 : UNIDAD TRANSFORMACION NO LINEAL DATOS

Fig. 31

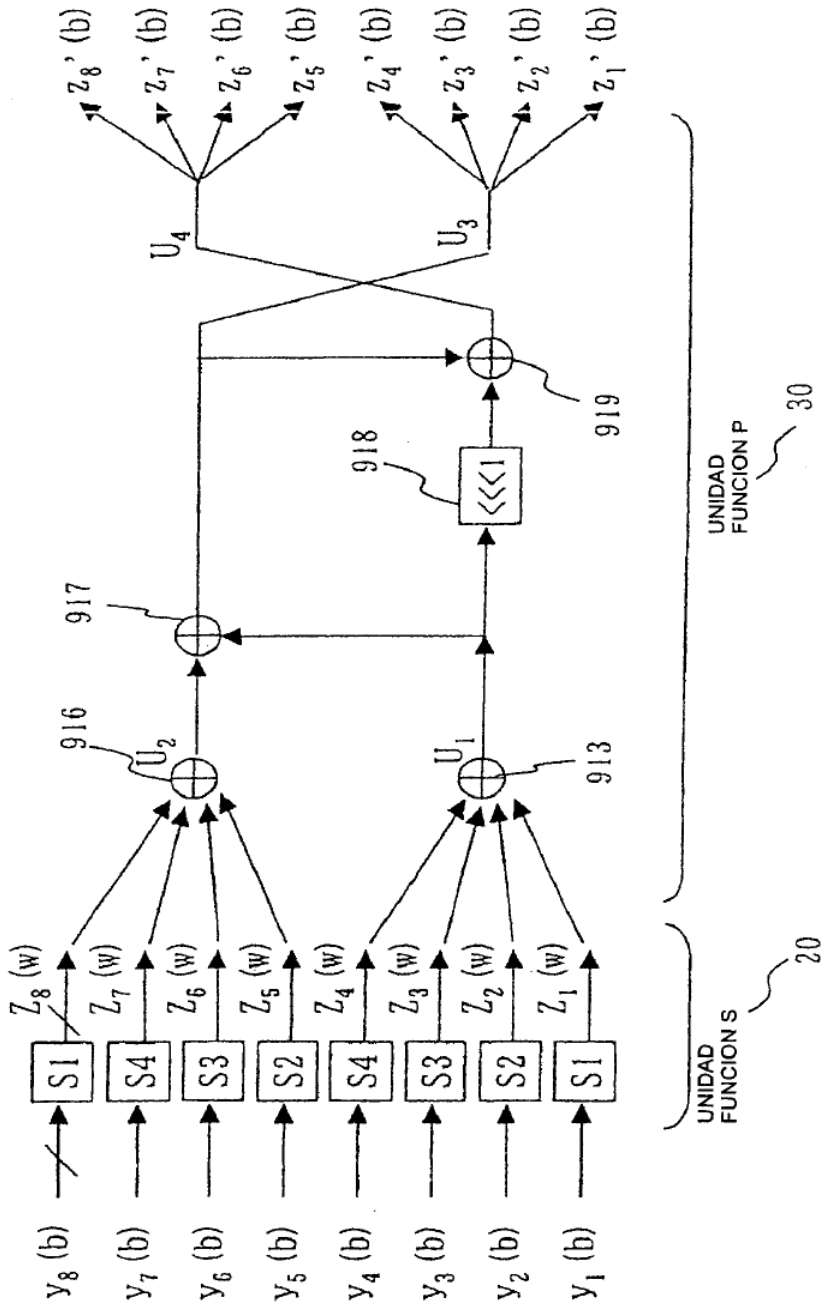

y8 (b)
y7 (b)
y6 (b)
y5 (b)
y4 (b)
y3 (b)
y2 (b)
y1 (b)
S1
S4
S3
S2
S4
S3
S2
S1
Z8 (w)
Z7 (w)
Z6 (w)
Z5 (w)
Z4 (w)
Z3 (w)
Z2 (w)
Z1 (w)
U2
U1
916
913
917
918
<<<1
919
U4
U3
z8' (b)
z7' (b)
z6' (b)
z5' (b)
z4' (b)
z3' (b)
z2' (b)
z1' (b)
UNIDAD FUNCION S
20
UNIDAD FUNCION P
30

Fig. 32

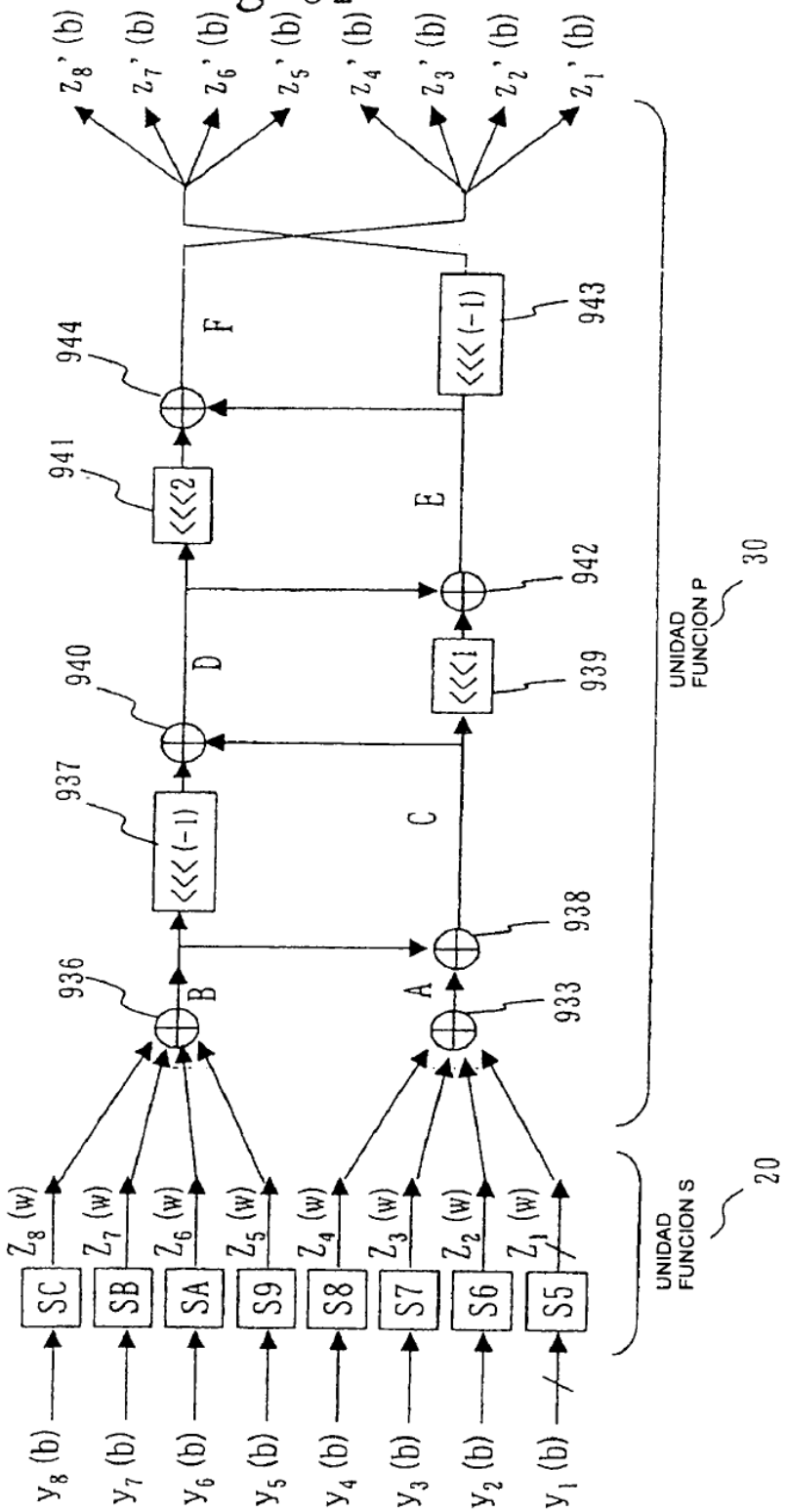

y8 (b)
y7 (b)
y6 (b)
y5 (b)
y4 (b)
y3 (b)
y2 (b)
y1 (b)
SC
SB
SA
S9
S8
S7
S6
S5
Z8 (w)
Z7 (w)
Z6 (w)
Z5 (w)
Z4 (w)
Z3 (w)
Z2 (w)
Z1 (w)
UNIDAD FUNCION S
20
936
B
937
<<<(-1)
940
D
941
<<<2
944
F
A
933
938
C
<<<1
939
942
E
<<<(-1)
943
UNIDAD FUNCION P
30
z8' (b)
z7' (b)
z6' (b)
z5' (b)
z4' (b)
z3' (b)
z2' (b)
z1' (b)

Fig. 33

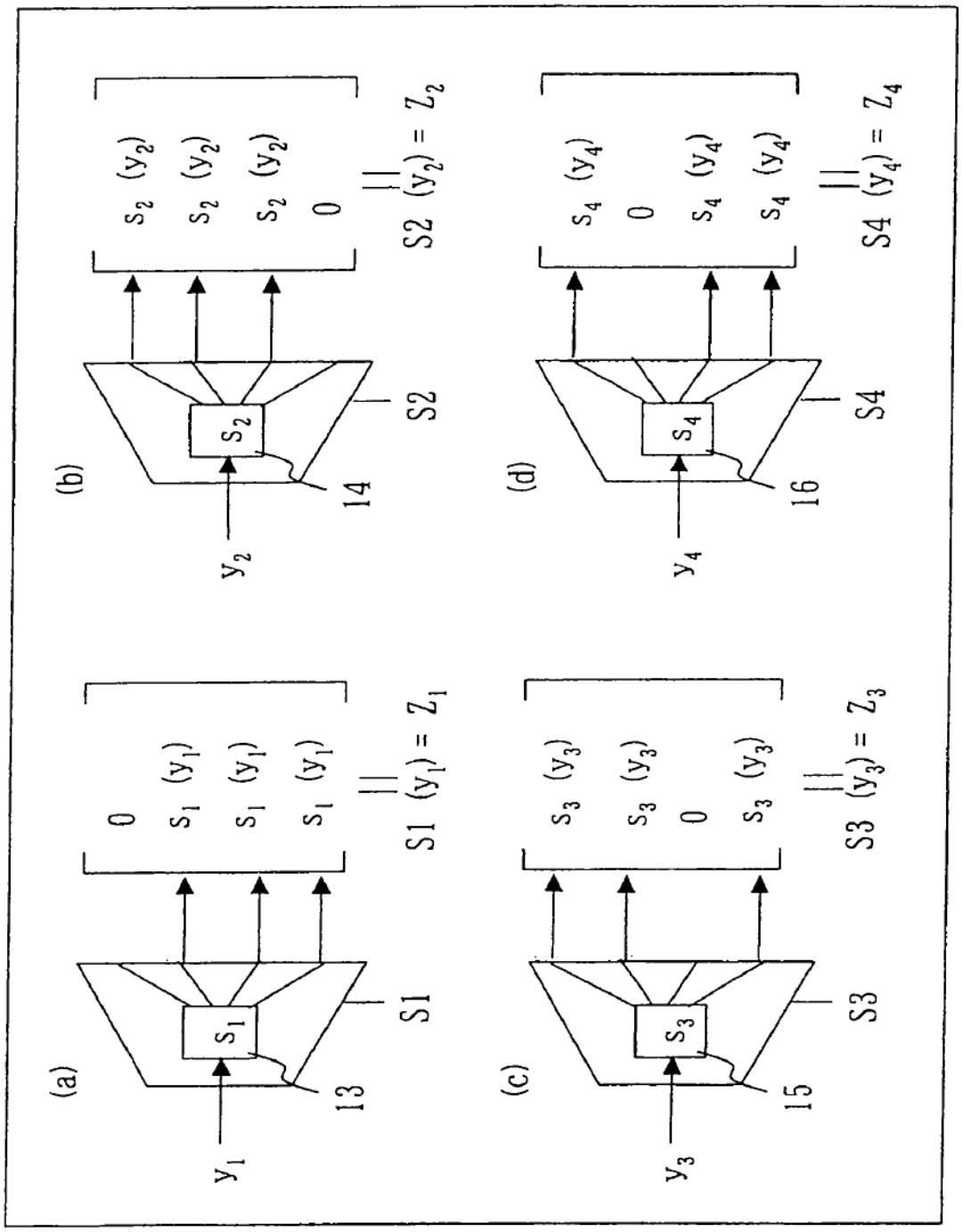

(a)
y1
s1
13
S1
0
s1 (y1)
s1 (y1)
s1 (y1)
S1 (y1) = Z1
(b)
y2
s2
14
S2
s2 (y2)
s2 (y2)
s2 (y2)
0
S2 (y2) = Z2
(c)
y3
s3
15
S3
s3 (y3)
s3 (y3)
0
s3 (y3)
S3 (y3) = Z3
(d)
y4
s4
16
S4
s4 (y4)
0
s4 (y4)
s4 (y4)
S4 (y4) = Z4

Fig.34

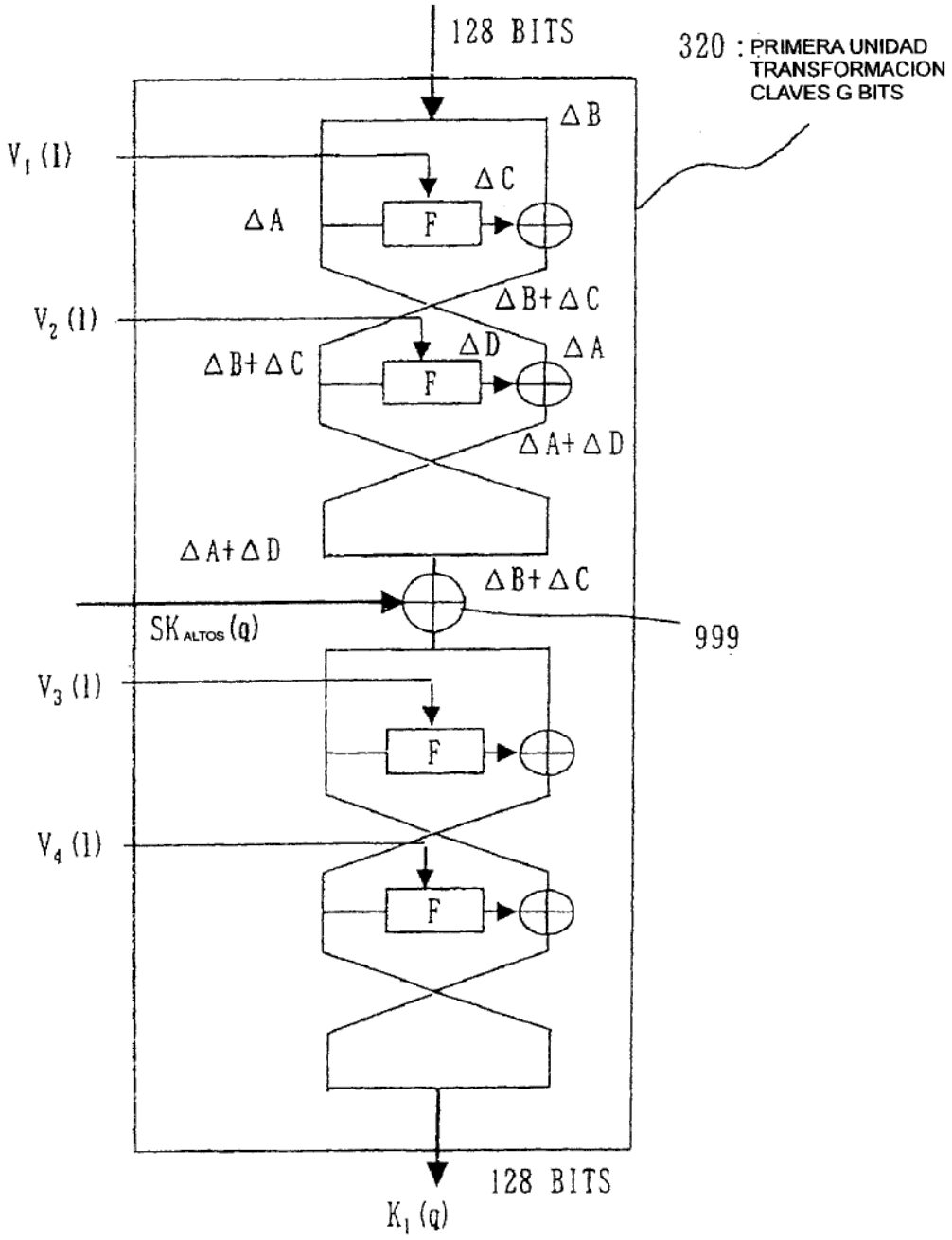

128 BITS
320 : PRIMERA UNIDAD TRANSFORMACION CLAVES G BITS
ΔB
V1 (I)
ΔC
ΔA
F
ΔB+ΔC
V2 (I)
ΔD
ΔA
ΔB+ΔC
F
ΔA+ΔD
ΔA+ΔD
ΔB+ΔC
SKALTOS (q)
999
V3 (I)
F
V4 (I)
F
128 BITS
K1 (q)

Fig.35

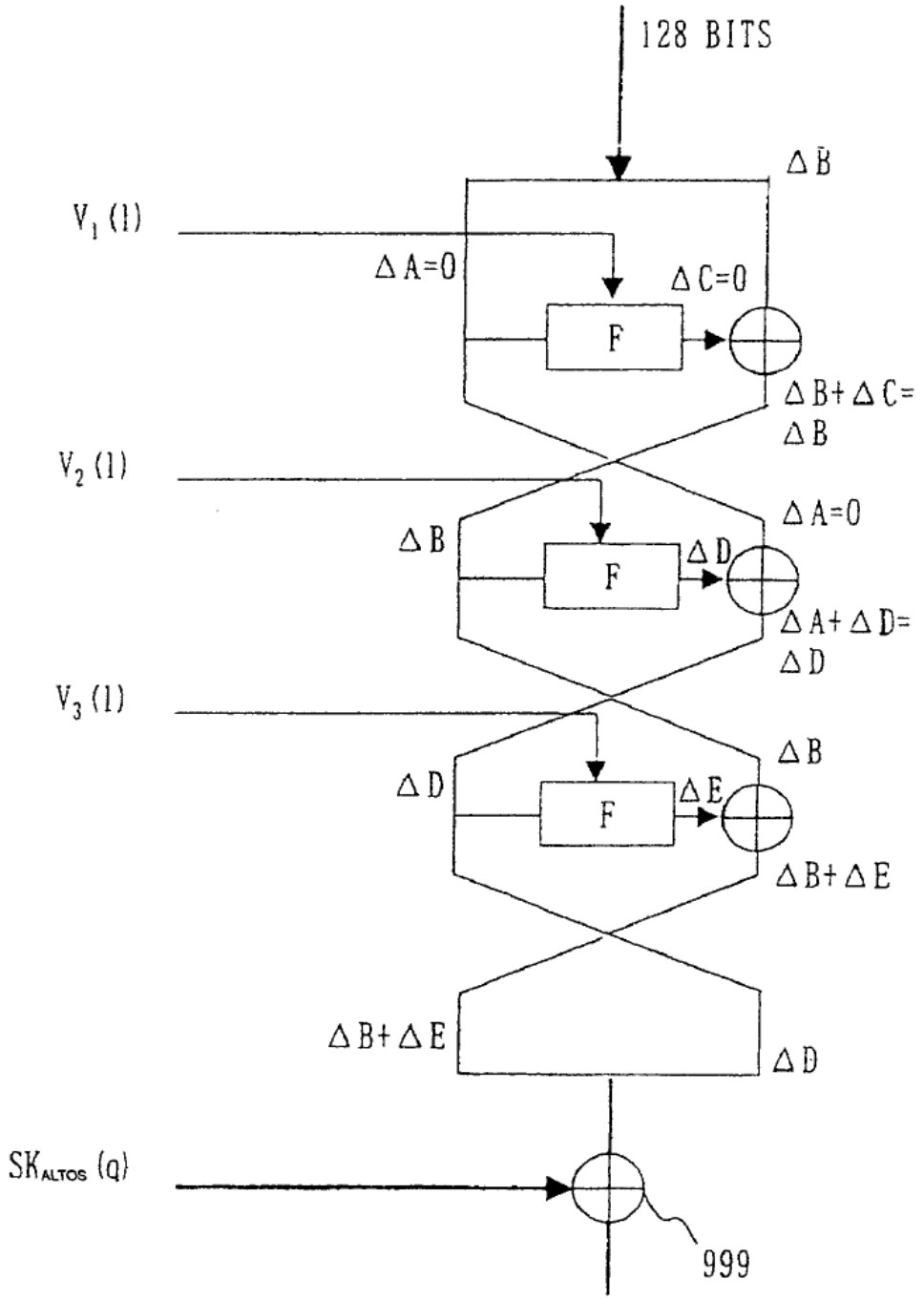

128 BITS
ΔB
V1 (1)
ΔA=0
ΔC=0
F
ΔB+ΔC=
ΔB
V2 (1)
ΔA=0
ΔB
F
ΔD
ΔA+ΔD=
ΔD
V3 (1)
ΔB
ΔD
F
ΔE
ΔB+ΔE
ΔB+ΔE
ΔD
SKALTOS (q)
999

Fig.36

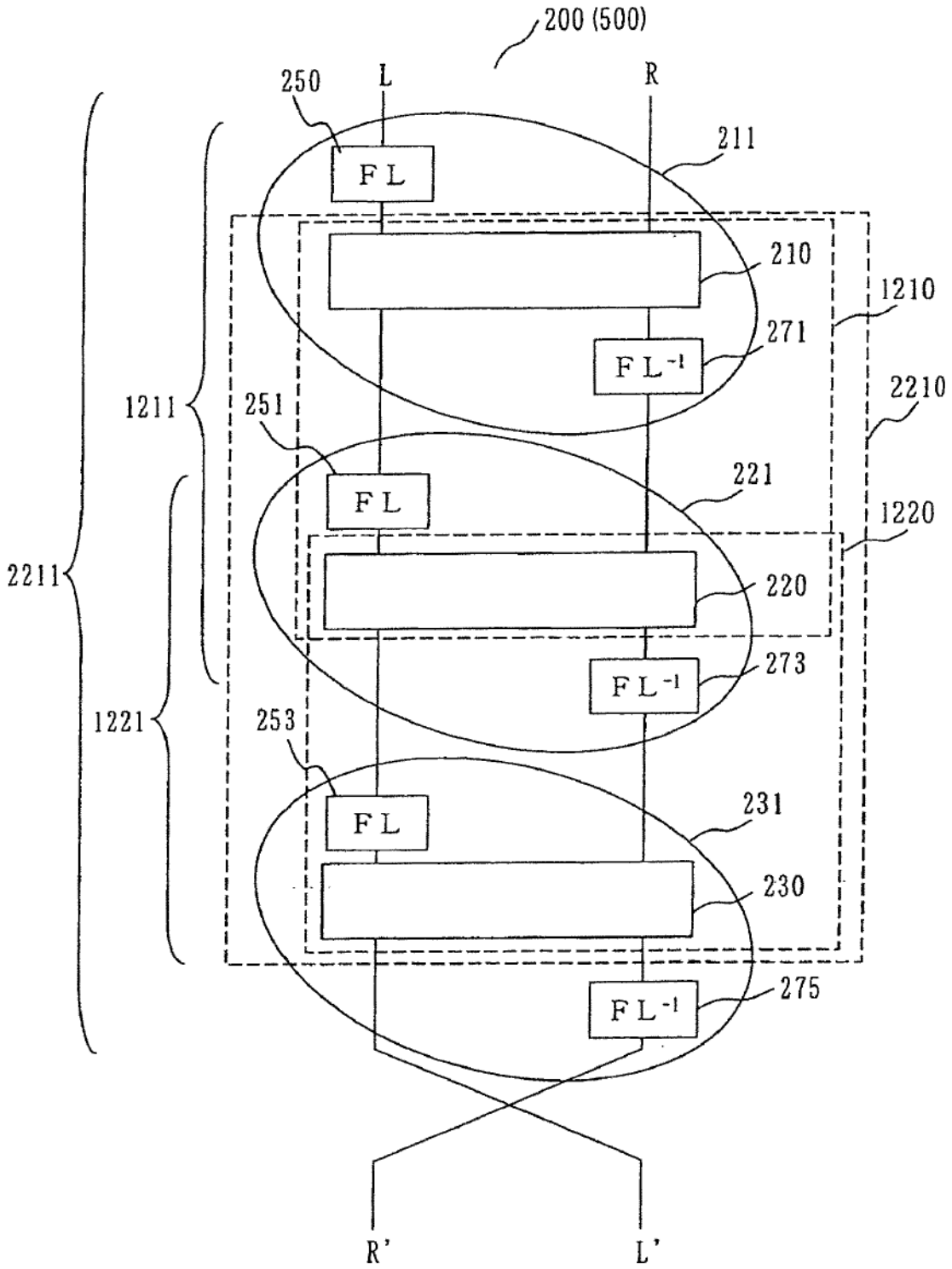

200 (500)
250
L
R
211
FL
210
1210
271
FL-1
2210
1211
251
221
FL
1220
2211
220
273
FL-1
1221
253
231
FL
230
275
FL-1
R'
L'

Fig. 37

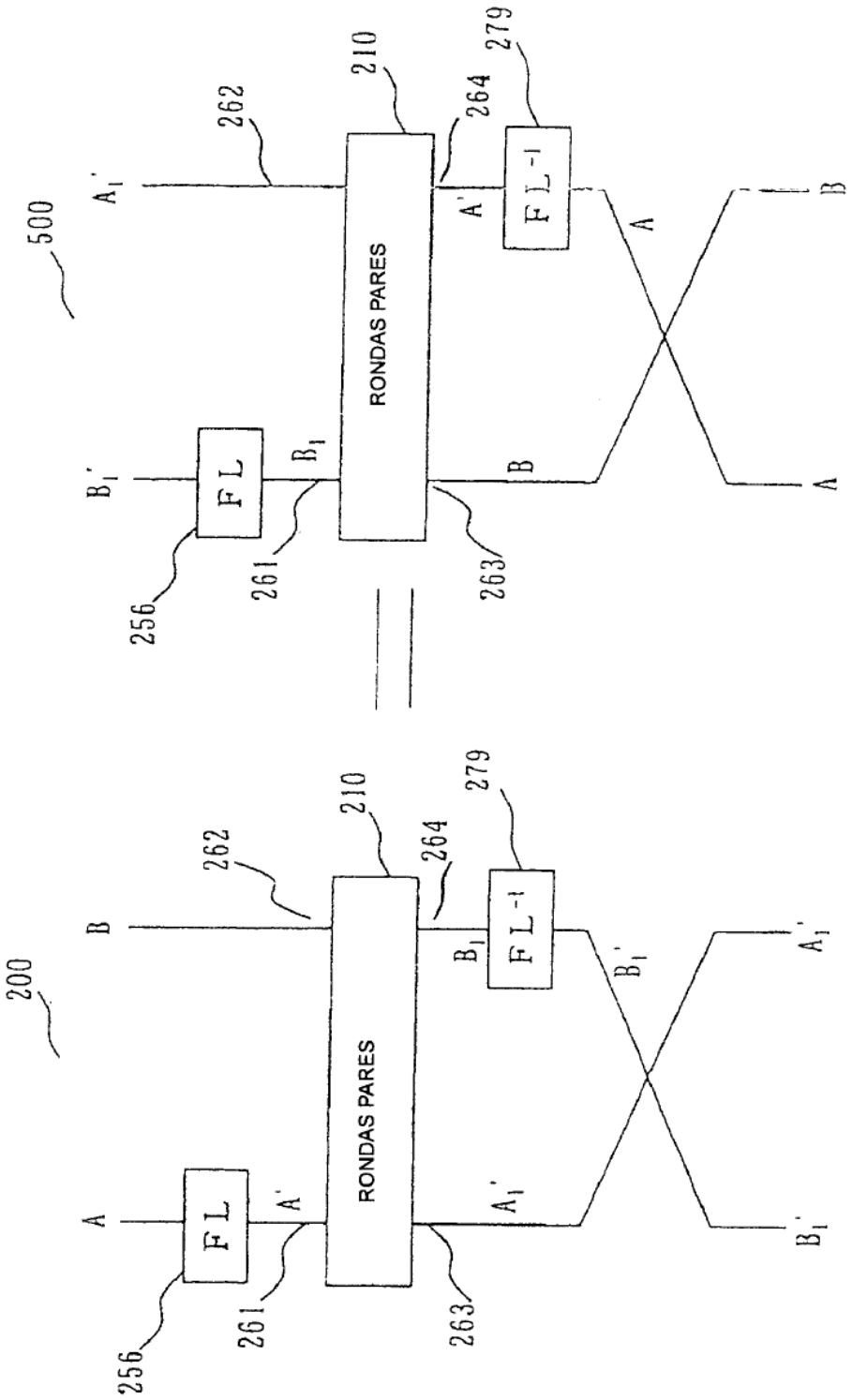

500
RONDAS PARES
FL
FL⁻¹
210
256
261
262
263
264
279
200
RONDAS PARES
FL
FL⁻¹
210
256
261
262
263
264
279

Fig. 38

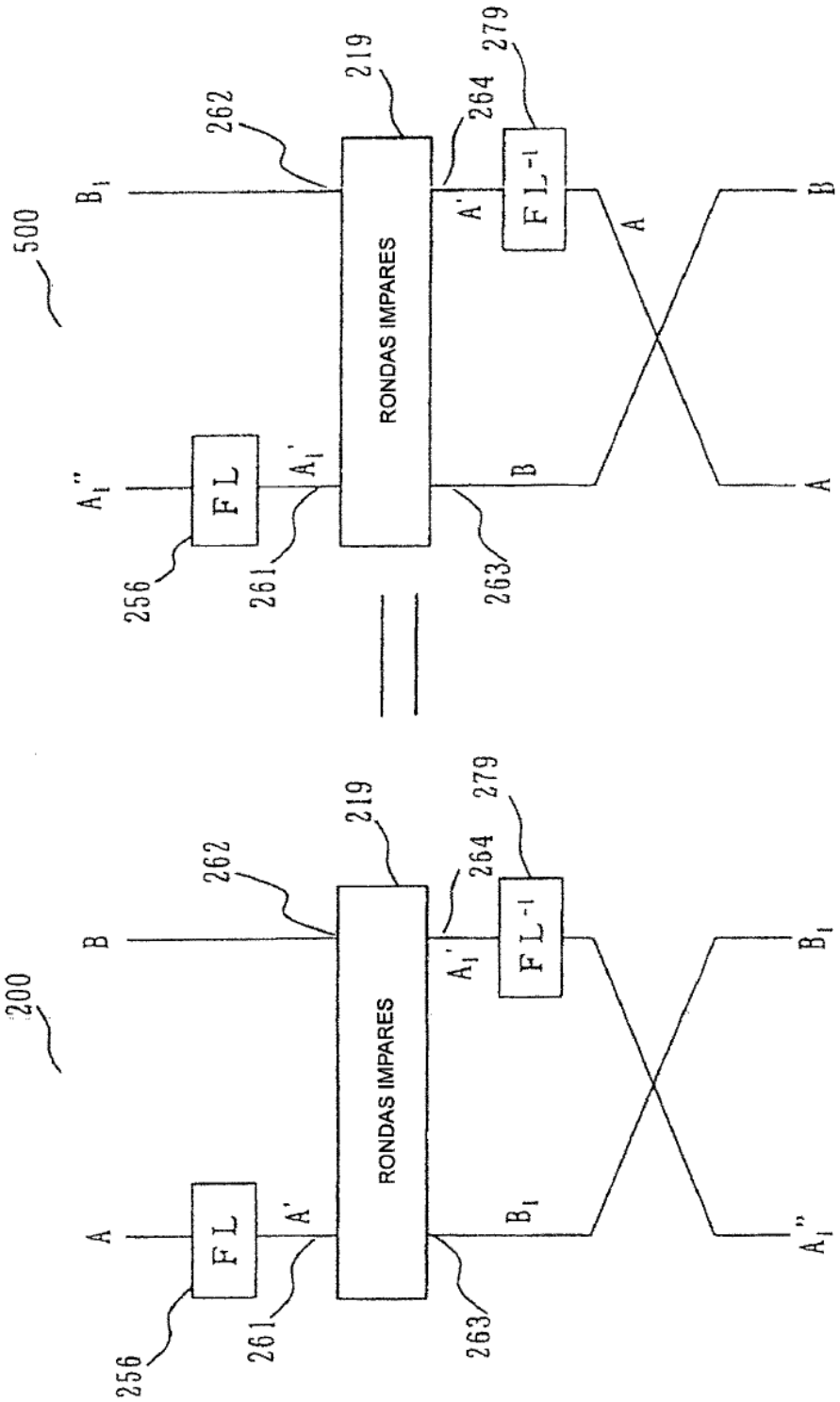

500
B_1
A_1''
FL
256
A_1'
261
262
219
RONDAS IMPARES
264
A'
FL^{-1}
279
A
B
263
B
A
200
B
A
FL
256
A'
261
262
219
RONDAS IMPARES
A_1'
264
FL^{-1}
279
B_1
B_1
A_1''
263

Fig. 39

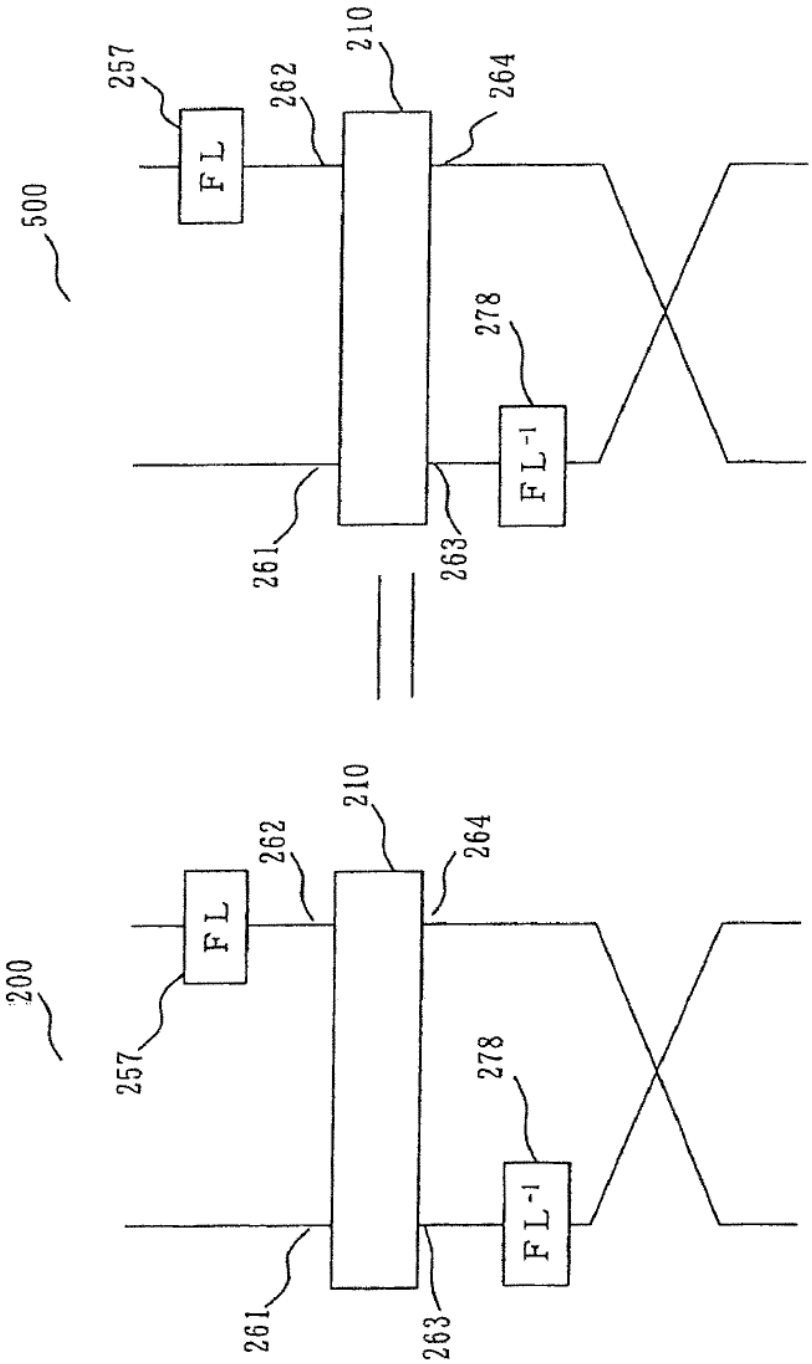

500
257
FL
262
210
264
261
263
278
FL-1
200
257
FL
262
210
264
261
263
278
FL-1

Fig. 40

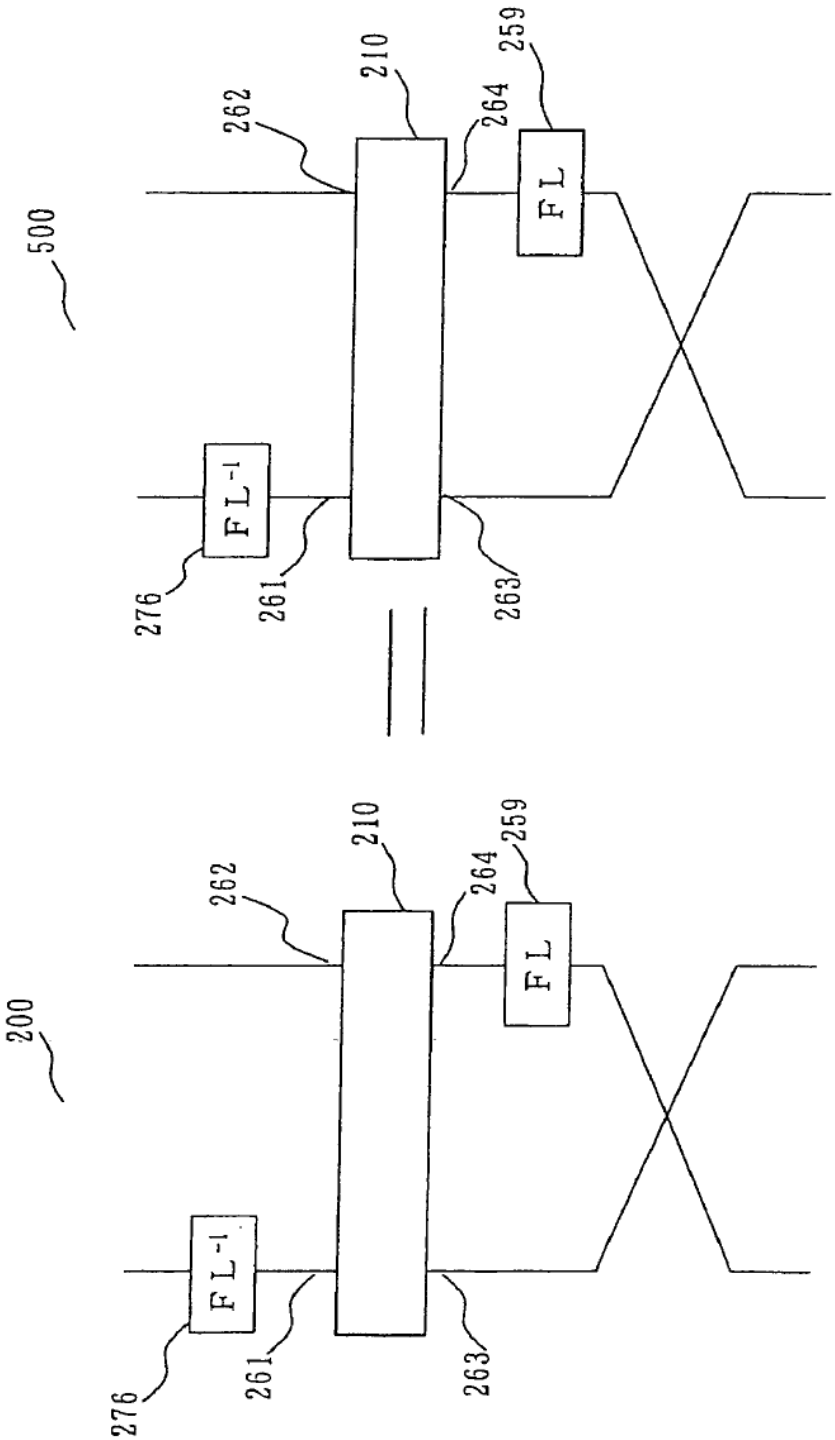

500
276
FL-1
261
262
210
263
264
FL
259
200
276
FL-1
261
262
210
263
264
FL
259

Fig. 41

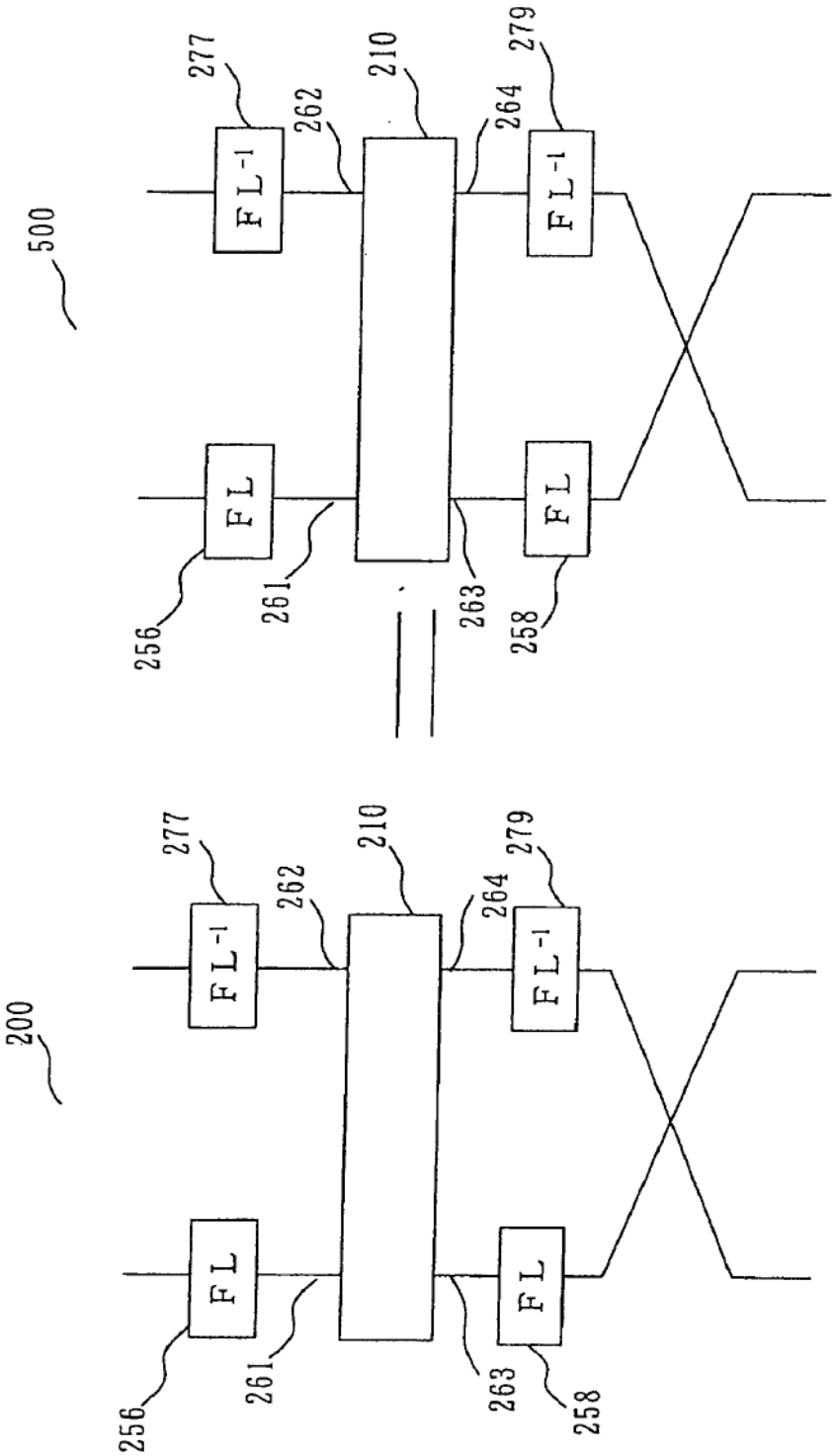


Fig. 42

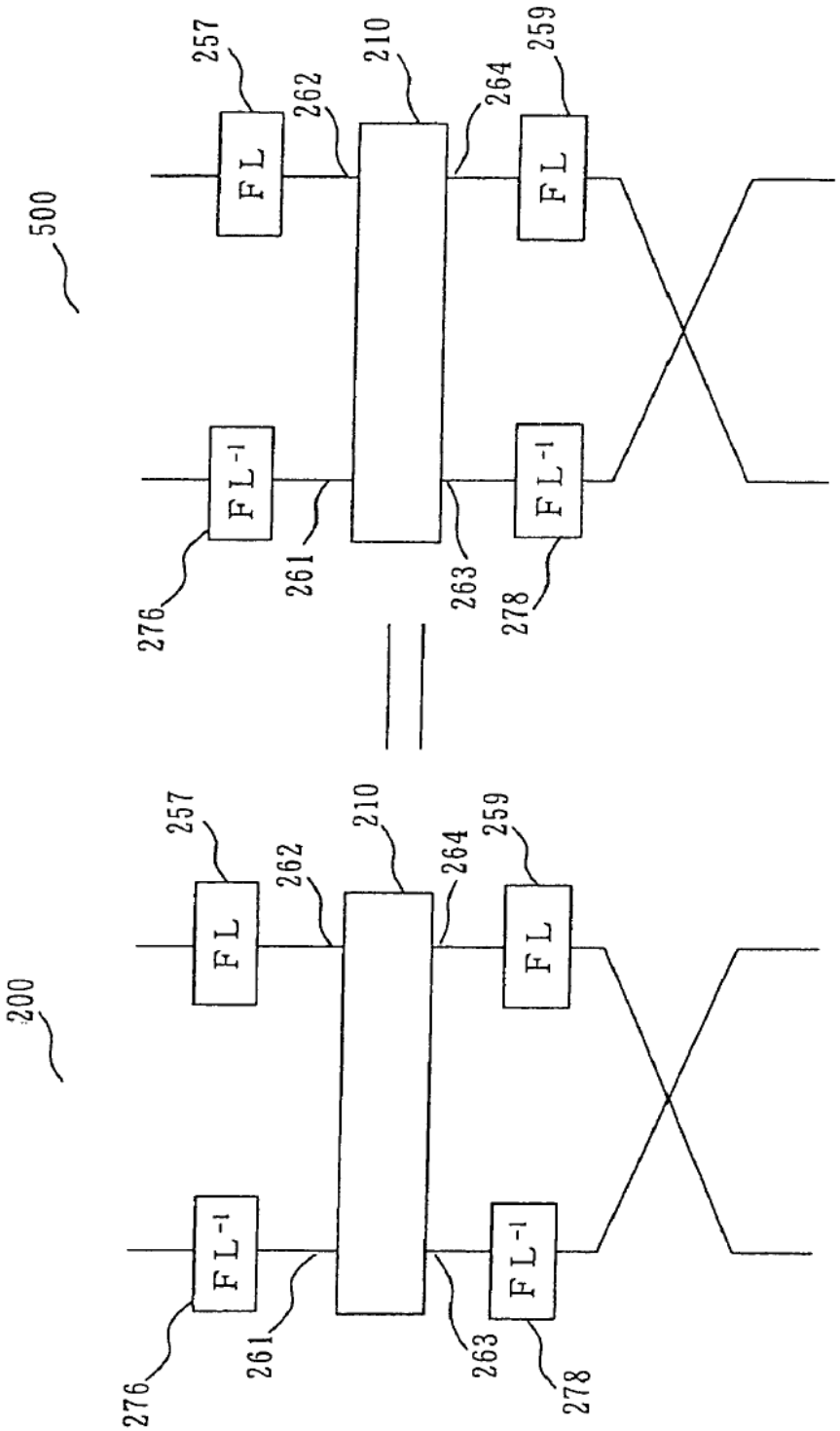


Fig. 43

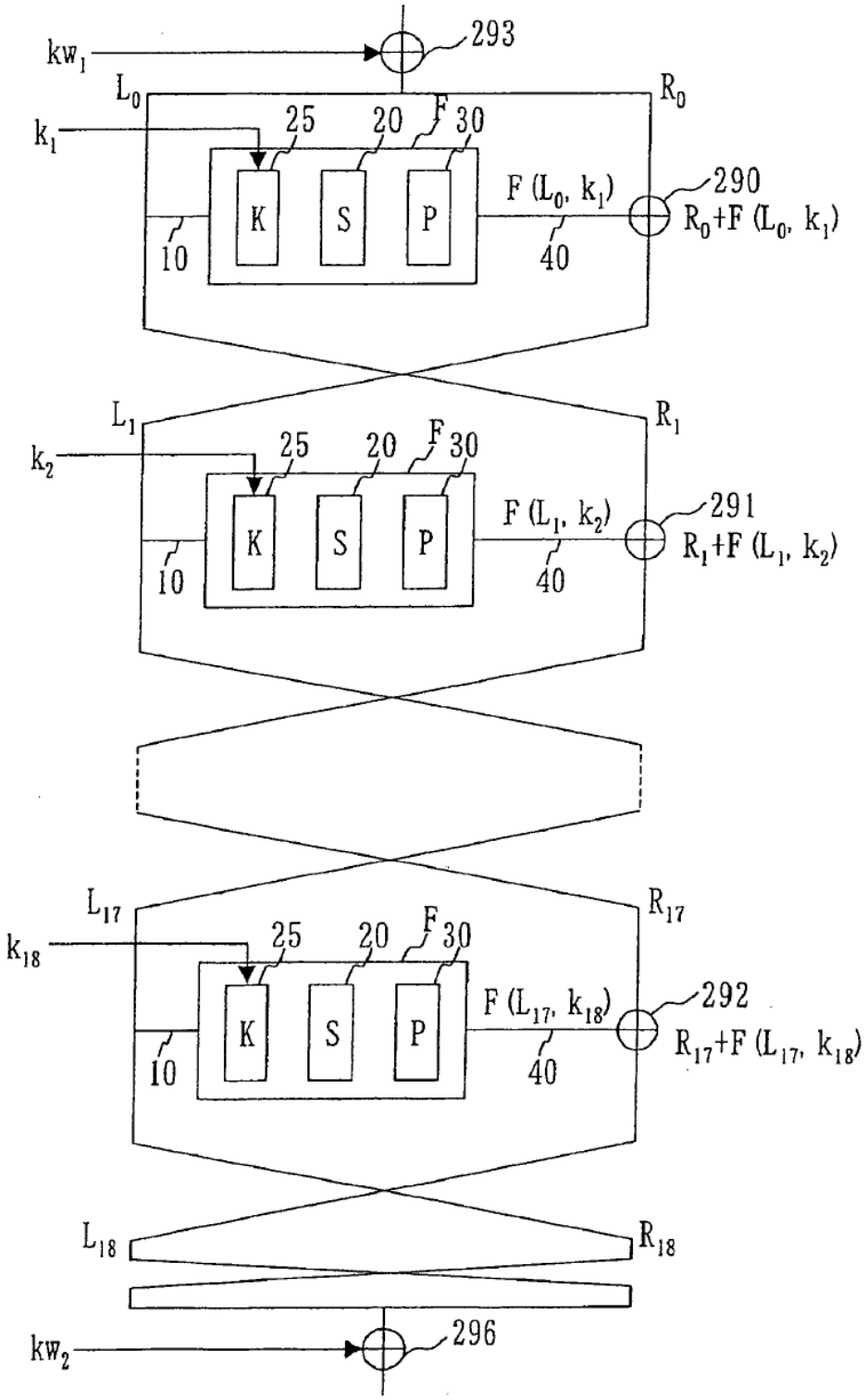

kw1
293
L0
R0
k1
25
20
F
30
K
S
P
F (L0, k1)
290
R0+F (L0, k1)
10
40
L1
R1
k2
F (L1, k2)
291
R1+F (L1, k2)
L17
R17
k18
F (L17, k18)
292
R17+F (L17, k18)
L18
R18
kw2
296

Fig. 44

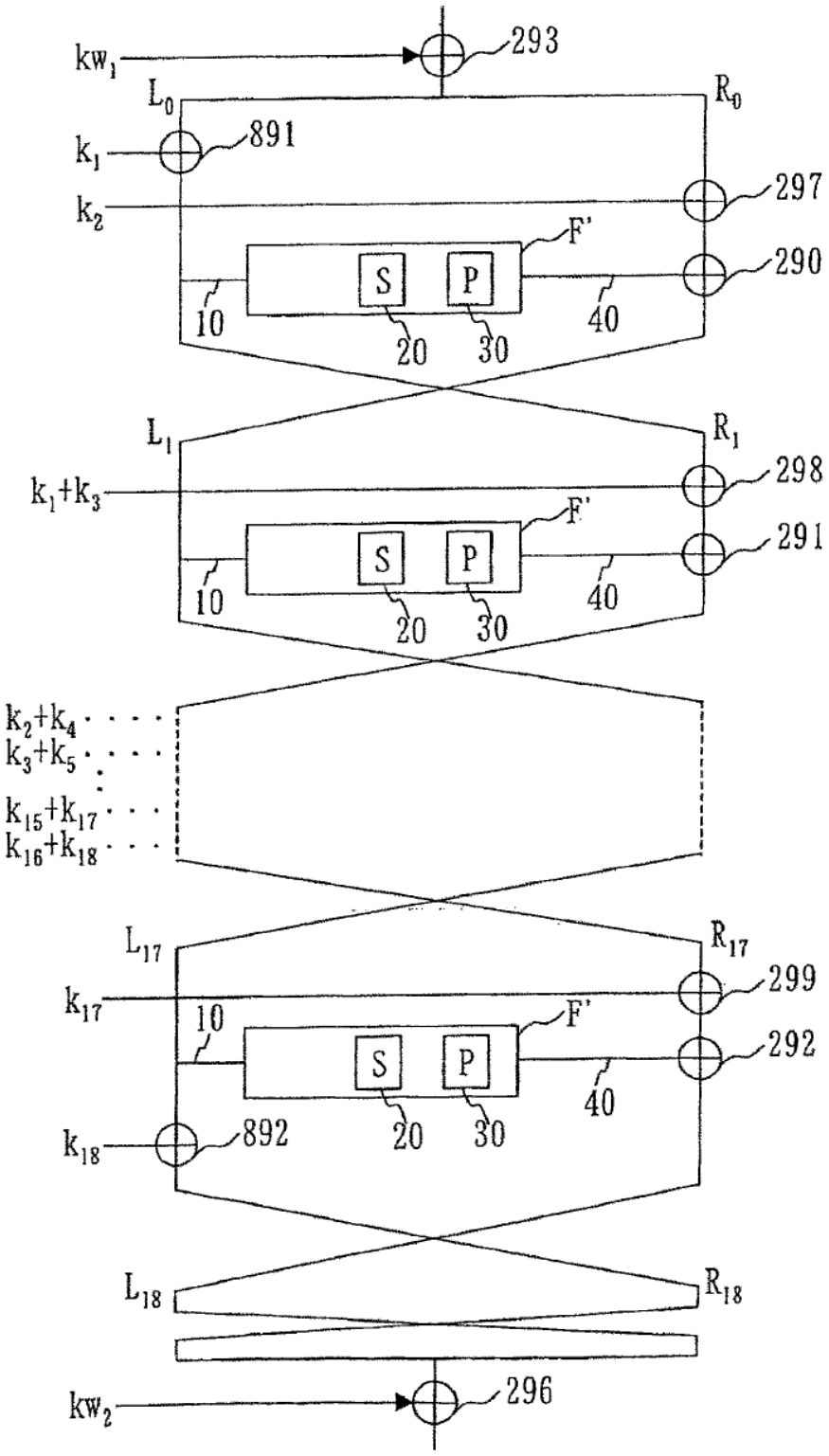

kw1
293
L0
R0
k1
891
k2
297
F'
290
S
P
10
20
30
40
L1
R1
k1+k3
298
F'
291
S
P
10
20
30
40
k2+k4
k3+k5
k15+k17
k16+k18
L17
R17
k17
299
10
F'
292
S
P
40
k18
892
20
30
L18
R18
kw2
296

Fig. 45

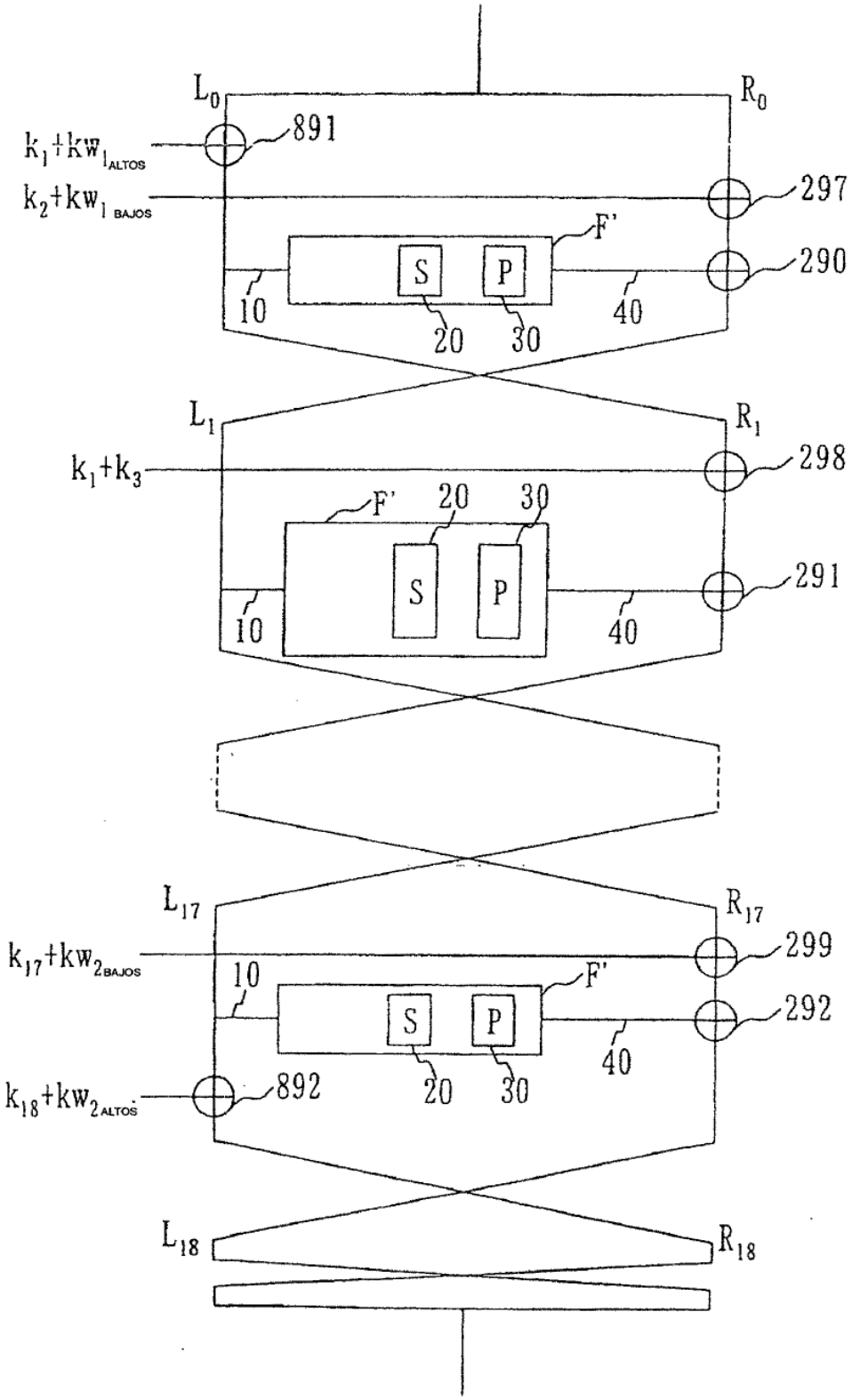

L0
R0
k1+kw1ALTOS
891
k2+kw1 BAJOS
297
F'
S
P
290
10
20
30
40
L1
R1
k1+k3
298
F'
20
30
S
P
291
10
40
L17
R17
k17+kw2BAJOS
299
10
F'
S
P
292
40
k18+kw2ALTOS
892
20
30
L18
R18

Fig. 46

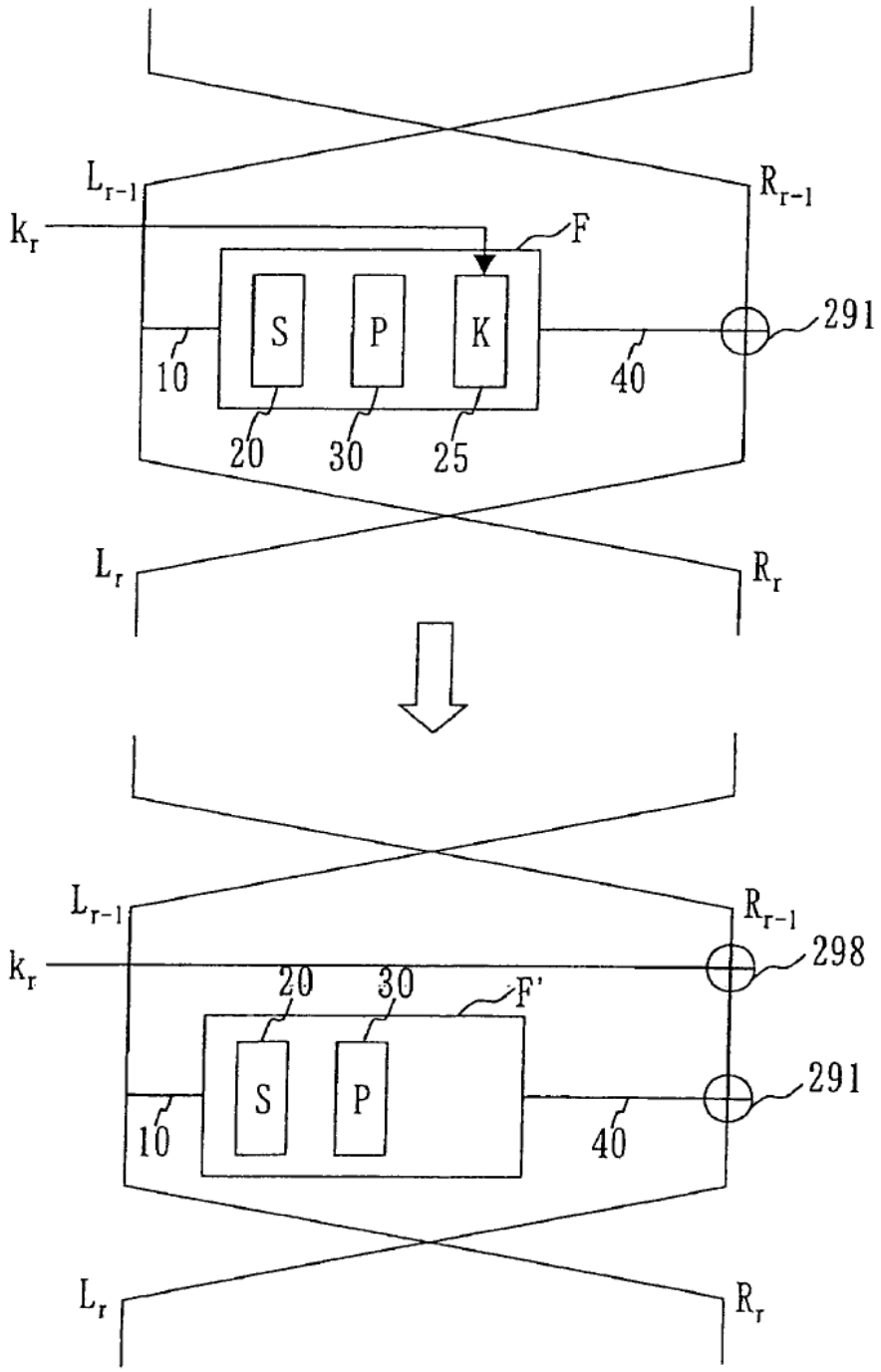
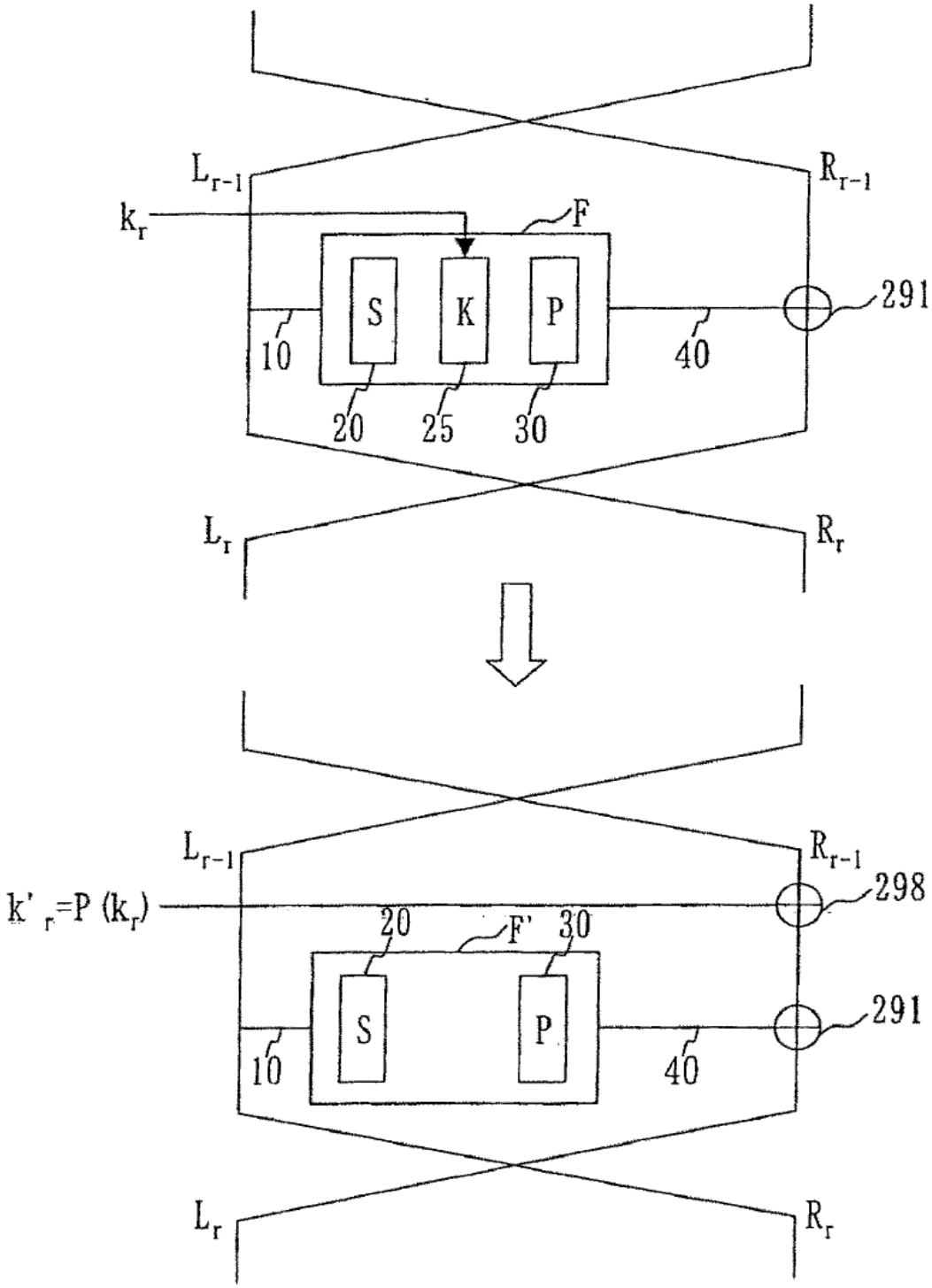

L_{r-1}
R_{r-1}
k_r
F
S
P
K
291
10
40
20
30
25
L_r
R_r
L_{r-1}
R_{r-1}
k_r
298
20
30
F'
S
P
291
10
40
L_r
R_r

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