(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 548 860**

(51) Int. CI.:

H04L 9/06 (2006.01)
H04L 9/32 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **22.12.2000 E 07075185 (4)**

(97) Fecha y número de publicación de la concesión europea: **15.07.2015 EP 1816782**

(54) Título: **Codificador, procedimiento de codificación, decodificador, procedimiento de decodificación y medio de grabación legible por ordenador que tiene almacenado un programa en el mismo**

(30) Prioridad:

14.01.2000 JP 2000005161

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
21.10.2015

(73) Titular/es:

MITSUBISHI DENKI KABUSHIKI KAISHA (100.0%)
2-3, MARUNOUCHI 2-CHOME, CHIYODA-KU
TOKYO 100-8310, JP

(72) Inventor/es:

SORIMACHI, TORU y
TOKITA, TOSHIO

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Codificador, procedimiento de codificación, decodificador, procedimiento de decodificación y medio de grabación legible por ordenador que tiene almacenado un programa en el mismo

Campo técnico

La presente invención versa sobre un aparato de codificación, un aparato de decodificación y procedimiento de codificación/decodificación; en particular, sobre una invención que permite codificar/decodificar otro dato mientras se codifica/decodifica cierto otro dato.

Técnica antecedente

La Fig. 43 muestra un diagrama de bloques de un codificador que realiza una codificación del modo de encadenamiento de bloques de cifrado (denominado en lo sucesivo modo CBC).

Una codificación del modo CBC se realiza como sigue: en primer lugar, la unidad de bloques introduce datos en bloques de texto en claro M_i de 64 bits; los datos de entrada son codificados por un módulo 51 de codificación usando una clave K de codificación; los datos en bloques de texto cifrado C_i y los datos en bloques de texto en claro M_{i+1}, después de los datos M_i, son objeto de una operación lógica de O excluyente; y el resultado objeto de una operación lógica de O excluyente es suministrado al módulo 51 de codificación para su codificación, usando la clave K de codificación, como siguiente entrada para el procedimiento de codificación. A continuación, este procedimiento se encadena reiteradamente, y la totalidad de los datos de texto en claro M será codificada formando los datos de texto cifrado C.

La Fig. 44 muestra un diagrama de bloques de un aparato de decodificación que realiza una decodificación del modo CBC.

El aparato de decodificación mostrado en la Fig. 44 es un aparato para decodificar los datos de texto cifrado codificados por el aparato de codificación mostrado en la Fig. 43. Los datos en bloques de texto cifrado C_1 son introducidos en un módulo 71 de decodificación para su decodificación, usando la clave K de codificación, objeto de una operación lógica de O excluyente con un valor inicial IV, y decodificados formando datos en bloques de texto en claro M_1. Cuando se introducen los datos en bloques de texto cifrado C_2, el módulo 71 de decodificación decodifica los datos en bloques C_2 usando la clave K de codificación, objeto de una operación lógica de O excluyente con los datos en bloques de texto cifrado C_1, que han sido previamente introducidos y almacenados en un registro 111, y decodificados formando datos en bloques de texto en claro M_2.

Aquí, se puede proporcionar el registro 111 dentro de un selector 73.

El modo CBC puede representarse por medio de las expresiones siguientes, en las que los datos en bloques de texto en claro son M_i (i = 1, 2, …, n), los datos en bloques de texto cifrado C_i (i = 1, 2, …, n), el procedimiento de codificación que usa la clave K de codificación está definido como E_k, y el procedimiento de decodificación que usa la clave K de codificación está definido como D_k:

$C_1 = E_k (M_1 \text{ EXR } IV)$
$C_i = E_k (M_i \text{ EXR } C_{i-1})$ (i = 2, 3, …, n)
$M_1 = D_k (C_1) \text{ EXR } IV$
$M_i = D_k (C_i) \text{ EXR } C_{i-1}$ (i = 2, 3, …, n)

Aquí, EXR representa una operación de O excluyente. IV representa un valor inicial que ha de usarse para una etapa inicial de los procedimientos de codificación y de decodificación. Se usa el mismo valor inicial IV tanto en el codificador como en el decodificador.

La Fig. 45 muestra un codificador que realiza una codificación del modo de retroalimentación de salidas (denominado en lo sucesivo modo OFB).

La Fig. 46 muestra un decodificador que realiza una decodificación del modo OFB.

La Fig. 47 muestra un codificador que realiza una codificación del modo de retroalimentación de cifrado (denominado en lo sucesivo modo CFB).

La Fig. 48 muestra un decodificador que realiza una decodificación según el modo CFB.

Aquí, se puede proporcionar el registro 111 dentro del selector 73.

La Fig. 49 es un diagrama de bloques que muestra un procedimiento para codificar datos de texto en claro M y datos de texto en claro N usando el codificador del modo CBC.

En lo que sigue se explicará un caso en el que los datos de texto en claro M incluyen datos en bloques de texto en claro M_1, datos en bloques de texto en claro M_2, y datos de texto en claro M_3, y los datos de texto en claro N incluyen únicamente datos en bloques de texto en claro N_1.

Cuando se inicia la codificación de datos en bloques de texto en claro M_1, se producen datos en bloques de texto cifrado C_1, y los datos en bloques de texto cifrado C_1 son también usados para el procedimiento de codificación de datos en bloques de texto en claro M_2. Así, los datos en bloques de texto cifrado C_i son retroalimentados al procedimiento de codificación de datos en bloques de texto en claro M_{i+1}, lo que forma un procedimiento en cadena. En consecuencia, no es posible codificar los datos en bloques de texto en claro N_1 a no ser que haya acabado el procedimiento de codificación de los datos en bloques de texto en claro M_1 hasta los datos en bloques de texto en claro M_3.

La Fig. 50 muestra el procedimiento de codificación del modo CBC, al igual que la Fig. 49.

En el caso de la Fig. 50, lleva mucho tiempo preparar cada uno de los datos en bloques de texto en claro M_1, los datos en bloques de texto en claro M_2, y los datos en bloques de texto en claro M_3. Mientras, la codificación ha terminado antes de que se preparen los siguientes datos en bloques de texto en claro M_{i+1}, lo que genera un tiempo en reposo (el tiempo de T1 a T2, T3 a T4). Así, aunque se genere un tiempo en reposo, tiene que llevarse a cabo el procedimiento en cadena, de modo que los datos en bloques de texto cifrado C_i deberían ser retroalimentados en el procedimiento de codificación de los datos de texto en claro M_{i+1}. Por lo tanto, no puede realizarse el procedimiento para los datos en bloques de texto en claro N_1 hasta que termine el procedimiento de codificación de los datos en bloques de texto en claro M_3.

La Fig. 51 muestra un procedimiento de confidencialidad de datos y un procedimiento de garantía de la integridad de datos. Los datos de texto en claro M, por ejemplo, son codificados en los datos de texto cifrado C por el codificador del modo OFB. El codificador del modo CBC calcula un código de autenticación de mensajes (MAC) P, que es añadido al último bit de los datos de texto cifrado C. En caso de recibir datos que están codificados y a los que se añada el MAC P, así como de que el decodificador del modo OFB decodifique los datos de texto cifrado C formando los datos de texto en claro M, el MAC P se calcula a partir de los datos de texto cifrado C por el decodificador del modo CBC. Es posible confirmar que los datos de texto cifrado C transmitidos no han sido manipulados comparando el MAC P obtenido con el MAC P transmitido y recibido.

La Fig. 52 muestra un procedimiento para el procedimiento de confidencialidad y el procedimiento del cálculo del MAC mostrados en la Fig. 51.

Los datos en bloques de texto en claro M_1 hasta los datos en bloques de texto en claro M_3 son codificados en serie formando los datos en bloques de texto cifrado C_1 hasta los datos en bloques de texto cifrado C_3. Posteriormente, se calcula el MAC P introduciendo en serie los datos en bloques de texto cifrado C_1 hasta los datos en bloques de texto cifrado C_3.

El codificador y el decodificador de cada modo mostrado en las Figuras 42 a 48 tienen el problema siguiente: los datos obtenidos por el procedimiento de codificación y de decodificación de los datos del bloque anterior deberían ser retroalimentados y usados para codificar y decodificar los datos del bloque siguiente; existe el problema de que, una vez que se inicie el procedimiento de codificación o el procedimiento de decodificación, no puede comenzar otro procedimiento de codificación u otro procedimiento de decodificación a no ser que finalicen todas las etapas del procedimiento de codificación o del procedimiento de decodificación. En consecuencia, si el procedimiento de codificación/decodificación que ya se ha iniciado previamente requiere mucho tiempo, el subsiguiente procedimiento de codificación/decodificación debería esperar mucho tiempo.

Además, en caso de realizar el procedimiento de confidencialidad y el procedimiento de garantía de la integridad, el procedimiento de garantía de la integridad debería llevarse a cabo tras realizar el procedimiento de confidencialidad, que lleva mucho tiempo de proceso.

El documento US 5 764 762 da a conocer un registro de paquetes de datos codificados para su uso en un sistema remoto de datos transaccionales medidos.

Es un objeto de la realización preferente de la presente invención obtener un codificador, un decodificador, un procedimiento de codificación y un procedimiento de decodificación que puedan realizar un procedimiento de codificación/decodificación de otro dato mientras se realiza el procedimiento de codificación/decodificación de cierto dato.

Además, es otro objeto de la realización preferente de la presente invención realizar la codificación/decodificación de los datos que tengan una prioridad mayor que otros datos.

Además, es otro objeto de la realización preferente de la presente invención realizar el procedimiento de confidencialidad y el procedimiento de garantía de la integridad en paralelo a velocidad elevada.

Divulgación de la invención

Según la presente invención, se proporcionan procedimientos, aparatos y medios de almacenamiento legibles por ordenador según las reivindicaciones independientes.

Breve descripción de los dibujos

La Fig. 1 muestra un codificador del modo CBC según la primera realización.
La Fig. 2 muestra un procedimiento de operación del codificador del modo CBC.
La Fig. 3 es un diagrama de flujo que muestra la operación del codificador del modo CBC.
La Fig. 4 es un diagrama de flujo que muestra la operación de un selector 54.
La Fig. 5 es un diagrama de flujo que muestra un procedimiento de interrupción de un conmutador 57.
La Fig. 6 muestra otro ejemplo de una memoria 55.
La Fig. 7 es un diagrama de flujo que muestra un procedimiento de interrupción de la memoria 55.
La Fig. 8 muestra otro ejemplo de la memoria 55.
La Fig. 9 muestra un procesamiento de prioridades.
La Fig. 10 muestra otro procesamiento de prioridades.
La Fig. 11 muestra otro procesamiento de prioridades.
La Fig. 12 muestra un caso en el que se proporciona la memoria 55 en paralelo con una línea 66 de retroalimentación.
La Fig. 13 muestra un procedimiento de operación del codificador de la Fig. 12.
La Fig. 14 muestra un caso en el que se proporciona la memoria 55 en paralelo con una línea 67 de retroalimentación.
La Fig. 15 muestra un procedimiento de operación del codificador de la Fig. 14.
La Fig. 16 muestra un codificador del modo OFB.
La Fig. 17 muestra un procedimiento de operación del codificador de la Fig. 16.
La Fig. 18 muestra un codificador del modo CFB.
La Fig. 19 muestra un procedimiento de operación del codificador de la Fig. 18.
La Fig. 20 muestra un decodificador del modo CBC.
La Fig. 21 muestra un procedimiento de operación del decodificador de la Fig. 20.
La Fig. 22 muestra un decodificador del modo OFB.
La Fig. 23 muestra un procedimiento de operación del decodificador de la Fig. 22.
La Fig. 24 muestra un decodificador del modo CFB.
La Fig. 25 muestra un procedimiento de operación del decodificador de la Fig. 24.
La Fig. 26 muestra un codificador del modo CBC que almacena una clave.
La Fig. 27 muestra un procedimiento de operación del codificador del modo CBC.
La Fig. 28 muestra un decodificador del modo CBC que almacena una clave.
La Fig. 29 muestra un procedimiento de operación del decodificador del modo CBC.
La Fig. 30 muestra un procedimiento de operación de un codificador que tiene una unidad 100 de codificación y un generador 200 de MAC.
La Fig. 31 muestra un diagrama de flujo de un codificador que tiene una unidad 100 de codificación y un generador 200 de MAC.
La Fig. 32 muestra un codificador en el que una unidad 100 de codificación y un generador 200 de MAC están unidos como una sola unidad.
La Fig. 33 muestra un procedimiento de operación del codificador en el que una unidad 100 de codificación y un generador 200 de MAC están unidos como una sola unidad.
La Fig. 34 muestra un decodificador que tiene una unidad 300 de decodificación y un generador 400 de MAC.
La Fig. 35 muestra un decodificador en el que una unidad 300 de decodificación y un generador 400 de MAC están unidos como una sola unidad.
La Fig. 36 muestra un procedimiento de operación del decodificador en el que una unidad 300 de decodificación y un generador 400 de MAC están unidos como una sola unidad.
La Fig. 37 muestra un codificador que tiene una unidad 100 de codificación y un generador 200 de MAC según la segunda realización.
La Fig. 38 muestra un decodificador que tiene una unidad 300 de decodificación y un generador 400 de MAC.
La Fig. 39 muestra una configuración modelo de un módulo 51 de codificación que usa una clave K de codificación.
La Fig. 40 muestra un ejemplo de implementación de un soporte físico de un codificador y un decodificador.
La Fig. 41 muestra un ejemplo de implementación de un soporte físico de un codificador y un decodificador.
La Fig. 42 muestra un caso en el que un programa 47 de cifrado es llamado por un programa 46 de aplicación.
La Fig. 43 muestra un codificador convencional del modo CBC.
La Fig. 44 muestra un decodificador convencional del modo CBC.
La Fig. 45 muestra un codificador convencional del modo OFB.
La Fig. 46 muestra un decodificador convencional del modo OFB.

La Fig. 47 muestra un codificador convencional del modo CFB.
La Fig. 48 muestra un decodificador convencional del modo CFB.
La Fig. 49 muestra un procedimiento convencional de codificación.
La Fig. 50 muestra un procedimiento convencional de codificación.
La Fig. 51 explica un procedimiento de confidencialidad y un procedimiento de garantía de la integridad.
La Fig. 52 muestra un procedimiento de operación de un procedimiento convencional de confidencialidad y de un procedimiento convencional de garantía de la integridad.

Mejor modo de realización de la invención

Realización 1

La Fig. 1 muestra un codificador del modo CBC según la presente realización.

El codificador de la presente realización está configurado por un selector 54, un circuito 58 de O excluyente, un módulo 51 de codificación que usa una clave K de codificación, y una memoria 55. Una unidad 52 de codificación incluye el circuito 58 de O excluyente y el módulo 51 de codificación que usa la clave K de codificación. El selector 54 y el módulo 51 de codificación que usa la clave K de codificación forman un bucle de retroalimentación con líneas 65, 66 y 67 de retroalimentación. Los datos en bloques de texto cifrado C_i codificados por el módulo 51 de codificación que usa la clave K de codificación son introducidos nuevamente en el circuito 58 de O excluyente a través del bucle de retroalimentación, y los datos de entrada del módulo S_i se generan en el circuito 58 de O excluyente. A continuación, los datos de entrada del módulo S_i generados son suministrados al módulo 51 de codificación que usa la clave K de codificación.

Se proporciona la memoria 55 en paralelo con la línea 65 de retroalimentación. La memoria 55 incluye un registro 56 y un conmutador 57. El conmutador 57 conmuta la entrada al registro 56 o que se ignore una salida del módulo 51 de codificación que usa la clave K de codificación. Esta conmutación es realizada, por ejemplo, por una interrupción IT. Cuando se genera la interrupción IT, el conmutador 57 se conecta a E, y cuando se resuelve la interrupción, el conmutador 57 se conecta a F. El registro 56 da entrada a los datos en bloques de texto cifrado C_i suministrados a través de E y los almacena. Los datos en bloques de texto cifrado C_i son dados al selector 54. Se proporcionan al selector 54 tres entradas A, B y C y él selecciona una de las tres. Esta selección depende de la interrupción IT.

La Fig.2 muestra un procedimiento de operación del codificador mostrado en la Fig. 1.

La Fig. 3 es un diagrama de flujo que muestra la operación del codificador mostrado en la Fig. 1.

La entrada del selector 54 se pone en A cuando se suministra energía eléctrica al codificador y el conmutador 57 está conectado a E. Además, cuando se solicita codificar datos de texto en claro N, se genera una interrupción IT. La interrupción IT se mantiene ACTIVADA a no ser que se resuelva la solicitud de codificar los datos de texto en claro N. Además, los datos de texto en claro M se codifican usando la clave K_1, y los datos de texto en claro N se codifican usando la clave K_2. Cuando se genera la interrupción IT o se resuelve la interrupción IT, vuelve a suministrarse nuevamente la clave K_1 o la clave K_2 al módulo de codificación.

En el instante T0, se suministra la clave K_1 y se inicia el procedimiento de codificación de los datos de texto en claro M_1. Cuando el procedimiento de codificación de los datos de texto en claro M_1 se inicia en el instante T0, la entrada del selector 54 es conmutada a B después de que se introduce una vez el valor inicial IT desde la entrada A del selector 54. Además, en el instante X, durante el cual se están codificando los datos de texto en claro M_1 usando la clave K_1, se supone que se genera una interrupción IT para solicitar la codificación de los datos en bloques de texto en claro N_1. Los datos en bloques de texto cifrado C_1 llegan a ser almacenados en la memoria 55 antes del instante T1. A continuación, en el instante T1, se suministra la clave K_2 al módulo 51 de codificación debido a la generación de la interrupción IT. Además, el selector 54 pone la entrada en A en el instante T1. El conmutador 57 se conecta a F en el instante T1. Después del instante T1, los datos en bloques de texto en claro N_1 son codificados usando la clave K_2, y se producen los datos en bloques de texto cifrado D_1. En el instante Y se supone que termina la codificación de los datos en bloques de texto en claro N_1 y se resuelve la interrupción IT. Debido a la resolución de la interrupción IT, en el instante T2, se suministra la clave K_1 al módulo 51 de codificación, la entrada del selector 54 se conmuta a C, y el conmutador 57 se conecta a E. Al conmutar el selector 54 a C, se introducen los datos en bloques de texto cifrado C_1 almacenados en la memoria 55 para codificar los datos en bloques de texto en claro M_2, los datos en bloques de texto en claro M_2 son codificados por el módulo de codificación usando la clave K_1, y se producen los datos en bloques de texto cifrado C_2. Antes del instante T3, la entrada del selector 54 se conmuta a B. En el caso de codificar los datos en bloques de texto en claro M_3, se retroalimentan los datos en bloques de texto cifrado C_2 desde una línea 65 de retroalimentación de un bucle de retroalimentación y son introducidos, los datos en bloques de texto en claro M_3 son codificados por el módulo de codificación usando la clave K_1, y se producen los datos en bloques de texto cifrado C_3.

Cuando se usan las mismas claves para codificar los datos de texto en claro M y los datos de texto en claro N ($K_1 = K_2$), es suficiente suministrar la clave una vez en el momento de inicio del procedimiento de codificación.

Se explicará la operación total con referencia al diagrama de flujo de la Fig. 3.

En la etapa S1, se inicia y prosigue el procedimiento de codificación de los datos de texto en claro M. Cuando acaban de procesarse los datos finales en bloques, el procedimiento de codificación termina. En la etapa S2, se observa una interrupción IT generada en un instante arbitrario. Cuando la interrupción IT se genera mientras se procesan los datos en bloques de texto en claro M_i, en la etapa S3, los datos en bloques de texto cifrado C_i que se están procesando son almacenados en el registro 56 de la memoria 55. En la etapa S4, se lleva a cabo el procedimiento de codificación de los datos de texto en claro N, que la interrupción IT solicita que se codifiquen. Este procedimiento de codificación de la etapa S4 se realiza de forma continua hasta que se libera la interrupción IT, según se muestra en la etapa S5. Cuando se libera la interrupción IT, en la etapa S6, los datos en bloques de texto en claro M_i son codificados usando los datos en bloques de texto cifrado C_i almacenados en el registro 56 de la memoria 55. Después, el procedimiento vuelve a la etapa S1, y se continuará el procedimiento de codificación.

La Fig. 4 muestra la operación del selector 54.

Cuando se ACTIVA la energía eléctrica, la entrada se pone en A según se muestra en la etapa S11. Cuando el procedimiento de codificación se inicia en la etapa S12, la entrada se pone en B en la etapa S13. Concretamente, se usan los datos en bloques de texto cifrado C_i retroalimentados desde la línea 65 de retroalimentación del bucle de retroalimentación. En la etapa S14, si se detecta que los datos en bloques que se están procesando son los datos finales, el procedimiento vuelve a la etapa S11, en la que el estado es igual, ya que la energía eléctrica está ACTIVADA. En la etapa S15, si se detecta que se genera la interrupción IT, la entrada se pone en A en la etapa S16, y si se inicia el procedimiento de codificación, la entrada se pone en B en la etapa S18. Hasta que se resuelva la interrupción IT, la entrada se mantiene en B. Es decir, se usan los datos en bloques de texto cifrado C_i retroalimentados desde la línea 65 de retroalimentación del bucle de retroalimentación. En la etapa S19, si se detecta que la interrupción IT está resuelta, la entrada se pone en C en la etapa S20. Al poner la entrada en C, se introducen los datos en bloques de texto cifrado C_i almacenados en la memoria 55. Cuando el procedimiento de codificación usa la entrada de C, el procedimiento vuelve a la etapa S13 y la entrada se pone en B.

Según se ha descrito más arriba, el selector 54 puede conmutarse en función de la generación de la interrupción IT.

El procedimiento de codificación de los datos de texto en claro M también se puede iniciar en un instante arbitrario en función de la generación de la interrupción IT.

La Fig. 5 es un diagrama de flujo que muestra el procesamiento de la interrupción por el conmutador 57.

Cuando se ACTIVA la energía eléctrica y en el caso del procedimiento de codificación de los primeros datos de texto en claro posteriores, el conmutador 57 se conecta a E. Cuando se genera la interrupción IT en la etapa S31, el conmutador 57 conmuta de E a F. Después, en la etapa S33, si se detecta que la interrupción IT está resuelta, el conmutador 57 conmuta de F a E. Así, el conmutador 57 ignora los datos en bloques de texto cifrado C_i desde la generación hasta la resolución de la interrupción. En consecuencia, el registro 56 de la memoria 55 mantiene los datos en bloques de texto cifrado C_i, que fueron generados en el momento de la generación de la interrupción IT.

Según se ha descrito más arriba, las operaciones del codificador ilustrado en las Figuras 1 a 5 muestran el mecanismo de procesamiento de interrupciones, que recibe la solicitud de codificar los datos de texto en claro N antes de la finalización de la codificación de los datos de texto en claro M en el codificador para codificar los datos en bloques de texto en claro M_i (i = 1, 2, 3, …) incluidos en los datos de texto en claro M y los datos en bloques de texto en claro N_j (j = 1, 2, 3, …) incluidos en los datos de texto en claro N.

Además, el codificador mostrado en las Figuras 1 a 5 incluye el módulo 51 de codificación para codificar los datos en bloques de texto en claro M_i y producir los datos en bloques de texto cifrado C_i, los bucles 65 y 66 de retroalimentación para retroalimentar los datos en bloques de texto cifrado C_i producidos en el módulo 51 de codificación a la unidad 52 de codificación a través de la línea 65 de retroalimentación, y la memoria 55, proporcionada en paralelo con la línea 65 de retroalimentación del bucle de retroalimentación, para recibir la solicitud de codificación de los datos de texto en claro N por la interrupción, y almacenar los datos en bloques de texto cifrado C_i retroalimentados si los datos en bloques de texto en claro M_{i+1} no son codificados con posterioridad a los datos en bloques de texto en claro M_i iniciando el procedimiento de codificación de cualquiera de los datos en bloques de texto en claro N.

Además, el codificador mostrado en las Figuras 1 a 5 incluye el selector 54 para seleccionar los datos en bloques de texto cifrado C_i, retroalimentados por la línea 65 de retroalimentación del bucle de retroalimentación, y suministrar los datos en bloques de texto cifrado C_i a través del bucle de retroalimentación cuando los datos en bloques de texto en claro M_{i+1} son codificados de forma subsiguiente a los datos en bloques de texto en claro M_i, y para seleccionar los datos en bloques de texto cifrado C_i almacenados en la memoria 55 y suministrarlos a la unidad 52 de codificación a través del bucle de retroalimentación cuando los datos en bloques de texto en claro M_{i+1} no son codificados a continuación de los datos en bloques de texto en claro M_i, y de cualquier dato de texto en claro N.

La memoria 55 almacena el estado del codificador en caso de que se genere la interrupción IT. Almacenando el estado del procedimiento de codificación, se hace posible volver al estado original de la codificación de cierto dato aunque se realice la codificación de otro dato durante el tiempo en que se codifica el cierto dato. Concretamente, usando los datos almacenados en la memoria, el estado del codificador puede volver al estado que es completamente idéntico al estado en el momento en que la codificación es interrumpida, lo que permite seguir el procedimiento de codificación interrumpido.

La Fig. 6 muestra otro ejemplo de configuración de la memoria 55.

La memoria 55 incluye una unidad 52 de control de interrupciones, un conmutador 96 de entrada, un conmutador 97 de salida, y varios registros (REG 1, 2, 3). Proporcionando los varios registros así, se hace posible recibir varias interrupciones.

La Fig. 7 muestra el procesamiento de la interrupción llevado a cabo por la memoria 55.

Cuando se genera la interrupción IT, en la etapa S41, se almacena el número k, que el el número del registro k que se usa en en ese momento. En la etapa S42, el conmutador 96 de entrada y el conmutador 97 de salida se conectan al registro 1, que es uno de los registros distintos del registro k. En este estado, se lleva a cabo el procedimiento de codificación de los datos de texto en claro N. Además, se observa si se genera otra interrupción durante el tiempo en que se codifican los datos de texto en claro N. Cuando se detecta que se genera otra interrupción IT en la etapa S43, vuelve a invocarse nuevamente la etapa S40, que es el procedimiento para procesar la interrupción. De esta manera, siempre que se genere la interrupción IT, se invoca de manera recursiva a la etapa S40. En consecuencia, pueden realizarse varios procesos jerárquicos para el procesamiento de la interrupción. En la etapa S44, se comprueba si está resuelta la interrupción. Cuando la interrupción está resuelta, el conmutador 96 de entrada y el conmutador 97 de salida se conmutan al registro k usando el número k almacenado en la memoria. En el caso de la Fig. 6, la memoria 55 incluye tres registros, de modo que pueden llevarse a cabo procesos jerárquicos de 3 capas para procesar la interrupción.

La Fig. 8 muestra otro ejemplo de configuración de la memoria 55.

La memoria 55 incluye una pila 64. La pila 64 es un registro en el que la primera entrada es la última salida (FILO). Cuando se genera la interrupción IT durante el tiempo en que se usa una pila 1, los datos almacenados en la pila 1 son transferidos a una pila 2, y los datos posteriores son apilados en la pila 1. Cuando se resuelve la interrupción IT, se da salida a los datos apilados en la pila 1, y los datos almacenados en la pila 2 son devueltos a la pila 1. La Fig. 8 muestra un caso en el que pueden llevarse a cabo procesos jerárquicos de 4 capas para procesar la interrupción.

Según se muestra en la Fig. 6, cuando es posible realizar varios procesos jerárquicos para procesar la interrupción, puede asignarse una prioridad a cada una de las interrupciones. Por ejemplo, se asigna una prioridad 1 a la interrupción IT1, y se asigna una prioridad 2, que significa una prioridad inferior a la prioridad 1, a la interrupción IT2. Asignando la prioridad de esta manera, es posible posponer el proceso para la prioridad 2 cuando se genere la interrupción IT1 que tiene la prioridad 1.

La Fig. 9 muestra un caso en el que el procedimiento de codificación que tiene la prioridad 1 se lleva a cabo antes que el procedimiento de codificación que tiene la prioridad 2. En este caso, el procedimiento de codificación que tiene la prioridad 1 termina primero.

La Fig. 10 muestra un caso en el que ambos procedimientos de codificación tienen la misma prioridad.

Cuando las prioridades son iguales, cada uno de los datos en bloques de texto en claro de ambos procedimientos de codificación es codificado de forma alterna.

La Fig. 11 muestra un caso en el que se codifican datos que tienen la prioridad 1 y datos que tienen la prioridad 2.

Asignando la prioridad a cada interrupción según se muestra en las Figuras 9 a 11, es posible llevar a cabo un procedimiento de codificación que resulte deseable para el usuario. En caso de procesar datos de un asunto urgente o datos de poca longitud, puede llevarse a cabo un procesamiento eficaz asignando una prioridad mayor a tales datos.

La Fig. 12 muestra un caso en el que la memoria 55 es puesta en paralelo con la línea 66 de retroalimentación.

El circuito 58 de O excluyente y el módulo 51 de codificación que usa la clave K de codificación constituyen la unidad 52 de codificación.

La Fig. 13 muestra un procedimiento de operación del codificador de la Fig. 12.

Cuando las siguientes conexiones son seleccionadas por el primer selector 61 y el segundo selector 62, se permite que estos selectores operen de la misma manera que el selector 54 de la Fig. 1.

el primer selector 61 + el segundo selector 62 = el selector 54

A + D = A
B + D = B
A + C = C
B + C = C

En la Fig. 13, cuando el segundo selector 62 selecciona D, se hace efectiva la selección (A o B) del primer selector 61, y cuando el segundo selector 62 selecciona C, se da salida al contenido de la memoria 55. Concretamente, el segundo selector 62 debería seleccionar C si se desea usar el contenido de la memoria 55 (cuando el procedimiento de codificación vuelve de los datos de texto en claro N a los datos de texto en claro M debido a la resolución de la interrupción IT).

La Fig. 14 muestra un caso en el que la memoria 55 es puesta en paralelo con la línea 67 de retroalimentación.

La Fig. 15 muestra un procedimiento de operación del codificador de la Fig. 14.

Si el instante X en que se genera la interrupción IT es anterior a la operación de O excluyente por parte del circuito 58 de O excluyente, la memoria 55 almacena los datos de entrada del módulo S_i objeto de una operación lógica de O excluyente por parte del circuito 58 de O excluyente. A continuación, los datos en bloques de texto en claro N_1 son codificados. Posteriormente, se hace que el segundo selector 62 seleccione e introduzca los datos de entrada del módulo S_i en el módulo 51 de codificación que usa la clave K de codificación, y que los codifique para producir los datos en bloques de texto cifrado C_1.

Según se muestra en las Figuras 1, 12 y 14, la memoria 55 puede ser puesta en paralelo con una cualquiera de la línea 65 de retroalimentación, la línea 66 de retroalimentación y la línea 67 de retroalimentación. La memoria 55 almacena el estado que es inmediatamente anterior a que el codificador empiece a codificar otro dato durante la codificación de cierto dato. La memoria 55 puede ser puesta en cualquier lugar, con la condición de que el codificador pueda volver al estado original usando los datos almacenados en la memoria 55 cuando el codificador acabe de codificar los otros datos. Además, puede proporcionarse la memoria 55 en varias ubicaciones.

Según se ha descrito más arriba, el codificador según la presente realización lleva a cabo el procedimiento de codificación de los primeros datos de procesamiento (los datos de texto en claro M), que incluyen al menos un dato en bloques M_i (i = 1, 2, 3, …, m), y de los segundos datos de procesamiento (los datos de texto en claro N), que incluyen al menos un dato en bloques N_j (j = 1, 2, 3, …, n), y el codificador incluye la memoria 55 para almacenar el estado del procedimiento de codificación. El codificador empieza a codificar los primeros datos en bloques de los segundos datos de procesamiento antes de codificar todos los datos en bloques (M_1-M_m) de los primeros datos de procesamiento. Y en el instante en que el codificador empieza a codificar los primeros datos en bloques N_1 de los segundos datos de procesamiento, se almacena en la memoria 55 el estado de la codificación de los primeros datos de procesamiento (por ejemplo, los datos en bloques de texto cifrado C_i). Cuando el codificador reinicia la codificación de los primeros datos de procesamiento, el estado de la codificación del codificador es devuelto al estado almacenado de la codificación de los primeros datos de procesamiento, y luego el codificador reinicia el procesamiento de la codificación de los primeros datos de procesamiento.

Además, el codificador reinicia la codificación de los primeros datos de procesamiento antes de completar la codificación de los datos de todos los bloques (N_1-N_n) de los segundos datos de procesamiento y, simultáneamente, la memoria 55 almacena el estado de la codificación de los segundos datos de procesamiento (por ejemplo, los datos en bloques de texto cifrado D_j) cuando el codificador reinicia la codificación de los primeros datos de procesamiento. Cuando el codificador reinicia la codificación de los segundos datos de procesamiento, el estado de la codificación del codificador es devuelto al estado almacenado de la codificación de los segundos datos de procesamiento, y después del codificador reinicia la codificación de los segundos datos de procesamiento.

La Fig. 16 muestra una configuración del codificador del modo OFB.

La figura se caracteriza por incluir, además, la memoria 55. La memoria 55 almacena los datos de salida del módulo T_1 suministrados desde el módulo 51 de codificación.

La Fig. 16 muestra un codificador para codificar datos en bloques de texto en claro M_i (i = 1, 2, 3, …) incluidos en los datos de texto en claro M y datos en bloques de texto en claro N_j (j = 1, 2, 3, …) incluidos en los datos de texto en claro N. El codificador incluye un mecanismo de procesamiento de la interrupción que recibe la solicitud de codificar los datos de texto en claro N durante la codificación de los datos de texto en claro M antes de completar la codificación de los datos de texto en claro M y el módulo 51 de codificación para producir datos codificados como los datos en bloques de salida del módulo T_i. El codificador incluye además, bucles 65 y 66 de retroalimentación para retroalimentar los datos en bloques de salida del módulo T_i suministrados desde el módulo 51 de codificación en el módulo de codificación a través de la línea 65 de retroalimentación, y la memoria 55 proporcionada en paralelo con la línea 65 de retroalimentación del bucle de retroalimentación y para recibir una solicitud de codificación de los datos de texto en claro N y almacenar los datos en bloques de salida del módulo T_i retroalimentados cuando los

datos en bloques de texto en claro M_{i+1} no son codificados de forma subsiguiente a los datos en bloques de texto en claro M_i porque el codificador empieza a codificar cualesquiera datos en bloques de texto en claro de los datos de texto en claro N. El codificador también incluye, además, el selector 54 que selecciona los datos en bloques de salida del módulo T_i retroalimentados por la línea 65 de retroalimentación del bucle de retroalimentación para suministrárselos al módulo 51 de codificación a través del bucle de retroalimentación cuando los datos en bloques de texto en claro M_i son codificados de forma subsiguiente a los datos en bloques de texto en claro M_i, y selecciona los datos en bloques de salida del módulo T_i almacenados en la memoria 55 para suministrárselos al módulo 51 de codificación a través del bucle de retroalimentación cuando los datos en bloques de texto en claro M_{i+1} no son codificados de forma subsiguiente a los datos en bloques de texto en claro M_i, sino después de cualquiera datos en bloques de texto en claro de los datos de texto en claro N.

La Fig. 17 explica el codificador del modo OFB mostrado en la Fig. 16.

En la Fig. 17, la operación del modo CBC de la Fig. 2 cambia a la operación del modo OFB, y las otras operaciones son iguales a las de la Fig. 2.

La Fig. 18 muestra un codificador del modo CFB.

En comparación con la Fig. 47, el codificador de la Fig. 18 incluye, además, la memoria 55. La memoria 55 almacena datos en bloques de texto cifrado C_i producidos en el circuito 58 de O excluyente.

Además, hay configurada una unidad 52 de codificación por el circuito 58 de O excluyente y el módulo 51 de codificación que usa la clave K de codificación.

La Fig. 18 muestra un codificador para codificar datos en bloques de texto en claro M_i (i = 1, 2, 3, …) incluidos en los datos de texto en claro M y datos en bloques de texto en claro N_j (j = 1, 2, 3, …) incluidos en los datos de texto en claro N. El codificador incluye un mecanismo de procesamiento de la interrupción que recibe la solicitud de codificar los datos de texto en claro N durante la codificación de los datos de texto en claro M antes de la finalización de la codificación de los datos de texto en claro M y la unidad 52 de codificación para codificar los datos en bloques de texto en claro M_i y producir los datos en bloques de texto cifrado C_i. El codificador incluye, además, los bucles 65 y 66 de retroalimentación para retroalimentar los datos en bloques de texto cifrado C_i suministrados desde la unidad 52 de codificación al módulo de codificación a través de la línea 65 de retroalimentación, y la memoria 55 proporcionada en paralelo con la línea 65 de retroalimentación del bucle de retroalimentación, para recibir una solicitud de codificación de los datos de texto en claro N y almacenar los datos en bloques de texto cifrado C_i retroalimentados cuando los datos en bloques de texto en claro M_{i+1} no son codificados de forma subsiguiente a los datos en bloques de texto en claro M_i porque el codificador empieza la codificación de ciertos datos en bloques de texto en claro de los datos de texto en claro N. El codificador también incluye, además, el selector 54 que selecciona los datos en bloques de texto cifrado C_i retroalimentados por la línea 65 de retroalimentación del bucle de retroalimentación para suministrárselos al módulo 51 de codificación a través del bucle de retroalimentación cuando los datos en bloques de texto en claro M_i son codificados de forma subsiguiente a los datos en bloques de texto en claro M_i, y selecciona los datos en bloques de texto cifrado C_i almacenados en la memoria 55 para suministrárselos al módulo 51 de codificación a través del bucle de retroalimentación cuando los datos en bloques de texto en claro M_{i+1} no son codificados de forma subsiguiente a los datos en bloques de texto en claro M_i, sino después de ciertos datos en bloques de texto en claro de los datos de texto en claro N.

La Fig. 19 explica el codificador del modo OFB mostrado en la Fig. 18.

En la Fig. 19, la operación del modo CBC de la Fig. 2 cambia a la operación del modo OFB, y las otras operaciones son iguales a las de la Fig. 2.

La Fig. 20 muestra un decodificador del modo CBC.

En comparación con la Fig. 44, el decodificador de la Fig. 20 incluye, además, la memoria 75.

La memoria 75 incluye un registro 76 y un conmutador 77.

Además, hay configurada una unidad 72 de decodificación por un circuito 78 de O excluyente y un módulo 71 de decodificación que usa la clave K.

Puede proporcionarse un registro 111 dentro de un selector 74.

El decodificador mostrado en la Fig. 20, que decodifica los datos en bloques de texto cifrado C_i (i = 1, 2, 3, …) incluidos en los datos de texto cifrado C y los datos en bloques de texto cifrado N_j (j = 1, 2, 3, …) incluidos en los datos de texto cifrado D, incluye un mecanismo de procesamiento de la interrupción que recibe una solicitud de decodificar los datos de texto cifrado D durante el procedimiento de decodificación de los datos de texto cifrado C.

Además, el decodificador mostrado en la Fig. 20 incluye, además, el módulo 71 de decodificación para producir datos decodificados de los datos en bloques de texto cifrado C_i como datos en bloques de salida del módulo T_i,

bucles 85, 111, 82 y 86 de retroalimentación para retroalimentar los datos en bloques de texto cifrado C_i a la unidad 72 de decodificación a través de las líneas 85, 111 y 82 de retroalimentación para decodificar datos en bloques de texto cifrado C_{i+1}. El decodificador incluye, además, la memoria 75 proporcionada en paralelo con las líneas 85, 111, 82 y 86 de retroalimentación del bucle de retroalimentación y para recibir un solicitud para decodificar los datos de texto cifrado D y almacenar los datos en bloques retroalimentados cuando los datos en bloques de texto cifrado C_{i+1} no son decodificados de forma subsiguiente a los datos en bloques de texto cifrado C_i porque el decodificador comienza a decodificar cualesquiera datos en bloques de texto cifrado de los datos de texto cifrado D.

Además, el decodificador mostrado en la Fig. 20 incluye el selector 74 que selecciona los datos en bloques de texto cifrado C_i retroalimentados por las líneas 85, 111, 82 de retroalimentación del bucle de retroalimentación para suministrárselos a la unidad 72 de codificación a través del bucle de retroalimentación cuando los datos en bloques de texto cifrado C_i son codificados de forma subsiguiente a los datos en bloques de texto cifrado C_i, y selecciona los datos en bloques de texto cifrado C_i almacenados en la memoria para suministrárselos a la unidad 72 de codificación a través del bucle de retroalimentación cuando los datos en bloques de texto cifrado C_{i+1} no son codificados de forma subsiguiente a los datos en bloques de texto cifrado C_i, sino después de cualesquiera datos en bloques de texto cifrado de los datos de texto cifrado D.

“Línea de retroalimentación” y “bucle de retroalimentación”, usados en la anterior explicación de la Fig. 20, no significan “retroalimentación” que “introduzca datos producidos de sí misma”. Aquí, “retroalimentación” significa suministrar nuevamente datos en bloques de texto cifrado C_i para decodificar los datos en bloques de texto cifrado C_{i+1} después de decodificar los datos en bloques de texto cifrado C_i.

La Fig. 21 muestra un procedimiento de operación del decodificador mostrado en la Fig. 20.

Cuando se genera la interrupción IT durante la decodificación de los datos en bloques de texto cifrado C_i usando la clave de codificación (también denominada clave de decodificación) K_1, los datos en bloques de texto cifrado C_i son almacenados en el registro 76 de la memoria 75. Después, los datos en bloques de texto cifrado D_i son decodificados usando la clave de codificación (también denominada clave de decodificación) K_2 formando los datos en bloques de texto en claro N_1. Luego, se leen los datos en bloques de texto cifrado C_1 almacenados en el registro 76 de la memoria 75, los datos en bloques de texto cifrado C_2 son decodificados formando los datos en bloques de texto en claro M_2. La operación del selector 74 es igual que la mostrada en la Fig. 4. Además, la operación del conmutador 77 es igual que la mostrada en la Fig. 5.

La Fig. 22 muestra el decodificador del modo OFB.

El decodificador mostrado en la Fig. 22, que decodifica los datos en bloques de texto cifrado C_i (i = 1, 2, 3, …) incluidos en los datos de texto cifrado C y los datos en bloques de texto cifrado D_j (j = 1, 2, 3, …) incluidos en los datos de texto cifrado D, incluye un mecanismo de procesamiento de la interrupción que recibe una solicitud para decodificar los datos de texto cifrado D durante la decodificación de los datos de texto cifrado C en un punto arbitrario en el tiempo. El decodificador incluye, además, el módulo 71 de decodificación para producir datos decodificados como datos en bloques de salida del módulo T_i, bucles 85, 86 de retroalimentación para retroalimentar los datos en bloques de salida del módulo T_i al módulo 71 de decodificación a través de las líneas 85 de retroalimentación. El decodificador incluye, además, la memoria 75 proporcionada en paralelo con la línea 85 de retroalimentación del bucle de retroalimentación, y para recibir una solicitud de decodificación de los datos de texto cifrado D y almacenar los datos en bloques de salida del módulo T_i retroalimentados cuando los datos en bloques de texto cifrado C_{i+1} no son decodificados de forma subsiguiente a los datos en bloques de texto cifrado C_i porque el decodificador comienza la decodificación de cualesquiera datos en bloques de texto cifrado de los datos de texto cifrado D. Además, el decodificador mostrado en la Fig. 22 incluye el selector 74 que selecciona los datos en bloques de salida del módulo T_i retroalimentados por la línea 85 de retroalimentación del bucle de retroalimentación para suministrárselos al módulo 71 de decodificación a través del bucle de retroalimentación cuando los datos en bloques de texto cifrado C_i son codificados de forma subsiguiente a los datos en bloques de texto cifrado C_i, y selecciona los datos en bloques de salida del módulo T_i almacenados en la memoria 75 para suministrárselos al módulo 71 de decodificación a través del bucle de retroalimentación cuando los datos en bloques de texto cifrado C_{i+1} no son codificados de forma subsiguiente a los datos en bloques de texto cifrado C_i, sino después de cualesquiera datos en bloques de texto cifrado de los datos de texto cifrado D.

La Fig. 23 explica la operación del decodificador del modo OFB mostrado en la Fig. 22.

La operación de la Fig. 23 es igual que la del decodificador del modo CBC mostrado en la Fig. 21, salvo que la operación se lleva a cabo en el modo OFB en lugar del modo CBC.

La Fig. 24 muestra un decodificador del modo CFB.

Se configura una unidad 72 de decodificación por medio del circuito 78 de O excluyente y un módulo 71 de decodificación.

Aquí, se puede proporcionar el registro 111 dentro del selector 74.

El decodificador mostrado en la Fig. 24, que decodifica los datos en bloques de texto cifrado C_i (i = 1, 2, 3, …) incluidos en los datos de texto cifrado C y los datos en bloques de texto cifrado D_j (j = 1, 2, 3, …) incluidos en los datos de texto cifrado D, incluye un mecanismo de procesamiento de la interrupción que recibe una solicitud de decodificación de los datos de texto cifrado D durante la decodificación de los datos de texto cifrado C en un punto arbitrario en el tiempo. El decodificador incluye, además, el módulo 71 de decodificación para producir datos decodificados como datos en bloques de salida del módulo T_i, bucles 85, 111, 82, 86 de retroalimentación para retroalimentar los datos en bloques de salida del módulo T_i al módulo 71 de decodificación a través de las líneas 85, 111, 82 de retroalimentación. El decodificador incluye, además, la memoria 75 proporcionada en paralelo con la línea 85, 111, 82 de retroalimentación del bucle de retroalimentación y para recibir una solicitud de decodificación de los datos de texto cifrado D y almacenar los datos en bloques de salida del módulo T_i retroalimentados cuando los datos en bloques de texto cifrado C_{i+1} no son decodificados de forma subsiguiente a los datos en bloques de texto cifrado C_i porque el decodificador empieza a decodificar cualesquiera datos en bloques de texto cifrado de los datos de texto cifrado D. Además, el decodificador mostrado en la Fig. 24 incluye el selector 74 que selecciona los datos en bloques de salida del módulo T_i retroalimentados por la línea 85 de retroalimentación del bucle de retroalimentación para suministrárselos al módulo 71 de decodificación a través del bucle de retroalimentación cuando los datos en bloques de texto cifrado C_i son codificados de forma subsiguiente a los datos en bloques de texto cifrado C_i, y selecciona los datos en bloques de salida del módulo T_i almacenados en la memoria 75 para suministrárselos al módulo 71 de decodificación a través del bucle de retroalimentación cuando los datos en bloques de texto cifrado C_{i+1} no son codificados de forma subsiguiente a los datos en bloques de texto cifrado C_i, sino después de cualesquiera datos en bloques de texto cifrado de los datos de texto cifrado D.

"Línea de retroalimentación" y "bucle de retroalimentación", usados en la anterior explicación de la Fig. 24, no significan "retroalimentación" que "introduzca datos producidos de sí misma". Aquí, "retroalimentación" significa suministrar nuevamente datos en bloques de texto cifrado C_i para decodificar los datos en bloques de texto cifrado C_{i+1} después de decodificar los datos en bloques de texto cifrado.

La Fig. 25 explica la operación del decodificador del modo CFB mostrado en la Fig. 24.

En la Fig. 25, la operación en el modo CBC mostrado en la Fig. 21 es sustituida con la operación en el modo CFB, y las otras operaciones son iguales que las mostradas en la Fig. 21.

La Fig. 26 muestra un ejemplo de mejora del codificador del modo CBC mostrado en la Fig. 1.

Se añaden un selector 154 y una memoria 155 al codificador de la Fig. 26. La Fig. 1 muestra un caso en el que la clave K_1 es suministrada desde el exterior si se resuelve la interrupción IT, mientras la clave K_1 suministrada desde el exterior es almacenada y usada aquí de nuevo.

La memoria 155 incluye un registro 156 y un conmutador 157. El conmutador 157 conmuta entre ignorar o introducir la clave K de codificación en el registro 156. Esta conmutación es llevada a cabo, por ejemplo, por la interrupción IT. Cuando se genera la interrupción IT, el conmutador 157 se conecta a E, y cuando se resuelve la interrupción IT, el conmutador 157 se conecta a F. El registro 156 introduce la clave K suministrada a través de E y la almacena. La clave K almacenada en el registro 156 es enviada al selector 154. El selector 154 tiene dos entradas A y C, de las cuales el selector 154 selecciona una. Esta selección depende de la interrupción IT, que será descrita posteriormente.

La Fig. 27 muestra un procedimiento de operación del codificador mostrado en la Fig. 26.

Cuando se suministra la energía eléctrica del codificador, las entradas al selector 54 y al selector 154 se ponen en A, y el conmutador 57 y el conmutador 157 se conectan a E. Además, cuando existe una solicitud de codificación de los datos de texto en claro N, se genera la interrupción IT y se la mantiene ACTIVADA hasta que se resuelve la solicitud de codificación de los datos de texto en claro N. Además, los datos de texto en claro M han de codificarse usando la clave K_1, y los datos de texto en claro N han de codificarse usando la clave K_2. Las claves K_1 y K_2 se suministran al módulo 51 de codificación.

En el instante T0, se suministra la clave K_1 desde el exterior como clave KI. Cuando el conmutador 157 se conecta a E, la clave K_1 se almacena en el registro 156. A continuación, se inicia el procedimiento de codificación para los datos en bloques de texto en claro M_1. Cuando comienzan los datos en bloques de texto en claro M_1 en el instante T0, el selector 54 introduce un valor inicial IV a través de A, y entonces el selector 54 se conmuta a B. En el instante X durante el procedimiento de codificación de los datos en bloques de texto en claro M_1 usando la clave K_1, se supone que se genera la interrupción IT para solicitar la codificación de los datos en bloques de texto en claro N_1. Hasta el instante T1, los datos en bloques de texto cifrado C_1 se almacenan en la memoria 55. A continuación, se suministra la clave K_2 al módulo 51 de codificación desde el exterior como clave KI en el instante T1 debido a la generación de la interrupción IT. En el instante T1, la entrada al selector 54 se pone en A. Y en el instante T1, el conmutador 57 y el conmutador 157 se conectan a F. En consecuencia, la clave K_2 no se almacena en el registro 156. Después del instante T1, se lleva a cabo la codificación de los datos en bloques de texto en claro N_1 usando la clave K_2, y se producen los datos en bloques de texto cifrado D_1. En el instante Y, finaliza la codificación de los datos en bloques de texto en claro N_1, y se resuelve la interrupción IT. Debido a esta resolución de la interrupción IT, en el

instante T2, la entrada al selector 54 se conmuta a C, y el conmutador 57 se conecta a E. En consecuencia, la clave K_1 es enviada al selector 154 desde el registro 156 como clave KL, y la clave K_1 es suministrada al módulo 51 de codificación desde el selector 154 como clave K_1. Además, cuando el selector 54 se conmuta a C, los datos en bloques de texto cifrado C_1 almacenados in la memoria 55 son introducidos para codificar los datos en bloques de texto en claro M_2, los datos en bloques de texto en claro M_2 son codificados por el módulo 51 de codificación usando la clave K_1, y se producen los datos en bloques de texto cifrado C_2. Antes del instante T3, la entrada al selector 54 se conmuta a B, y cuando se codifican los datos en bloques de texto en claro M_3, se introducen los datos en bloques de texto cifrado C_2 retroalimentados desde la línea 65 de retroalimentación del bucle de retroalimentación, los datos en bloques de texto en claro M_3 son codificados por el módulo 51 de codificación usando la clave K_1, y se producen los datos en bloques de texto cifrado C_3.

Además, antes del instante T3, la entrada al selector 154 se conmuta a A.

Se describirá la operación del selector 154.

Cuando se ACTIVA la energía eléctrica, la entrada al selector 154 se pone en A. Además, cuando también se detecta la generación de la interrupción IT, la entrada se mantiene configurada en A. Hasta que se resuelva la interrupción IT, el selector 154 opera con la configuración de su entrada en A. Cuando se detecta la resolución de la interrupción IT, el selector 154 configura la entrada a C. Debido a la configuración de la entrada a C, la clave K_1 almacenada en la memoria 55 es introducida en el módulo 51 de codificación como clave K. Cuando se inicia la codificación usando la clave introducida desde C, el selector configura la entrada en A.

Según se ha descrito más arriba, el selector 154 puede conmutarse en función de la generación de la interrupción IT.

A continuación se explicará la operación del procesamiento de la interrupción del conmutador 157.

Cuando se ACTIVA la energía eléctrica, y en el primer procedimiento de codificación de los datos de texto en claro M, el conmutador 157 se conecta a E, y la clave K_1 para los datos de texto en claro M se almacena en el registro 156. Y cuando se genera la interrupción IT en el instante X, el conmutador 157 se conecta a F desde E en el instante T1, y se ignora la clave K_2 para los datos de texto en claro N. Además, cuando se detecta la resolución de la interrupción IT en el instante Y, el conmutador 157 se conecta a E desde F en el instante T2. De esta forma, el conmutador 157 ignora la clave K_2 para los datos de texto en claro N desde la generación hasta la resolución de la interrupción IT. En consecuencia, la clave K_1 para los datos de texto en claro M se mantiene almacenada en el registro 156 de la memoria 155.

La Fig. 28 muestra la configuración del decodificador mostrado en la Fig. 20 cuando se almacena la clave K_1 para ser reutilizada.

La Fig. 28 muestra un caso en el que se añaden un selector 174 y una memoria 175 al decodificador de la Fig. 20. Las operaciones del selector 174 y de la memoria 175 son iguales que las del selector 154 y de la memoria 155 mostrados en la Fig. 26.

La memoria 55 y la memoria 155 son ejemplos de la memoria para almacenar el estado del codificador cuando se genera la interrupción IT. De esta manera, el estado del procedimiento de codificación se almacena en la memoria 55 y en la memoria 155, lo que permite que el codificador vuelva al estado de codificación de cierto dato incluso cuando se lleve a cabo la codificación de otro dato durante el tiempo en que se codifica el cierto dato. Concretamente, usando los datos almacenados en la memoria 55 y la clave K almacenada en la memoria 155, el estado del codificador puede ser devuelto al estado idéntico cuando la codificación se interrumpe, y el procedimiento de codificación interrumpido puede continuar.

La memoria 155 y la memoria 175 pueden ser configuradas idénticamente a la memoria 55 mostrada en las Figuras 6 y 8. O la clave K_1 puede almacenarse añadiendo la configuración, tal como las mostradas en las Figuras 26 y 28 a las Figuras 16, 18, 22 y 24.

Además, dado que las memorias 55 y 155 mostradas en la Fig. 26 operan igual, estas memorias pueden ser integradas en una sola memoria.

Según se ha expuesto, el decodificador en relación con la presente realización lleva a cabo la decodificación de los primeros datos de procesamiento (datos de texto cifrado C), que incluyen al menos los datos en bloques C_i (i = 1, 2, 3, …, m) , y de los segundos datos de procesamiento (datos de texto cifrado D), que incluyen al menos un dato en bloque D_j (J = 1, 2, 3, …, n). El decodificador incluye la memoria 75 que almacena el estado del procedimiento de decodificación. El decodificador inicia el procedimiento de decodificación de unos datos iniciales en bloques D_1 de los segundos datos de procesamiento antes de la finalización del procedimiento de decodificación de todos los datos en bloques (C_1 a C_n) de los primeros datos de procesamiento. Cuando el decodificador inicia el procedimiento de decodificación de los datos iniciales en bloques D_1 de los segundos datos de procesamiento, el decodificador hace que la anterior memoria almacene el estado de decodificación de los primeros datos de procesamiento, y cuando el

decodificador reinicia la decodificación de los primeros datos de procesamiento, el estado del decodificador es devuelto al estado de decodificación del decodificador almacenado en la memoria 75 y el decodificador reinicia el procedimiento de decodificación de los primeros datos de procesamiento.

Además, el decodificador reinicia el procedimiento de decodificación de los primeros datos de procesamiento antes de la finalización de todos los datos en bloques (D_1 a D_n) de los segundos datos de procesamiento, y la memoria 74 almacena el estado de decodificación de los segundos datos de procesamiento cuando el decodificador reinicia el procedimiento de decodificación de los primeros datos de procesamiento. Cuando el decodificador reinicia el procedimiento de decodificación de los segundos datos de procesamiento, el estado del decodificador es devuelto al estado de decodificación de los segundos datos de procesamiento almacenados en la memoria y el decodificador reinicia el procedimiento de decodificación de los segundos datos de procesamiento.

Aquí, el estado del procedimiento de codificación es, por ejemplo,
Datos en bloques codificados C_i (y la clave K_1) en el caso del modo CBC mostrado en la Fig. 1,
Datos de salida del módulo T_i (y la clave K_1) en el caso del modo OFB mostrado en la Fig. 16, y
Datos en bloques codificados C_i (y la clave K_1) en el caso del modo CFB mostrado en la Fig. 18.

El estado de decodificación es, por ejemplo,
Datos en bloques codificados C_i (y la clave K_1) en el caso del modo CBC mostrado en la Fig. 20,
Datos de salida del módulo T_i (y la clave K_1) en el caso del modo OFB mostrado en la Fig. 22, y
Datos en bloques codificados C_i (y la clave K_1) en el caso del modo CFB mostrado en la Fig. 24.

En la descripción anterior, el codificador y el decodificador han sido explicados, respectivamente, en casos de tres modos. Los tres modos son únicamente ejemplos; la presente realización puede ser aplicada al codificador y al decodificador en otros modos, tal como un modo mejorado o un modo transformado. En particular, las características de la realización es que el procedimiento de codificación/decodificación, en el que los datos en bloques C_i, M_i, o T_i generados en el momento de la codificación/decodificación de los datos anteriores son usados para el procedimiento de codificación/decodificación de los datos del bloque siguiente M_{i+1} o C_{i+1} como datos de retroalimentación, se proporciona la memoria 55 para almacenar el estado del procedimiento de codificación/decodificación, para que el estado del codificador/decodificador pueda ser devuelto al estado original usando los datos en bloques C_i, M_i, o T_i después del procedimiento de codificación/decodificación de otro dato. En consecuencia, qué modo de codificación/decodificación se use carece de importancia.

Aquí, en lugar de la interrupción IT, puede recibirse la solicitud de codificación usando otro mecanismo, tal como un sistema de interrogación secuencial o un sistema de obtención de testigo, y puede llevarse a cabo un procesamiento interactivo en paralelo de al menos dos procedimientos de codificación/decodificación.

Además, en la realización anterior, se usa la clave K de codificación para el procedimiento de codificación/decodificación; sin embargo, la realización puede ser aplicada al procedimiento de codificación/decodificación sin usar la clave de codificación.

Realización 2

En la siguiente realización se explicará otro caso, en el que el codificador lleva a cabo un procedimiento de confidencialidad y un procedimiento de garantía de la integridad de datos.

El procedimiento de confidencialidad de datos significa codificar datos para hacer los datos ininteligibles incluso cuando los datos sean objeto de pinchazo en la línea o robados. Además, el procedimiento de garantía de la integridad de datos significa garantizar que los datos no son sustituidos por nadie. En el caso de transferir datos, a veces es preciso garantizar la integridad de los datos, así como llevar a cabo el procedimiento de confidencialidad de los datos. El procedimiento de confidencialidad de datos se lleva a cabo codificando los datos. El procedimiento de garantía de la integridad de datos se lleva a cabo añadiendo un MAC (código de autenticación de mensajes) al último bit de los datos y comprobando el MAC para detectar la manipulación.

La Fig. 29 muestra un caso en el que una unidad 100 de codificación del modo OFB realiza el procedimiento de confidencialidad, y un generador 200 de MAC del modo CBC genera el MAC.

La Fig. 29 muestra el codificador que codifica los datos de texto en claro, que incluyen al menos un dato en bloques de texto en claro, usando el módulo de codificación y genera el MAC para garantizar la integridad de los datos de texto cifrado. El codificador incluye una unidad 100 de codificación que tiene un primer bucle de retroalimentación 65 que retroalimenta los datos en bloques de salida del módulo T_i suministrados al módulo 51 de codificación desde el módulo 51 de codificación en la codificación de los datos en bloques de texto en claro por el módulo 51 de codificación. La unidad 100 de codificación introduce los datos en bloques de texto en claro, retroalimenta los datos en bloques de salida del módulo T_i usando el primer bucle 65 de retroalimentación para realizar el procedimiento de codificación para producir datos en bloques de texto cifrado C_i. El codificador incluye un generador 200 de MAC que tiene un segundo bucle 66 de retroalimentación que retroalimenta un resultado de MAC intermedio calculado T_i. El generador 200 de MAC introduce los datos en bloques de texto cifrado C_i en todas las salidas de los datos en

bloques de texto cifrado C_i desde la unidad 100 de codificación, calcula el MAC, hace que el resultado de MAC intermedio calculado T_i sea retroalimentado usando el segundo bucle 66 de retroalimentación, y genera un MAC P para garantizar la integridad de los datos de texto cifrado.

La Fig. 30 muestra un procedimiento de operación del codificador mostrado en la Fig. 29.

Los datos en bloques de texto en claro M_1 son codificados en primer lugar formando los datos en bloques de texto cifrado C_1. A continuación, los datos en bloques de texto en claro M_2 son introducidos para ser codificados formando los datos en bloques de texto cifrado C_2. Simultáneamente a la codificación de los datos en bloques de texto en claro M_1, los datos en bloques de texto cifrado C_1 son introducidos y se inicia el cálculo del MAC. Entre el instante T1 y el instante T2, se llevan a cabo el procedimiento de codificación de los datos en bloques de texto en claro M_2 y el cálculo del MAC en función de los datos en bloques de texto cifrado C_1. Entre el instante T2 y el instante T3, se llevan a cabo el procedimiento de codificación de los datos en bloques de texto en claro M_3 y el cálculo del MAC en función de los datos en bloques de texto cifrado C_2. En el instante T3, se lleva a cabo el cálculo del MAC en función de los datos en bloques de texto cifrado C_3 y se produce el MAC P.

La configuración de la Fig. 29 se caracterizada porque los datos en bloques de texto cifrado C_i producidos en el circuito 58 de O excluyente son introducidos en el circuito 59 de O excluyente por una línea 69 de alimentación. La línea 69 de alimentación combina los procedimientos de codificación del modo OFB y del modo CBC para que el procedimiento de confidencialidad y el procedimiento de garantía de la integridad se lleven a cabo mediante procesamiento en serie. En el caso de la Fig. 52, el proceso en el instante T6 requiere mucho tiempo de proceso; sin embargo, en el caso de la Fig. 30, el procesamiento acaba en el instante T4, lo que muestra que se ha realizado un procesamiento a velocidad elevada.

La Fig. 31 es un diagrama de flujo que muestra la operación del codificador mostrado en la Fig. 29.

En S51, se inicializa a 1 un contador i de datos en bloques. S52 muestra la operación de la unidad 100 de codificación. La unidad 100 de codificación da entrada a los datos en bloques de texto en claro M_i, codifica los datos de texto en claro M_i formando los datos en bloques de texto cifrado C_i, y produce los datos en bloques de texto cifrado C_i. S53 muestra la operación del generador 200 de MAC. El generador 200 de MAC da entrada a los datos en bloques de texto cifrado C_i y los codifica y calcula el MAC. S54 comprueba si el contador i de datos en bloques indica el último dato n del bloque. Cuando el contador no lo indica, el contador i de datos en bloques es incrementado en S55, y la operación vuelve al procedimiento de S52. Concretamente, se repiten los procedimientos de la unidad 100 de codificación y del generador 200 de MAC. Cuando se detecta en S54 que se ha completado el procedimiento del último dato del bloque, el último MAC calculado en S53 se convierte en el MAC final, y el MAC es añadido al último bit de los datos en bloques de texto cifrado C_i en S56. Según se muestra en la Fig. 31, en cada generación de los datos en bloques de texto cifrado C_i por la unidad 100 de codificación, el generador 200 de MAC introduce los datos en bloques de texto cifrado C_i para calcular el MAC, lo que permite el procesamiento en serie a velocidad elevada.

La Fig. 32 muestra una configuración que combina la unidad 100 de codificación y el generador 200 de MAC mostrados en la Fig. 29. Es decir, el módulo 51 de codificación es compartido por la unidad 100 de codificación y el generador 200 de MAC, y el circuito de O excluyente es usado como el circuito 58 de O excluyente de la unidad 100 de codificación y el circuito 59 de O excluyente del generador 200 de MAC. Además, la línea de retroalimentación es usada como tanto la línea 65 de retroalimentación de la unidad 100 de codificación como la línea 66 de retroalimentación del generador 200 de MAC.

Un primer selector 61 selecciona un valor inicial IV en el momento de inicio del procedimiento de confidencialidad. Un segundo selector selecciona el valor inicial IV en el momento de inicio del procedimiento de garantía de la integridad. Un tercer selector 63 selecciona de forma alterna el procedimiento de confidencialidad y el procedimiento de garantía de la integridad. El procedimiento de confidencialidad puede realizarse poniendo la entrada del tercer selector en E. Además, el procedimiento de garantía de la integridad puede realizarse poniendo la entrada del tercer selector en F.

Una memoria 93 almacena los datos de salida del módulo T_i producidos en el módulo 51 de codificación que usa la clave K de codificación. La memoria 93 incluye un conmutador 96 de entrada, un conmutador 97 de salida, un primer registro 98 y un segundo registro 99. El conmutador 96 de entrada y el conmutador 97 de salida están sincronizados con la conmutación del tercer selector 63. En cada conmutación del tercer selector 63, se conmutan tanto el conmutador 96 de entrada como el conmutador 97 de salida.

La Fig. 33 muestra un procedimiento de operación del codificador mostrado en la Fig. 32.

Entre el instante T0 y el instante T1, se lleva a cabo el procedimiento de confidencialidad de los datos en bloques de texto en claro M_1. Los datos de salida del módulo generados durante el procedimiento de confidencialidad son almacenados en el primer registro 98. Entre el instante T1 y el instante T2, se calcula el MAC en función de los datos en bloques de texto cifrado C_1. El resultado de MAC intermedio calculado generado por el procedimiento de garantía de la integridad se almacena en el segundo registro 99. A continuación, entre el instante T2 y el instante T3, se lleva

a cabo el procedimiento de confidencialidad de los datos en bloques de texto en claro M_2 en función de los datos de salida del módulo almacenados en el primer registro 98 y los datos en bloques de texto en claro M_2. A continuación, entre el instante T3 y el instante T4, el resultado de MAC intermedio calculado almacenado en el segundo registro 99 y se introducen los datos en bloques de texto cifrado C_2 y se calcula el MAC. Repitiendo estas operaciones, se completan el procedimiento de confidencialidad y el procedimiento de garantía de la integridad y se producen los datos de texto cifrado y el MAC P. En el caso de la Fig. 33, el procedimiento finaliza en el instante T6 y no se reduce el tiempo de procesamiento. Sin embargo, según se muestra en la Fig. 32, el módulo 51 de codificación que usa la clave K de codificación, el circuito 58 de O excluyente, y la línea 67, 68 de retroalimentación (bucle de retroalimentación) son compartidos por la unidad de codificación y el generador de MAC, para que pueda reducirse la escala del circuito.

La Fig. 34 muestra un decodificador que incluye una unidad 300 de decodificación del modo OFB y un generador 400 de MAC del modo CBC.

El generador 400 de MAC está configurado igual que el generador 200 de MAC.

La Fig. 34 muestra el decodificador que decodifica los datos de texto cifrado, que incluyen al menos un dato en bloques de texto cifrado, formando los datos de texto en claro y genera el MAC para garantizar la integridad de los datos de texto cifrado. El decodificador incluye una unidad 300 de decodificación que tiene un primer bucle de retroalimentación 65 que retroalimenta los datos en bloques de salida del módulo T_i generados en el procedimiento de decodificación de los datos en bloques de texto cifrado C_i usando el módulo 71 de decodificación. La unidad 300 de decodificación introduce los datos en bloques de texto cifrado C_i, hace que los datos en bloques de salida del módulo T_i sean retroalimentados por el primer bucle 65 de retroalimentación para la decodificación y produce los datos en bloques de texto en claro M_i. El decodificador incluye, además, un generador 400 de MAC que tiene un segundo bucle de retroalimentación 66 que retroalimenta el resultado de MAC intermedio calculado T_i. El generador 400 de MAC introduce los mismos datos en bloques de texto cifrado con los datos en bloques de texto cifrado C_i introducidos en la unidad 300 de decodificación, realiza el cálculo de MAC para producir el resultado de MAC intermedio calculado, hace que el segundo bucle de retroalimentación 66 retroalimente el resultado de MAC intermedio calculado T_i, y genera el MAC Q para garantizar la integridad de los datos de texto cifrado.

Los datos en bloques de texto cifrado C_i son introducidos en el circuito 78 de O excluyente de la unidad 300 de decodificación y, al mismo tiempo, los datos en bloques de texto cifrado C_i son introducidos en el generador 400 de MAC por la línea 69 de alimentación. Por medio de esta configuración, se llevan a cabo simultáneamente los procedimientos de la unidad 300 de decodificación y del generador 400 de MAC, de modo que puede aumentar la velocidad de procesamiento.

La Fig. 35 muestra una configuración en la que se integran la unidad 300 de decodificación y el generador 400 de MAC del decodificador mostrados en la Fig. 34.

La Fig. 35 muestra un caso en el que el módulo 71 de decodificación y las líneas de retroalimentación 87, 88 (bucle de retroalimentación) son compartidos.

Un primer selector 81 selecciona el valor inicial IV en el momento de inicio del procedimiento de decodificación. Un segundo selector 82 selecciona el valor inicial IV en el momento de inicio del procedimiento de garantía de la integridad. Un tercer selector 83 selecciona de forma alterna el procedimiento de decodificación y el procedimiento de garantía de la integridad. La decodificación puede realizarse poniendo la entrada del tercer selector 83 en E. Además, puede realizarse el procedimiento de garantía de la integridad poniendo la entrada del tercer selector 83 en F.

La memoria 93 almacena los datos de salida del módulo T_i producidos en el módulo 51 de codificación que usa la clave K de codificación. La memoria 93 incluye un conmutador 96 de entrada, un conmutador 97 de salida, un primer registro 98 y un segundo registro 99. El conmutador 96 de entrada y el conmutador 97 de salida están sincronizados con la conmutación del tercer selector 83. En cada conmutación del tercer selector 83, también se conmutan el conmutador 96 de entrada y el conmutador 97 de salida.

La Fig. 36 muestra un procedimiento de operación del decodificador mostrado en la Fig. 35.

El decodificador introduce los datos de texto cifrado y el MAC P.

Entre el instante T0 y el instante T1, los datos en bloques de texto cifrado C_i son decodificados y los datos en bloques de texto cifrado C_i son almacenados en el registro 111. Los datos de salida del módulo generados durante el procedimiento de decodificación son almacenados en el registro 98. Entre el instante T1 y el instante T2, se calcula el MAC en función de los datos en bloques de texto cifrado C_i almacenados en el registro 111. El resultado de MAC intermedio calculado generado durante el procedimiento de garantía de la integridad es almacenado en el segundo registro 99. A continuación, entre el instante T2 y el instante T3, los datos en bloques de texto cifrado C_2 son almacenados en el registro 111, se lleva a cabo el procedimiento de decodificación de los datos en bloques de texto en claro M_2 en función de los datos de salida del módulo almacenados en el primer registro 98 y los datos en

bloques de texto cifrado C_2. Acto seguido, entre el instante T3 y el instante T4, se introducen el resultado de MAC intermedio calculado almacenado en el segundo registro 99 y los datos en bloques de texto cifrado C_2 almacenados en el registro 111 y se calcula el MAC. Repitiendo estas operaciones, se producen los datos de texto en claro y el MAC Q. El MAC Q es comparado con el MAC P. Si el MAC P coincide con el MAC Q, puede garantizarse la integridad de los datos. Así se completan el procedimiento de decodificación y el procedimiento de garantía de la integridad.

La Fig. 37 muestra una configuración en la que se usa la unidad 100 de codificación del modo CBC en lugar de la unidad 100 de codificación del modo OFB mostrada en la Fig. 29.

La Fig. 37 muestra el codificador, que codifica los datos de texto en claro, incluyendo al menos un dato en bloques de texto en claro, usando el módulo de codificación y genera el MAC para garantizar la integridad de los datos de texto cifrado. El codificador incluye una unidad 100 de codificación que tiene un primer bucle 65 de retroalimentación que retroalimenta los datos en bloques de texto cifrado C_i producidos en el módulo 51 de codificación en el momento de codificación de los datos en bloques de texto en claro por la unidad 52 de codificación. La unidad 100 de codificación introduce los datos en bloques de texto en claro M_i, hace que los datos en bloques de texto cifrado C_i sean retroalimentados usando el primer bucle 65 de retroalimentación para realizar el procedimiento de codificación, y produce los datos en bloques de texto cifrado C_i. El codificador incluye, además, un generador 200 de MAC que tiene un segundo bucle 66 de retroalimentación que retroalimenta un resultado de MAC intermedio calculado T_i. El generador 200 de MAC introduce los datos en bloques de texto cifrado C_i en todas las salidas de los datos en bloques de texto cifrado C_i desde la unidad 100 de codificación, calcula el MAC, hace que el resultado de MAC intermedio calculado T_i sea retroalimentado usando el segundo bucle de retroalimentación 66, y genera un MAC P para garantizar la integridad de los datos de texto cifrado.

La Fig. 38 muestra una configuración en la que se proporciona la unidad 300 de decodificación del modo CBC en lugar de la unidad 300 de decodificación del modo OFB mostrada en la Fig. 34.

La Fig. 38 muestra el decodificador que decodifica los datos de texto cifrado, que incluyen al menos un dato en bloques de texto cifrado, formando los datos de texto en claro, y genera el MAC para garantizar la integridad de los datos de texto cifrado. El decodificador incluye una unidad 300 de decodificación que tiene unos primeros bucles 85, 82 de retroalimentación para retroalimentar los datos en bloques de texto cifrado C_i, y la unidad 300 de decodificación introduce los datos en bloques de texto cifrado C_i y hace que los datos en bloques de texto cifrado C_i sean retroalimentados por los primeros bucles 85, 82 de retroalimentación para la decodificación, y produce los datos en bloques de texto en claro M_i. El decodificador incluye, además, un generador 400 de MAC que tiene un segundo bucle de retroalimentación 66 para retroalimentar el resultado de MAC intermedio calculado T_i, y el generador 400 de MAC introduce los datos en bloques de texto cifrado C_i que son idénticos a los datos en bloques de texto cifrado C_i introducidos en la unidad 300 de decodificación, calcula el MAC, produce el resultado de MAC intermedio calculado T_i, hace que el resultado de MAC intermedio calculado T_i sea retroalimentado por el segundo bucle de retroalimentación, y genera el MAC para garantizar la integridad de los datos de texto cifrado.

Según se ha descrito más arriba, las Figuras 29 y 37 muestran los codificadores, cada uno de los cuales incluye la unidad de codificación que da entrada a datos que han de codificarse y produce los datos, y el generador de MAC que da entrada a los datos codificados producidos por la unidad de codificación y genera el MAC para garantizar la integridad de los datos de texto cifrado, iniciando el generador de MAC la generación del MAC antes de que la unidad de codificación complete la codificación de los datos.

Además, las Figuras 34 y 38 muestran los decodificadores, cada uno de los cuales incluye la unidad de decodificación que da entrada a los datos que han de decodificarse y da produce los datos, y el generador de MAC, que da entrada a los datos introducidos por la unidad de decodificación y genera el MAC para garantizar la integridad de los datos de texto cifrado, iniciando el generador de MAC la generación del MAC antes de que la unidad de decodificación complete la decodificación de los datos.

La unidad 100 de codificación o la unidad 300 de decodificación del modo OFB, que no se muestran en las Figuras, pueden ser usadas en el codificador/decodificador anterior.

El generador 200 de MAC del modo OFB o del modo CFB, que no se muestra en las Figuras, puede ser usado en el codificador/decodificador anterior.

La Fig. 39 muestra una configuración del módulo 51 de codificación o del módulo 71 de decodificación.

El módulo 51 de codificación incluye un planificador 511 de claves y un aleatorizador 512 de datos. El planificador 511 de claves introduce una clave K para generar un número n de claves extendidas $ExtK_1$ a $ExtK_n$. El aleatorizador 512 de datos genera un número aleatorio usando una función F y un circuito de O excluyente. La función F introduce la clave extendida y lleva a cabo una transformación no lineal de los datos.

En el módulo 51 de codificación del codificador anterior, puede usarse un algoritmo de cifrado de bloques, tal como:

(1) DES (Estándar de Codificación de Datos),
(2) MISTY, el algoritmo de cifrado de bloques dado a conocer por la publicación de patente internacional nº WO97/9705 (solicitud de patente estadounidense nº 08/83640),
(3) KASUMI, cifrado de bloques de 64 bits desarrollado en función del algoritmo de cifrado de bloques MISTY, que se determinó para ser aplicado al cifrado internacional estándar para el teléfono celular de la siguiente generación (IMT2000) (para más detalles, conviene visitar http://www.3gpp.org/About_3GPP/3gpp.htm), o
(4) Camellia, el algoritmo de cifrado en bloques dado a conocer en la solicitud de patente japonesa nº 2000-64614 (presentada el 9 de marzo de 2000). Además, en el módulo de decodificación del decodificador, puede aplicarse un algoritmo de cifrado de bloques, tal como DES, MISTY, KASUMI o Camellia.

La Fig. 40 muestra una forma de implementación del codificador o del decodificador.

La Fig. 40 muestra un caso en el que el codificador y el decodificador están instalados en una FPGA, un IC o una LSI. Concretamente, el codificador y el decodificador anteriormente mencionados pueden ser implementados mediante soporte físico. Además, el codificador y el decodificador pueden ser implementados mediante una placa de circuito impreso, que no se muestra en la figura.

La Fig. 41 muestra un caso en el que el codificador y el decodificador son implementados mediante soporte lógico.

El anterior codificador puede ser implementado mediante un programa 47 de cifrado. El programa 47 de cifrado se almacena en ROM (memoria de solo lectura) 42 (un ejemplo de almacenamiento). El programa 47 de cifrado puede almacenarse en otro tipo de almacenamiento, tal como RAM (memoria de acceso aleatorio), en un disco flexible, o en un disco duro. Además, el programa 47 de cifrado puede ser descargado de un ordenador servidor. El programa 47 de cifrado opera como una subrutina. El programa 47 de cifrado es invocado desde un programa 46 de aplicación almacenado en la RAM 45 por medio de una llamada a una subrutina, y se ejecuta el programa 47 de cifrado. O el programa 47 de cifrado puede ser activado mediante la generación de una interrupción recibida por la unidad 43 de control de interrupciones. La memoria 55 puede ser una parte de la RAM 45. El programa 46 de aplicación y el programa 47 de cifrado son programas ejecutados por la CPU.

La Fig. 42 muestra un mecanismo mediante el cual el programa 46 de aplicación llama al programa 47 de cifrado.

El programa 46 de aplicación llama al programa 47 de cifrado usando la clave K, el valor inicial IV, los datos de texto en claro M, y los datos de texto cifrado C como parámetros. El programa 47 de cifrado toma como entradas la clave K, el valor inicial IV y los datos de texto en claro M y devuelve los datos de texto cifrado C. Cuando el programa 47 de cifrado y el programa de descifrado son el mismo, el programa de cifrado es llamada usando la clave K, el valor inicial IV, los datos de texto cifrado C y los datos de texto en claro M como parámetros.

El programa 47 de cifrado puede ser implementado por un procesador de señales digitales y un programa que es leído y ejecutado por el procesador de señales digitales. Concretamente, el programa 47 de cifrado puede ser implementado mediante la combinación de soporte físico y soporte lógico.

Las Figuras 40, 41 y 42 explican principalmente casos para el codificador; sin embargo, el decodificador puede ser implementado de la misma manera.

El codificador y el decodificador mostrados en las Figuras 40 y 41 pueden ser instalados en un dispositivo electrónico. Por ejemplo, el codificador y el decodificador pueden ser instalados en todo tipo de dispositivos electrónicos tales como un ordenador personal, una máquina de fax, un teléfono celular, una videocámara, una cámara digital, una cámara de TV. En particular, las características de la presente realización pueden ser utilizadas de forma efectiva en el caso de la codificación/decodificación de los datos de varios canales. O la aplicación de la realización es efectiva cuando se reciben de varios usuarios varios datos que han de ser decodificados, cuando se generan aleatoriamente varios datos a partir de varios usuarios y los datos deban ser codificados en tiempo real. Concretamente, el codificador y el decodificador de la realización son realmente eficaces cuando el número de los dispositivos para codificar/decodificar es pequeño en comparación con el número de datos que han de codificarse/decodificarse. Por ejemplo, el codificador y el decodificador son muy eficaces para el ordenador servidor que requiere soportar muchos ordenadores cliente, para una estación base o un controlador de líneas que requiera recoger datos de muchos teléfonos celulares.

En lugar del procesamiento en paralelo de procedimientos de codificación y procedimientos de decodificación, el procedimiento de codificación y el procedimiento de decodificación pueden realizarse en paralelo.

Además, la anterior explicación muestra un caso de combinación de la unidad de codificación (o unidad de decodificación) del modo OFB y el generador de MAC del modo CBC; sin embargo, puede usarse cualquier combinación de modos, tales como el modo OFB, el modo CBC, el modo CFB, un modo mejorado de estos modos, etcétera.

Además, la anterior explicación muestra un caso en el que el generador de MAC realiza la codificación usando la clave K de codificación; sin embargo, el generador de MAC puede realizar la codificación de datos, el cálculo de datos u otro procesamiento de datos.

Aplicabilidad industrial

Como se ha descrito, según la realización preferente de la presente invención, el procedimiento de codificación de los datos de texto en claro N puede iniciarse durante el procedimiento de codificación de los datos de texto en claro M. Además, el procedimiento de decodificación de los datos de texto cifrado D puede iniciarse durante el procedimiento de decodificación de los datos de texto cifrado C.

Además, según la realización preferente de la presente invención, pueden asignarse prioridades a los datos que han de codificarse/decodificarse y puede realizarse un procesamiento a velocidad elevada en función de las prioridades de los datos.

Además, según la realización preferente de la presente invención, también pueden llevarse a cabo en paralelo el procedimiento de confidencialidad y el procedimiento de garantía de la integridad, lo que permite un procesamiento a velocidad elevada. Además, el procedimiento de confidencialidad y el procedimiento de garantía de la integridad pueden ser realizados por un soporte físico integrado.

REIVINDICACIONES

1. Un aparato de codificación que comprende:

una unidad (100) de codificación para introducir datos que han de codificarse y producir datos codificados, y un generador (200) de códigos de autenticación de mensajes (MAC) para introducir los datos codificados producidos por la unidad de codificación y generar un MAC para garantizar una integridad de los datos codificados,
en el que el generador de MAC inicia la generación del MAC antes de que la unidad de codificación acabe de codificar los datos y en el que se codifican datos de texto en claro que incluyen al menos un dato en bloques de texto en claro formando datos de texto cifrado usando la unidad de codificación y se genera un código de autenticación de mensajes (MAC) para garantizar una integridad de los datos de texto cifrado,
en el que la unidad (100) de codificación tiene un primer bucle de retroalimentación para retroalimentar a la unidad (100) de codificación los datos en bloques de texto cifrado C_i producidos por la unidad (100) de codificación cuando la unidad (100) de codificación codifica los datos en bloques de texto en claro, para introducir los datos de texto en claro, realizando un procedimiento de codificación retroalimentando los datos en bloques de texto cifrado C_i a través del primer bucle de retroalimentación, y produciendo los datos en bloques de texto cifrado C_i,
en el que el aparato de codificación comprende, además, una línea (69) de alimentación para introducir los datos en bloques de texto cifrado C_i desde la unidad (100) de codificación y producir los datos en bloques de texto cifrado C_i para el generador (200) de códigos de autenticación de mensajes (MAC), y
en el que el generador (200) de códigos de autenticación de mensajes (MAC) tiene un segundo bucle de retroalimentación para retroalimentar un resultado de MAC intermedio calculado T_i, para introducir los datos en bloques de texto cifrado C_i desde la línea (69) de alimentación, calcular un resultado de MAC intermedio en función de los datos en bloques de texto cifrado C_i introducidos, retroalimentar el resultado de MAC intermedio como el resultado de MAC intermedio calculado T_i por el segundo bucle de retroalimentación, y generar el MAC para garantizar la integridad de los datos de texto cifrado,
caracterizado porque
la unidad de codificación y el generador de MAC realizan alternativamente el procedimiento de codificación y un procedimiento de generación de MAC compartiendo un módulo de codificación y un bucle de retroalimentación (67, 68), en el que el bucle de retroalimentación (67, 68) incluye:

una memoria para, respectivamente, almacenar y producir resultados del procedimiento de codificación y del procedimiento de generación de MAC, y
un selector para seleccionar alternativamente los resultados del procedimiento de codificación y del procedimiento de generación de MAC de la memoria para realizar, alternativamente, el procedimiento de codificación y el procedimiento de generación de MAC.

2. Un aparato de codificación que comprende:

una unidad (100) de codificación para introducir datos que han de codificarse y producir datos codificados, y un generador (200) de códigos de autenticación de mensajes (MAC) para introducir los datos codificados producidos por la unidad de codificación y generar un MAC para garantizar una integridad de los datos codificados,
en el que el generador de MAC inicia la generación del MAC antes de que la unidad de codificación acabe de codificar los datos y en el que se codifican datos de texto en claro que incluyen al menos un dato en bloques de texto en claro y se genera un código de autenticación de mensajes (MAC) para garantizar una integridad de los datos de texto cifrado,
en el que la unidad (100) de codificación tiene un primer bucle de retroalimentación para retroalimentar al módulo (51) de codificación datos en bloques de salida del módulo T_i producidos por el módulo (51) de codificación de la unidad (100) de codificación cuando la unidad (100) de codificación codifica los datos en bloques de texto en claro, para introducir los datos de texto en claro, realizando un procedimiento de codificación retroalimentando los datos en bloques de salida del módulo T_i a través del primer bucle de retroalimentación, y produciendo los datos en bloques de texto cifrado,
en el que el aparato de codificación comprende, además, una línea (69) de alimentación para introducir los datos en bloques de texto cifrado desde la unidad (100) de codificación y producir los datos en bloques de texto cifrado para el generador (200) de códigos de autenticación de mensajes (MAC), y
en el que el generador (200) de MAC tiene un segundo bucle de retroalimentación para retroalimentar un resultado de MAC intermedio calculado, para introducir los datos en bloques de texto cifrado desde la línea (69) de alimentación, calcular un resultado de MAC intermedio en función de los datos en bloques de texto cifrado introducidos, retroalimentar el resultado de MAC intermedio como el resultado de MAC intermedio calculado a través del segundo bucle de retroalimentación, y generar el MAC para garantizar la integridad de los datos de texto cifrado,
caracterizado porque

la unidad de codificación y el generador de MAC comparten un módulo de codificación y un bucle de retroalimentación (67, 68) para realizar alternativamente el procedimiento de codificación y un procedimiento de generación de MAC, en el que el bucle de retroalimentación incluye:

una memoria para, respectivamente, almacenar y producir resultados del procedimiento de codificación y del procedimiento de generación de MAC, y
un selector para seleccionar alternativamente los resultados del procedimiento de codificación y del procedimiento de generación de MAC de la memoria para realizar, alternativamente, el procedimiento de codificación y el procedimiento de generación de MAC.

3. Un procedimiento de codificación que comprende:

una etapa de codificación para introducir datos que han de codificarse y producir datos codificados, y
una etapa de generación de MAC para introducir los datos codificados producidos en la etapa de codificación y generar un MAC para garantizar una integridad de los datos codificados,
en el que la etapa de generación de MAC inicia la generación del MAC antes de que la etapa de codificación acabe de codificar los datos y en el que se codifican datos de texto en claro que incluyen al menos un dato en bloques de texto en claro formando datos de texto cifrado usando una unidad de codificación y se genera un código de autenticación de mensajes (MAC) para garantizar una integridad de los datos de texto cifrado,
en el que la etapa de codificación incluye una primera etapa de retroalimentación para retroalimentar datos en bloques de texto cifrado C_i producidos por la unidad de codificación cuando la unidad de codificación codifica los datos en bloques de texto en claro, introducir los datos en bloques de texto en claro, realizar un procedimiento de codificación retroalimentando los datos en bloques de texto cifrado C_i a través de un primer bucle de retroalimentación, y producir los datos en bloques de texto cifrado C_i,
en el que los datos en bloques de texto cifrado C_i producidos por la etapa de codificación son alimentados, por una etapa de alimentación, a la etapa de generación de MAC, y
en el que la etapa de generación de MAC incluye una segunda etapa de retroalimentación para retroalimentar un resultado de MAC intermedio calculado, introducir los datos en bloques de texto cifrado C_i en la etapa de alimentación, calcular un resultado de MAC intermedio en función de los datos en bloques de texto cifrado C_i introducidos, retroalimentar el resultado de MAC intermedio como el resultado de MAC intermedio calculado a través de la segunda etapa de retroalimentación, y generar el MAC para garantizar la integridad de los datos de texto cifrado,
caracterizado porque
la etapa de codificación y la etapa de generación de MAC realizan alternativamente el procedimiento de codificación y un procedimiento de generación de MAC compartiendo un módulo de codificación y un bucle de retroalimentación, en el que el bucle de retroalimentación incluye:

una memoria para, respectivamente, almacenar y producir resultados del procedimiento de codificación y del procedimiento de generación de MAC, y
un selector para seleccionar alternativamente los resultados del procedimiento de codificación y del procedimiento de generación de MAC de la memoria para realizar, alternativamente, el procedimiento de codificación y el procedimiento de generación de MAC.

4. Un procedimiento de codificación que comprende:

una etapa de codificación para introducir datos que han de codificarse y producir datos codificados, y
una etapa de generación de MAC para introducir los datos codificados producidos en la etapa de codificación y generar un MAC para garantizar una integridad de los datos codificados,
en el que la etapa de generación de MAC inicia la generación del MAC antes de que la etapa de codificación acabe de codificar los datos y en el que se codifican datos de texto en claro que incluyen al menos un dato en bloques de texto en claro formando datos de texto cifrado usando una unidad de codificación y se genera un código de autenticación de mensajes (MAC) para garantizar una integridad de los datos de texto cifrado,
en el que la etapa de codificación tiene una primera etapa de retroalimentación para retroalimentar datos en bloques de salida del módulo T_i producidos por un módulo de codificación cuando los se codifican los datos en bloques de texto en claro, para introducir los datos en bloques de texto en claro, realizar un procedimiento de codificación retroalimentando los datos en bloques de salida del módulo T_i a través de un primer bucle de retroalimentación, y producir los datos en bloques de texto cifrado,
en el que los datos en bloques de texto cifrado producidos por la etapa de codificación son alimentados, por una etapa de alimentación, a la etapa de generación de MAC, y
en el que la etapa de generación de MAC tiene una segunda etapa de retroalimentación para retroalimentar un resultado de MAC intermedio calculado, para introducir los datos en bloques de texto cifrado de la etapa de alimentación, calcular un resultado de MAC intermedio en función de los datos en bloques de texto cifrado introducidos, retroalimentar el resultado de MAC intermedio como el resultado de MAC intermedio

calculado a través de la segunda etapa de retroalimentación, y generar el MAC para garantizar la integridad de los datos de texto cifrado,
caracterizado porque
la etapa de codificación y la etapa de generación de MAC comparten un módulo de codificación y un bucle de retroalimentación para realizar alternativamente el procedimiento de codificación y un procedimiento de generación de MAC, en el que el bucle de retroalimentación incluye:

una memoria para, respectivamente, almacenar y producir resultados del procedimiento de codificación y del procedimiento de generación de MAC, y
un selector para seleccionar alternativamente los resultados del procedimiento de codificación y del procedimiento de generación de MAC de la memoria para realizar, alternativamente, el procedimiento de codificación y el procedimiento de generación de MAC.

5. Un aparato de decodificación que comprende:

una unidad (300) de decodificación para introducir datos que han de decodificarse y producir datos decodificados, y
un generador (400) de códigos de autenticación de mensajes (MAC) para introducir los datos decodificados producidos por la unidad de decodificación y generar un MAC para garantizar una integridad de los datos codificados,
en el que el generador de MAC inicia la generación del MAC antes de que la unidad de decodificación acabe de decodificar los datos y en el que se decodifican datos de texto cifrado que incluyen al menos un dato en bloques de texto cifrado formando datos de texto en claro y se genera un código de autenticación de mensajes (MAC) para garantizar una integridad de los datos de texto cifrado,
en el que la unidad (300) de decodificación incluye un primer bucle de retroalimentación para retroalimentar datos en bloques de salida del módulo generados en la decodificación de datos por un módulo de decodificación, para introducir los datos en bloques de texto cifrado C_i, decodificar los datos en bloques de texto cifrado C_i usando los datos en bloques de salida del módulo retroalimentados a través del primer bucle de retroalimentación, y producir datos en bloques de texto en claro M_i,
en el que el aparato de decodificación comprende, además, una línea (69) de alimentación para introducir los datos en bloques de texto cifrado C_i que se introducen a la unidad (300) de decodificación y producir los datos en bloques de texto cifrado C_i para el generador (400) de códigos de autenticación de mensajes (MAC), y
en el que el generador (400) de MAC incluye un segundo bucle de retroalimentación para retroalimentar un resultado de MAC intermedio calculado T_i, para introducir los datos en bloques de texto cifrado C_i desde la línea (69) de alimentación, calcular un resultado de MAC intermedio en función de los datos en bloques de texto cifrado C_i introducidos, producir el resultado de MAC intermedio como el resultado de MAC intermedio calculado T_i, retroalimentar el resultado de MAC intermedio calculado T_i a través del segundo bucle de retroalimentación, y generar el MAC para garantizar la integridad de los datos de texto cifrado,
caracterizado porque
la unidad de decodificación y el generador de MAC comparten un módulo de decodificación y un bucle de retroalimentación y realizan alternativamente un procedimiento de decodificación y un procedimiento de generación de MAC, en el que el bucle de retroalimentación incluye:

una memoria que almacena y produce resultados del procedimiento de decodificación y del procedimiento de generación de MAC, y
un selector para seleccionar alternativamente los resultados del procedimiento de decodificación y del procedimiento de generación de MAC para dárselos al módulo de decodificación para realizar, alternativamente, el procedimiento de decodificación y el procedimiento de generación de MAC.

6. Un aparato de decodificación que comprende:

una unidad (300) de decodificación para introducir datos que han de decodificarse y producir datos decodificados, y
un generador (400) de códigos de autenticación de mensajes (MAC) para introducir los datos que han de decodificarse y generar un MAC para garantizar una integridad de los datos codificados,
en el que el generador de MAC inicia la generación del MAC antes de que la unidad de decodificación acabe de decodificar los datos y en el que se decodifican datos de texto cifrado que incluyen al menos un dato en bloques de texto cifrado formando datos de texto en claro usando un módulo de decodificación y se genera un código de autenticación de mensajes (MAC) para garantizar una integridad de los datos de texto cifrado,
en el que la unidad (300) de decodificación tiene un primer bucle de retroalimentación para retroalimentar datos en bloques de texto cifrado C_i, para introducir los datos en bloques de texto cifrado, realizar un procedimiento de decodificación retroalimentando los datos en bloques de texto cifrado C_i a través del primer bucle de retroalimentación, y producir datos en bloques de texto en claro,

en el que el aparato de decodificación comprende, además, una línea (69) de alimentación para introducir los datos en bloques de texto cifrado que se introducen a la unidad (300) de decodificación y producir los datos en bloques de texto cifrado para el generador (400) de códigos de autenticación de mensajes (MAC), y

en el que el generador (400) de códigos de autenticación de mensajes (MAC) tiene un segundo bucle de retroalimentación para retroalimentar un resultado de MAC intermedio calculado, para introducir los datos en bloques de texto cifrado desde la línea (69) de alimentación, calcular un resultado de MAC intermedio en función de los datos en bloques de texto cifrado introducidos, producir el resultado de MAC intermedio como el resultado de MAC intermedio calculado, retroalimentar el resultado de MAC intermedio calculado a través del segundo bucle de retroalimentación, y generar el MAC para garantizar la integridad de los datos de texto cifrado,

caracterizado porque

la unidad de decodificación y el generador de MAC comparten un módulo de decodificación y un bucle de retroalimentación para realizar alternativamente el procedimiento de decodificación y un procedimiento de generación de MAC, en el que el bucle de retroalimentación incluye:

una memoria para, respectivamente, almacenar y producir resultados del procedimiento de decodificación y del procedimiento de generación de MAC, y

un selector para seleccionar alternativamente los resultados del procedimiento de decodificación y del procedimiento de generación de MAC de la memoria para realizar, alternativamente, el procedimiento de decodificación y el procedimiento de generación de MAC.

7. Un procedimiento de decodificación que comprende:

una etapa de decodificación para introducir datos que han de decodificarse y producir datos decodificados, y

una etapa de generación de MAC para introducir los datos decodificados producidos en la etapa de decodificación y generar un MAC para garantizar una integridad de los datos codificados,

en el que la etapa de generación de MAC inicia la generación del MAC antes de que la etapa de decodificación acabe de decodificar los datos y en el que se decodifican datos de texto cifrado que incluyen al menos un dato en bloques de texto cifrado formando datos de texto en claro y se genera un código de autenticación de mensajes (MAC) para garantizar una integridad de los datos de texto cifrado,

en el que la etapa de decodificación incluye una primera etapa de retroalimentación para retroalimentar datos en bloques de salida del módulo generados en la decodificación de datos por un módulo de decodificación, introducir los datos en bloques de texto cifrado C_i, decodificar los datos en bloques de texto cifrado C_i usando los datos en bloques de salida del módulo retroalimentados a través del primer bucle de retroalimentación, y producir datos en bloques de texto en claro M_i,

en el que los datos en bloques de texto cifrado C_i introducidos en la etapa de de codificación son alimentados, por una etapa de alimentación, a la etapa de generación de MAC, y

en el que la etapa de generación de MAC incluye una segunda etapa de retroalimentación para retroalimentar un resultado de MAC intermedio calculado T_i, introducir los datos en bloques de texto cifrado C_i en la etapa de alimentación, calcular un resultado de MAC intermedio en función de los datos en bloques de texto cifrado C_i introducidos, producir el resultado de MAC intermedio como el resultado de MAC intermedio calculado T_i, retroalimentar el resultado de MAC intermedio calculado T_i por un segundo bucle de retroalimentación, y generar el MAC para garantizar la integridad de los datos de texto cifrado,

caracterizado porque

la etapa de decodificación y la etapa de generación de MAC comparten un módulo de decodificación y un bucle de retroalimentación y realizan alternativamente el procedimiento de decodificación y un procedimiento de generación de MAC, en el que el bucle de retroalimentación incluye:

una memoria que almacena y produce resultados del procedimiento de decodificación y del procedimiento de generación de MAC, y

un selector para seleccionar alternativamente los resultados del procedimiento de decodificación y del procedimiento de generación de MAC para dárselos al módulo de decodificación para realizar, alternativamente, el procedimiento de decodificación y el procedimiento de generación de MAC.

8. Un procedimiento de decodificación que comprende:

una etapa de decodificación para introducir datos que han de decodificarse y producir datos decodificados, y

una etapa de generación de MAC para introducir los datos que han de decodificarse y generar un MAC para garantizar una integridad de los datos codificados,

en el que la etapa de generación de MAC inicia la generación del MAC antes de que la etapa de decodificación acabe de decodificar los datos y en el que se decodifican datos de texto cifrado que incluyen al menos un dato en bloques de texto cifrado formando datos de texto en claro usando una unidad de

decodificación y se genera un código de autenticación de mensajes (MAC) para garantizar una integridad de los datos de texto cifrado,
en el que la etapa de decodificación tiene una primera etapa de retroalimentación para retroalimentar datos en bloques de texto cifrado C_i, para introducir los datos en bloques de texto cifrado, realizar un procedimiento de decodificación de los datos en bloques de texto cifrado C_i retroalimentados a través de un primer bucle de retroalimentación, y producir datos en bloques de texto en claro,
en el que los datos en bloques de texto cifrado introducidos en la etapa de decodificación son alimentados, por una etapa de alimentación, a la etapa de generación de MAC, y
en el que la etapa de generación de MAC tiene una segunda etapa de retroalimentación para retroalimentar un resultado de MAC intermedio calculado, introducir los datos en bloques de texto cifrado C_i en la etapa de alimentación, calcular un resultado de MAC intermedio en función de los datos en bloques de texto cifrado C_i introducidos, producir el resultado de MAC intermedio como el resultado de MAC intermedio calculado, retroalimentar el resultado de MAC intermedio calculado a través de la segunda etapa de retroalimentación, y generar el MAC para garantizar la integridad de los datos de texto cifrado,
caracterizado porque
la etapa de decodificación y la etapa de generación de MAC comparten un módulo de decodificación y un bucle de retroalimentación para realizar alternativamente el procedimiento de decodificación y un procedimiento de generación de MAC, en el que el bucle de retroalimentación incluye:

una memoria para, respectivamente, almacenar y producir resultados del procedimiento de decodificación y del procedimiento de generación de MAC, y
un selector para seleccionar alternativamente los resultados del procedimiento de decodificación y del procedimiento de generación de MAC de la memoria para realizar, alternativamente, el procedimiento de decodificación y el procedimiento de generación de MAC.

9. Un medio de almacenamiento legible por ordenador que almacena un programa para hacer que un ordenador ejecute etapas del procedimiento de codificación descrito en una de las reivindicaciones 3 a 4.

10. Un medio de almacenamiento legible por ordenador que almacena un programa para hacer que un ordenador ejecute etapas del procedimiento de decodificación descrito en una de las reivindicaciones 7 a 8.

Fig. 1

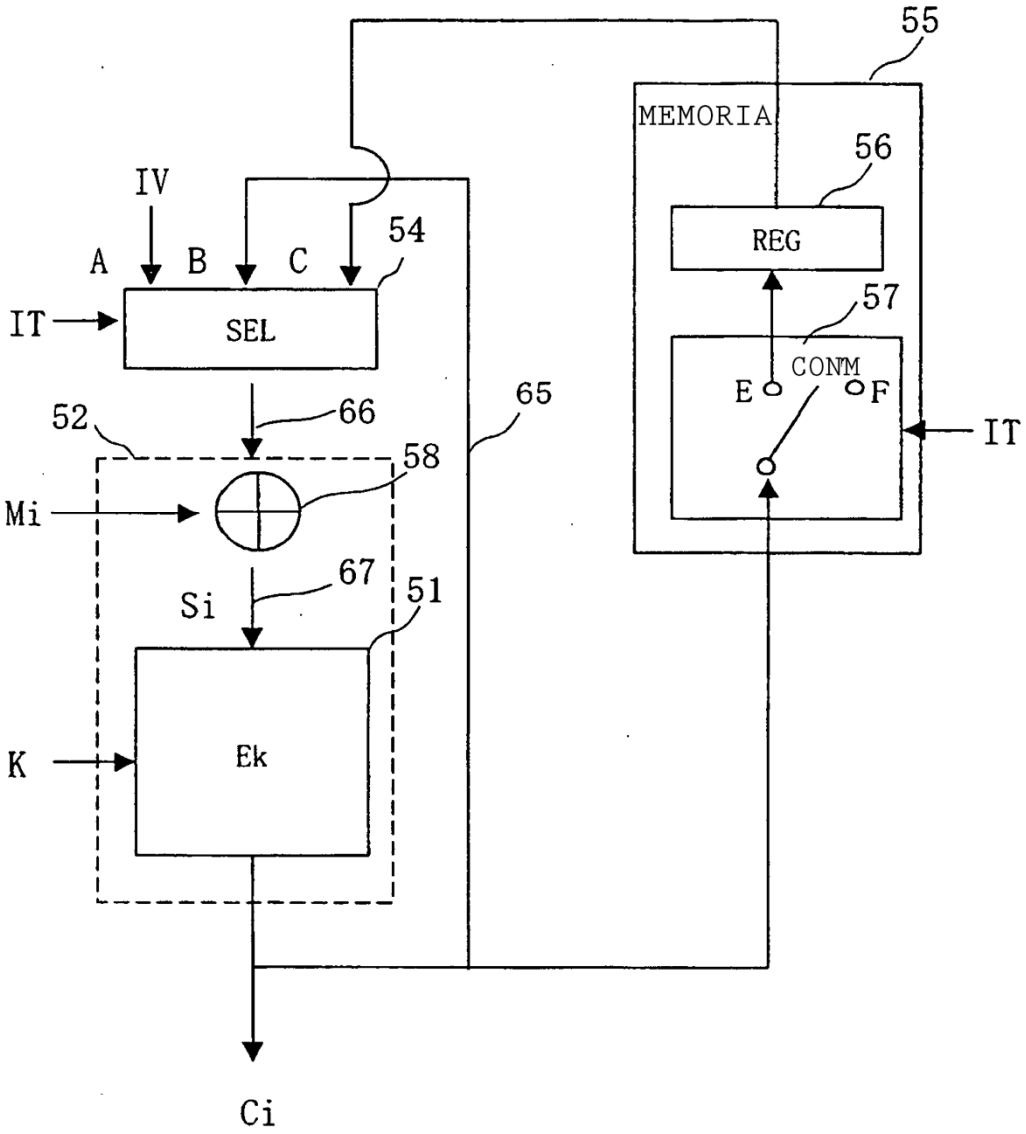
55
MEMORIA
56
REG
57
CONM
E
F
IT
IV
A
B
C
54
IT
SEL
65
52
66
58
Mi
67
Si
51
K
Ek
Ci

Fig. 2

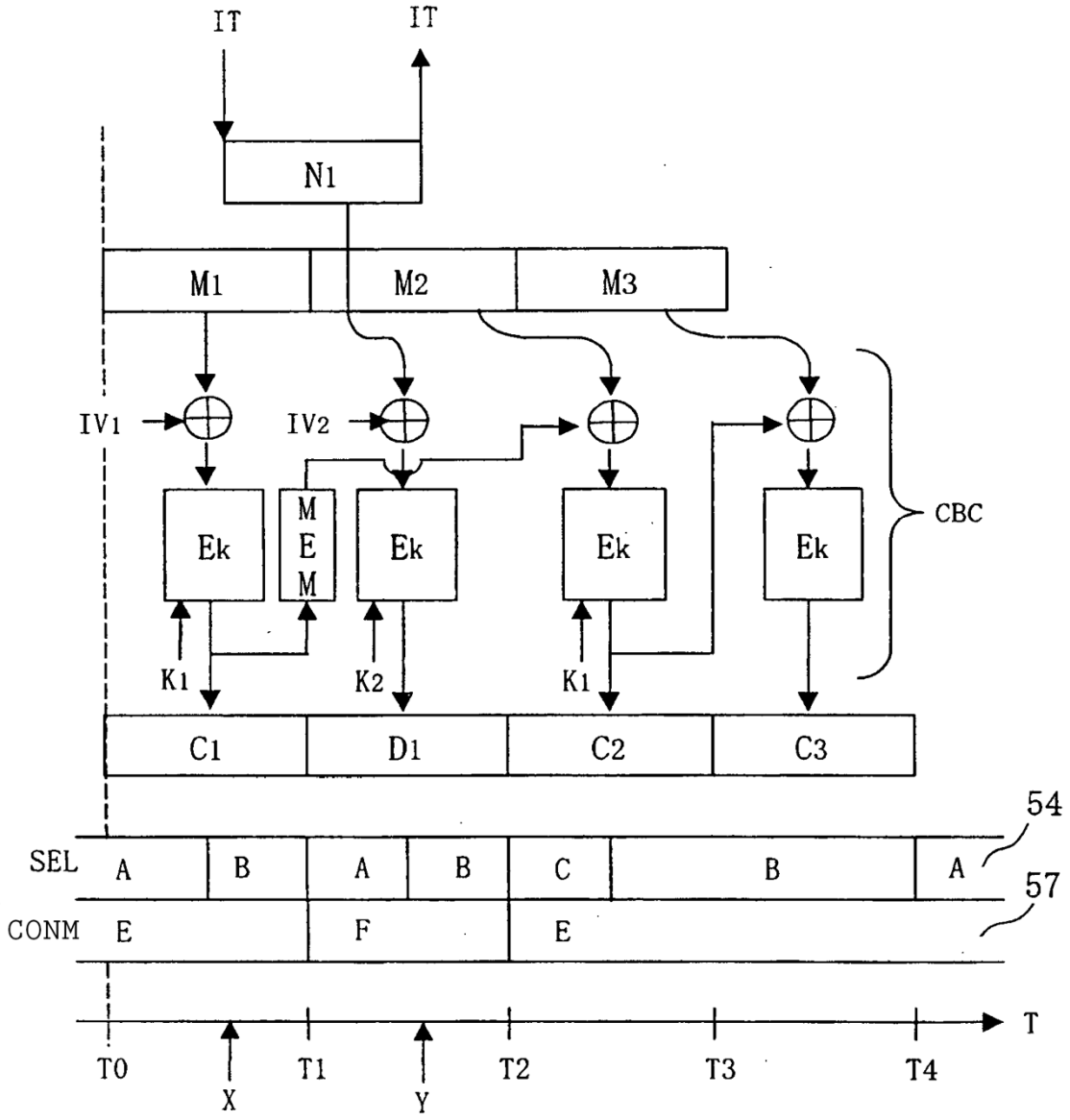
IT
IT
N1
M1
M2
M3
IV1
IV2
Ek
MEM
Ek
Ek
Ek
CBC
K1
K2
K1
C1
D1
C2
C3
54
SEL
A
B
A
B
C
B
A
57
CONM
E
F
E
T
T0
X
T1
Y
T2
T3
T4

Fig. 3

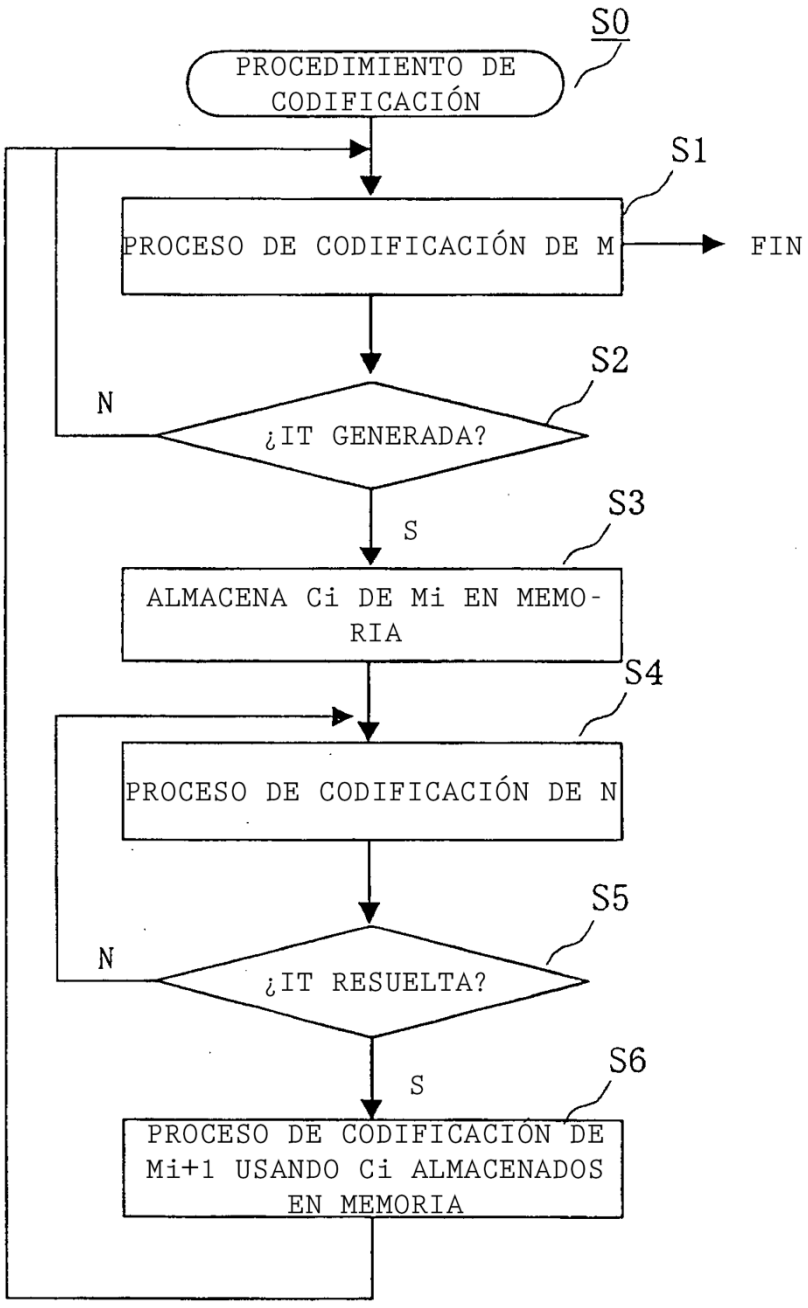
S0
PROCEDIMIENTO DE CODIFICACIÓN
S1
PROCESO DE CODIFICACIÓN DE M
FIN
S2
N
¿IT GENERADA?
S
S3
ALMACENA Ci DE Mi EN MEMORIA
S4
PROCESO DE CODIFICACIÓN DE N
S5
N
¿IT RESUELTA?
S
S6
PROCESO DE CODIFICACIÓN DE Mi+1 USANDO Ci ALMACENADOS EN MEMORIA

Fig. 4

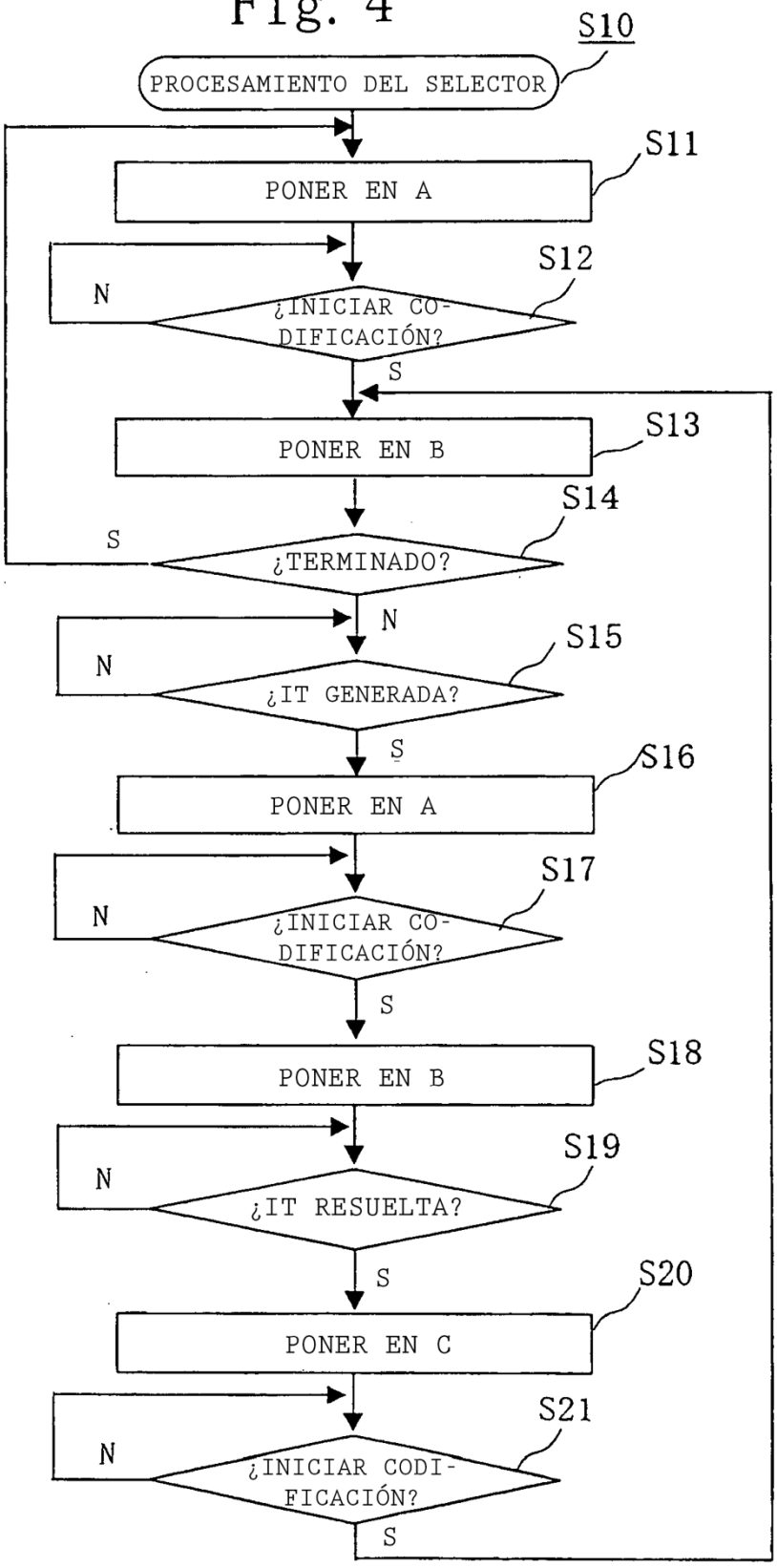
S10
PROCESAMIENTO DEL SELECTOR
S11
PONER EN A
S12
N
¿INICIAR CO-DIFICACIÓN?
S
S13
PONER EN B
S14
S
¿TERMINADO?
N
S15
N
¿IT GENERADA?
S
S16
PONER EN A
S17
N
¿INICIAR CO-DIFICACIÓN?
S
S18
PONER EN B
S19
N
¿IT RESUELTA?
S
S20
PONER EN C
S21
N
¿INICIAR CODI-FICACIÓN?
S

Fig. 5

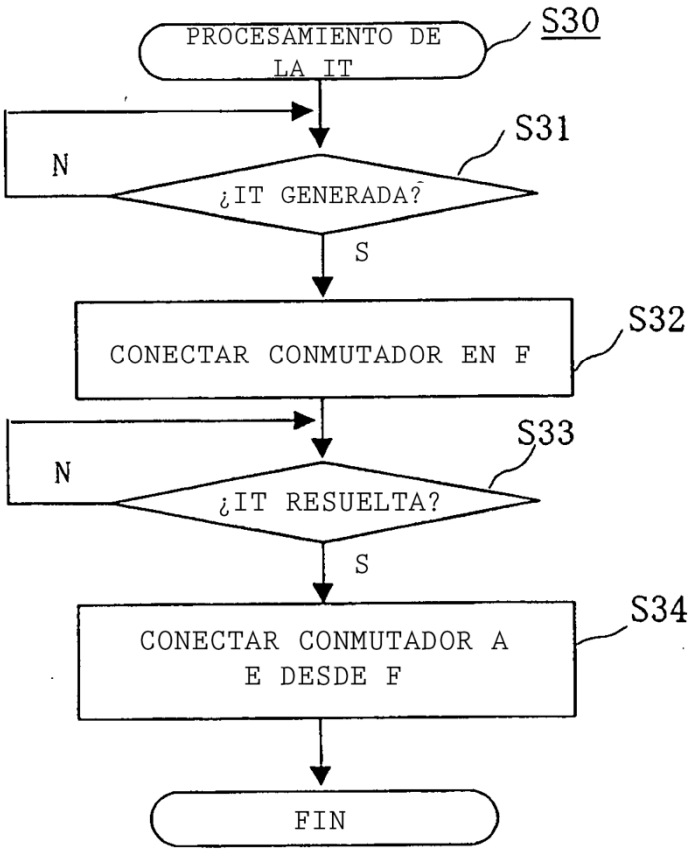
PROCESAMIENTO DE LA IT
S30
S31
N
¿IT GENERADA?
S
CONECTAR CONMUTADOR EN F
S32
S33
N
¿IT RESUELTA?
S
CONECTAR CONMUTADOR A E DESDE F
S34
FIN

Fig. 6

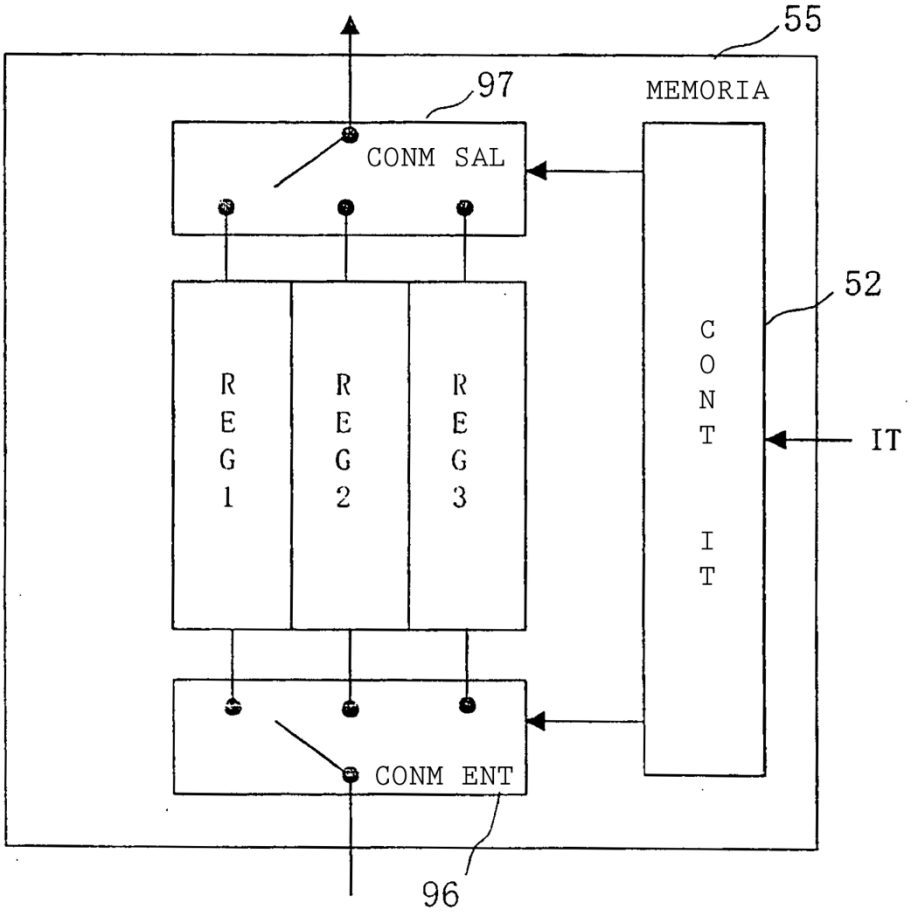

Fig. 7

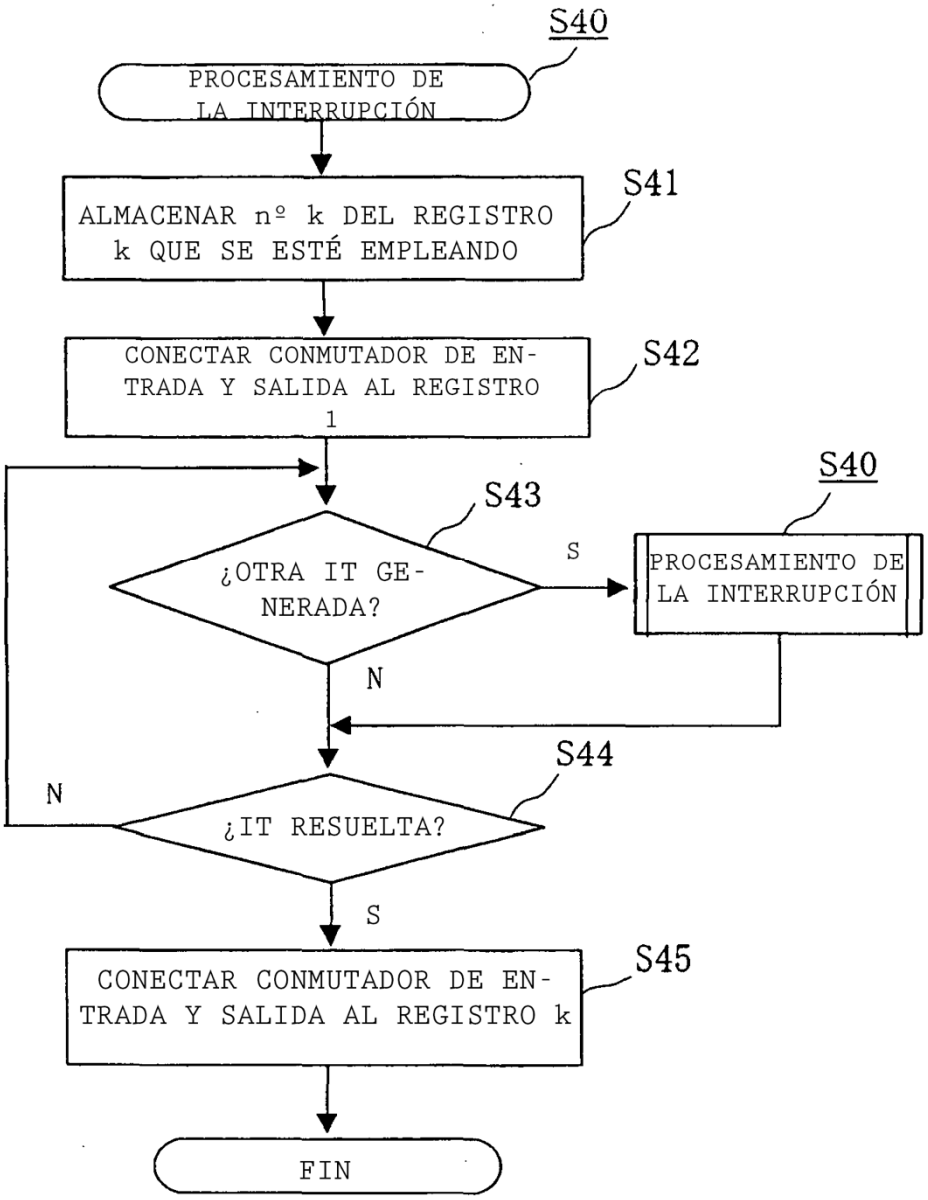
S40
PROCESAMIENTO DE LA INTERRUPCIÓN
S41
ALMACENAR nº k DEL REGISTRO k QUE SE ESTÉ EMPLEANDO
S42
CONECTAR CONMUTADOR DE ENTRADA Y SALIDA AL REGISTRO 1
S43
¿OTRA IT GENERADA?
S
S40
PROCESAMIENTO DE LA INTERRUPCIÓN
N
S44
¿IT RESUELTA?
N
S
S45
CONECTAR CONMUTADOR DE ENTRADA Y SALIDA AL REGISTRO k
FIN

Fig. 8

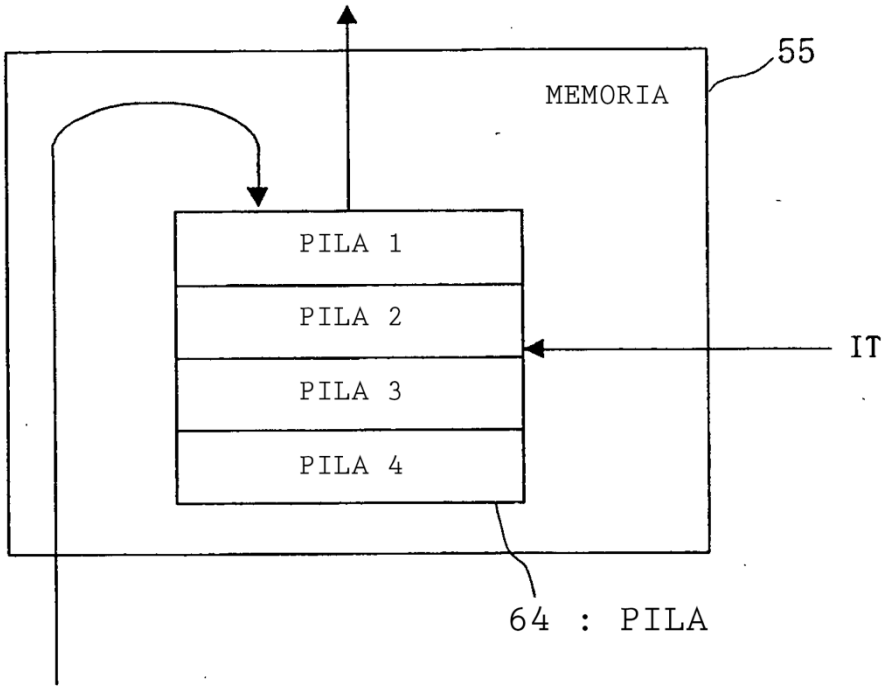
55
MEMORIA
PILA 1
PILA 2
PILA 3
PILA 4
IT
64 : PILA

Fig. 9

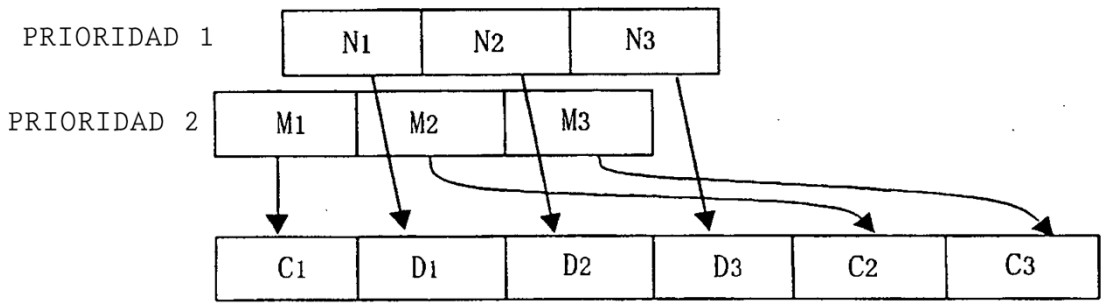
PRIORIDAD 1
PRIORIDAD 2
N1
N2
N3
M1
M2
M3
C1
D1
D2
D3
C2
C3

Fig. 10

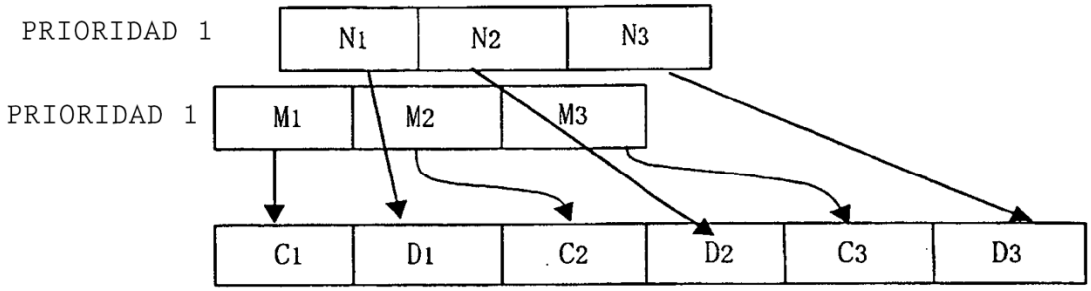
PRIORIDAD 1
PRIORIDAD 1
N1
N2
N3
M1
M2
M3
C1
D1
C2
D2
C3
D3

Fig. 11

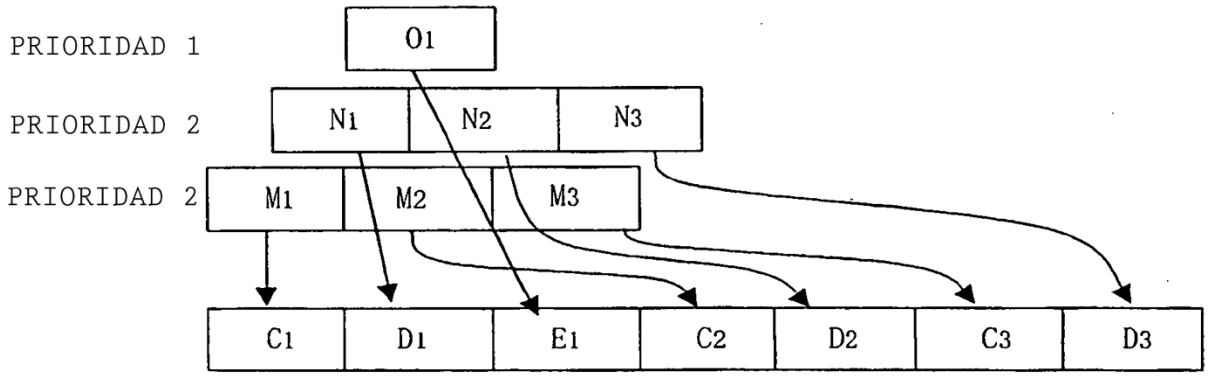
PRIORIDAD 1
PRIORIDAD 2
PRIORIDAD 2
O1
N1
N2
N3
M1
M2
M3
C1
D1
E1
C2
D2
C3
D3

Fig. 12

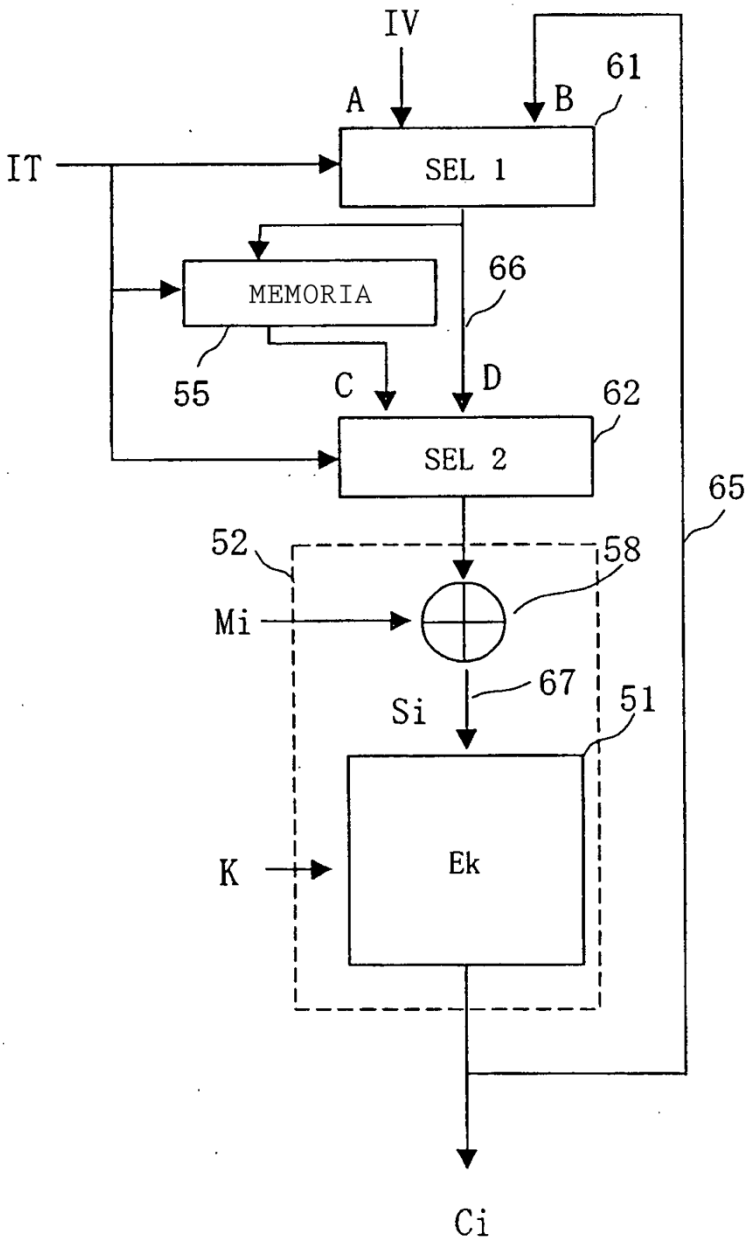
IV
A
B
61
IT
SEL 1
MEMORIA
66
55
C
D
62
SEL 2
65
52
58
Mi
67
Si
51
K
Ek
Ci

Fig. 13

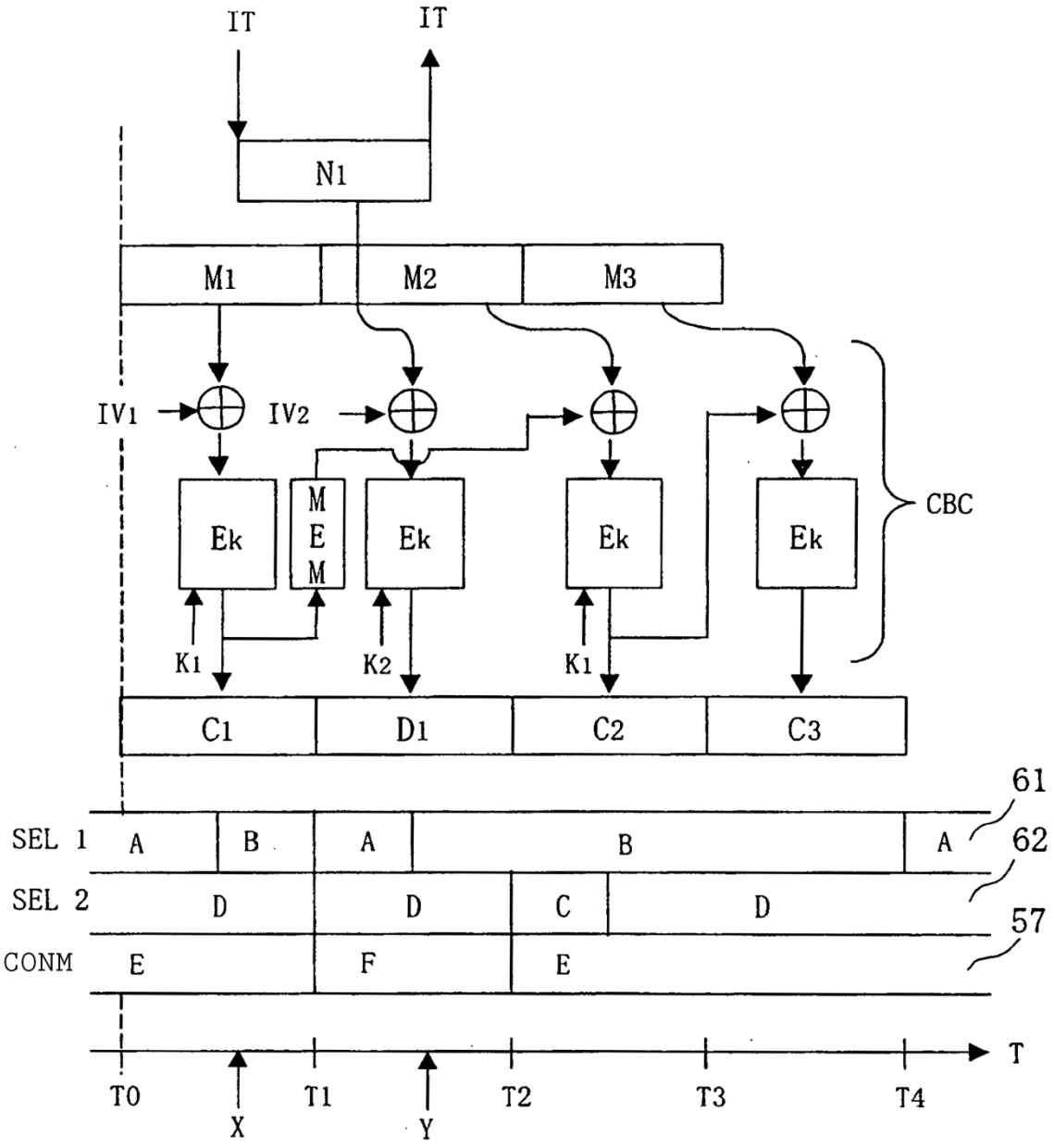
IT
IT
N1
M1
M2
M3
IV1
IV2
Ek
MEM
Ek
Ek
Ek
CBC
K1
K2
K1
C1
D1
C2
C3
61
SEL 1
A
B
A
B
A
62
SEL 2
D
D
C
D
57
CONM
E
F
E
T
T0
T1
T2
T3
T4
X
Y

Fig. 14

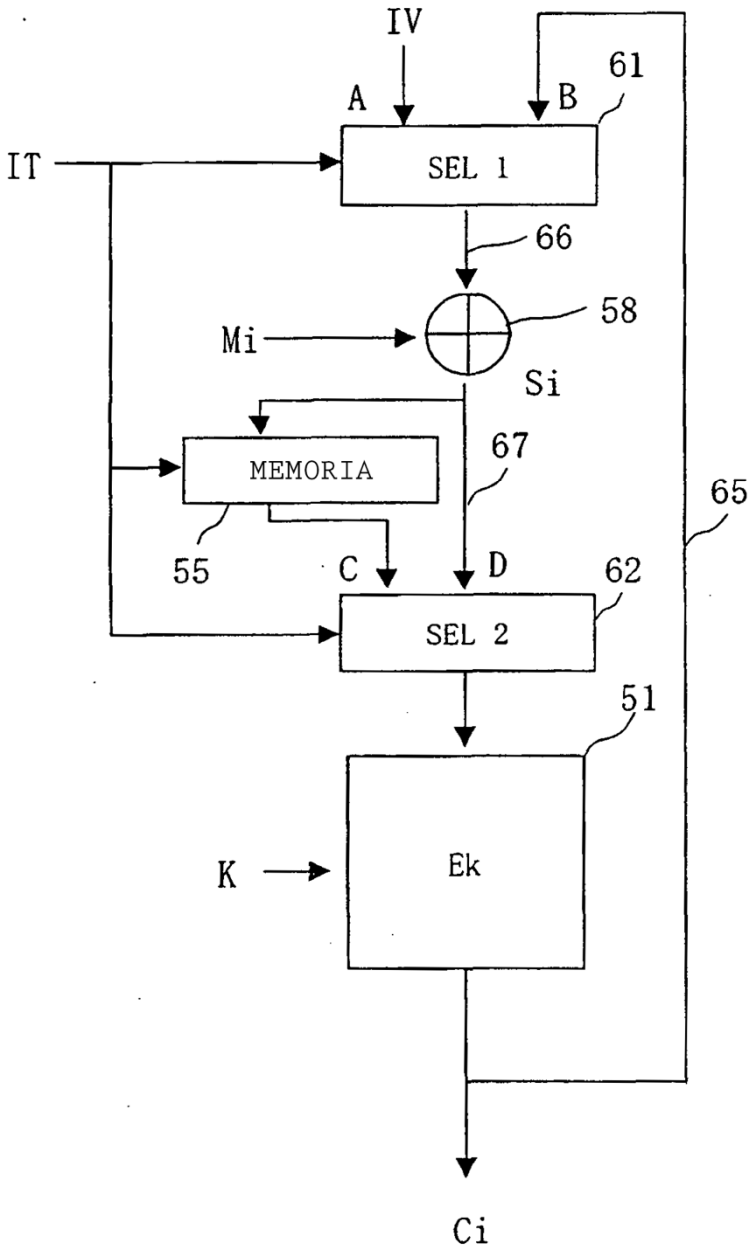

Fig. 15

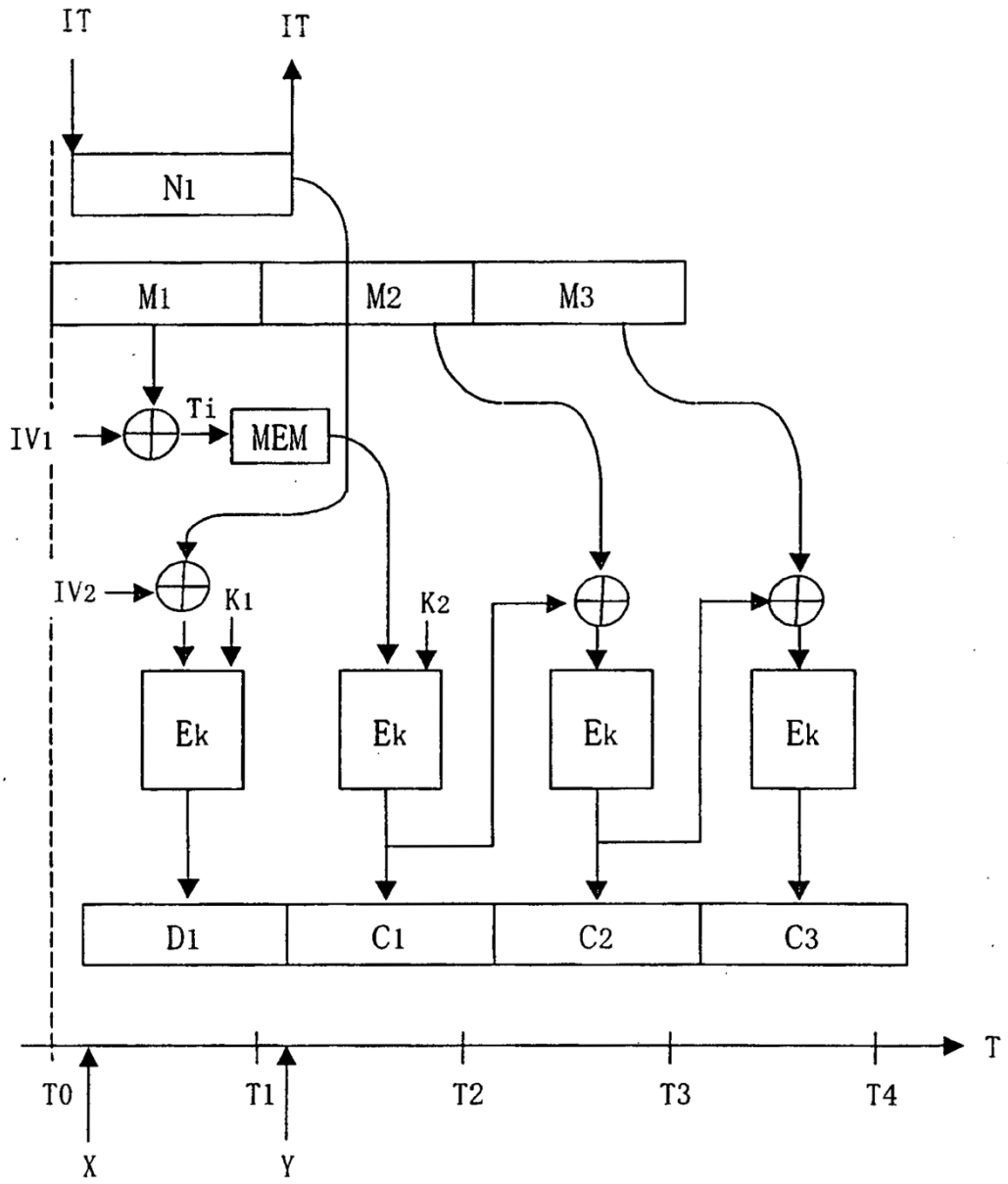
IT
IT
N1
M1
M2
M3
IV1
Ti
MEM
IV2
K1
K2
Ek
Ek
Ek
Ek
D1
C1
C2
C3
T
T0
T1
T2
T3
T4
X
Y

Fig. 16

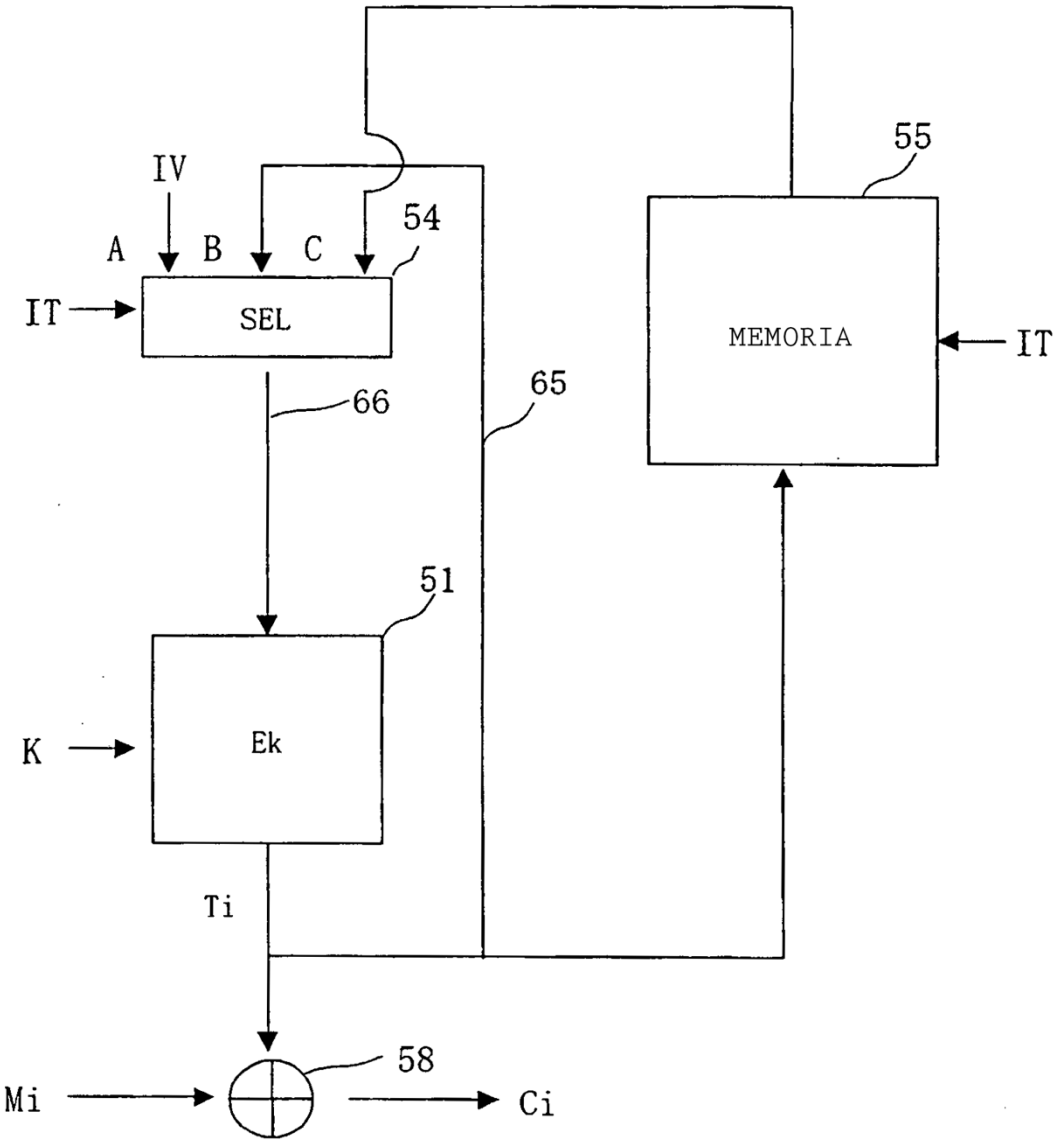

Fig. 17

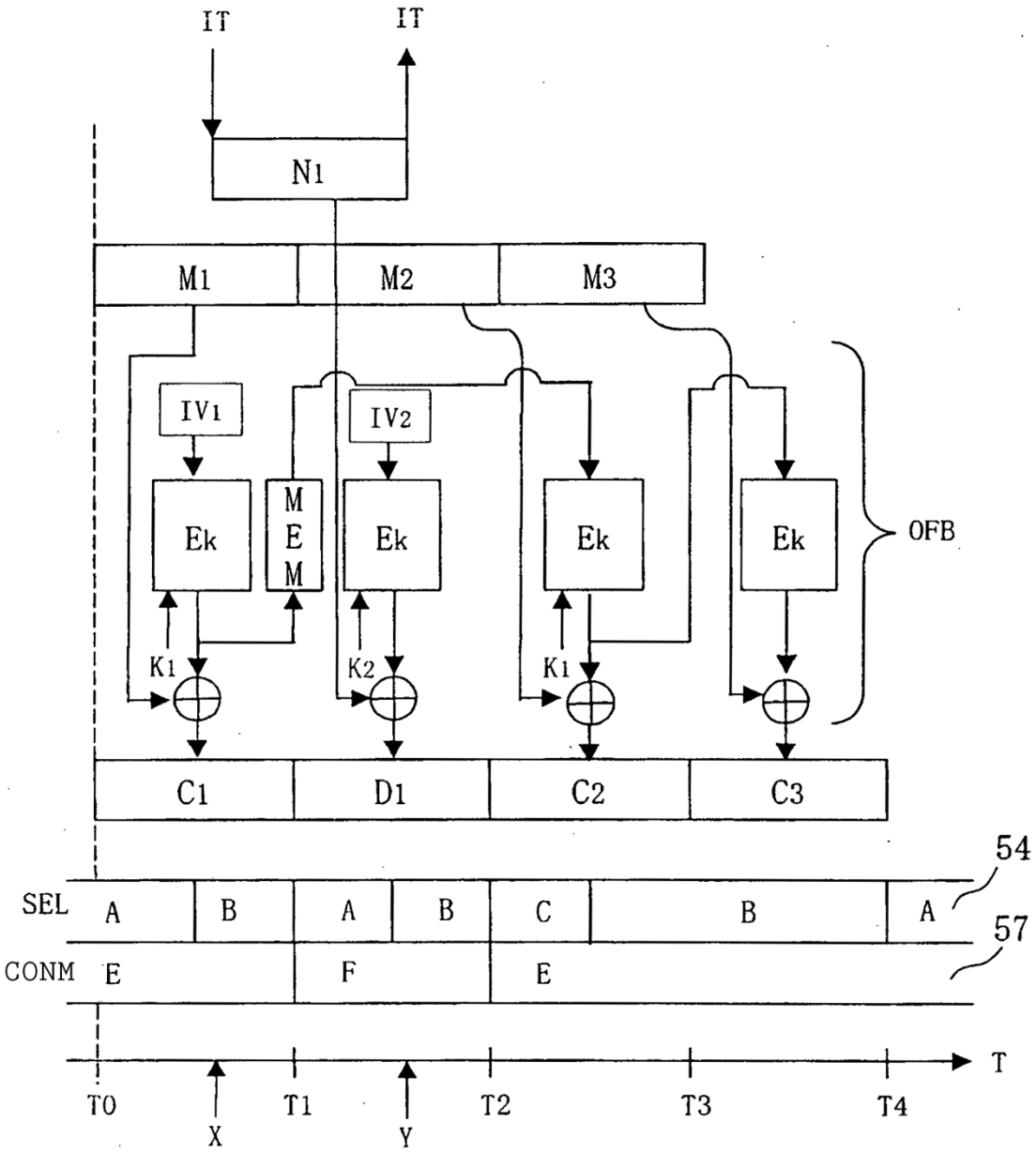
IT
IT
N1
M1
M2
M3
IV1
IV2
Ek
MEM
Ek
Ek
Ek
OFB
K1
K2
K1
C1
D1
C2
C3
54
SEL
A
B
A
B
C
B
A
57
CONM
E
F
E
T
T0
X
T1
Y
T2
T3
T4

Fig. 18

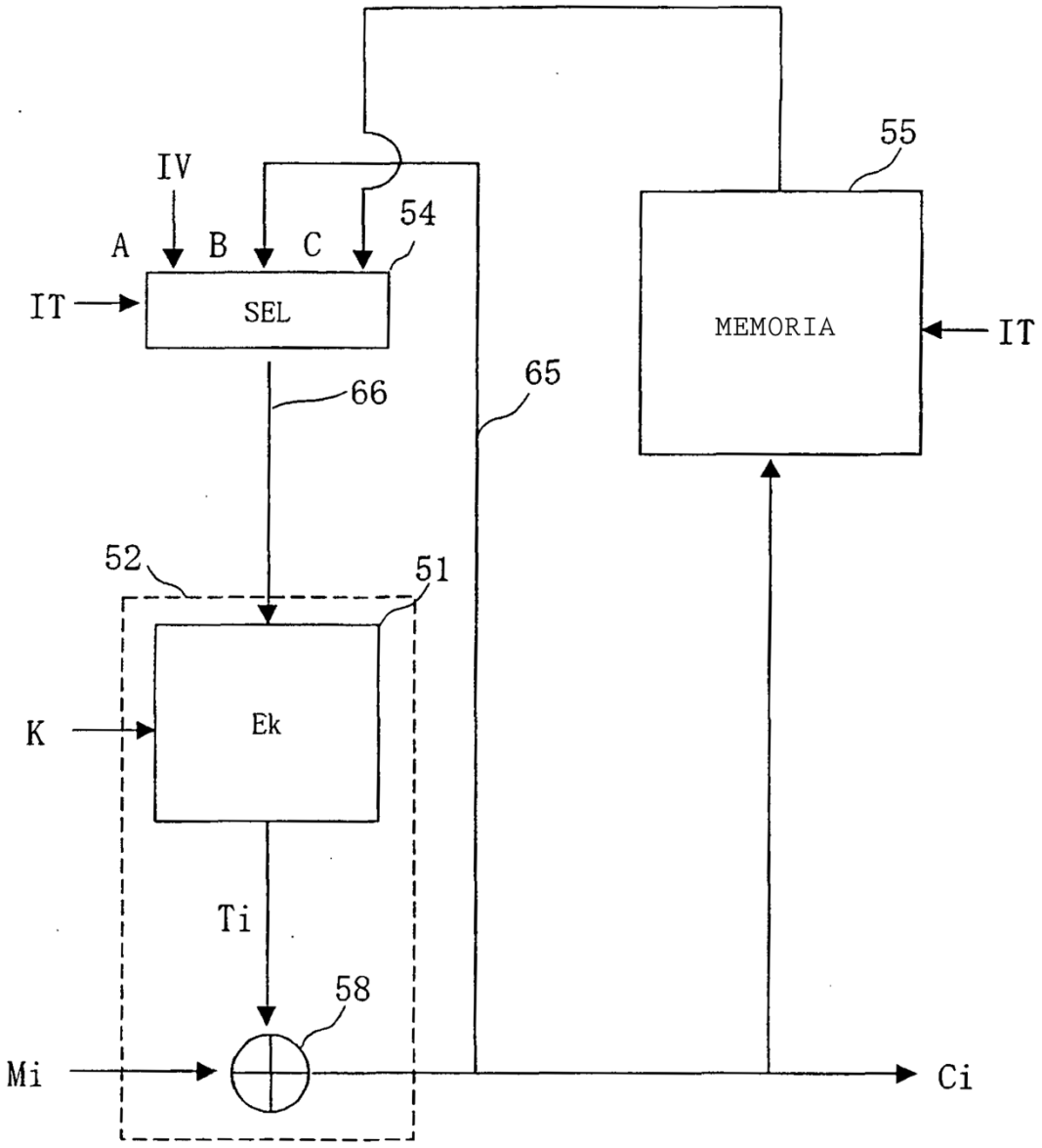

Fig. 19

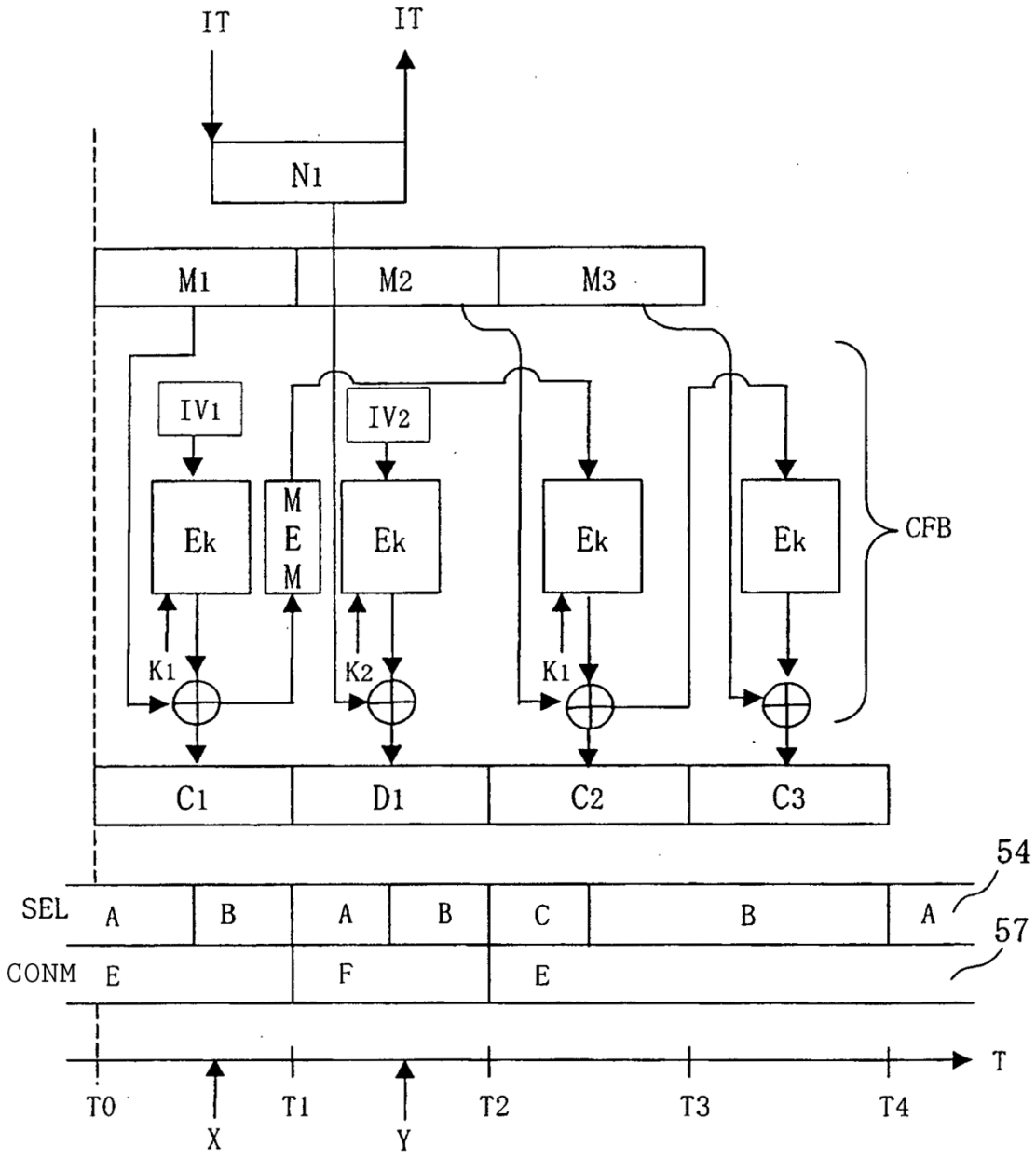
IT
IT
N1
M1
M2
M3
IV1
IV2
Ek
MEM
Ek
Ek
Ek
CFB
K1
K2
K1
C1
D1
C2
C3
54
SEL
A
B
A
B
C
B
A
57
CONM
E
F
E
T
T0
X
T1
Y
T2
T3
T4

Fig.20

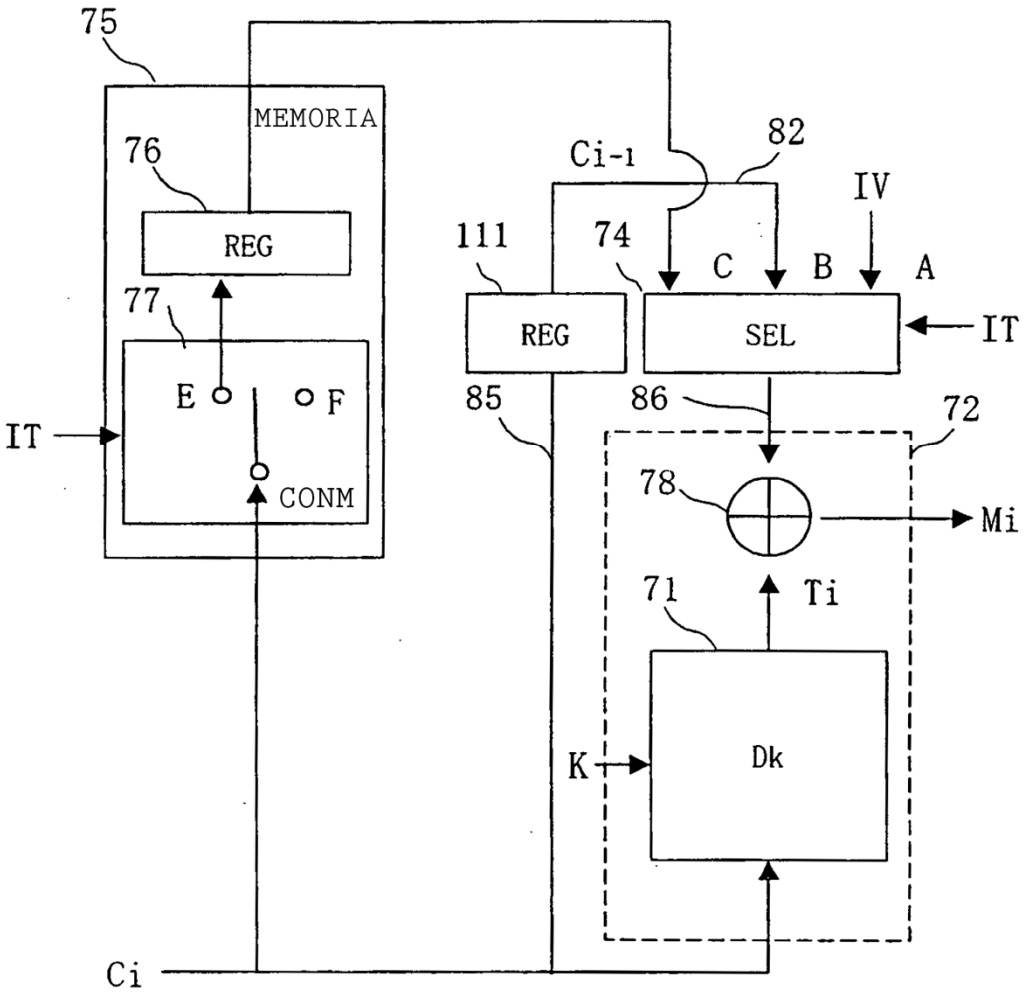
75
MEMORIA
76
REG
77
E
F
CONM
IT
Ci
Ci-1
82
IV
111
74
C
B
A
REG
SEL
IT
85
86
72
78
Mi
71
Ti
K
Dk

Fig. 21

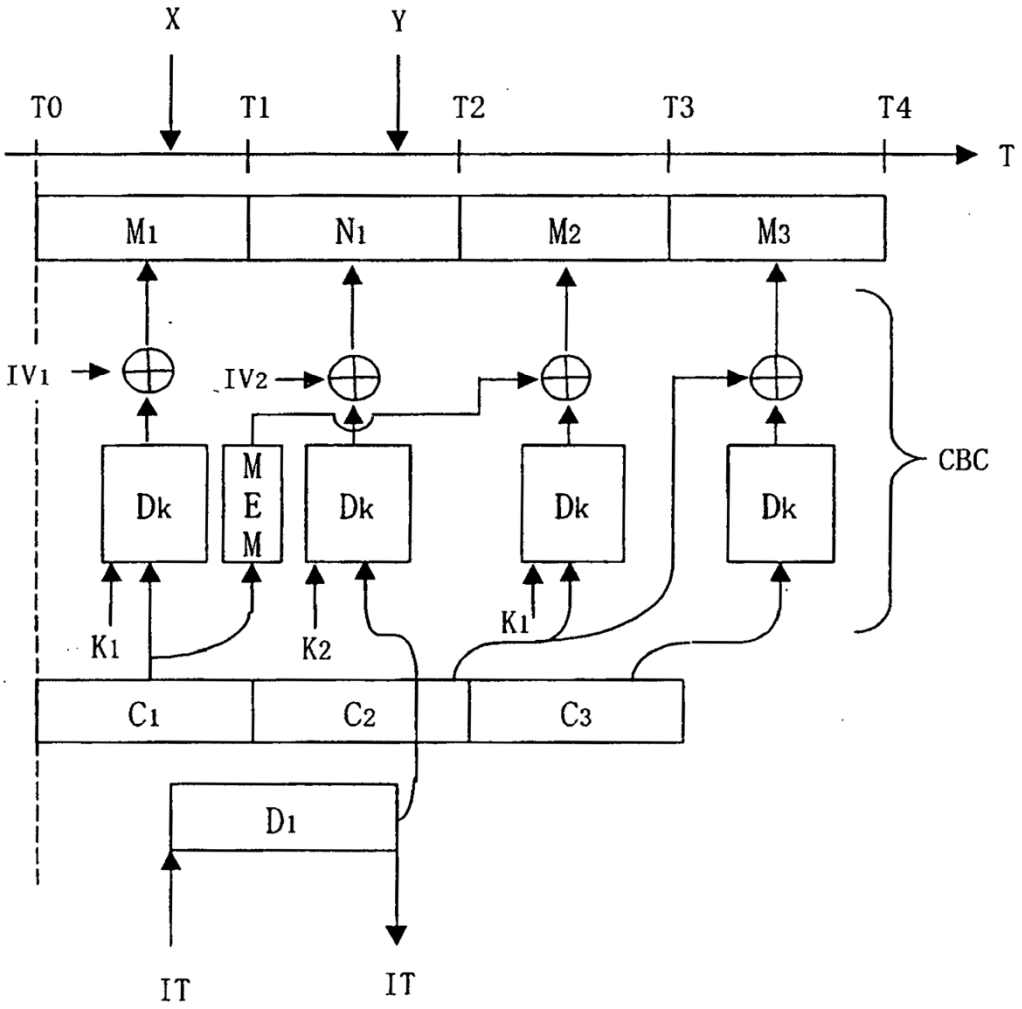
X
Y
T0
T1
T2
T3
T4
T
M1
N1
M2
M3
IV1
IV2
Dk
MEM
Dk
Dk
Dk
CBC
K1
K2
K1
C1
C2
C3
D1
IT
IT

Fig. 22

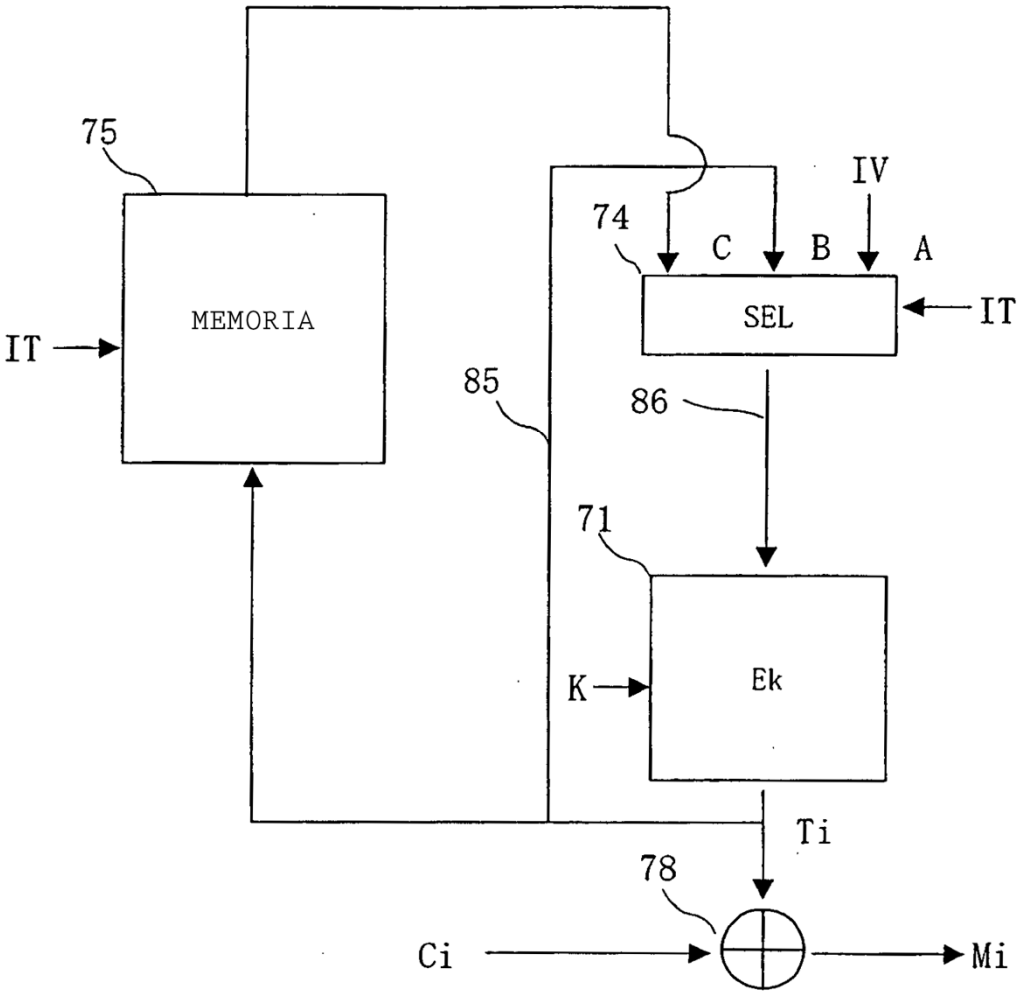

Fig. 23

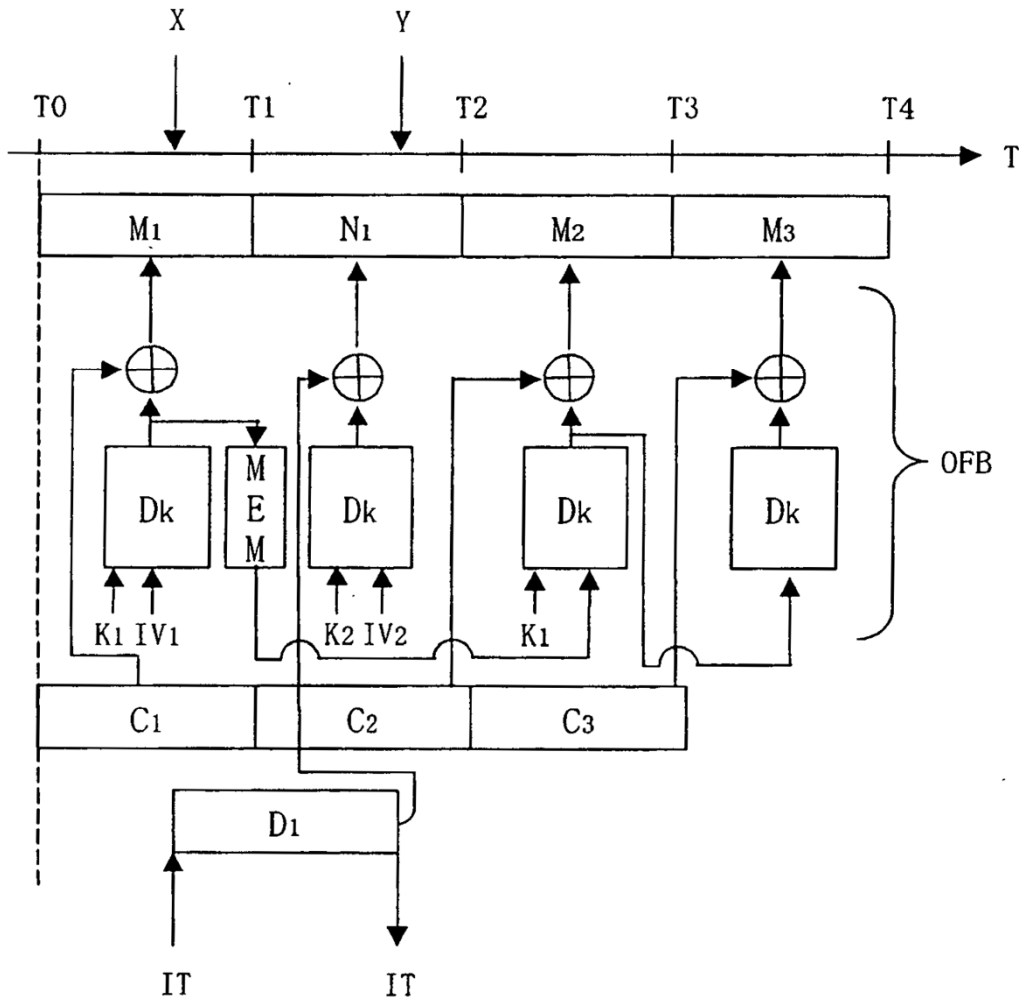
X
Y
T0
T1
T2
T3
T4
T
M1
N1
M2
M3
Dk
MEM
Dk
Dk
Dk
OFB
K1 IV1
K2 IV2
K1
C1
C2
C3
D1
IT
IT

Fig.24

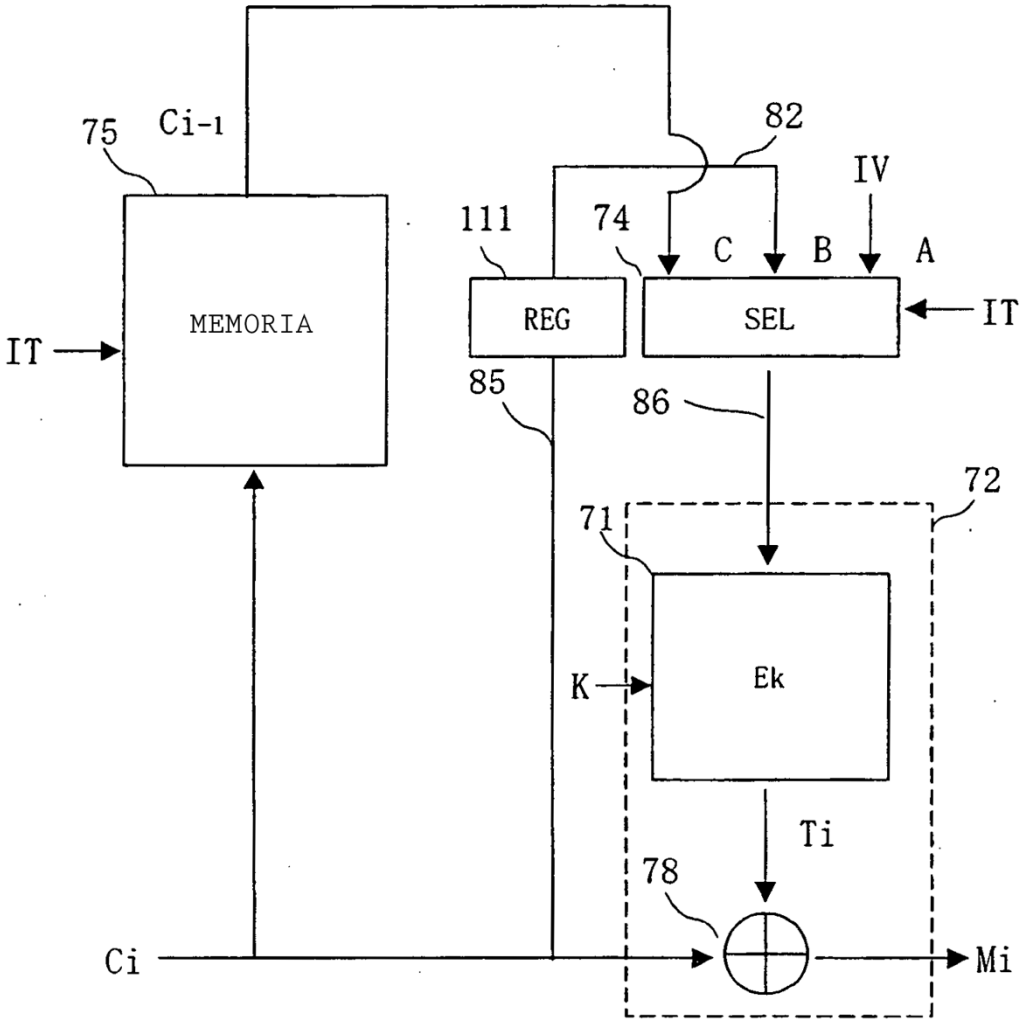
75
Ci-1
82
IV
111
74
C
B
A
MEMORIA
REG
SEL
IT
IT
85
86
72
71
K
Ek
Ti
78
Ci
Mi

Fig.25

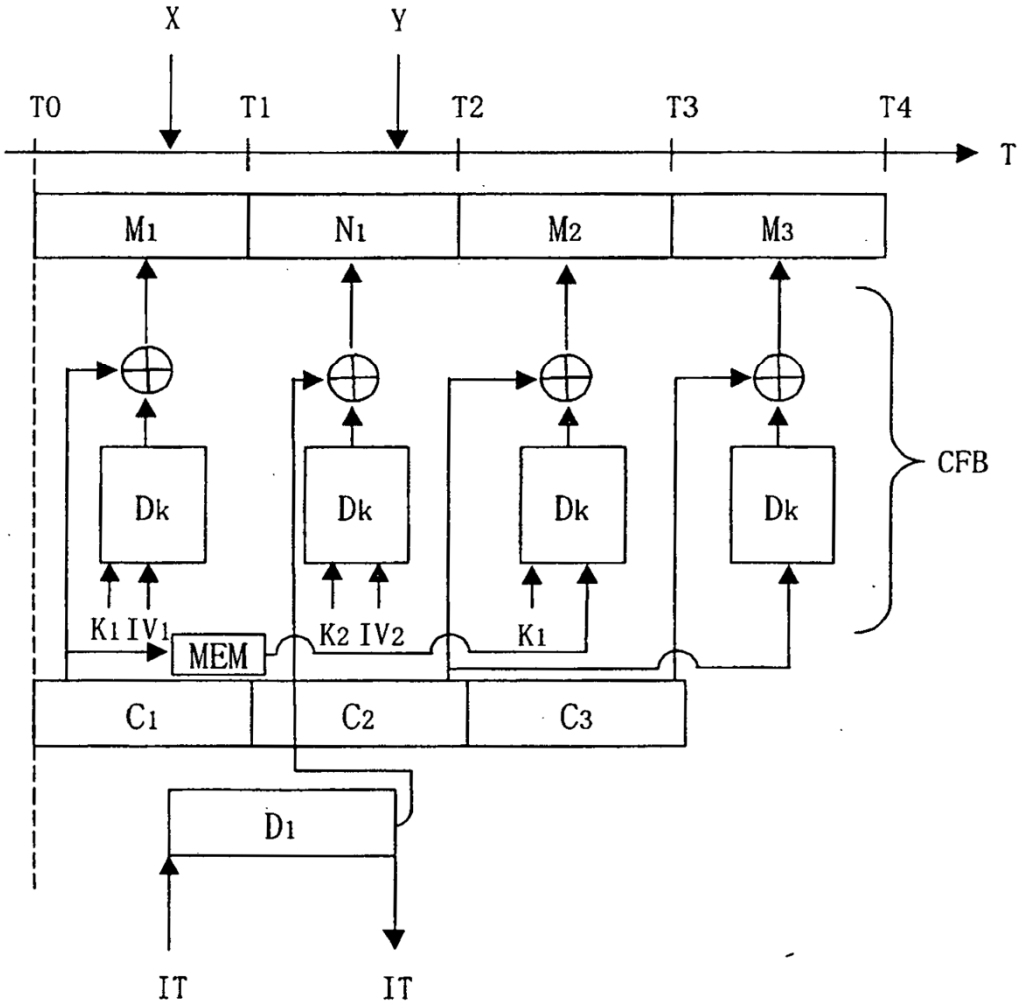
X
Y
T0
T1
T2
T3
T4
T
M1
N1
M2
M3
Dk
Dk
Dk
Dk
CFB
K1 IV1
MEM
K2 IV2
K1
C1
C2
C3
D1
IT
IT

Fig.26

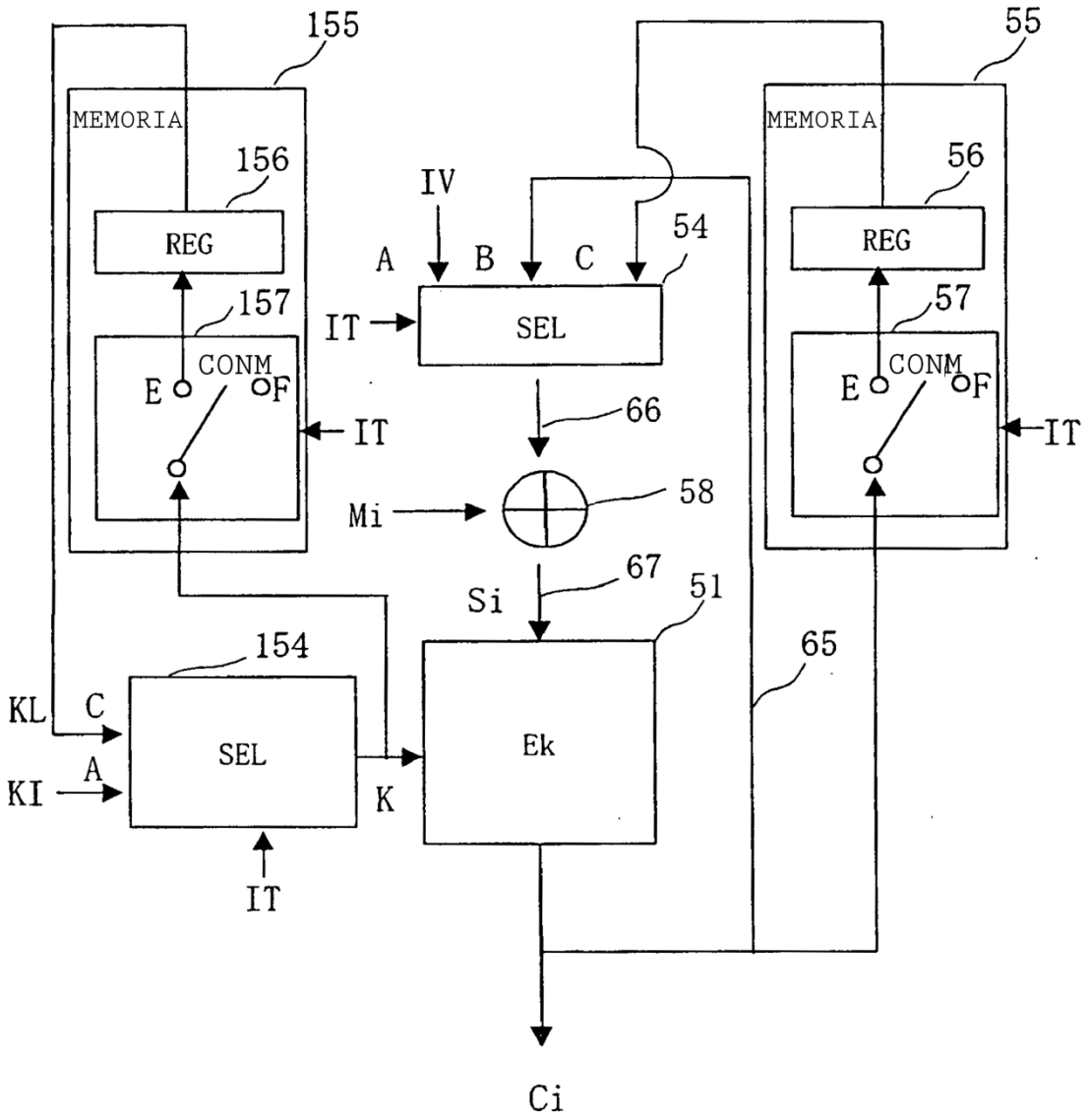
155
MEMORIA
156
REG
157
CONM
E
F
IT
55
MEMORIA
56
REG
57
CONM
E
F
IT
IV
A
B
C
54
IT
SEL
66
Mi
58
67
Si
51
65
154
KL
C
A
KI
SEL
K
Ek
IT
Ci

Fig.27

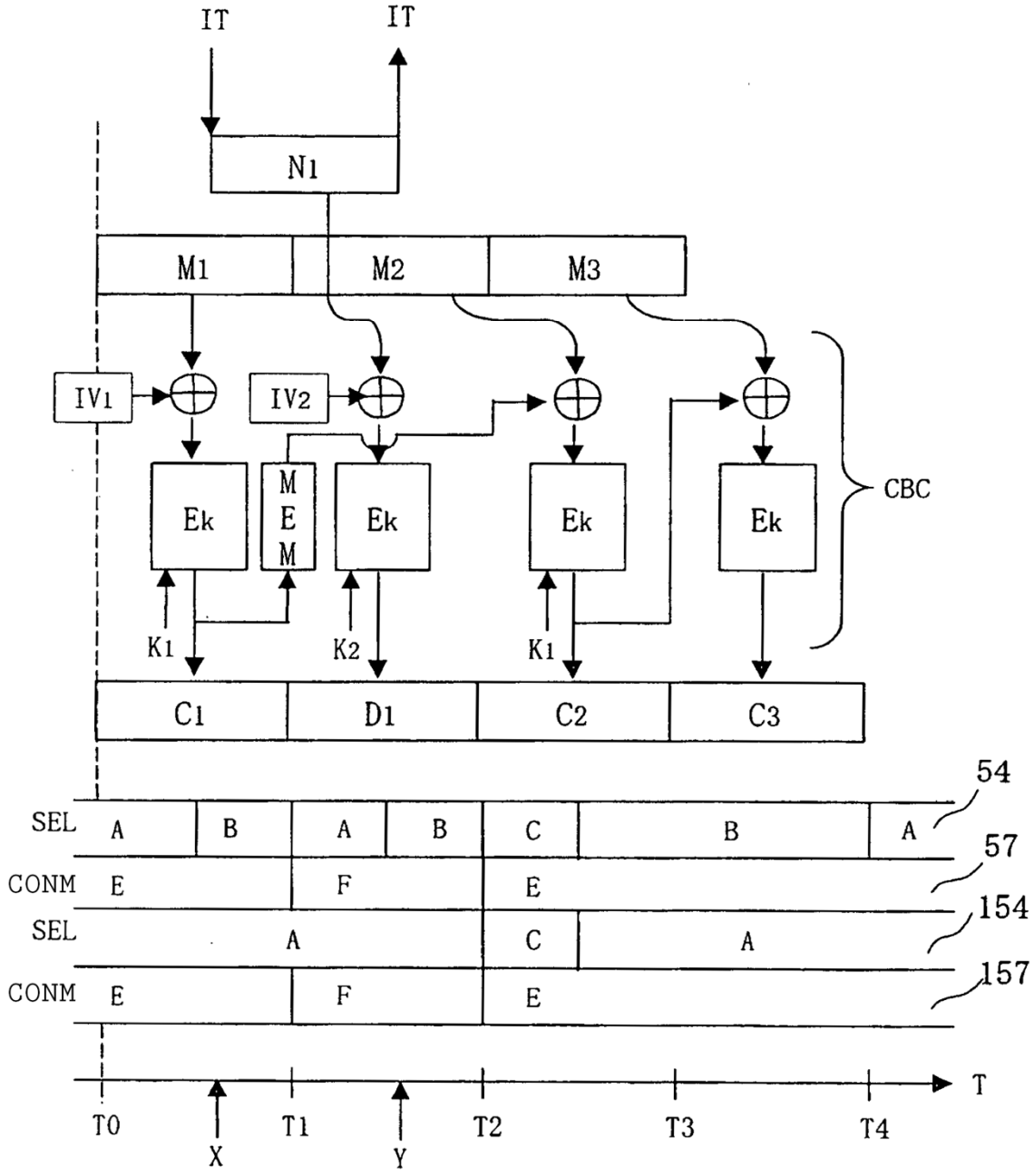
IT
IT
N1
M1
M2
M3
IV1
IV2
Ek
MEM
Ek
Ek
Ek
CBC
K1
K2
K1
C1
D1
C2
C3
54
SEL
A
B
A
B
C
B
A
57
CONM
E
F
E
154
SEL
A
C
A
157
CONM
E
F
E
T
T0
X
T1
Y
T2
T3
T4

Fig.28

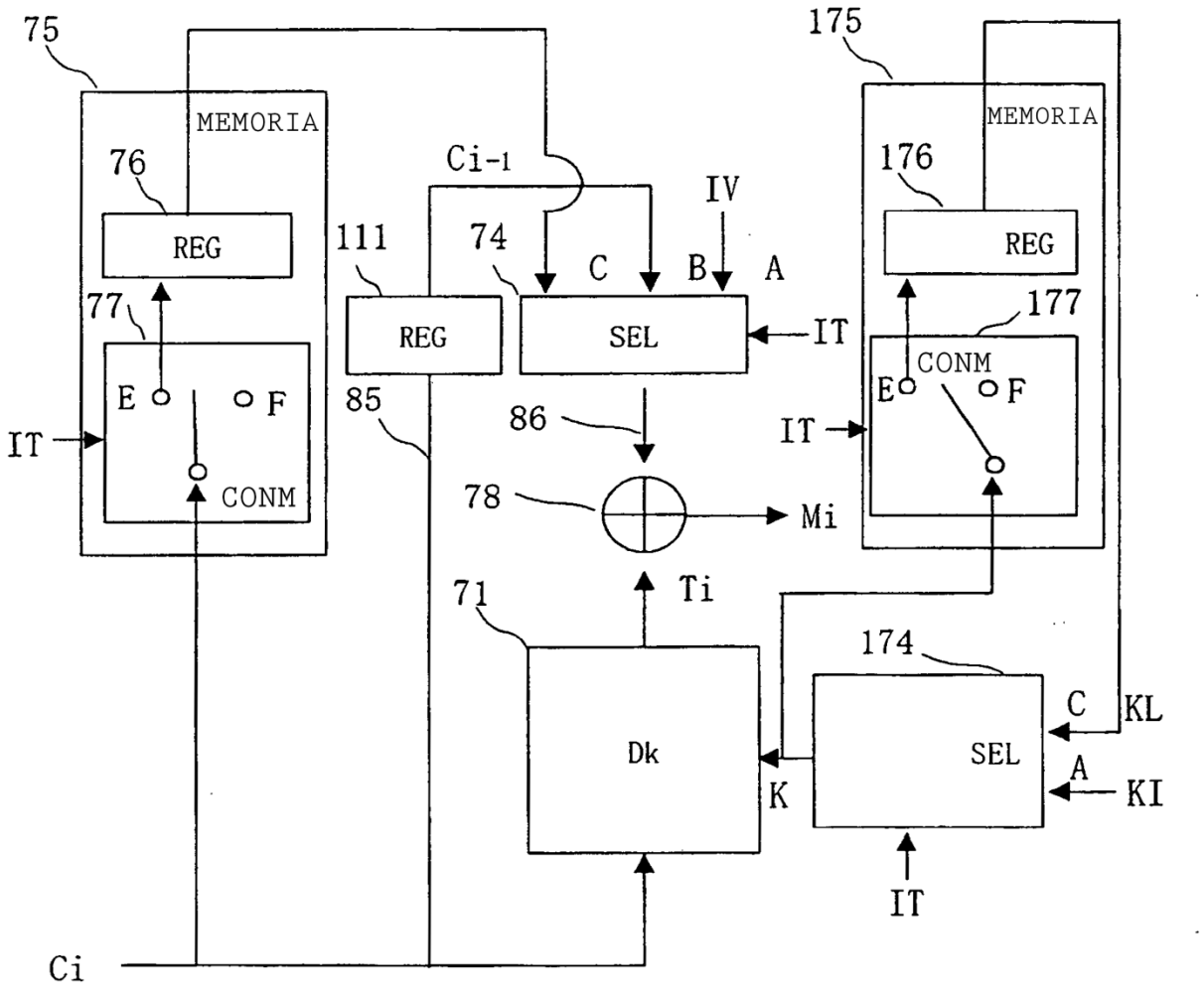
75
MEMORIA
76
REG
77
E
F
CONM
IT
Ci
111
REG
Ci-1
74
C
B
A
IV
SEL
IT
85
86
78
Mi
Ti
71
Dk
K
175
MEMORIA
176
REG
177
CONM
E
F
IT
174
SEL
C
KL
A
KI
IT

Fig.29

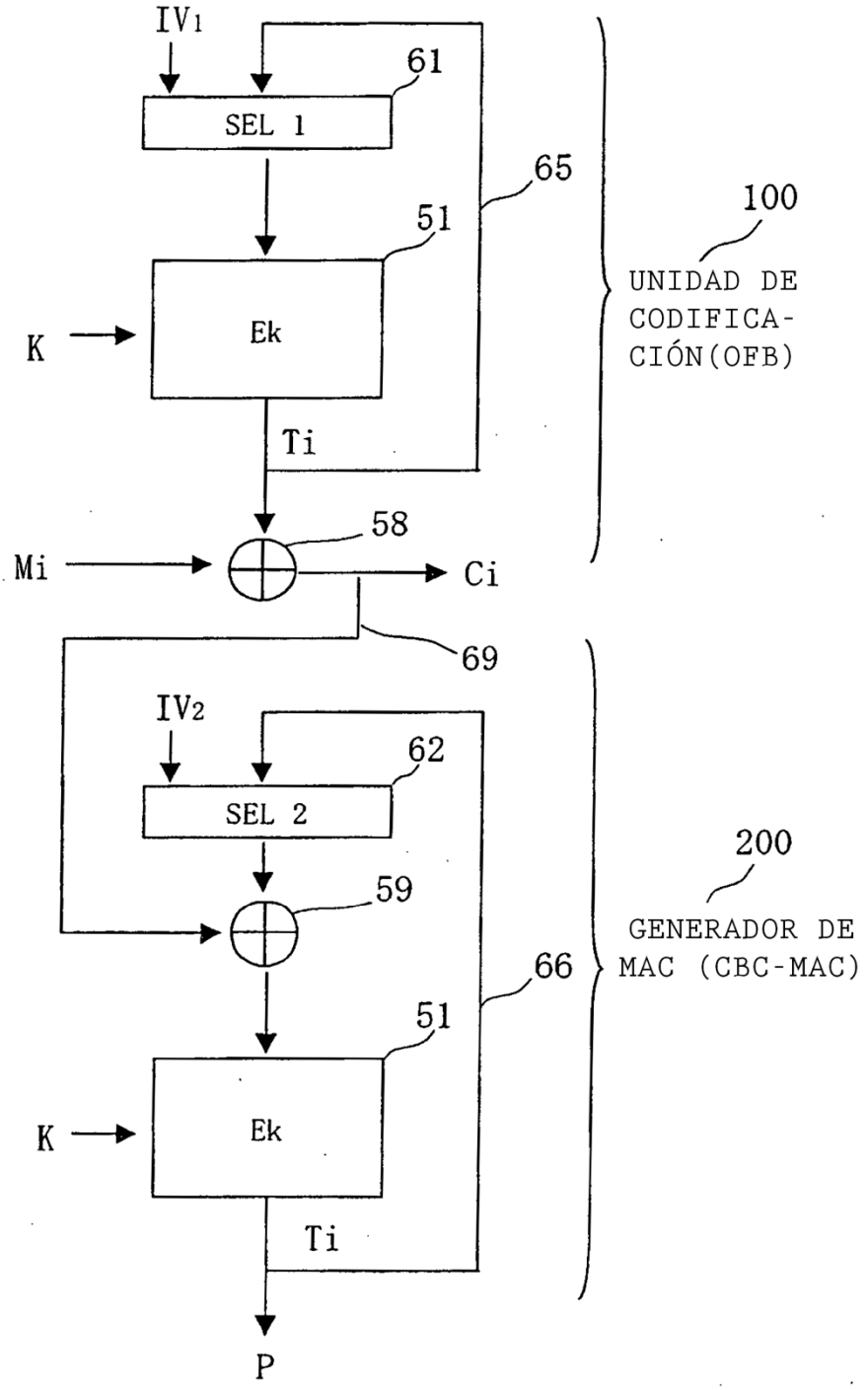
IV1
61
SEL 1
65
100
51
UNIDAD DE CODIFICA-CIÓN(OFB)
K
Ek
Ti
58
Mi
Ci
69
IV2
62
SEL 2
200
59
GENERADOR DE MAC (CBC-MAC)
66
51
K
Ek
Ti
P

Fig.30

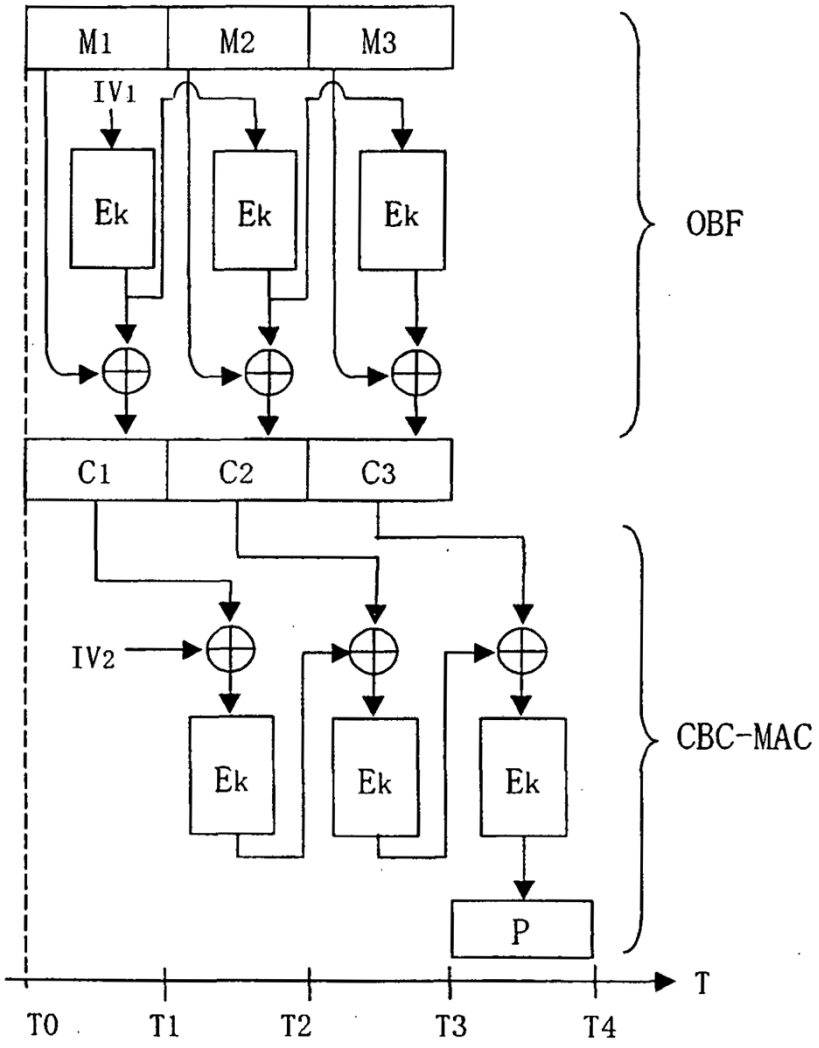
M1
M2
M3
IV1
Ek
Ek
Ek
OBF
C1
C2
C3
IV2
Ek
Ek
Ek
CBC-MAC
P
T
T0
T1
T2
T3
T4

Fig.31

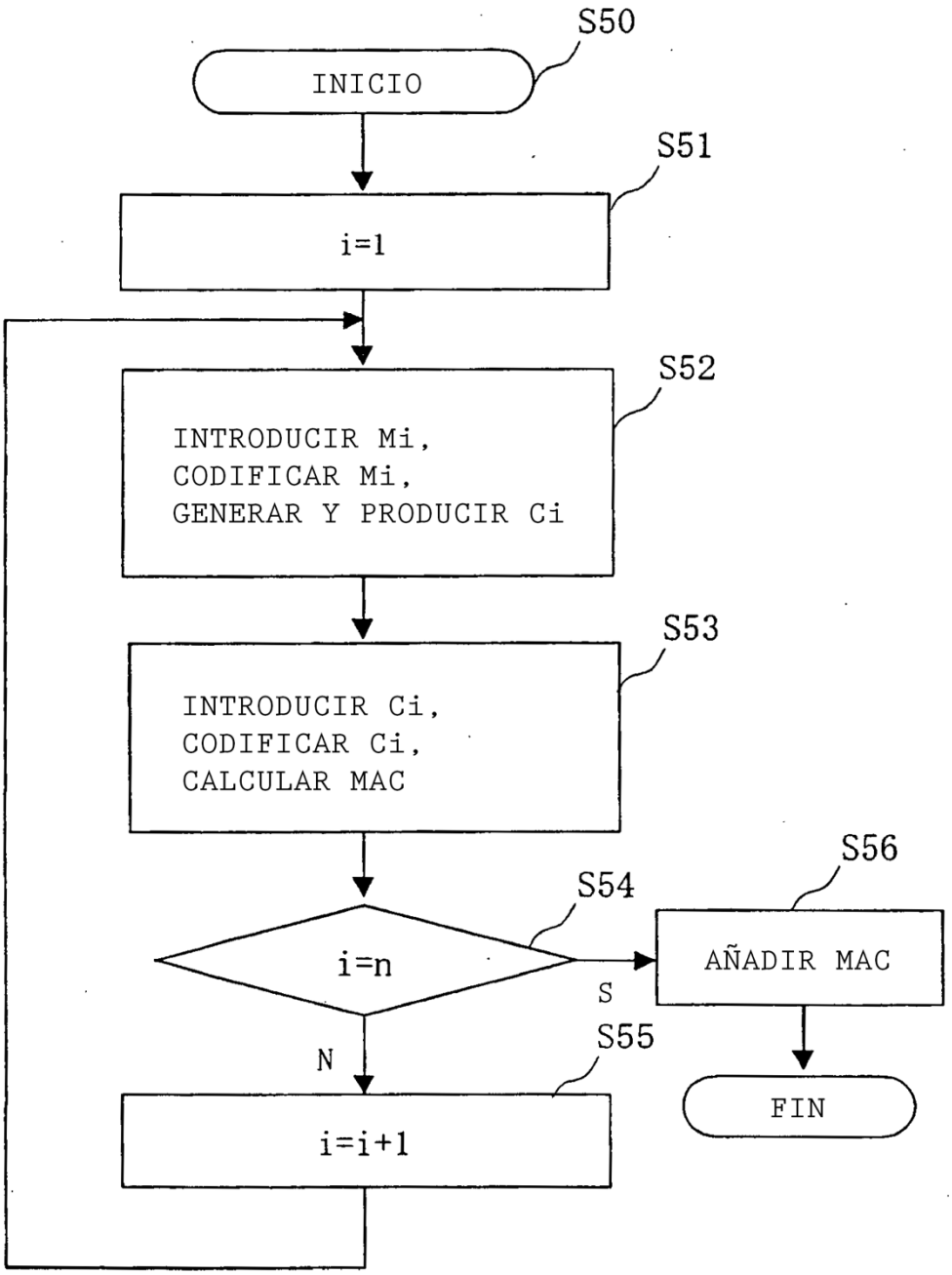
S50
INICIO
S51
i=1
S52
INTRODUCIR Mi,
CODIFICAR Mi,
GENERAR Y PRODUCIR Ci
S53
INTRODUCIR Ci,
CODIFICAR Ci,
CALCULAR MAC
S54
i=n
S
N
S55
i=i+1
S56
AÑADIR MAC
FIN

Fig. 32

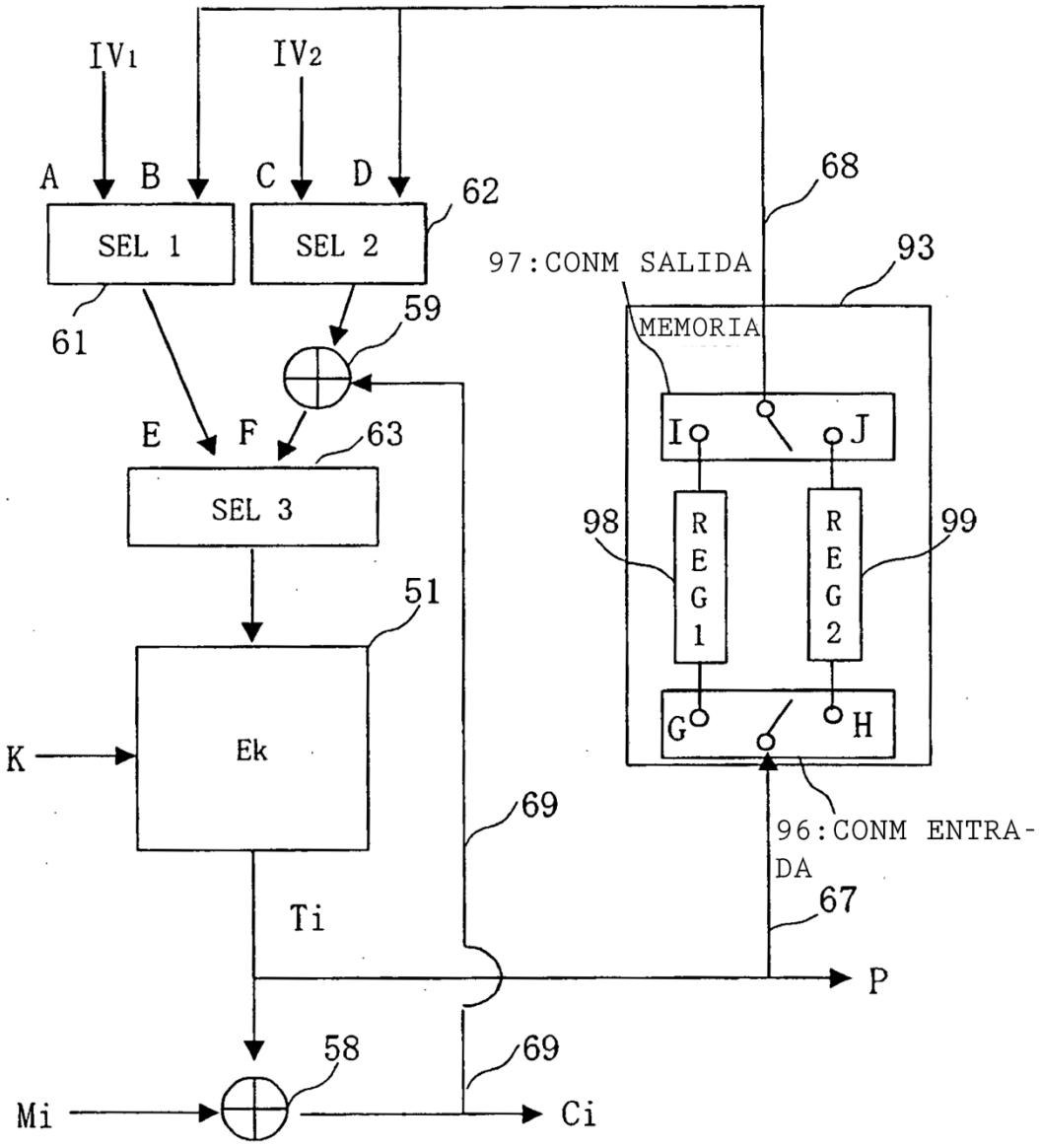
IV1
IV2
A
B
C
D
62
SEL 1
SEL 2
68
97:CONM SALIDA
93
MEMORIA
61
59
E
F
63
SEL 3
I
J
98
REG1
REG2
99
51
K
Ek
69
96:CONM ENTRADA
Ti
67
P
58
69
Mi
Ci

Fig. 33

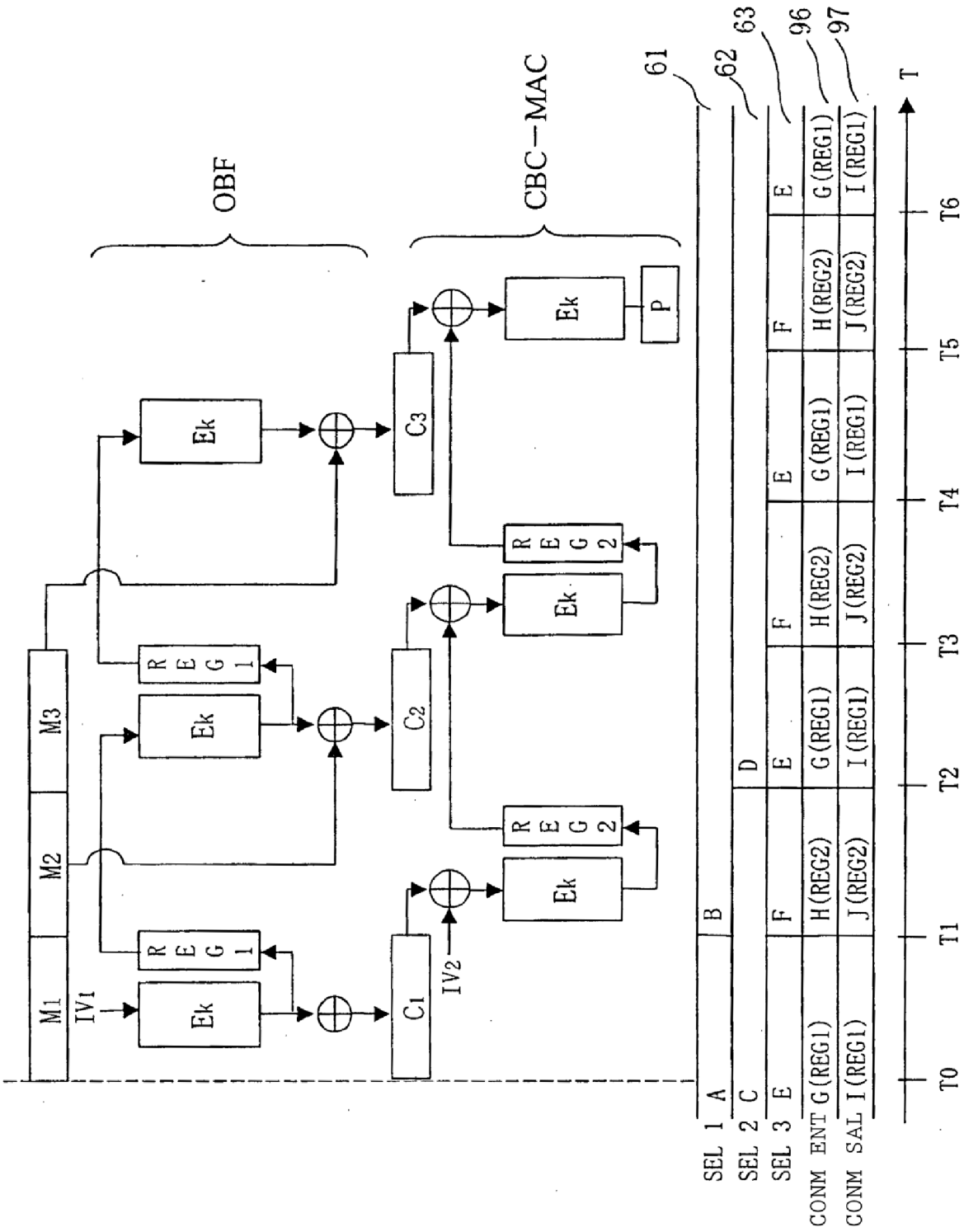
OBF
CBC−MAC
M1
M2
M3
IV1
IV2
Ek
REG1
REG2
C1
C2
C3
P
61
62
63
96
97
T
T0
T1
T2
T3
T4
T5
T6
SEL 1 A B
SEL 2 C D
SEL 3 E F E F E E
CONM ENT G(REG1) H(REG2) G(REG1) H(REG2) G(REG1) G(REG1)
CONM SAL I(REG1) J(REG2) I(REG1) J(REG2) I(REG1) I(REG1)

Fig. 34

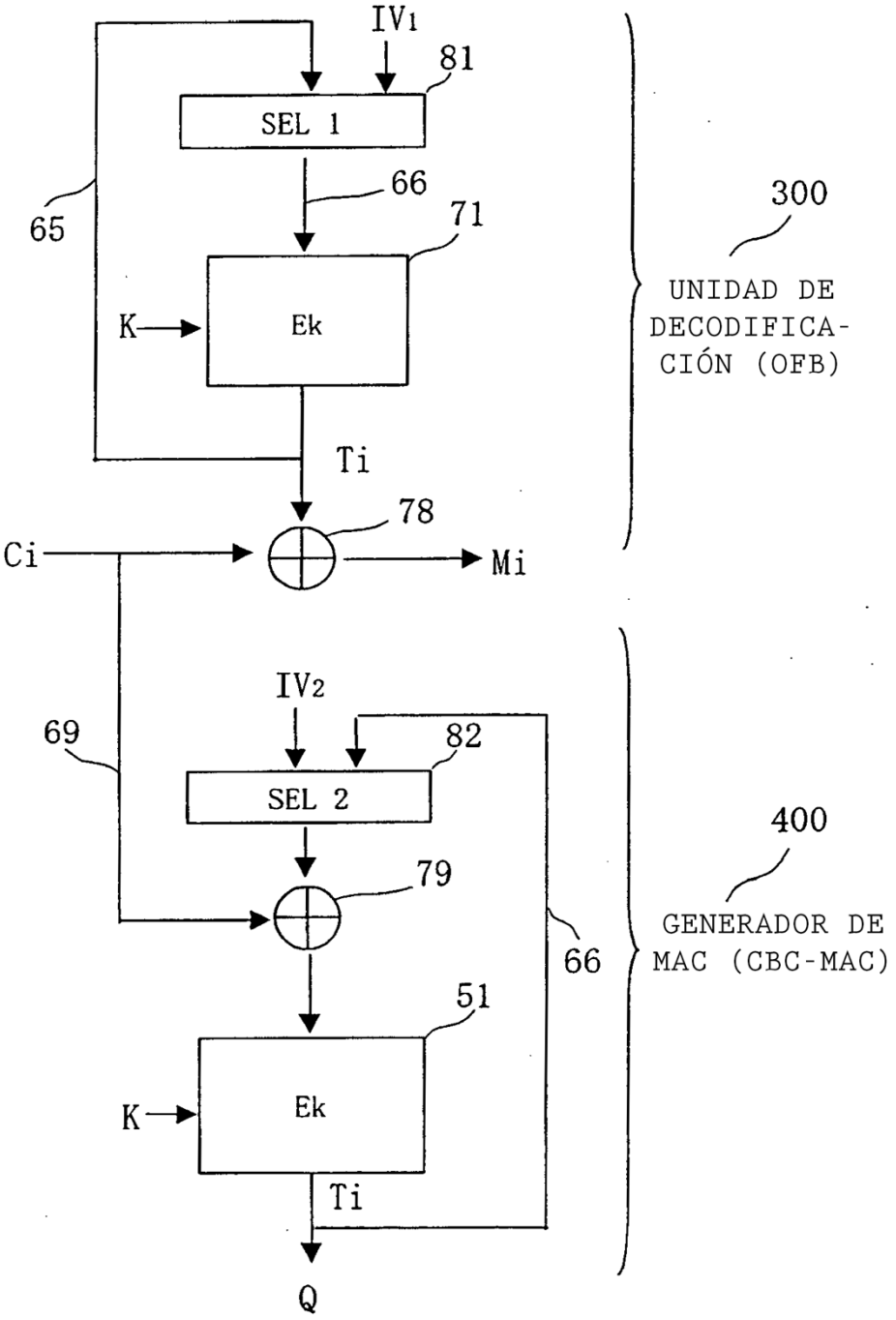
IV1
81
SEL 1
66
65
71
300
K
Ek
UNIDAD DE DECODIFICACIÓN (OFB)
Ti
78
Ci
Mi
IV2
69
82
SEL 2
400
79
66
GENERADOR DE MAC (CBC-MAC)
51
K
Ek
Ti
Q

Fig. 35

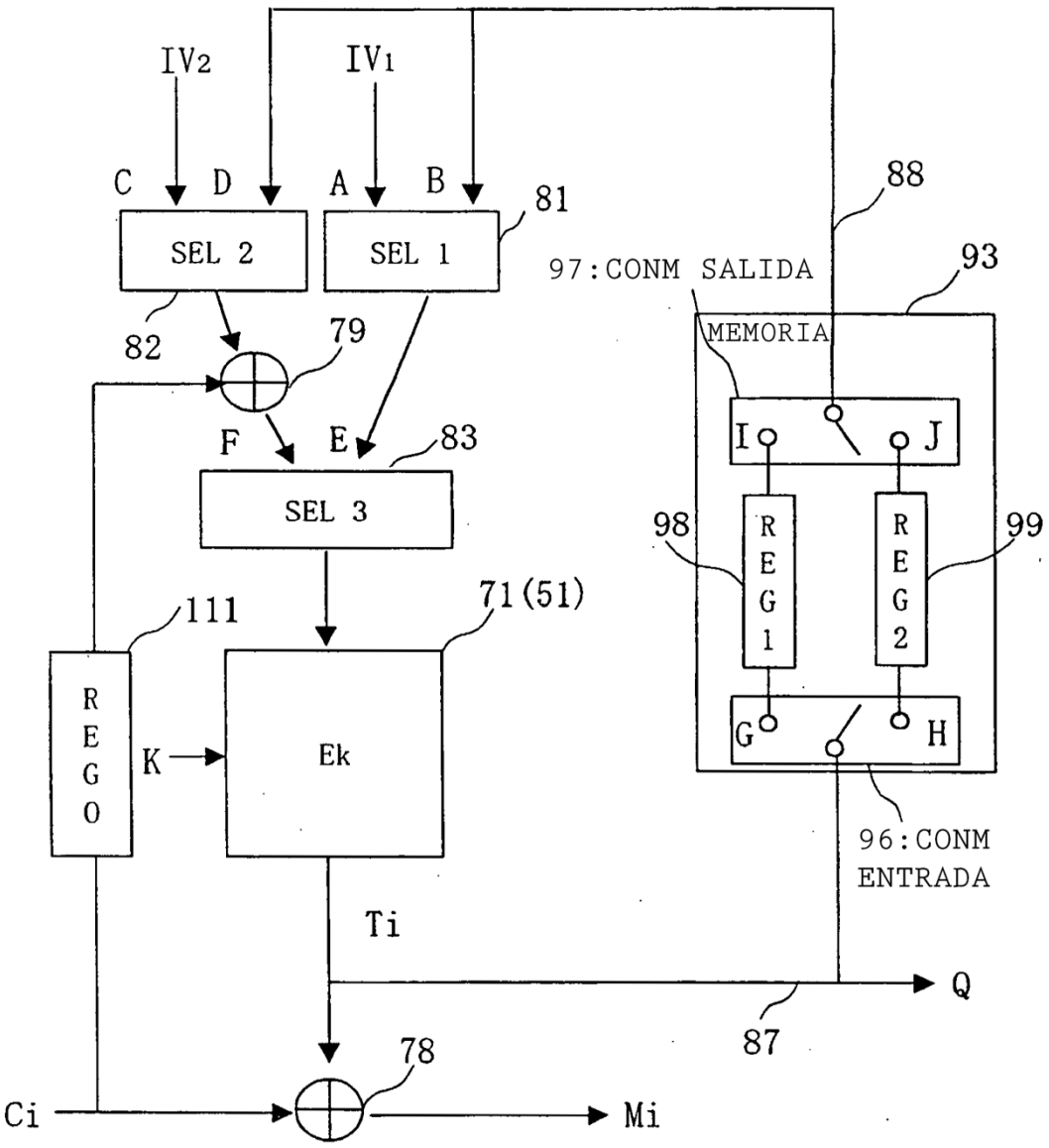
IV2
IV1
C
D
A
B
81
SEL 2
SEL 1
82
79
F
E
83
SEL 3
111
71(51)
REG0
K
Ek
Ti
Ci
78
Mi
Q
87
88
97:CONM SALIDA
93
MEMORIA
I
J
98
REG1
REG2
99
G
H
96:CONM
ENTRADA

Fig. 36

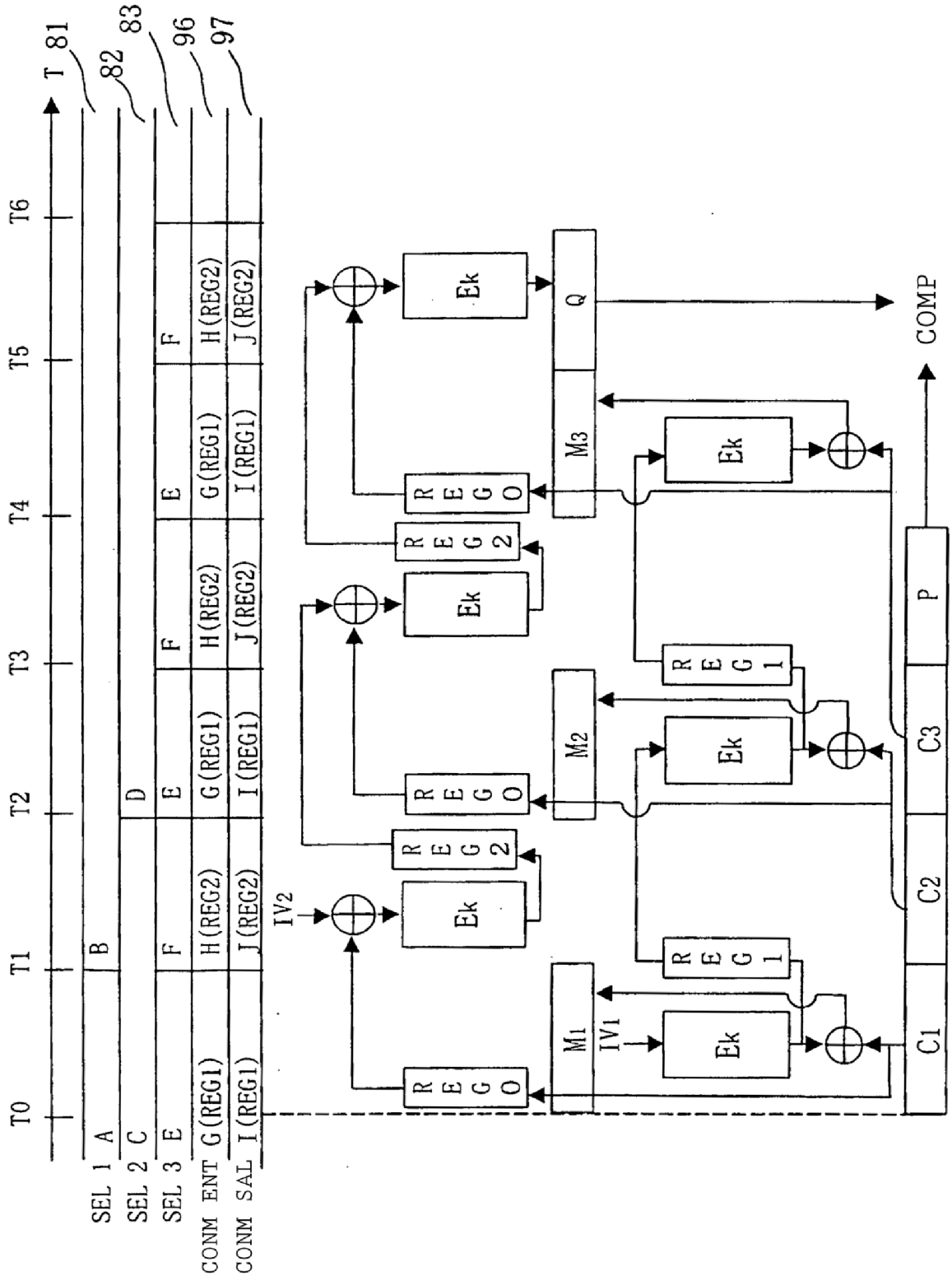
T
81
82
83
96
97
T0
T1
T2
T3
T4
T5
T6
SEL 1 A
SEL 2 C
SEL 3 E
CONM ENT G(REG1)
CONM SAL I(REG1)
B
D
E
F
H(REG2)
J(REG2)
G(REG1)
I(REG1)
IV2
IV1
M1
M2
M3
Q
P
C1
C2
C3
COMP
Ek
REG0
REG1
REG2

Fig. 37

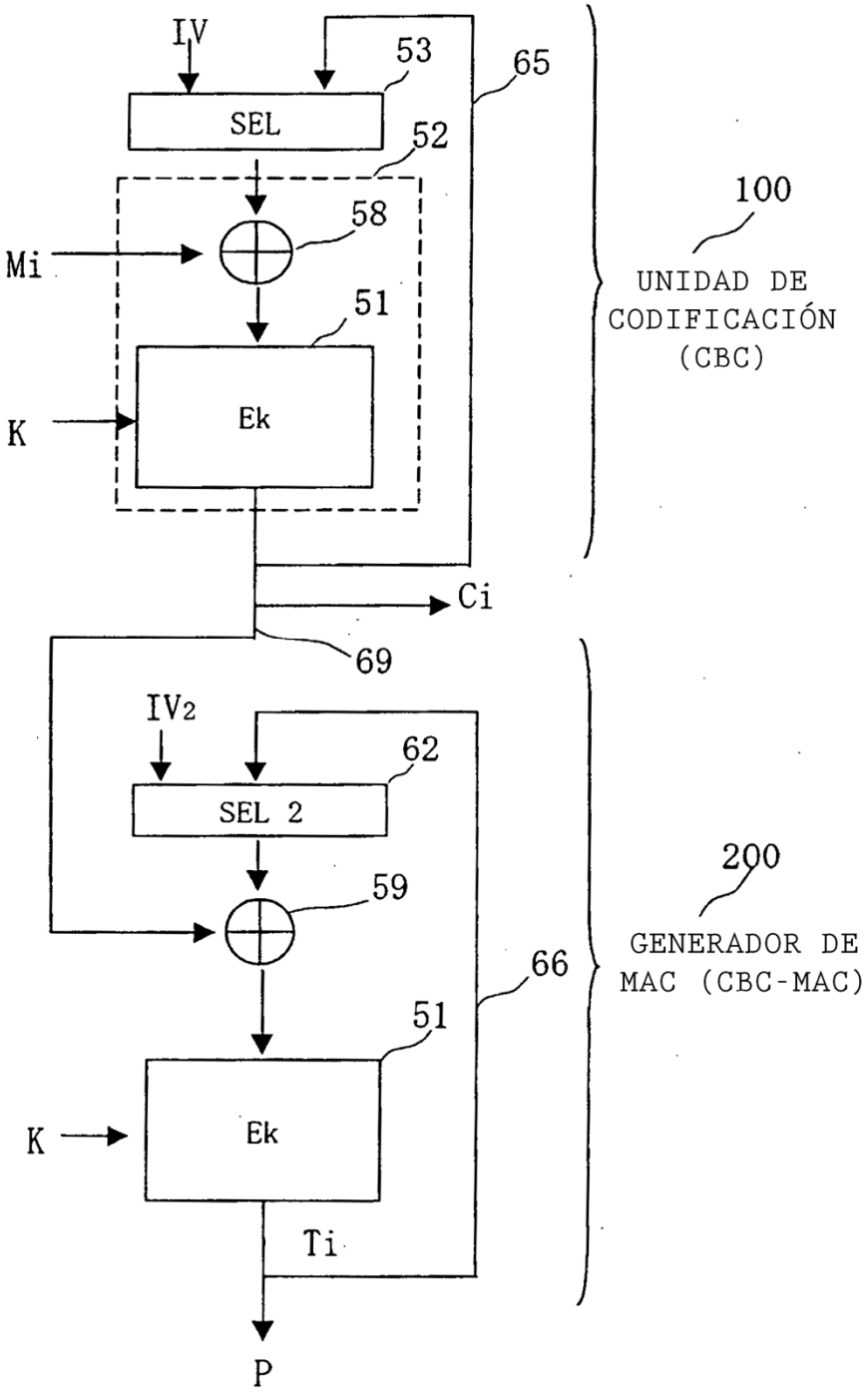
IV
53
65
SEL
52
100
58
Mi
51
UNIDAD DE CODIFICACIÓN (CBC)
Ek
K
Ci
69
IV2
62
SEL 2
200
59
GENERADOR DE MAC (CBC-MAC)
66
51
Ek
K
Ti
P

Fig. 38

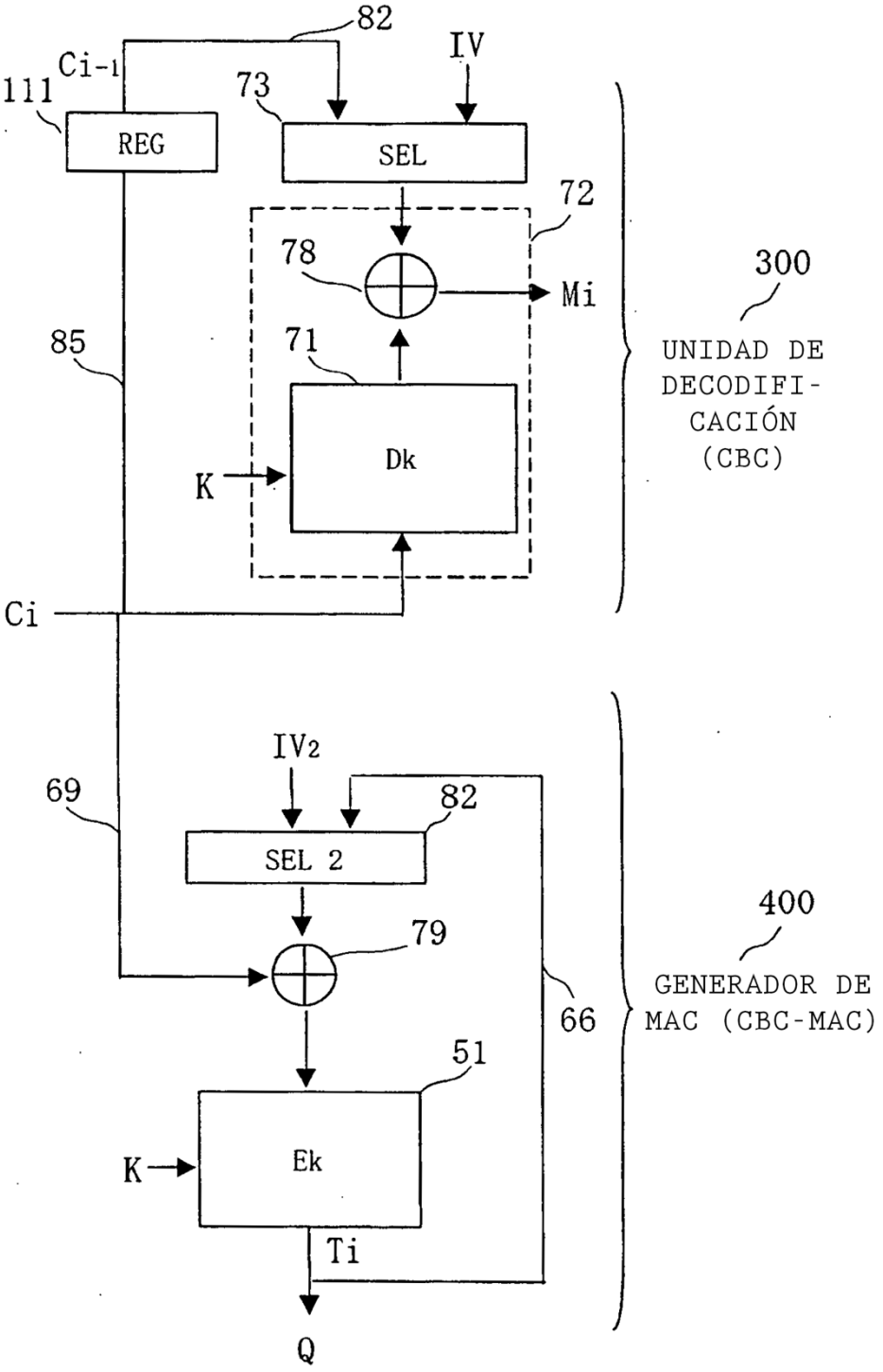
82
IV
Ci-1
111
73
REG
SEL
72
78
Mi
300
85
71
UNIDAD DE DECODIFI-CACIÓN (CBC)
Dk
K
Ci
IV2
69
82
SEL 2
400
79
GENERADOR DE MAC (CBC-MAC)
66
51
Ek
K
Ti
Q

Fig. 39

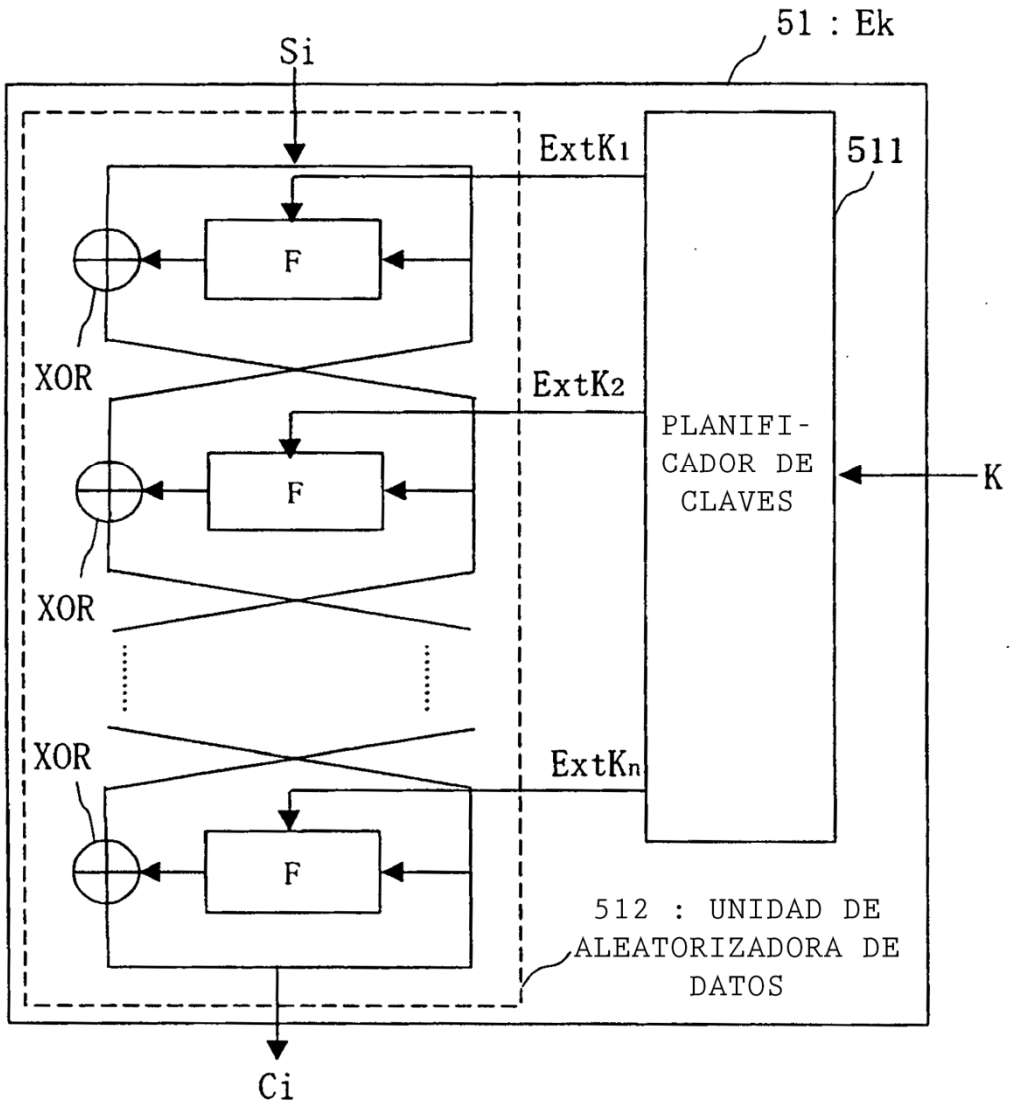
51 : Ek
Si
ExtK1
511
F
XOR
ExtK2
PLANIFI-
CADOR DE
CLAVES
K
F
XOR
XOR
ExtKn
F
512 : UNIDAD DE
ALEATORIZADORA DE
DATOS
Ci

Fig. 40

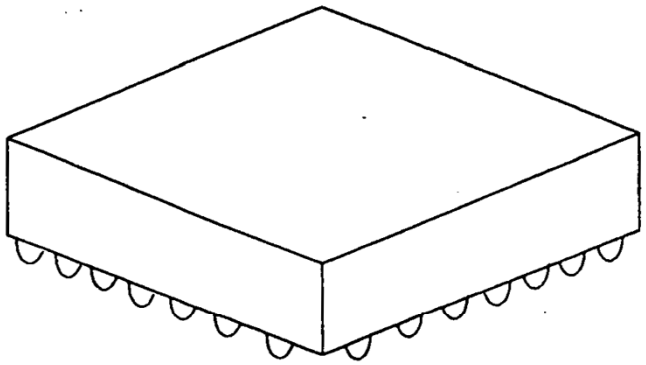

Fig. 41

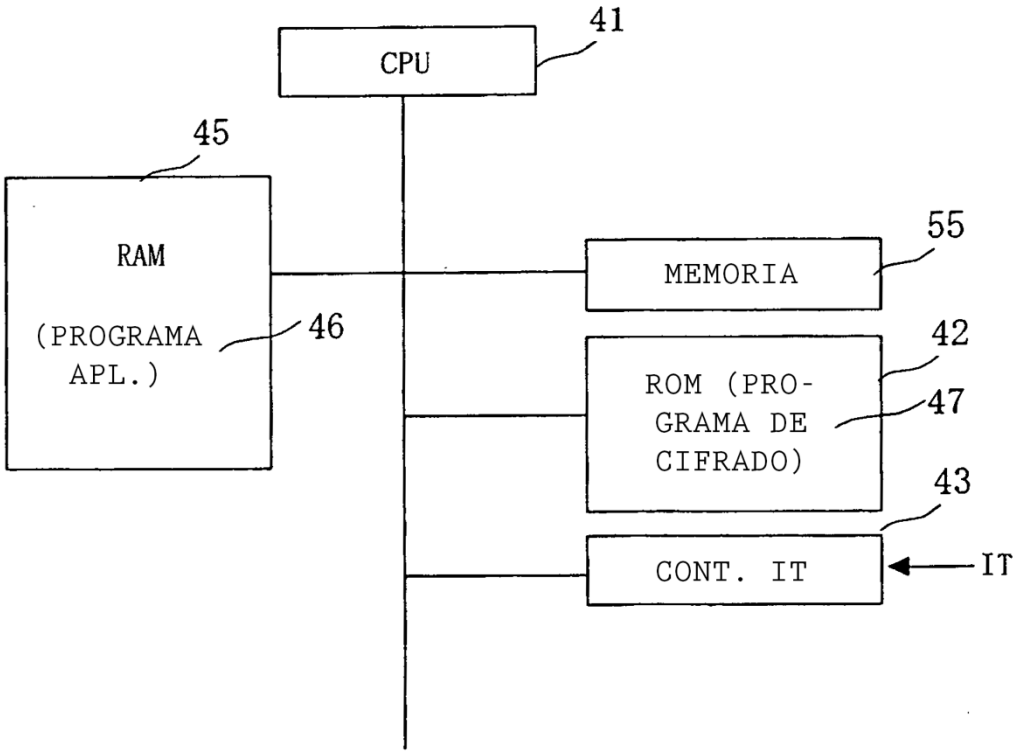
41
CPU
45
RAM
(PROGRAMA APL.)
46
55
MEMORIA
42
ROM (PRO-GRAMA DE CIFRADO)
47
43
CONT. IT
IT

Fig. 42

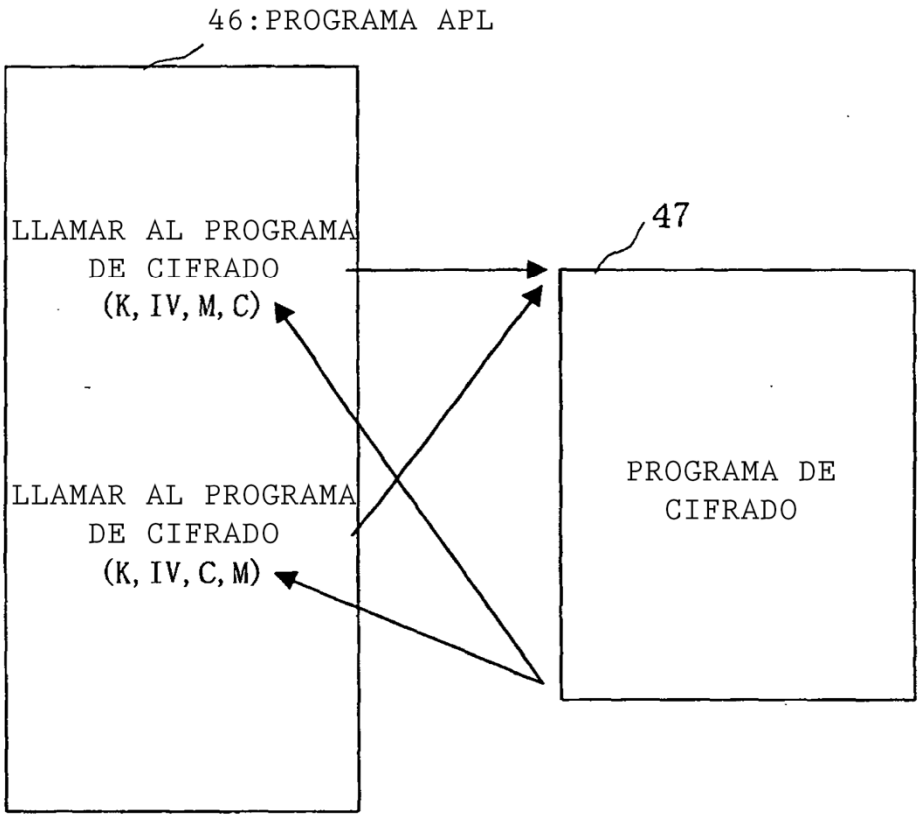
46 : PROGRAMA APL
LLAMAR AL PROGRAMA
DE CIFRADO
(K, IV, M, C)
LLAMAR AL PROGRAMA
DE CIFRADO
(K, IV, C, M)
47
PROGRAMA DE
CIFRADO

Fig. 43

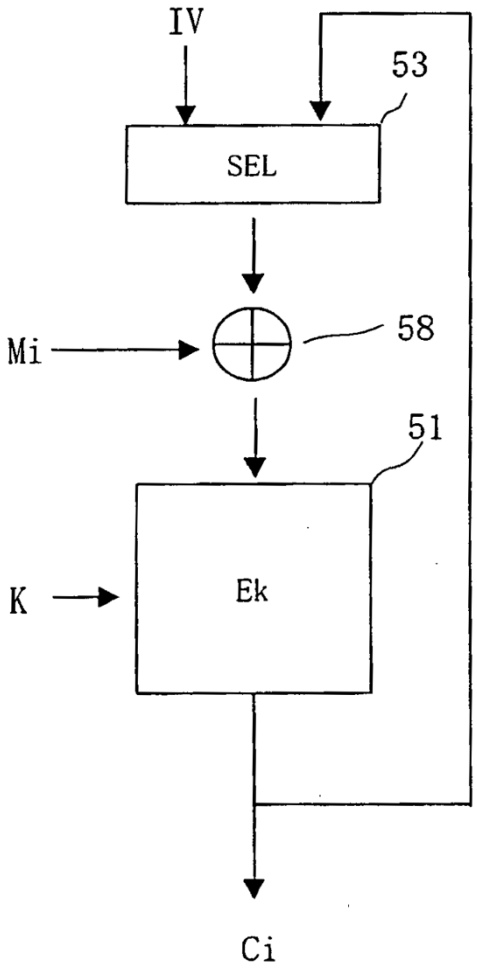

Fig.44

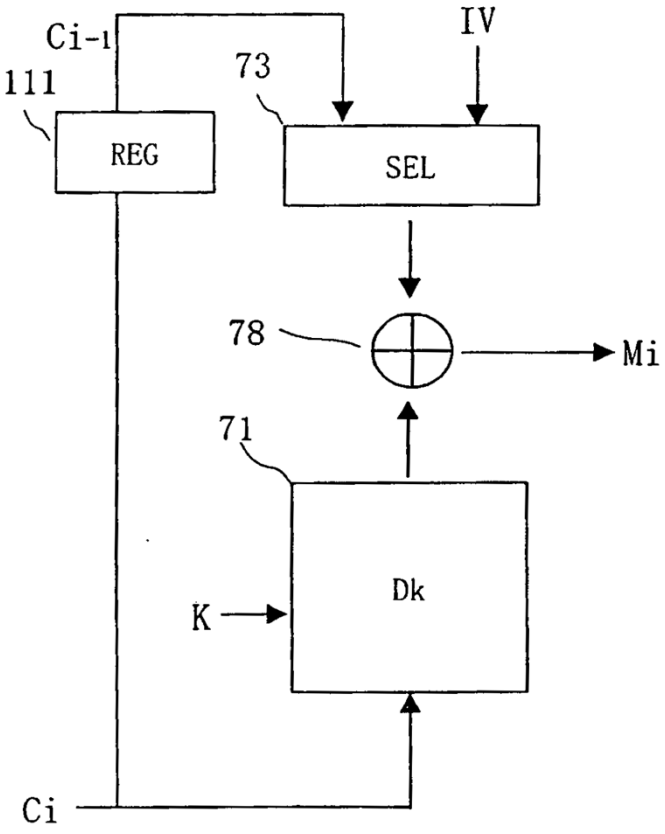

Fig. 45

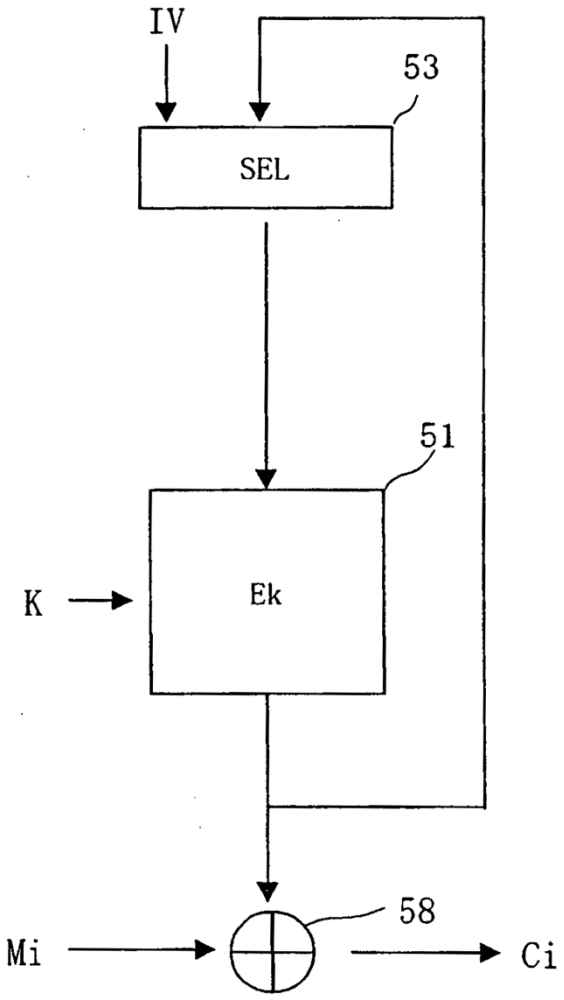

Fig. 46

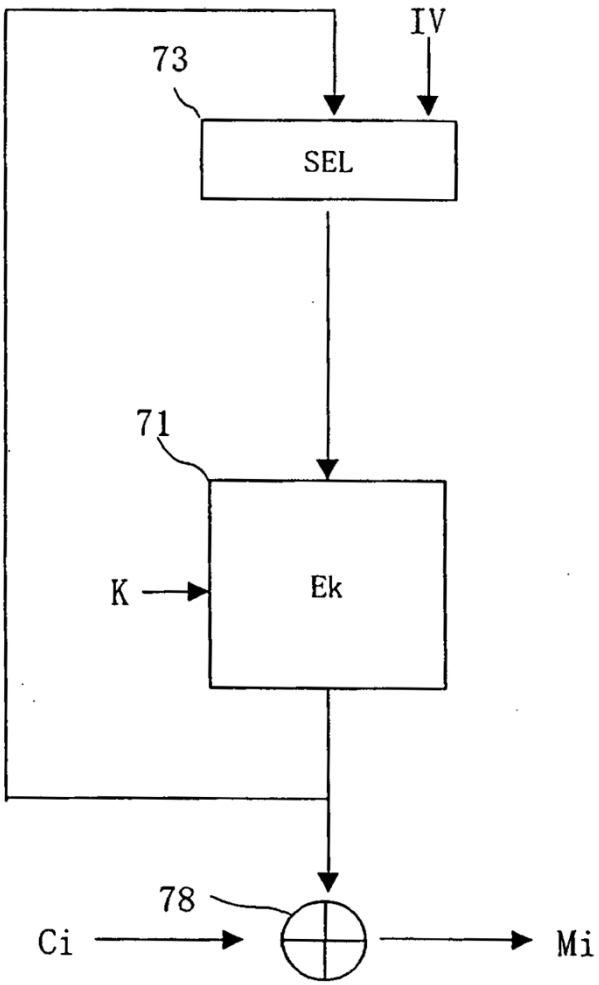

Fig.47

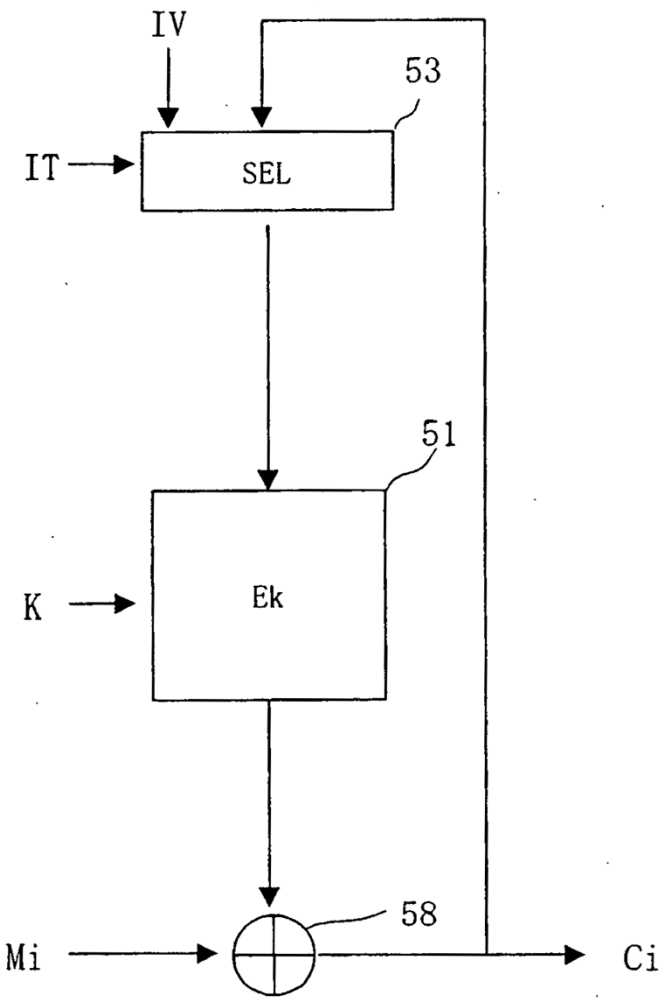

Fig. 48

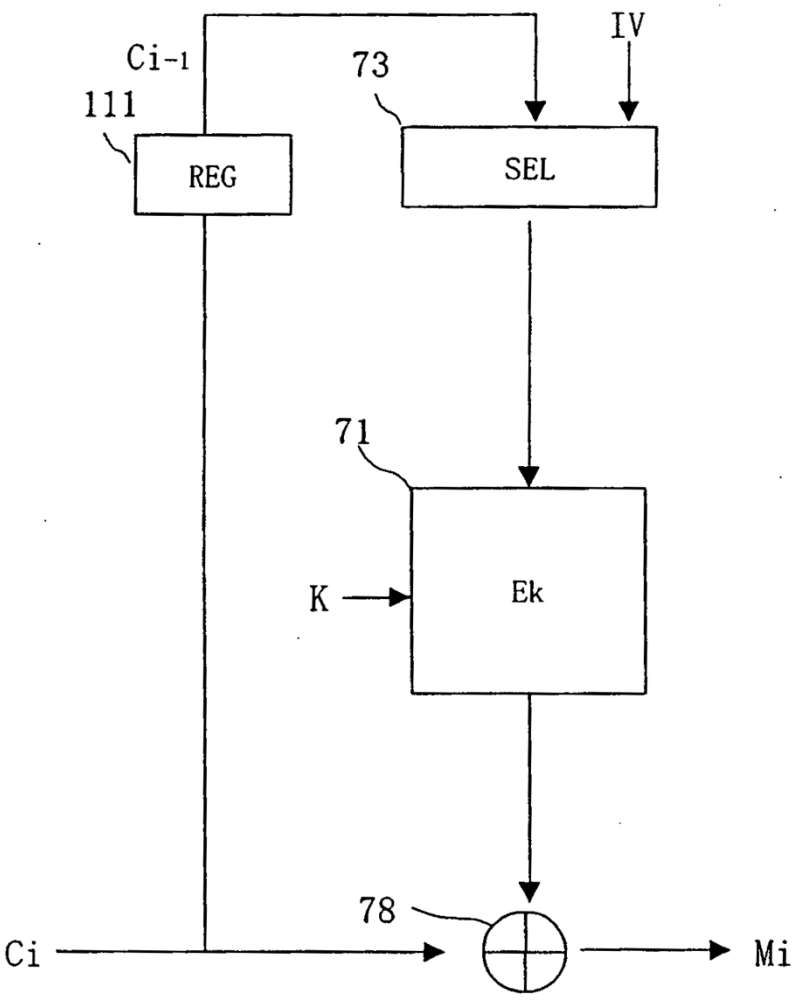

Fig. 49

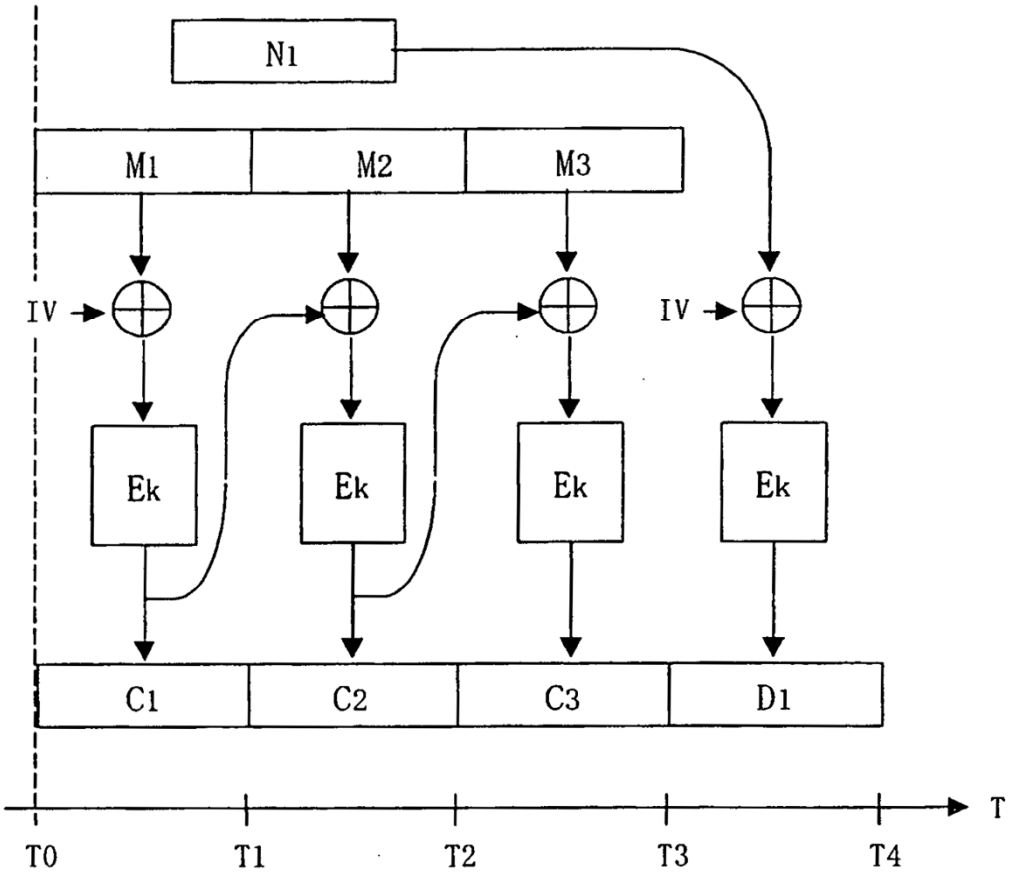

Fig. 50

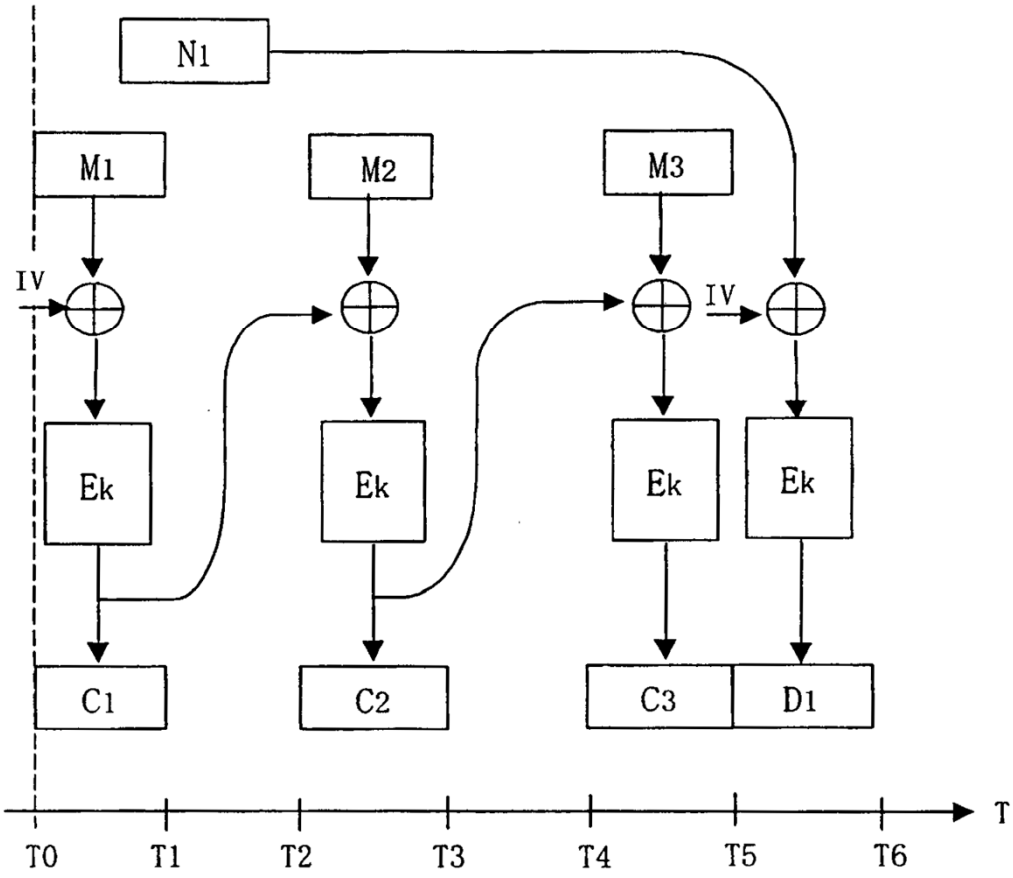

Fig.51

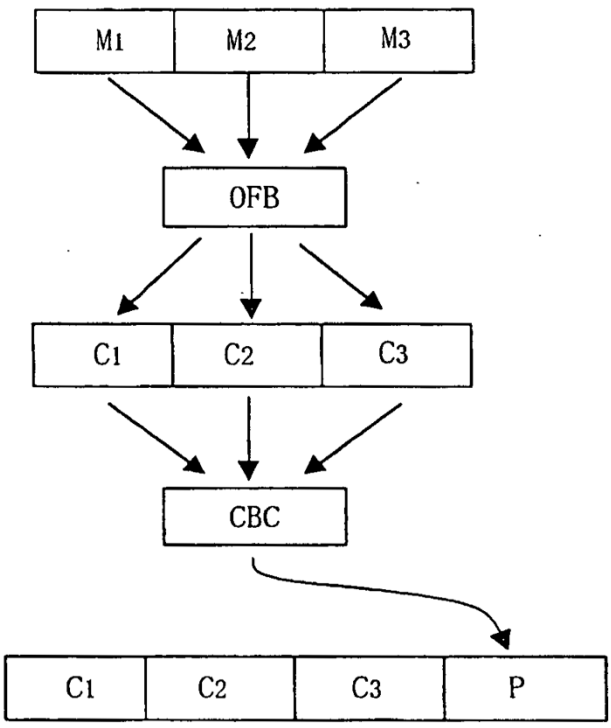

Fig. 52