(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 958 182**

(51) Int. Cl.:

G06F 21/71 (2013.01)
G06F 21/64 (2013.01)
H04L 9/32 (2006.01)
H04L 9/08 (2006.01)
G06F 21/44 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **20.05.2020 E 20175606 (1)**

(97) Fecha y número de publicación de la concesión europea: **06.09.2023 EP 3913517**

(54) Título: **Elemento seguro para procesamiento seguro de información digital**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
05.02.2024

(73) Titular/es:

NAGRAVISION SÀRL (100.0%)
Route de Genève 22-24
1033 Cheseaux-sur-Lausanne, CH

(72) Inventor/es:

GREMAUD, FABIEN;
VILLEGAS, KARINE;
HAUTIER, ROAN y
FUCHS, PASCAL

(74) Agente/Representante:

DEL VALLE VALIENTE, Sonia

DESCRIPCIÓN

Elemento seguro para procesamiento seguro de información digital

Campo técnico

La presente descripción se refiere al campo de la seguridad informática y, con mayor precisión, a procesar de manera segura información digital, tal como códigos (aplicaciones) y/o datos.

Antecedentes

Generalmente, un dispositivo de procesamiento que puede procesar de manera segura información digital, también conocido como un elemento seguro, por ejemplo una tarjeta inteligente o un chip, incluye recursos de hardware tales como uno o más procesadores (por ejemplo, un procesador huésped y un procesador seguro), y una o más memorias (memoria volátil, memoria caché y memoria no volátil). El dispositivo de procesamiento generalmente opera bajo el control de un sistema operativo y ejecuta instrucciones de programa usando uno o más componentes o aplicaciones de software. Cuando se ejecuta una aplicación, la información digital (código y datos) requerida para la aplicación y los datos producidos por la aplicación se pueden almacenar en una memoria no volátil y, cuando se requiere que se ejecuten o se procesen, la información digital puede cargarse en una memoria volátil o memoria caché.

El elemento seguro puede estar incorporado en un SoC (Sistema en Chip) que tiene varios módulos de procesamiento, varias memorias y varias funcionalidades. El SoC puede integrarse en un módulo más grande, por ejemplo, un dispositivo de IoT (dispositivo de Internet de las cosas).

En un entorno de restricción como un sistema de IoT, el elemento seguro puede tener solo una memoria no volátil pequeña, tal como una memoria OTP (una vez programable), que se usa para almacenar una cantidad limitada de información, típicamente contadores y/o teclas. El elemento seguro no se utiliza para almacenar la información digital (por ejemplo, código y/o datos). Para mantener el coste del elemento seguro bajo, la información digital se almacena en una memoria no volátil externa ubicada fuera del elemento seguro (por ejemplo, una memoria flash en el SoC o en el módulo fuera del SoC). Esta memoria externa tiene una gran capacidad de almacenamiento, por ejemplo, de varios megabytes, más altas que la capacidad de almacenamiento interna del elemento seguro. La información digital almacenada en la memoria externa estará protegida con un alto nivel de seguridad equivalente al nivel de seguridad del procesador seguro. Más precisamente, debe asegurarse en privacidad, integridad, autenticidad y renovación. Con respecto a la renovación, significa que la información digital debe protegerse contra un ataque de repetición (también conocido como ataque de reproducción), que es una forma de ataque de red en el que una transmisión de datos válida se repite o retrasa de manera deficiente o fraudulenta.

El elemento seguro debe protegerse contra ataques maliciosos, en particular frente a ataques de repetición. Un ataque de repetición (también conocido como ataque de reproducción) es un ataque malicioso en el que un hacker escucha a escondidas una transmisión de datos válida, la intercepta y luego la retrasa o la repite de manera fraudulenta para desviar a un receptor (en este caso, el elemento seguro) para que haga lo que quiere el hacker.

El elemento seguro tiene un cargador que importa de manera segura la información digital externa en una memoria interna, tal como una RAM (Memoria de Acceso Aleatorio). Cuando se importa una pieza de información digital, se verifica si ha sido sustituido o no por una versión previa de dicha pieza de información digital que ya no es válida.

El documento EP 2-860-660 describe un sistema de procesamiento de datos que incluye un SoC que tiene un procesador huésped (HCPLI) y memorias huésped asociadas, un procesador seguro (SCPLI) y memorias seguras asociadas, una memoria caché segura que almacena una pluralidad de líneas de caché y un controlador de caché que controla la carga de las líneas de caché desde el procesador huésped al procesador seguro. La memoria caché almacena una tabla de integridad que contiene una huella digital de cada línea de caché, que se ha calculado usando una función hash. Cuando se carga una línea de caché en el procesador seguro, su integridad se verifica comparando con éxito una huella digital de la línea de caché calculada por el procesador seguro y la huella digital correspondiente extraída de la tabla de integridad.

Dicha solución requiere cargar una tabla de integridad en la memoria interna (por ejemplo, una RAM) del elemento seguro. Por ejemplo, en caso de que la memoria externa almacene 1MB de información digital y la información digital esté segmentada en piezas de 1KB cada una con una huella digital o TAG de 32B, la tabla de integridad que se va a almacenar en la memoria RAM interna del elemento seguro debe tener una capacidad de almacenamiento de 32 KB. Para un dispositivo de procesamiento que tenga poca capacidad de almacenamiento, como un dispositivo de IoT, dicha solución puede no ser apropiada porque requiere demasiada capacidad de almacenamiento en el elemento seguro.

La presente descripción pretende mejorar la situación.

Resumen

La presente descripción se refiere a un elemento seguro para procesar de forma segura información digital, como se define en la reivindicación 1.

Por lo tanto, el algoritmo criptográfico usa el número de versión del segmento de información digital almacenado en la memoria externa como una entrada. El número de versión del segmento de información se usa criptográficamente para procesar previamente el segmento de la dicha información digital. Está asociado con el segmento de información digital en la memoria externa y se carga en el elemento seguro. El número de versión puede tener un valor inicial (por ejemplo, cero) e incrementarse cada vez que el segmento de la información digital se modifica, por ejemplo, durante el procesamiento por el elemento seguro. El segmento de información digital se enlaza criptográficamente a su número de versión. El elemento seguro almacena internamente este número de versión, lo que requiere una baja capacidad de memoria. Gracias a eso, el elemento seguro está protegido contra ataques de repetición maliciosos sin requerir un almacenamiento de alta capacidad. Por lo tanto, la presente descripción puede aplicarse a dispositivos de bajo extremo que necesitan un buen nivel de seguridad.

Ventajosamente, el sistema está configurado para determinar la información de identificación del segmento de información digital, la determinación de dicha información de identificación incluyendo además obtener un identificador de segmento de dicho segmento de información digital, y para ejecutar el algoritmo criptográfico usando dicha información de identificación, siendo dicho identificador de segmento un índice asignado al segmento correspondiente de información digital en la memoria externa o un identificador presente en un encabezamiento del segmento correspondiente de información digital. En otras palabras, el algoritmo criptográfico usa el número de versión y el identificador del segmento de la información digital como entradas.

El identificador del segmento de información digital se puede usar criptográficamente, además de su número de versión, para pre-procesar la información digital.

El identificador es una pieza de información corta, por ejemplo, un índice de la pieza de información digital en la memoria externa. Típicamente, la información digital se divide en segmentos de información digital, que están indexadas y almacenadas en la memoria externa. Los segmentos de información digital pueden apilarse o disponerse más generalmente en posiciones dadas en la memoria externa. Los índices que se asignan a los segmentos de información digital pueden depender de las posiciones de estos segmentos de información digital en la memoria externa.

Ventajosamente, dicho elemento seguro almacena al menos una clave criptográfica única, que se generó de manera única para el elemento seguro, para ser utilizada por el sistema en ejecución del algoritmo criptográfico.

El uso del número de versión evita una sustitución del segmento de información digital por una versión anterior de la información digital. El uso del identificador del segmento de la información digital evita una sustitución de la información digital por otra información digital válida cargada en el elemento seguro. El uso de una clave única evita una sustitución de la información digital por una información digital de otro elemento o dispositivo seguro. El uso combinado del número de versión, el identificador y la clave única permite tener una solución equivalente a la solución de TAG de la técnica anterior, que protege implícitamente de muchos ataques diferentes, pero sin requerir una alta capacidad de almacenamiento.

El sistema puede comprender una función de derivación de clave para generar al menos una clave criptográfica, para ser utilizada por el algoritmo criptográfico, usando la al menos una clave criptográfica única y la información de identificación de la información digital como entradas.

El algoritmo criptográfico puede comprender un algoritmo de autenticación para autenticar la información digital que se ha cargado. Alternativamente, o adicionalmente, el algoritmo criptográfico puede comprender un algoritmo de descifrado para descifrar la información digital que se ha cargado. Por lo tanto, el número de versión y/o el identificador del segmento de la información digital se puede usar no solo para la autenticación sino también para el descifrado de la información digital, lo que aumenta la seguridad.

Por ejemplo, el sistema está configurado para generar un vector de inicialización, para ser utilizado por el algoritmo de descifrado, usando la información de identificación del segmento de la información digital como una entrada.

La información digital almacenada en la memoria externa puede protegerse mediante un mecanismo de cifrado autenticado, por ejemplo, basado en un enfoque "Cifrado-entonces-MAC" *(MAC = Message Authentication Code - Código de Autenticación de Mensaje).* En ese caso, se requiere autenticarse y luego descifrar la información digital, antes de procesar de manera segura la información digital en el elemento seguro. Con tal configuración, el uso del número de versión y el identificador del segmento de la información digital como entradas para su autenticación proporciona el nivel más alto de seguridad. De hecho, el segmento de la información digital no se descifrará si esta autenticación no tiene éxito.

Los índices de los segmentos de información digital en la memoria externa se pueden usar para indexar los números de versión de los segmentos de información digital correspondientes en la tabla de versión.

Ventajosamente, el sistema está configurado además para verificar una información de renovación de dicha tabla de versión comparando dicha información de renovación de la tabla de versión con un contador de renovación almacenado en una memoria no volátil interna del elemento seguro.

La renovación o la coherencia de todas los segmentos de información digital se autentica por la tabla de versión. El uso de los números de versión de los segmentos de información digital, en lugar de sus TAG (huellas digitales), permite intercambiar recursos de almacenamiento internos en el elemento seguro. Por ejemplo, en el caso de una memoria externa que almacena 1MB de información digital almacenada en piezas o fragmentos de 1KB cada uno con un TAG como un MAC de 32B, el tamaño de la tabla de integridad que contiene los TAG de todos los segmentos de información digital sería de 32 KB. En la presente descripción, si se supone que el número de versión tiene un tamaño de unos pocos bytes, por ejemplo 3B, el tamaño de la tabla de versión que necesita almacenarse en el elemento seguro es solo de 3 KB. Por lo tanto, la presente descripción permite intercambiar 29KB de capacidad de almacenamiento interno.

La verificación de la información de renovación de la tabla de versión garantiza la seguridad construyendo una cadena de confianza para renovación (anti-repetición).

La memoria no volátil interna puede ser una memoria de OTP (One Time Programable).

El sistema puede configurarse para cargar la tabla de versión en un procedimiento de inicio del elemento seguro.

Ventajosamente, el sistema está configurado para autenticar y descifrar la tabla de versión que se ha cargado.

El sistema puede configurarse para leer la información de renovación de un encabezamiento de la tabla de versión.

Ventajosamente, el elemento seguro comprende un módulo de actualización configurado, en caso de que el segmento de la información digital se modifique cuando sea procesado por el procesador seguro, para controlar una operación de escritura de la información digital modificada y una tabla de versión actualizada en la memoria externa, e incrementar el contador de renovación en la memoria interna.

Un segundo aspecto de la descripción se refiere a un sistema que incluye el elemento seguro como se ha definido anteriormente y una memoria externa para almacenar la información digital.

Un tercer aspecto de la presente descripción se refiere a un método para procesar de manera segura información digital mediante un elemento seguro, como se define en la reivindicación 13.

Breve descripción de los dibujos

Otras características, propósitos y ventajas de la descripción se harán más explícitos por medio de la lectura de la declaración detallada de las realizaciones no restrictivas hechas con referencia a los dibujos adjuntos.

La Figura 1 muestra un elemento seguro incorporado en un SoC (Sistema en Chip), según una primera realización.

La Figura 2 muestra el elemento seguro y una memoria externa que opera durante un procedimiento de arranque, de manera simplificada, según la primera realización.

La Figura 3 muestra el elemento seguro y la memoria externa que opera durante un procedimiento de carga y pre-procesamiento (o lectura) de la información digital de la memoria externa, de manera simplificada, según la primera realización.

La Figura 4 muestra el elemento seguro y la memoria externa que opera durante un procedimiento de escritura de una pieza modificada de información digital y una tabla de versión actualizada en la memoria externa, de manera simplificada, según la primera realización.

La Figura 5 muestra un diagrama de flujo de un procedimiento para procesar de forma segura información digital, realizado por el elemento seguro, estando almacenada la información digital en la memoria externa, según la primera realización.

La Figura 6 muestra un diagrama de flujo de un procedimiento de actualización ejecutado cuando una pieza de información digital ha sido modificado por el elemento seguro, según la primera realización.

La Figura 7 representa esquemáticamente el elemento seguro según la primera realización.

Descripción detallada

La Figura 1 muestra un sistema que incluye un elemento seguro 100 y una memoria externa 200 (es decir, una memoria externa al elemento seguro 100), según una primera realización. El elemento seguro 100 es, por ejemplo, una tarjeta o un chip inteligente. Puede incorporarse en un SoC (Sistema en Chip) 300 que tiene varias unidades de procesamiento, varias memorias y varias funcionalidades (no representadas). El SoC 300 puede integrarse en un módulo más grande 400. La memoria externa 200 es externa al elemento seguro 100. Puede construirse sobre un sustrato de silicio diferente de un procesador seguro del elemento seguro 100. Por ejemplo, la memoria externa está dispuesta en el módulo 400, fuera del SoC 300. Sin embargo, la memoria externa 200 podría estar dispuesta en el SoC 300. Por ejemplo, este módulo más grande 400 podría ser un dispositivo de IoT (típicamente provisto de un identificador único y la capacidad de transferir datos a través de una red), un aparato de telecomunicaciones, un sistema de ubicación, un vehículo como un automóvil o un avión, etc. Los ejemplos de dispositivos de IoT incluyen medidores inteligentes, cámaras inteligentes, sensores, rastreadores, etiquetas, detectores, monitores, artículos y prendas portátiles, dispositivos de hogar inteligente, dispositivos médicos y sanitarios, dispositivos de ciencia de vida, decodificadores, y dispositivos de borde en redes de telecomunicaciones tales como redes 5G.

A continuación se proporcionan diferentes casos de uso ilustrativos (no limitativos) del elemento seguro 100.

En un primer caso de uso, el elemento seguro 100 puede integrarse en un módem de un sistema o aparato de telecomunicaciones. En tal caso, el elemento seguro 100 puede manejar aplicaciones seguras de autenticación y descarga de red.

En un segundo caso de uso, el elemento seguro 100 puede integrarse en un tacógrafo y manejar de forma segura datos de ubicación.

En un tercer caso de uso, el elemento seguro 100 puede integrarse en un vehículo, por ejemplo, un automóvil o un avión, para asegurar y gestionar el transporte de datos de seguridad.

La memoria externa 200 almacena información digital. Los términos "información digital" designan datos propensos a ser cargados en el elemento seguro 100, tal como código ejecutable o información generada por código ejecutable o utilizada por código ejecutable, o cualquier otro dato para ser utilizado o procesado por el elemento seguro 100.

El elemento seguro 100 está destinado a cargar, procesar previamente (o leer) y procesar de forma segura la información digital almacenada en la memoria externa 200, como se explica más adelante en la descripción.

La memoria externa 200 puede ser una memoria no volátil.

En la presente realización, la segmentación de memoria se usa en la memoria externa 200. Significa que la información digital se segmenta y se almacena en segmentos (también llamados "fragmentos"). Un segmento de información digital es un pieza de información digital resultante de una segmentación. La segmentación se utiliza para almacenar, transferir y procesar previamente la información digital.

Por ejemplo, la memoria externa 200 almacena M segmentos de información digital denominados 'Si', con $1 \leq i \leq M$. Los M segmentos Si pueden incluir segmentos M1 de código y segmentos M2 de datos, con $M1 \geq 0$ y $M2 \geq 0$. Los segmentos de información digital pueden tener el mismo tamaño o tener tamaños respectivos que pueden ser diferentes, dependiendo de la implementación. En algunas realizaciones, el elemento seguro 100 tiene una implementación en caché. En ese caso, el elemento seguro 100 puede tener un controlador de caché (típicamente un componente de hardware de memoria caché) configurado para cargar y almacenar líneas de caché (correspondientes a segmentos de información digital) de un tamaño predeterminado. En otras realizaciones, los segmentos de información digital podrían ser piezas de información digital solicitadas por el procesador seguro, cuando se ejecuta un componente de software o una aplicación. En tal caso, los segmentos podrían tener diferentes tamaños respectivos.

Los segmentos de información digital Si están protegidos por un algoritmo de cifrado autenticado para asegurar la seguridad durante el almacenamiento en la memoria externa 200 y durante la transferencia desde la memoria externa 200 al elemento seguro 100. En la primera realización, el cifrado autenticado usado para proteger la información digital se basa en el enfoque bien conocido de "Cifrado-entonces-MAC" (EtM). El enfoque EtM se considera un enfoque muy robusto para el cifrado autenticado. Para cada segmento de información digital Si, la memoria externa 200 almacena el elemento [Si] |MAC[Si], que contiene el segmento Si cifrado con una clave de cifrado (descrita más adelante) y concatenada con el elemento de autenticación MAC[Si] del segmento cifrado [Si] calculado con una clave de autenticación (también descrita más adelante). El elemento de autenticación MAC[Si] puede calcularse mediante una función MAC.

En la presente descripción, los corchetes '[ ]' representan la forma cifrada de un elemento y el símbolo '|' representa la concatenación de dos elementos.

Un número de versión Vi se atribuye a cada segmento de información digital Si y se incrementa cuando este segmento de información digital Si es modificado por el elemento seguro 100. El tamaño del número de versión Vi de un segmento de información digital Si puede ser pequeño, generalmente unos pocos bytes, por ejemplo 3 bytes.

Inicialmente, cuando la información digital se almacena en la memoria externa 200, los números de versión Vi de todos los segmentos Si de información digital con $1 \leq i \leq M$ se pueden establecer en un valor inicial, por ejemplo, cero (pero podría ser uno o cualquier otro valor). A continuación, cada vez que se modifica un segmento específico (aquí denominado 'Sj') entre los segmentos M, Si, su número de versión Vj se incrementa, por ejemplo, en uno (es decir, Vj=Vj+1).

Además, en la primera realización, cada segmento de información digital tiene un identificador de segmento asociado denominado IDSi. Este identificador de segmento IDSi es una pieza de información corta que identifica el segmento. El identificador de segmento IDSi puede ser un índice del segmento Si en la memoria externa 200. Típicamente, los índices se asignan a los segmentos de información digital Si en la memoria externa 200, por ejemplo, dependiendo de las posiciones respectivas de los segmentos Si almacenados en la memoria externa 200. Estos índices pueden usarse como identificadores de segmento IDSi por el elemento seguro 100. También se pueden usar para indexar los números de versión Vi de los segmentos Si en la tabla de versión. En el caso de una implementación de caché en el elemento seguro 100, las direcciones de bloque de memoria caché se dan por direcciones para extraer código o datos de carga y pueden usarse como identificadores de segmento (o de bloque) IDSi. Estas direcciones de bloque de memoria caché corresponden a los índices de los bloques (segmentos) en la memoria externa 200. Las mismas direcciones de bloques de memoria caché se pueden usar para indexar los números de versión en la tabla de versión.

En la presente realización, el número de versión Vi y el identificador IDSi de cada segmento de información digital Si almacenada en la memoria externa 200 se utilizan en el algoritmo de encriptación autenticado como entradas. Por ejemplo, el algoritmo de autenticación, tal como un algoritmo MAC, usado para autenticar el segmento Si, usa una clave de autenticación k1_Si que se deriva de una clave maestra k1 (o clave de origen k1) mediante una función de derivación de clave (KDF) que toma el número de versión Vi y el identificador IDSi del segmento Si como entradas, como se expresa a continuación:

$$k1\_Si = KDF(k1, Vi, ID_{Si})$$

Además, para aumentar la seguridad, el número de versión Vi y el identificador IDSi del segmento Si puede usarse también para cifrar el segmento de información digital Si. Por ejemplo, el algoritmo de cifrado usa una clave de cifrado k2_Si que se deriva de una clave maestra k2 (o clave de origen k2) mediante una función de derivación de clave (KDF) que toma el número de versión Vi y el identificador IDSi del segmento Si como entradas:

$$k2\_Si = KDF(k2, Vi, ID_{Si})$$

Las claves k1 y k2 pueden ser claves secretas. Ventajosamente, estas claves k1 y k2 se generan de manera única para el elemento seguro 100. Pueden generarse por el propio elemento seguro 100. Las claves k1 y k2 son diferentes en la primera realización. Sin embargo, podrían ser la misma clave en otras realizaciones.

Todos los segmentos de MAC cifrados y autenticados [Si] |MAC[Si] (con $1 \leq i \leq M$) se almacenan en la memoria externa 200. Pueden apilarse (y ordenarse) en un área de memoria de la memoria externa 200.

La memoria externa 200 almacena además una tabla de versión, denominada "CtrlObj" en las Figuras 2-4. La tabla de versión CtrlObj contiene los respectivos números de versión de corriente (válidos) Vi de todos los segmentos de información digital Si (con $1 \leq i \leq M$) almacenado en la memoria externa 200. Los números de versión Vi contenidos en la tabla de versión CtrlObj se pueden disponer en el mismo orden que los segmentos Si están dispuestos en la memoria externa 200. Más generalmente, los números de versión Vi en la tabla de versión CtrlOb pueden indexarse de la misma manera que los segmentos Si correspondientes están indexados en la memoria externa 200. Esto permite identificar los segmentos Si por sus números o índices de ordenación. Alternativamente, los segmentos de información digital podrían tener identificadores de segmento en un segmento. Estos identificadores de segmento presentes en los encabezamientos de los segmentos podrían asociarse con los números de versión Vi en la tabla de versión CtrlObj.

Además, una información de renovación se atribuye a la tabla de versión CtrlObj. Típicamente, esta información de renovación es un número de versión de la tabla de versión, denominada 'CtrlObj_Version'. Esta información de renovación tiene un valor inicial, por ejemplo cero, que se atribuye a la tabla de versión CtrlObj cuando la información digital se almacena inicialmente en segmentos en la memoria externa 200. Se puede incrementar cada vez que se modifica el contenido de la tabla de versión CtrlObj, como se explicará más adelante. La información de renovación CtrlObj_Version de la tabla de versión CtrlObj se almacena en la memoria externa 200, por ejemplo, en un encabezamiento de la tabla de versión CtrlObj.

La tabla de versión CtrlObj también puede protegerse para su almacenamiento en la memoria externa 200 y su transferencia entre el elemento seguro 100 y la memoria externa 200, por ejemplo, usando un algoritmo de cifrado autenticado. Por ejemplo, la tabla de versión CtrlObj se cifra con la clave k1 y luego se calcula un elemento de autenticación de la tabla de versión CtrlObj (aquí en forma cifrada) con un algoritmo de autenticación, tal como un algoritmo MAC, usando la clave k2. La memoria externa 200 almacena la tabla de versión cifrada [CtrlObj]k1 y su elemento de autenticación MAC[CtrlObj]. Puede usarse cualquier otro algoritmo de cifrado autenticado.

El elemento seguro 100 comprende una unidad de procesamiento central segura CPU (o procesador seguro) 110, una pluralidad de memorias internas 121, 122, 130 y un sistema de carga y pre-procesamiento 140.

La CPU segura 110, también conocida como un procesador seguro, tiene características de seguridad que tienen una certificación de nivel de garantía alta para aplicaciones críticas de seguridad. Puede ser un núcleo de propiedad intelectual de hardware integrado (núcleo IP).

Una primera memoria interna 121 es para almacenar segmentos de información digital Si antes del procesamiento por la unidad central de procesamiento 110. Es por ejemplo una memoria caché.

Una segunda memoria interna 122 es para almacenar la tabla de versión CtrlObj, de los segmentos de información digital solicitados por la CPU segura 110. Puede ser una memoria volátil, tal como una RAM.

La primera y segunda memorias internas 121, 122 son parte de un sistema de memoria volátil interno 120. Pueden ser memorias separadas o áreas diferentes de una memoria.

Una tercera memoria interna 130 es una memoria persistente (no volátil) para almacenar claves similares a datos y contadores. Más precisamente, la memoria interna 130 puede almacenar claves, como las claves principales k1 y k2, para usarse en los algoritmos criptográficos ejecutados por el sistema de carga y pre-procesamiento 140 o por el procesador seguro 110 y el contador de renovación 'FC' relacionado con la tabla de versión CtrlObj. La memoria interna persistente 130 puede ser una memoria no volátil OTP (One Time Programable).

El sistema 140 de carga y pre-procesamiento tiene varias funciones que incluyen:

- la función de cargar la tabla de versión CtrlObj de la memoria externa 200 al elemento seguro 100,

- la función de cargar piezas (segmentos) de información digital desde la memoria externa 200 al elemento seguro 100,

- la función de pre-procesamiento de la tabla de versión CtrlObj a almacenar en la segunda memoria interna (RAM) 122,

- la función del pre-procesamiento de las piezas (segmentos) de información digital a almacenar en la primera memoria interna 121 (memoria caché) y para ser procesada por el procesador seguro 110, y

- la función de comprobar la renovación de la tabla de versión cargada CtrlObj.

Estas funciones del sistema de carga y pre-procesamiento 140 se implementan por los componentes 141-145, descritos a continuación. Pueden ser componentes de hardware, o componentes de software, o mediante una mezcla de componentes de hardware y componentes de software. El procesador seguro 110 está dispuesto para ejecutar los componentes de software. En la Figura 7, los componentes 141-145 están representados fuera del procesador seguro 110 pero podrían estar dispuestos al menos parcialmente dentro del procesador seguro 110 (en particular, los componentes del software).

El sistema de carga y pre-procesamiento 140 tiene un primer cargador 141, un segundo cargador 142, un módulo de verificación de autenticación 143, un módulo de descifrado 144 y un módulo de comprobación de renovación 145 (o módulo de comprobación de anti-repetición).

El primer cargador 141 es para cargar la tabla de versión CtrlObj desde la memoria externa 200. El segundo cargador 142 es para cargar segmentos de información digital Si desde la memoria externa 200.

El módulo de verificación de autenticación 143 está configurado para autenticar datos cargados, tales como la tabla de versión CtrlObj y segmentos de información digital Si. En la presente realización, se implementa una función MAC y una función de derivación de clave KDF para derivar claves de la clave maestra k1, usando los números de versión y los identificadores de estos segmentos como entradas, para usarse como claves de autenticación para autenticar los segmentos de información digital.

El módulo de descifrado 144 está configurado para descifrar datos cargados, tales como la tabla de versión CtrlObj y segmentos de información digital Si. En la presente realización, se implementa un algoritmo de descifrado y una función de derivación de clave KDF para derivar claves de la clave maestra k2, usando los números de versión y los identificadores de estos segmentos como entradas, para usarse como claves de descifrado para descifrar los segmentos de información digital.

Alternativamente, la función de derivación de clave podría implementarse en un componente separado 146 (representado por un bloque discontinuo en la Figura 7), utilizado tanto por el módulo de verificación de autenticación 143 como por el módulo de descifrado 144.

El módulo de comprobación de renovación 145 está configurado para verificar la renovación de la tabla de versión CtrlObj con respecto al contador de renovación interno FC en la memoria de OTP 130, cuando esta tabla de versión CtrlObj se carga desde la memoria externa 200, como se describirá más adelante en la descripción del procedimiento de procesamiento seguro de información digital.

El elemento seguro 100 puede incluir además un módulo de actualización 150 para actualizar la memoria externa 200, cuando el procesador seguro 110 ha modificado uno o más segmentos de información digital, como se explica más adelante en la descripción.

Un procesador o CPU huésped (no representado) podría proporcionarse fuera del elemento seguro 100, por ejemplo, en el SoC 300, para ejecutar algunas operaciones bajo el control del elemento seguro 100, por ejemplo, cargando la tabla o los segmentos de versión. Por lo tanto, la carga podría realizarse a través de un bus de sistema o una CPU huésped, tras la solicitud del elemento seguro 100.

El elemento seguro 100 tiene otras funciones criptográficas, incluyendo una función de cifrado criptográfico y una función de autenticación como una función MAC, que permite que la CPU segura 110 encripte y autentique los datos (típicamente segmentos de información digital y la tabla de versión CtrlObj), como se explica más adelante. La función de cifrado criptográfico y la función de autenticación se implementan por los componentes 148, 149. Estos últimos son componentes de software, o componentes de hardware, o una mezcla de componentes de hardware y software.

El procesador seguro, o la unidad de procesamiento central segura (CPU) 110 se configura para procesar o usar de manera segura los segmentos de información digital cargados en la memoria interna (caché) 121. Se pretende ejecutar aplicaciones críticas de seguridad. El procesador seguro 110 también está configurado para controlar las operaciones del elemento seguro, tal como el descifrado autenticado y el cifrado autenticado. Además, cuando un segmento de información digital se ha modificado durante el procesamiento, la CPU segura 110 puede controlar la actualización de la tabla de versión CtrlObj en la memoria interna 122 y también la actualización de las memorias persistentes 200 y 130 (es decir, la actualización de los segmentos modificados y la tabla de versión CtrlObj en la memoria externa 200 y la actualización del contador de renovación FC en la memoria interna 130) que tiene que llevarse a cabo por una transacción atómica.

El módulo de actualización 150 tiene la función de incrementar el contador de renovación en la memoria de OTP 130 y actualizar la memoria externa 200, cuando se ha modificado un segmento de información digital durante el procesamiento, bajo el control del procesador seguro 110, como se explicará más adelante. Esto se puede basar principalmente en software. La acción de incrementar el contador de renovación en la memoria de OTP interna 130 y la acción de actualizar la memoria externa 200 escribiendo el segmento modificado de información digital y la tabla de versión actualizada en la memoria externa 200 se ejecutan ventajosamente juntas mediante una única transacción atómica, por razones funcionales.

El elemento seguro 100 también tiene una interfaz de hardware 160 para acceder directamente a la memoria no volátil externa 200.

Los componentes del elemento seguro 100 son confiables ventajosamente, lo que significa que no se pueden manipular para inducir fallas.

El procedimiento para procesar de manera segura información digital almacenada en la memoria externa 200 por el elemento seguro 100 se describirá ahora con referencia a las Figuras 2-6. El procedimiento puede dividirse en tres partes:

- un procedimiento de inicio, que se ejecuta después de cambiar en el elemento seguro 100 (Figuras 2 y 5);

- un procedimiento de lectura o pre-procesamiento, que sigue al procedimiento de inicio y se ejecuta antes de procesar de forma segura información digital por el procesador seguro 110 (Figuras 3 y 5); y

- un procedimiento de actualización, que se ejecuta cuando un pieza de información digital se modifica durante el procesamiento por el procesador seguro (Figuras 4 y 6).

Procedimiento de inicio (Figuras 2 y 5)

Inicialmente, el elemento seguro 100 se enciende. Posteriormente, el elemento seguro 100 ejecuta un procedimiento de arranque o inicialización. Durante este procedimiento de inicialización, el elemento seguro 100 puede ejecutar un protocolo de seguridad (por ejemplo, una inicialización segura) que puede asegurar el procedimiento de arranque. En el procedimiento de inicialización, el primer cargador 141 carga la tabla de versión CtrlObj leída de la memoria externa 200, en una etapa S50. Más precisamente, la tabla de versión CtrlObj en forma cifrada y concatenada con su MAC, [CtrlObj]k1|MAC[CtrlObj], se carga desde la memoria externa 200 al elemento seguro 100. Entonces, el módulo de verificación de autenticación 143 autentica la tabla de versión CtrlObj, en una etapa S51. Para este fin, el módulo de

verificación de autenticación 143 calcula un MAC de la tabla de versión cifrada cargada, [CtrlObj]k1, usando la clave k2, y compara el MAC calculado con el MAC cargado con la tabla de versión MAC[CtrlObj]. Si la autenticación se logra con éxito, la tabla de versión cifrada [CtrlObj]k1 es descifrada por el módulo de descifrado 144, en una etapa S52, usando la clave k1. Si la autenticación no tiene éxito, la etapa S51 es seguida por una etapa de error S70. Por ejemplo, en la etapa de error S70, el procedimiento para procesar la información digital se aborta (es interrumpido).

Después de la autenticación con la clave k2 y el descifrado con la clave k1 de la tabla de versión CtrlObj, la renovación de la tabla de versión CtrlObj se verifica mediante el módulo de verificación de renovación 145, en una etapa S53. Para este fin, la información de renovación CtrlObj_Version, incluida en el encabezamiento de la tabla de versión CtrlObj, se compara con el contador de renovación FC almacenado en la memoria de OTP 130.

Si la información de renovación CtrlObj_Version de la tabla de versión CtrlObj coincide con el contador de renovación FC almacenado en la memoria de OTP 130, se verifica con éxito la renovación de la tabla de versión CtrlObj y el procedimiento continúa con las etapas S54 a S59 de segmentos de carga y pre-procesamiento de información digital Si. Si la comprobación de la renovación de la tabla de versión cargada no tiene éxito, el procedimiento va a una etapa de error S71. Por ejemplo, en la etapa de error S71, el procedimiento para procesar la información digital se aborta (es interrumpido).

Procedimiento de lectura o pre-procesamiento (Figuras 3 y 5)

El procedimiento de inicio es seguido por un procedimiento de lectura o pre-procesamiento, realizado por el elemento seguro 100, para leer o procesar segmentos de información digital desde la memoria externa 200.

El procedimiento de carga y pre-procesamiento incluye las etapas S54 a S59 (descritas a continuación) que se repiten iterativamente para cada una de una pluralidad de segmentos de información digital Si, con $1 \leq i$. Durante el procedimiento de carga y pre-procesamiento, el índice 'i' representa el orden en el que algunos segmentos de información digital se extraen sucesivamente en la memoria externa 200 y luego se pre-procesan por el elemento seguro 100 a lo largo del tiempo. Este orden de extracción (o el orden de carga) no corresponde necesariamente al orden en el que los segmentos de información digital están apilados en la memoria externa 200. En otras palabras, cuando se supone que Si es un segmento de información digital cargado actualmente y pre-procesado por el elemento seguro 100, Si+1 es el siguiente segmento de información digital que se va a cargar y pre-procesar. Pero Si y Si+1 pueden no ser segmentos apilados consecutivamente en la memoria externa.

En la etapa S54, se carga un segmento de información digital Si en forma protegida. Más precisamente, el segundo cargador 142 carga el elemento [Si]|MAC[Si] que contiene el segmento de información digital Si cifrada con la clave k1_Si y concatenado con el elemento de autenticación MAC[si] del segmento cifrado [Si] calculado con la clave k2_Si.

En la etapa S55, el módulo de verificación de autenticación 143 obtiene el número de versión Vi de la tabla de versión CrtObj almacenada en la memoria interna 122. También obtiene el identificador de segmento IDSi desde la tabla de versión CtrlObj ya que este IDSi corresponde al índice del número de versión Vi en la tabla de versión CtrlObj. A continuación, en una etapa S56, el módulo de verificación de autenticación 143 autentica el segmento cifrado [Si], calculando un MAC de [Si] y comparándolo con el MAC[Si] cargado con [Si]. Para calcular este MAC, el módulo 143 obtiene la clave k2_Si de la clave k2 maestra leída en la memoria 130 y utiliza el número de versión Vi y el identificador IDSi del segmento Si (leído en la etapa S55) como entradas de la función de derivación de clave. Si la autenticación se logra con éxito (es decir, el MAC calculado coincide con el MAC cargado), el segmento encriptado [Si] es descifrado entonces por el módulo de descifrado 143, en una etapa S58. El descifrado utiliza una clave k1_Si calculada a partir de la clave maestra k1 (leída de la memoria 130), y utiliza el número de versión Vi y el identificador IDSi del segmento Si, obtenido de la tabla de versión CtrlObj almacenada en la memoria 122 en la etapa S57, como una entrada de la función de derivación de clave (KDF). Si la autenticación no tiene éxito, la etapa S56 es seguida por una etapa de error S72. Por ejemplo, en la etapa de error S72, el procedimiento para procesar la información digital se aborta (es interrumpido). En cambio, el segmento de información digital Si podría eliminarse.

Durante el procedimiento de pre-procesamiento, en las etapas S55 y S57, el sistema de carga y pre-procesamiento 140 obtiene el número de versión Vi y el identificador IDSi del segmento de información digital Si desde la memoria interna 122 (por ejemplo, RAM), y proporciona esta información de identificación del segmento Si como entrada en la ejecución de los algoritmos criptográficos de autenticación y descifrado utilizados para pre-procesar el segmento Si. Alternativamente, la información de identificación del segmento Si de la tabla de versión CtrlObj almacenada en la memoria interna 122 puede usarse solo por uno de los dos algoritmos criptográficos, preferentemente por el algoritmo de autenticación. Sin embargo, la seguridad aumenta si el descifrado usa también la información de identificación del segmento Si.

En lugar de usar el número de versión de segmento y el identificador de segmento como entradas para generar la clave de descifrado, el número de versión Vi y el identificador del segmento IDSi pueden usarse para generar un vector de inicialización IV para ser utilizado por el módulo de descifrado 144 para descifrar el segmento de información digital [Si].

Después de la autenticación y el descifrado del segmento de información digital Si, el segmento Si se carga en la memoria caché 121 del procesador seguro 110 en una etapa S59.

Entonces, en una etapa S60, el procesador seguro 110 puede acceder al segmento de información digital Si y puede procesarse de forma segura por el procesador seguro 110. Por ejemplo, el procesador seguro ejecuta una aplicación segura (o crítica para la seguridad) procesando el segmento de información digital Si.

Las etapas S54 a S59 se repiten iterativamente para segmentos de información digital Si cargados sucesivamente desde la memoria externa 200 al elemento seguro 100, a menos que el procedimiento se aborte de antemano debido a un error.

Procedimiento de actualización (Figuras 4 y 6)

Cuando un segmento de información digital (aquí denominado "Sj" con $1 \leq j \leq M$) es procesado por el procesador seguro 110, puede modificarse durante su procesamiento, en una etapa S80. La versión modificada del segmento de información digital Sj se denomina Sj'. En tal caso, se logra un procedimiento de actualización y se describirá ahora con referencia a las Figuras 4 y 6.

Cuando el segmento inicial Sj se cambia en el segmento modificado Sj', el procesador seguro 110 actualiza la tabla de versión CtrlObj almacenada en la memoria interna 122 incrementando el número de versión del segmento modificado Sj' en la tabla de versión CtrlObj almacenada en la memoria interna 122, en una etapa S81. Si se supone que el número de versión del segmento inicial Sj es Vj, el nuevo número de versión del segmento modificado Sj' es Vj+1. En la etapa S81 de actualización de la tabla de versión CtrlObj en la memoria interna 122, el procesador seguro 110 también actualiza el número de versión de la tabla de versión CtrlObj, ctrlObj_Version, aquí en el encabezamiento de la tabla de versión CtrlObj almacenada en la memoria interna 122. El número de versión de la tabla de versión CtrlObj, ctrlObj_Version, se incrementa en uno (CtrlObj_Version=CtrlObj_Versión+1). La tabla de versión actualizada (almacenada en la memoria 122 interna) se denomina CtrlObj'.

El elemento seguro 100 también necesita actualizar la memoria externa 200. Más precisamente, necesita escribir la pieza modificada de la información digital Sj' y la tabla de versión actualizada CtrlObj' en la memoria externa 200. Para este fin, el procesador seguro 110 envía la tabla de versión actualizada CtrlObj' y el segmento modificado Sj' a un algoritmo de cifrado autenticado, en una etapa S82, antes de controlar una transferencia del segmento modificado Sj' y la tabla de versión actualizada CtrlObj' a la memoria externa 200. Se usa el mismo algoritmo de cifrado autenticado que el usado para preparar los datos almacenados en la memoria externa. En la presente realización, el algoritmo de cifrado autenticado sigue un enfoque Cifrado-entonces-Mac. El cifrado de la tabla de versión CtrlObj' utiliza la clave k1, y la función MAC aplicada a [CtrlObj']k1 utiliza la clave k2. El cifrado del segmento modificado de información digital Sj' utiliza la clave derivada k1_Vj+1, donde k1_Vj+1=KDF (k1, Vj+1,IDSj). Debe observarse que el identificador de segmento del segmento modificado Sj' es el mismo que el identificador de segmento del segmento correspondiente Sj (antes de la modificación). La autenticación del segmento modificado de información digital en forma cifrada [Sj'] usa la clave derivada k2_Vj+1, donde k2_Vj+1=KDF(k2, Vj+1,$ID_{Sj'}$).

Entonces, en una etapa S83, bajo el control del procesador seguro 110, el módulo de actualización 150 incrementa el contador de renovación FC de la tabla de versión en la memoria de OTP 130, y escribe la nueva tabla de versión CtrlObj' y el segmento de alterado de información digital Sj' en la memoria externa 200. El contador de renovación FC se incrementa aquí en uno. Todas estas acciones (incrementar FC en la memoria interna 130 y escribir CtrlObj' y Sj' en la memoria externa 200) se ejecutan juntas mediante una única transacción atómica. Dicha operación tiene solo dos estados persistentes: un estado original sin la modificación y un estado final con la modificación. La operación puede ser rápida o tomar mucho tiempo.

Un procedimiento análogo al procedimiento de actualización anterior puede llevarse a cabo inicialmente por el elemento seguro 100 para almacenar los segmentos de información digital en forma cifrada y autenticadas en la memoria externa 200. El elemento seguro 100 puede tener un cargador seguro de firmware configurado para cargar de manera segura la información digital que va a almacenarse en la memoria externa desde un proveedor de información digital, por ejemplo, a través de una red de comunicación o a través de una conexión local a otra máquina. El elemento seguro se puede configurar para

- segmentar la información digital cargada en segmentos de información digital, autenticar y cifrar estos segmentos (de la misma manera que se describe en el procedimiento de actualización para el segmento modificado Sj') y escribirlos en la memoria externa,

- generar una tabla de versión CtrlObj, autenticar y cifrar esta tabla de versión (de la misma manera que se describe en el procedimiento de actualización) y luego escribirla en la memoria externa.

Una segunda realización se basa en la primera realización solo difiere de ella por el algoritmo de encriptación autenticado. En la segunda realización ilustrativa, el cifrado autenticado sigue un enfoque del tipo "MAC-entonces-Cifrado-entonces-MAC". Este enfoque consiste, para cada segmento de información digital Si, en el cálculo de un

primer MAC 'MACi1' del segmento Si en claro usando una primera clave k0_Vi (derivada de una clave maestra k0 y usando el número de versión Vi como una entrada), cifrando a continuación el segmento Si y el primer MAC concatenados juntos (es decir, Si|MACi1) con una segunda clave k1_Si (derivada de una clave maestra k1 y usando el número de versión Vi como una entrada), y luego calcular un segundo MAC 'MACi2' del resultado del cifrado (es decir, [Si]|[MACi1]) usando una tercera clave k2_Si (derivada de una clave maestra k2 y usando el número de versión Vi como una entrada), para obtener el segmento protegido [Si]|[MACi1]|MACi2.

Una tercera realización se basa en la primera o la segunda realización y difiere de ellas solo en que la información de identificación del segmento de información digital utilizada en la operación de autenticación y/o en la operación de descifrado como una entrada, solo contiene el número de versión del segmento de información digital. Tal realización es apropiada, por ejemplo, en caso de que el elemento seguro cargue solo un segmento o pieza de información digital.

Podrían usarse diferentes tipos de función MAC. Por ejemplo, puede usarse una función CMAC (Cipher-based Message Authentication Code - Código de Autenticación de Mensaje basado en Cifrado), es un algoritmo de código de autenticación de mensaje basado en cifrado de bloque, o un algoritmo HMAC (Hash-based Message Authentication Code - Código de Autenticación de Mensaje Basado en Hash), que es un tipo específico de código de autenticación de mensaje (MAC - Message Authentication Code) que implica una función hash criptográfica y una clave criptográfica secreta. En lugar de una función MAC, podría usarse cualquier otra función de autenticación.

REIVINDICACIONES

1. Un elemento seguro (100) para procesar de forma segura información digital, estando segmentada y almacenada dicha información digital en una pluralidad de segmentos M en una memoria (200) externa al elemento seguro (100), que incluye:

-un procesador seguro (110) para procesar de forma segura la pluralidad de segmentos M de información digital;
-un sistema (140) configurado para cargar un segmento de la pluralidad de segmentos M de información digital (Si) de la memoria (200) externa en el elemento (100) seguro, y preprocesar dicho segmento de información digital ejecutando un algoritmo criptográfico antes de procesar dicho segmento de información digital por el procesador (110) seguro;
en donde el sistema (140) está configurado para determinar una información de identificación del segmento de información digital que ha sido cargado, la determinación de dicha información de identificación, incluyendo la obtención de un número de versión de dicha información digital, y para ejecutar el algoritmo criptográfico usando dicha información de identificación.
caracterizado por que el sistema (140) está configurado para cargar una tabla de versión (CtrlObj) que contiene los respectivos números de versión de la pluralidad de M segmentos de información digital, desde la memoria externa (200) a una memoria interna (122) del elemento seguro (100), y para leer de la tabla de versión (CtrlObj) almacenada en la memoria interna (122), el número de versión (Vi) de cada segmento de información digital que se carga.

2. El elemento de seguridad según la reivindicación 1, en donde el sistema (140) está configurado para determinar la información de identificación del segmento de información digital, la determinación de dicha información de identificación que además incluye, obtener un identificador de segmento de dicho segmento de información digital, y para ejecutar el algoritmo criptográfico usando dicha información de identificación, siendo dicho identificador de segmento un índice asignado al segmento correspondiente de información digital en la memoria externa o un identificador presente en un encabezamiento del segmento correspondiente de información digital.

3. El elemento seguro según cualquiera de las reivindicaciones 1 y 2, caracterizado porque dicho elemento seguro almacena al menos una clave criptográfica única, que se generó de manera única para el elemento seguro, para ser utilizada por el sistema en la ejecución del algoritmo criptográfico.

4. El elemento seguro según la reivindicación 3, en donde el sistema (140) comprende una función de derivación de clave para generar al menos una clave criptográfica, para ser utilizada por el algoritmo criptográfico, usando la al menos una clave criptográfica única y la información de identificación de la información digital (Si) como entradas.

5. El elemento seguro según cualquiera de las reivindicaciones 1 a 4, en donde el algoritmo criptográfico comprende un algoritmo de autenticación (MAC) para autenticar el segmento de la información digital (Si) que se ha cargado.

6. El elemento seguro según cualquiera de las reivindicaciones 1 a 5, en donde el algoritmo criptográfico comprende un algoritmo de descifrado para descifrar el segmento de la información digital (Si) que se ha cargado.

7. El elemento seguro según la reivindicación 6, en donde el sistema (140) está configurado para generar un vector de inicialización (IV), para ser utilizado por el algoritmo de descifrado (144), usando la información de identificación de la información digital (Si) como una entrada.

8. El elemento seguro según las reivindicaciones 1 a 7, en donde el sistema (140) está configurado además para verificar una información de renovación (CtrlObj_Version) de dicha tabla de versión (CtrlObj) comparando dicha información de renovación (CtrlObj_Version) de la tabla de versión (CtrlObj) con un contador de renovación (FC) almacenado en una memoria no volátil interna (130) del elemento seguro (100).

9. El elemento seguro según cualquiera de las reivindicaciones 1 a 8, en donde el sistema (140) está configurado para cargar la tabla de versión (CtrlObj) en un procedimiento de inicio del elemento seguro (100).

10. El elemento seguro según cualquiera de las reivindicaciones 1 a 9, en donde el sistema (140) está configurado para autenticar y descifrar la tabla de versión (CtrlObj) que se ha cargado.

11. El elemento seguro según cualquiera de las reivindicaciones 1 a 10, que comprende además un módulo de actualización (150) configurado, en caso de que un segmento de información digital (Sj) se modifique cuando sea procesado por el procesador seguro (110), para controlar una operación de escritura del segmento

modificado de información digital (Sj') y una tabla de versión actualizada (CtrlObj') en la memoria externa (200), e incrementar el contador de renovación en la memoria no volátil interna (130).

12. Un sistema que incluye el elemento seguro (100) según cualquiera de las reivindicaciones 1 a 11 y una memoria externa (200) para almacenar la información digital.

13. Un método para procesar de manera segura información digital mediante un elemento seguro (100), estando segmentada y almacenada dicha información digital en una pluralidad de segmentos M en una memoria (200) externa al elemento seguro (100), incluyendo las siguientes etapas, realizadas por el elemento seguro (100), de:

-cargar (S54) un segmento de la pluralidad de segmentos M
de información digital ([Si]k1|MAC[Si]) de la memoria externa (200) al elemento seguro (100);
-determinar una información de identificación del segmento de información digital que ha sido cargado obteniendo un número de versión (Vi) de dicho segmento de información digital;
-pre-procesar (S56-S58) dicho segmento de información digital ([Si]k_1|MACi) ejecutando un algoritmo criptográfico antes de procesar dicho segmento de información digital (Si) mediante un procesador seguro (110) del elemento seguro (100);
-procesar de forma segura (S59) el segmento de información digital (Si) por el procesador seguro (110);
caracterizado porque el método comprende además:
una etapa de carga de una tabla de versión (CtrlObj) que contiene números de versión respectivos de la pluralidad de segmentos M de información digital desde la memoria externa (200) en una memoria interna (122) del elemento seguro (100) y una etapa de lectura>de la tabla de versión (CtrlObj) almacenada en la memoria interna (122), el número de versión (Vi) de cada segmento de información digital que se carga.

14. El método según la reivindicación 13, que comprende además una etapa de verificación de información de renovación (CtrlObj_Version) de dicha tabla de versión (CtrlObj) comparando dicha información de renovación (CtrlObj_Version) de la tabla de versión (CtrlObj) con un contador de renovación (FC) almacenado en una memoria no volátil interna (130) del elemento seguro (100).

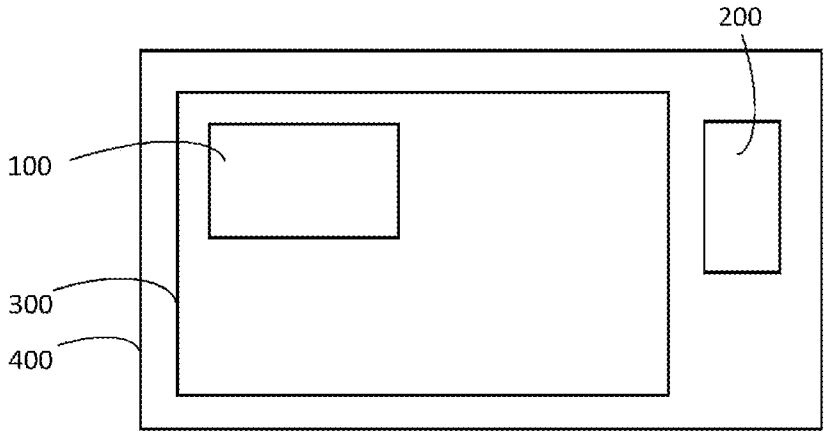

Figura 1

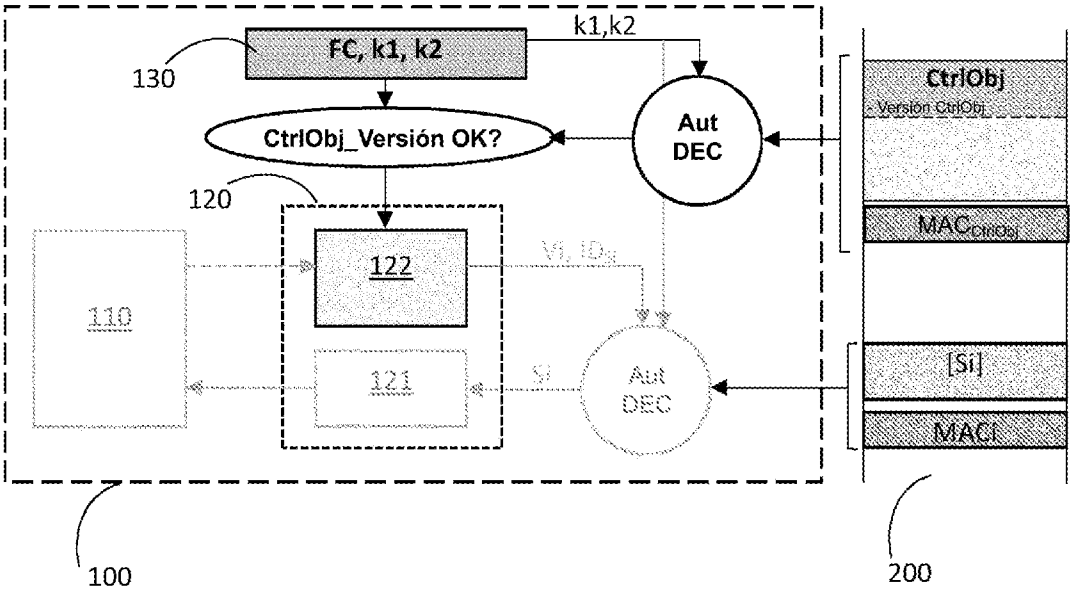

Figura 2

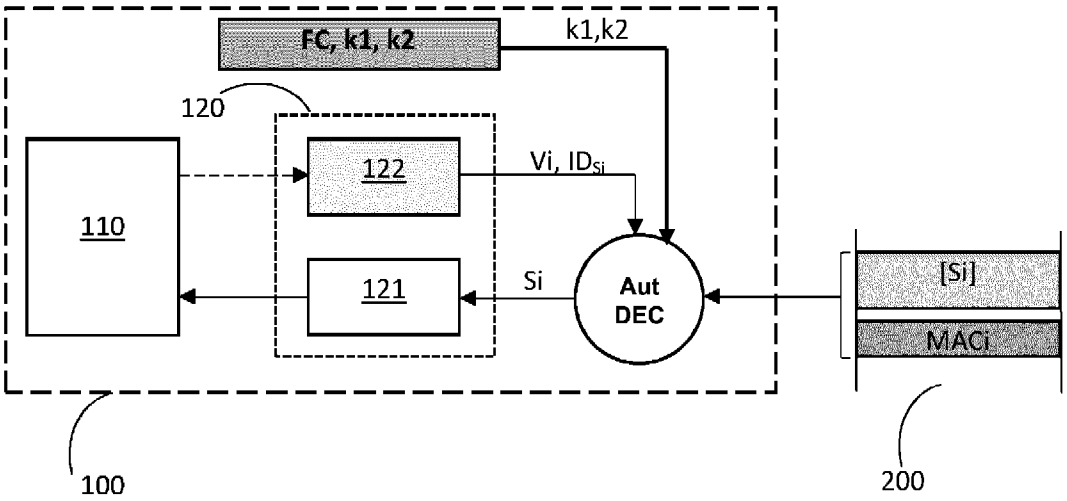

Figura 3

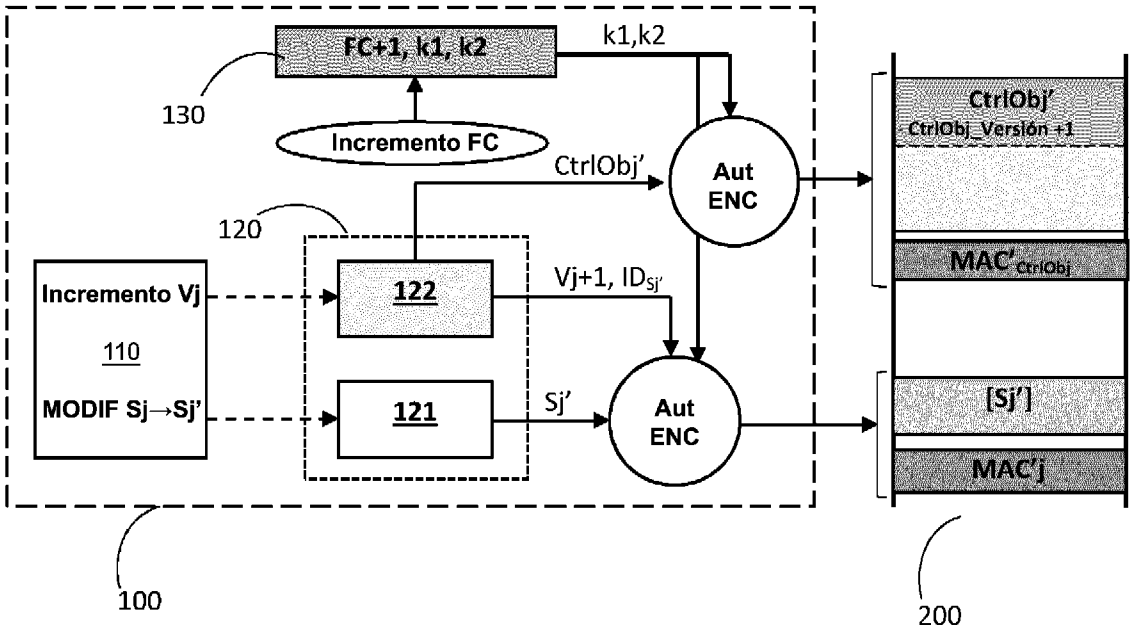

Figura 4

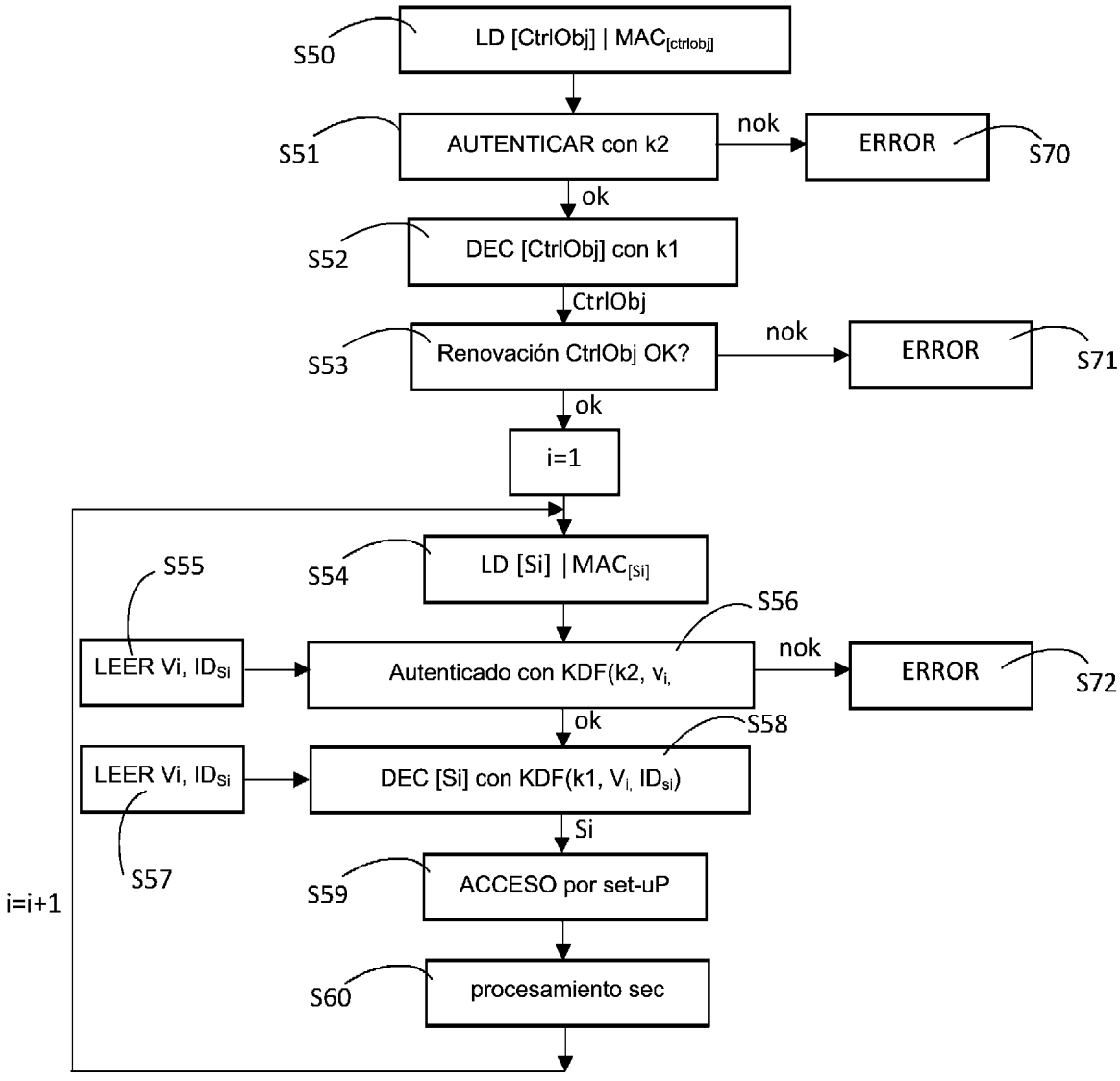
S50
LD [CtrlObj] | MAC[ctrlobj]
S51
AUTENTICAR con k2
nok
ERROR
S70
ok
S52
DEC [CtrlObj] con k1
CtrlObj
S53
Renovación CtrlObj OK?
nok
ERROR
S71
ok
i=1
S55
S54
LD [Si] |MAC[Si]
S56
LEER Vi, IDSi
Autenticado con KDF(k2, vi,
nok
ERROR
S72
ok
S58
LEER Vi, IDSi
DEC [Si] con KDF(k1, Vi, IDsi)
Si
S57
S59
ACCESO por set-uP
i=i+1
S60
procesamiento sec

Figura 5

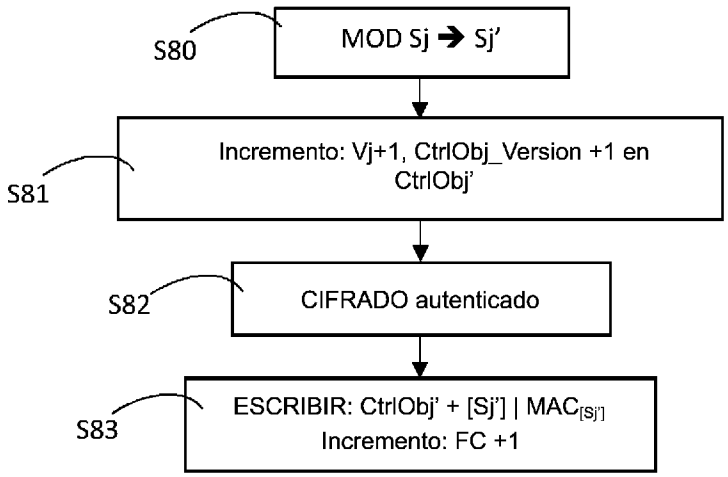

Figura 6

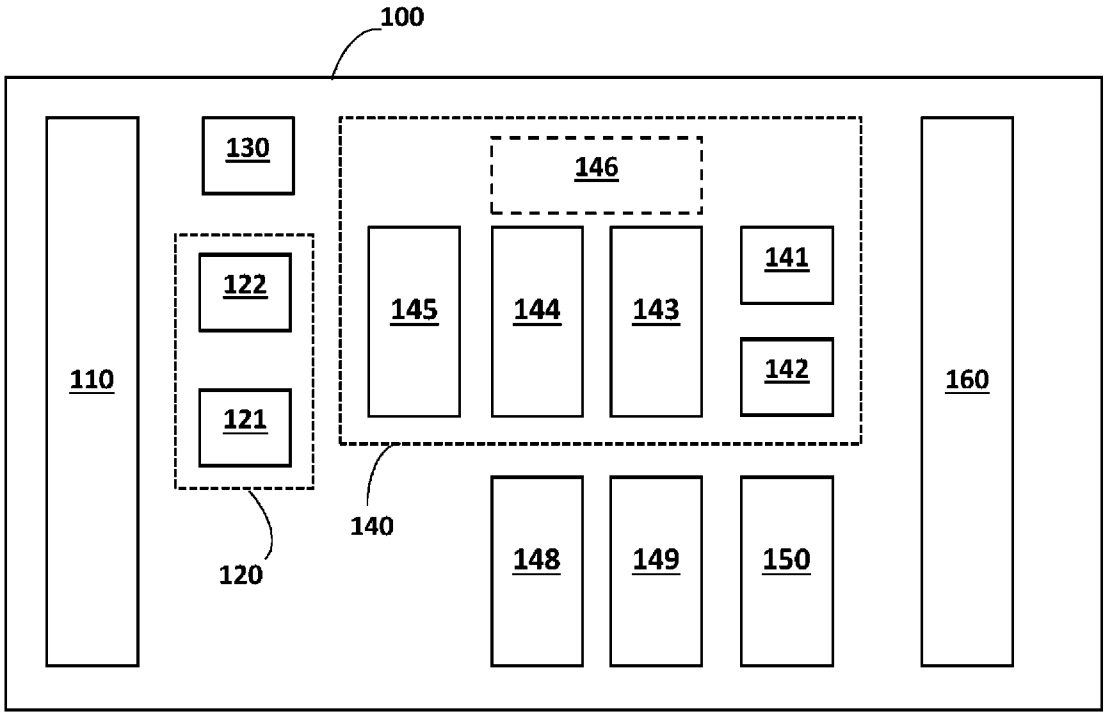

Figura 7