(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 969 160**

(51) Int. Cl.:

G06F 21/57 (2013.01)
G06F 9/4401 (2008.01)
G06F 9/455 (2008.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **17.02.2020 PCT/EP2020/054101**

(87) Fecha y número de publicación internacional: **17.09.2020 WO20182420**

(96) Fecha de presentación y número de la solicitud europea: **17.02.2020 E 20705701 (9)**

(97) Fecha y número de publicación de la concesión europea: **20.12.2023 EP 3935545**

(54) Título: **Descifrado incremental y verificación de integridad de una imagen de sistema operativo seguro**

(30) Prioridad:

08.03.2019 US 201916296334

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
16.05.2024

(73) Titular/es:

INTERNATIONAL BUSINESS MACHINES CORPORATION (100.0%)
New Orchard Road
Armonk, New York 10504, US

(72) Inventor/es:

BUENDGEN, REINHARD;
BORNTRAEGER, CHRISTIAN;
BRADBURY, JONATHAN;
BUSABA, FADI;
HELLER, LISA y
MIHAJLOVSKI, VIKTOR

(74) Agente/Representante:

ELZABURU, S.L.P

DESCRIPCIÓN

Descifrado incremental y verificación de integridad de una imagen de sistema operativo seguro

Un sistema operativo invitado se refiere al software del sistema operativo instalado, por ejemplo, en una máquina virtual. En informática, una máquina virtual es una emulación de un sistema informático. Las máquinas virtuales, que pueden ser creadas por un hipervisor, se basan en una arquitectura informática y proporcionan la funcionalidad de un ordenador físico. Sus implementaciones pueden implicar hardware especializado, software, o una combinación de los mismos. La tecnología de virtualización permite que un ordenador ejecute más de un sistema operativo individual a la vez.

En los entornos informáticos actuales, el software del sistema, tal como el sistema operativo, debe ser confiable, debido a que tiene control completo sobre la aplicación que se va a ejecutar y los datos. Tradicionalmente, el sistema operativo (así como el hipervisor) pueden acceder o modificar los datos de cualquier aplicación o, potencialmente manipular una característica de seguridad implementada por la aplicación sin ser detectado. En consecuencia, el software subyacente debe ser parte de una base informática confiable.

En los entornos compartidos, los clientes de aplicaciones se ven obligados a confiar en que las entidades que desarrollan, configuran, implementan y controlan el software del sistema no son maliciosas. Las aplicaciones de los clientes también deben confiar en que el software del sistema es invulnerable a los ataques que aumentan los privilegios y comprometen la integridad de la información de la aplicación. Este amplio requisito de confianza a veces puede ser difícil de justificar y conlleva un riesgo significativo, especialmente para los clientes de aplicaciones que adoptan, por ejemplo, servicios en la nube pública.

El documento EP 3367287 A1 es otro estado de la técnica.

Sumario

La presente invención está definida por las reivindicaciones independientes adjuntas. Las realizaciones preferidas se exponen en las reivindicaciones dependientes.

Se superan ciertas deficiencias de la técnica anterior y se proporcionan ventajas adicionales mediante la provisión de un sistema informático para facilitar el procesamiento seguro dentro de un entorno informático. El sistema informático incluye una memoria y un procesador acoplado a la memoria, y el sistema informático está configurado para realizar un método. El método incluye descifrar de manera incremental una imagen del sistema operativo seguro. El descifrado incremental incluye, para una página de una pluralidad de páginas de la imagen del sistema operativo seguro: recibir una dirección de página de la página y un valor de ajuste utilizado durante el cifrado de la página; determinar que el valor de ajuste no se ha utilizado previamente durante el descifrado de otra página de la pluralidad de páginas de la imagen del sistema operativo seguro; y descifrar el contenido de la página de memoria en la dirección de la página utilizando una clave de cifrado de imagen y el valor de ajuste para facilitar la obtención de una imagen descifrada del sistema operativo seguro. El método también incluye verificar la integridad de la imagen del sistema operativo seguro y, en base a la verificación de la integridad de la imagen del sistema operativo seguro, iniciar la ejecución de la imagen descifrada del sistema operativo seguro. Ventajosamente, el descifrado incremental de las páginas de una imagen del sistema operativo seguro, donde las páginas se cifran por separado utilizando una clave de cifrado de imagen y un valor de ajuste único (también conocido como vector de inicialización), proporciona una seguridad mejorada dentro del entorno informático, por ejemplo, evitando el análisis estadístico de la imagen del sistema operativo seguro para obtener datos significativos. Además, el enfoque resumido también permite que los datos almacenados en diferentes ubicaciones se cifren de manera diferente.

En una o más realizaciones, el descifrado incremental y la verificación de la integridad se realizan mediante un control de interfaz segura del sistema informático. El control de interfaz segura es una entidad segura y confiable del sistema informático, y la dirección de la página y el valor de ajuste son recibidos por el control de interfaz segura desde un hipervisor. Ventajosamente, el control de interfaz segura presenta un entorno de ejecución seguro, en donde la memoria de un sistema operativo invitado y sus aplicaciones que se ejecutan en máquinas virtuales seguras están protegidas del hipervisor, así como otras máquinas virtuales y cualquier programa que esté ejecutándose en el mismo sistema anfitrión que la máquina o máquinas virtuales.

En una o más realizaciones, las direcciones de página de la pluralidad de páginas están en un orden fijo (por ejemplo, de menor a mayor), y el descifrado incremental de la imagen del sistema operativo seguro incluye determinar que la dirección de la página es diferente (por ejemplo, mayor en el caso en donde el orden sea de menor a mayor) a cualquier dirección de página anterior de páginas previamente descifradas de la pluralidad de páginas de la imagen del sistema operativo seguro. Además, en una o más implementaciones, el descifrado incremental de la imagen del sistema operativo seguro incluye además determinar que el valor de ajuste es diferente (por ejemplo, mayor en el caso de un orden de menor a mayor) a cualquiera de los valores de ajuste previos utilizados durante el descifrado de las páginas previamente descifradas de la pluralidad de páginas de la imagen del sistema operativo seguro.

En una realización, la verificación de la integridad incluye emplear, en parte, un hash de contenido acumulativo obtenido utilizando el contenido de la página de memoria de la pluralidad de páginas y un hash de dirección

acumulativo obtenido utilizando las direcciones de página de la pluralidad de páginas. Por ejemplo, la verificación de la integridad de la imagen del sistema operativo seguro puede incluir comparar el hash de contenido acumulativo con un hash de contenido de integridad recibido por el control de interfaz segura con los metadatos del hipervisor, y comparar el hash de dirección acumulativo con un hash de dirección de integridad recibido por el control de interfaz segura con los metadatos del hipervisor. Por ejemplo, el control de interfaz segura puede extraer el hash de contenido de integridad y el hash de la dirección de integridad, así como la clave de cifrado de la imagen, de los metadatos recibidos del hipervisor. Ventajosamente, el control de interfaz segura, es decir, una entidad segura del sistema anfitrión, es capaz de verificar la integridad de la imagen del sistema operativo a fin de evitar la ejecución de una imagen que ha sido manipulada. Para este propósito, se utilizan el valor de hash calculado a partir del contenido de las páginas de memoria, así como el valor de hash de la lista de las direcciones de páginas de memoria de la imagen del sistema operativo y se comparan con los valores hash de integridad respectivos del contenido y las direcciones que se han proporcionado en el orden de las direcciones de las páginas. Obsérvese, a este respecto, que el orden de las direcciones de las páginas de menor a mayor facilita tanto la validez de ajuste como la verificación de la integridad. Se señala también que las páginas de memoria no necesitan estar contiguas utilizando los procesos mencionados anteriormente.

En una o más realizaciones, el control de interfaz segura incluye elementos de firmware del sistema informático, y la imagen del sistema operativo seguro que se va a ejecutar como una imagen del sistema operativo invitado seguro dentro del entorno informático. En una o más realizaciones específicas, un par de claves privada-pública está asociado con el sistema informático, siendo la clave privada conocida solo por el control de interfaz segura y no accesible al software que se ejecuta en el sistema informático. Un propietario de la imagen del sistema operativo seguro utiliza la clave pública para generar datos del acuerdo de la clave a fin de comunicar de forma segura la clave de cifrado de la imagen utilizada para cifrar la imagen del sistema operativo seguro con el control de interfaz segura y almacenar los datos del acuerdo de la clave en una estructura de metadatos. El control de interfaz segura puede recibir la estructura de metadatos con una llamada desde el hipervisor. La estructura de metadatos puede incluir además valores hash de integridad utilizados en la verificación de la integridad de la imagen del sistema operativo seguro. Ventajosamente, este proceso da como resultado un entorno de ejecución seguro, en donde se preserva la confidencialidad de la imagen del sistema operativo seguro y se verifica al mismo tiempo la integridad del sistema operativo. En este proceso, el propietario de la imagen del sistema operativo seguro elige la clave secreta, es decir, la clave de cifrado de la imagen, y se supone que todos los componentes del sistema operativo están cifrados, así como los datos en reposo utilizados por la imagen del sistema operativo después de que se haya arrancado.

También se describen y reivindican en el presente documento productos de programa informático y métodos implementados por ordenador que se relacionan con uno o más aspectos. Además, también se describen y pueden reivindicarse en el presente documento servicios relacionados con uno o más aspectos.

Se obtienen características y ventajas adicionales a través de las técnicas de la presente descripción. Otras realizaciones y aspectos de la invención se describen en detalle en el presente documento y se consideran una parte de la invención reivindicada.

Breve descripción de los dibujos

Uno o más aspectos de la presenta invención se señalan particularmente y se reivindican claramente como ejemplos en las reivindicaciones al final de la memoria descriptiva. Los objetos, características y ventajas anteriores de la invención y otros son evidentes a partir de la siguiente descripción detallada tomada junto con los dibujos adjuntos, en los que:

La FIG. 1 representa un ejemplo de un entorno informático para incorporar y utilizar uno o más aspectos del descifrado de imagen del sistema operativo seguro y verificación de la integridad, de acuerdo con uno o más aspectos de la presente invención;

La FIG. 2 representa una realización de un proceso de descifrado incremental y verificación de la integridad de una imagen de sistema operativo seguro, de acuerdo con uno o más aspectos de la presente invención;

La FIG. 3A representa una realización del contenido de la memoria de una máquina virtual (VM) después de una carga de la imagen del sistema operativo seguro, de acuerdo con uno o más aspectos de la presente invención;

La FIG. 3B representa una realización de metadatos utilizados para descomprimir una imagen de un sistema operativo seguro conforme a un proceso de descifrado incremental y verificación de la integridad, de acuerdo con uno o más aspectos de la presente invención;

La FIG. 4A representa una realización de un proceso de descompresión de una imagen de inicio, que muestra las estructuras de datos y operaciones de un espacio de dirección de invitado, hipervisor y ultravisor (o control de interfaz segura) conforme a un proceso de descifrado incremental y verificación de la integridad, de acuerdo con uno o más aspectos de la presente invención;

La FIG. 4B representa detalles adicionales de una realización de un proceso de descompresión entre el espacio de dirección de invitado, hipervisor y ultravisor de la FIG. 4A, de acuerdo con uno o más aspectos de la presente invención;

La FIG. 4C representa una realización del espacio de dirección de invitado, hipervisor y ultravisor de las FIGS. 4A y 4B tras la finalización de la descompresión de la imagen del sistema operativo, de acuerdo con uno o más aspectos de la presente invención;

La FIGS. 5A-5B representan un ejemplo de descifrado incremental y verificación de la integridad de una imagen de sistema operativo seguro, de acuerdo con uno o más aspectos de la presente invención;

la FIG. 6A representa otro ejemplo de un entorno informático para incorporar o utilizar uno o más aspectos de la presente invención;

la FIG. 6B representa detalles adicionales de la memoria de la FIG. 6A, de acuerdo con uno o más aspectos de la presente invención;

La FIG. 7 representa una realización de un entorno de informática en la nube; y

La FIG. 8 representa un ejemplo de capas de modelo de abstracción.

Descripción detallada

De acuerdo con un aspecto de la presente invención, se proporciona una capacidad para facilitar el procesamiento seguro dentro de un entorno informático. En un ejemplo, la capacidad incluye descifrar de manera incremental una imagen del sistema operativo seguro página por página, donde se supone que cada página se ha cifrado utilizando una clave de cifrado de imagen, es decir, una clave secreta y un valor de ajuste único respectivo. El descifrado incremental incluye determinar que el valor de ajuste único para una página particular no se ha utilizado previamente durante el descifrado de otra página de la imagen del sistema operativo seguro. Adicionalmente, la verificación de la integridad de la imagen del sistema operativo seguro se realiza de forma proporcional al descifrado de las páginas cifradas de la imagen. Además, en base a la integridad de la verificación de la imagen del sistema operativo seguro, se inicia la ejecución de la imagen.

Un ejemplo de un entorno informático para incorporar y utilizar uno o más aspectos de un proceso de descifrado incremental y verificación de la integridad tal como se describe en el presente documento se describe con referencia a la FIG. 1. Haciendo referencia a la FIG. 1, en un ejemplo, un entorno informático 100 se basa en z/Architecture, ofrecida por International Business Machines (IBM®) Corporation, Armonk, Nueva York. La z/Architecture se describe en una publicación de IBM® titulada "z/Architecture Principles of Operation", publicación de IBM n.º SA22-7832-11, 12ª edición, septiembre de 2017.

Z/ARCHITECTURE, IBM, Z/VM y Z/OS (referenciadas en el presente documento) son marcas comerciales registradas de International Business Machines Corporation, Armonk, Nueva York. Otros nombres utilizados en el presente documento pueden ser marcas comerciales registradas, marcas comerciales o nombres de productos de International Business Machines Corporation u otras compañías.

En otro ejemplo, el entorno informático se basa en la Power Architecture, ofrecida por International Business Machines Corporation, Armonk, Nueva York. Una realización de la Power Architecture se describe en "Power ISA™ Version 2.07B", International Business Machines Corporation, 9 de abril de 2015.

POWER ARCHITECTURE es una marca comercial registrada de International Business Machines Corporation, Armonk, Nueva York, EE. UU.

El entorno informático 100 incluye un complejo de procesador central (CPC) 102 que proporciona soporte a la máquina virtual. El CPC 102 se acopla a uno o más dispositivos de entrada/salida (E/S) 106 a través de una o más unidades de control 108. El complejo de procesador central 102 incluye, por ejemplo, una memoria de procesador 104 (también conocida como memoria principal, almacenamiento principal, almacenamiento central) acoplada a uno o más procesadores centrales (también conocidos como unidades centrales de procesamiento (CPU)) 110 y un subsistema de entrada/salida 111, cada uno de los cuales se describe a continuación.

La memoria del procesador 104 incluye, por ejemplo, una o más máquinas virtuales 112, un gestionador de máquina virtual, tal como un hipervisor 114 que gestiona las máquinas virtuales, un ultravisor 115 (o control de interfaz segura) y el firmware del procesador 116. Un ejemplo de hipervisor 114 es z/VM®, ofrecido por International Business Machines Corporation, Armonk, Nueva York. El hipervisor algunas veces es referido como el anfitrión. Además, como se utiliza en el presente documento , el firmware incluye, por ejemplo, el microcódigo y/o milicódigo del procesador. Incluye, por ejemplo, las instrucciones a nivel de hardware y/o las estructuras de datos utilizadas en la implementación del código de máquina de nivel superior. En una realización, incluye, por ejemplo, código de propietario que normalmente se entrega como microcódigo que incluye software confiable o microcódigo específico para el hardware subyacente y controla el acceso del sistema operativo al hardware del sistema. En una o más realizaciones, el control de interfaz segura (ultravisor 115) puede implementarse, al menos en parte, en hardware y/o firmware configurados para realizar, por ejemplo, procesos tales como los descritos en el presente documento.

El soporte de la máquina virtual del CPC proporciona la capacidad de operar grandes números de máquinas virtuales

112, cada una capaz de operar con diferentes programas 120 y ejecutar un sistema operativo invitado 122, tal como Linux. Cada máquina virtual 112 es capaz de funcionar como un sistema separado. Es decir, cada máquina virtual se puede restablecer de forma independiente, ejecutar un sistema operativo invitado y operar con diferentes programas. Un sistema operativo o un programa de aplicación que se ejecuta en una máquina virtual parece tener acceso a un sistema total y completo, pero en realidad solo una parte de él está disponible.

La memoria del procesador 104 está acoplada a procesadores centrales (CPU) 110, que son recursos de procesador físicos asignables a máquinas virtuales. Por ejemplo, la máquina virtual 112 incluye uno o más procesadores lógicos, cada uno de los cuales representa todo o una parte de un recurso de procesador físico 110 que puede asignarse dinámicamente a la máquina virtual.

Adicionalmente, en una realización, cada CPU 110 es un hilo de hardware que se ejecuta dentro de un núcleo de procesamiento (también conocido como núcleo) 132. Un núcleo incluye uno o más hilos y, en este ejemplo, el núcleo 132 incluye cuatro hilos de hardware. En otros ejemplos, el entorno informático puede incluir uno o más núcleos, y cada núcleo puede incluir uno o más hilos de hardware.

Además, la memoria del procesador 104 está acoplada a un subsistema de E/S 111. El subsistema de entrada/salida 111 dirige el flujo de información entre las unidades de control de entrada/salida 108 y los dispositivos 106 y el almacenamiento principal 104. Está acoplado al complejo de procesamiento central, ya que puede ser parte del complejo de procesamiento central o estar separado del mismo.

En un ejemplo particular, el modelo de máquinas virtuales es un modelo V=V, en donde la memoria real o absoluta de una máquina virtual está respaldada por la memoria virtual del anfitrión, en lugar de por la memoria real o absoluta. Cada máquina virtual tiene un espacio de memoria virtual contiguo. Los recursos físicos son gestionados por el anfitrión de hipervisor 114, y los recursos físicos compartidos son enviados por el anfitrión a los sistemas operativos invitados, según sea necesario, para satisfacer sus demandas de procesamiento. Este modelo de máquina virtual V=V (es decir, invitado paginable) supone que las interacciones entre los sistemas operativos invitados y los recursos de máquina compartidos físicos están controladas por el anfitrión, ya que el gran número de invitados normalmente impide que el anfitrión simplemente divida y asigne los recursos de hardware a los invitados configurados.

Como se observa, la memoria 104 se acopla al subsistema de E/S 111. El subsistema de E/S 111 puede ser parte del complejo de procesamiento central o estar separado del mismo. Puede dirigir el flujo de información entre el almacenamiento principal 104 y las unidades de control de entrada/salida 108 y los dispositivos de entrada/salida (E/S) 106 acoplados al complejo de procesamiento central.

Se pueden utilizar muchos tipos de dispositivos de E/S. Un tipo particular es un dispositivo de almacenamiento de datos 140. El dispositivo de almacenamiento de datos 140 puede almacenar uno o más programas 142, una o más instrucciones de programa legibles por ordenador 144 y/o datos, etc. Las instrucciones de programa legibles por ordenador pueden configurarse para llevar a cabo funciones de realizaciones de aspectos de la invención.

Las instrucciones de programa legibles por ordenador configuradas para llevar a cabo funciones de realizaciones de aspectos de la invención también pueden o alternativamente incluirse en la memoria 104. Son posibles muchas variaciones.

Como se describe en el presente documento, la memoria de un sistema operativo invitado y sus aplicaciones que se ejecutan en una o varias máquinas virtuales seguras están protegidas de un hipervisor que no es confiable, así como de otras máquinas virtuales y cualquiera de los programas que se están ejecutando en el mismo sistema anfitrión que la máquina virtual. La protección la aplica una entidad confiable, que incluye elementos de hardware y/o firmware de un sistema anfitrión seguro. La entidad confiable es referida en el presente documento como un control de interfaz segura o ultravisor. En una o más implementaciones, el control de interfaz segura (o ultravisor) se ejecuta en un firmware confiable. El firmware confiable reside en el Área del Sistema de Hardware (HSA), que está altamente protegido y solo se puede acceder por los niveles más bajos de firmware. Este código es propiedad, en una o más realizaciones, del fabricante del sistema informático y es enviado por el fabricante, manteniéndose de forma segura por el fabricante. La máquina lee los contenidos de este código para ejecutar el código, pero no se descarga ni se extrae de la máquina. El control de interfaz segura es, en una o más realizaciones, una extensión del hardware utilizado para implementar instrucciones que pueden ser demasiado complicadas para implementarse únicamente en el hardware. Por el contrario, el hipervisor referirse a cualquier software que pueda alojar múltiples invitados (también conocidos como máquinas virtuales). El hipervisor no necesita pertenecer al firmware del sistema que está cargado con cuidado, y es protegido del acceso por el software que se ejecuta en el sistema. En general, el hipervisor puede ser, en una o más realizaciones, software de terceros, con amplias interfaces de gestión, que incluyen interfaces de red. Por tanto, el hipervisor no es confiable y puede estar sujeto a vulnerabilidades explotables a través de una interfaz de red, o a una gestión maliciosa o negligente.

Una imagen del sistema operativo destinada a ejecutarse como un invitado seguro en un entorno informático se supone en el presente documento que está protegida contra la inspección y la manipulación, incluso antes de que esté disponible para el sistema anfitrión con el fin de asegurar la confidencialidad y la integridad. Para lograr esto, la imagen del sistema operativo se cifra y su integridad se protege mediante una clave secreta elegida por el propietario de la

imagen. La clave secreta también es referida en el presente documento como clave de cifrado de imagen. Se supone que todos los componentes de la imagen del sistema operativo están cifrados, así como todos los datos en reposo utilizadas por el sistema operativo después que el sistema se haya arrancado.

El cifrado de los datos no requerido para arrancar el sistema operativo se puede realizar independientemente del cifrado de la propia imagen del sistema operativo. El descifrado de los datos adicionales puede realizarlo el sistema operativo que se ejecuta como invitado seguro. Por ejemplo, una imagen cifrada del sistema operativo Linux puede contener un núcleo, un disco RAM (ramdisk) inicial y una línea de parámetros del núcleo. El disco RAM inicial puede contener la clave secreta que se utiliza para descifrar los datos que residen en un volumen de disco cifrado asignado a la máquina virtual.

El sistema anfitrión que ejecuta la máquina virtual necesita ser capaz de descifrar la imagen del sistema operativo seguro para poder arrancarla. Para asegurar la confidencialidad, esta capacidad está reservada en el presente documento al hardware y firmware confiables del sistema anfitrión (el "control de interfaz segura" o "ultravisor"). Para este propósito, se asocia un par de claves pública-privada al sistema anfitrión. La clave privada solo se conocen por el hardware y el firmware confiable del sistema anfitrión (es decir, el control de interfaz segura o ultravisor) y no es accesible por ningún software que se ejecute en el anfitrión. La clave pública se utiliza por el propietario de la imagen cifrada del sistema operativo para generar datos del acuerdo de la clave para comunicar de forma segura la clave de cifrado de la imagen utilizada para cifrar la imagen del sistema operativo seguro con el control de interfaz segura y almacenar los datos del acuerdo de la clave en una estructura de metadatos (referida en el presente documento como un Encabezado de Ejecución Segura (SE)). A modo de ejemplo, se pueden utilizar métodos de envoltura de claves RSA o de intercambio de claves de tipo Diffie-Hellman/KEM. El tamaño del encabezado SE es independiente del tamaño de la imagen del sistema operativo.

Además, como se describe en el presente documento, el sistema anfitrión también verifica la integridad de la imagen del sistema operativo para evitar la ejecución de una imagen que ha sido manipulada. Para este propósito, se calcula un valor de hash de integridad para el contenido de las páginas de memoria, así como un valor de hash de integridad para la lista de direcciones de página de memoria de la imagen del sistema operativo. En una o más realizaciones, para facilitar el cálculo de ambos valores hash, las páginas y sus direcciones se proporcionan en el orden de las direcciones de página, por ejemplo, del más bajo al más alto. Estos valores hash de integridad se comparan con los que se pueden almacenar y proporcionar al control de interfaz segura o ultravisor a través de la estructura de metadatos (Encabezado SE).

El proceso de descifrar de manera incremental y facilitar la verificación de la integridad de la imagen del sistema operativo seguro es referido en el presente documento como descompresión. Uno o más aspectos descritos en el presente documento cubren un proceso dentro del sistema anfitrión seguro que descifra la imagen del sistema operativo de una manera que preserva la confidencialidad y verifica la integridad de la imagen del sistema operativo al mismo tiempo. En una o más realizaciones, esto puede facilitarse mediante la aplicación del orden de páginas de memoria durante el proceso de descompresión. Otra ventaja de los aspectos inventivos descritos en el presente documento es la capacidad de manejar imágenes del sistema operativo guardadas de forma no contigua.

Adicionalmente, en uno o más aspectos, el enfoque del descifrado y verificación de la integridad de la imagen del sistema operativo seguro descrito hace que el análisis estadístico de la imagen del sistema operativo seguro y los ataques a la imagen basados en el análisis estadístico de la misma más difíciles de realizar.

Como se observa, en una o más realizaciones, las imágenes seguras del sistema operativo especialmente preparadas se ejecutan en el presente documento de una manera que no permitirá la observación del contenido de la memoria en el estado seguro. Para tener una seguridad intrínseca, se supone que la imagen del sistema operativo está cifrada con una clave, referida en el presente documento como clave secreta o clave de cifrado de imagen, elegida por un usuario o propietario de la imagen del sistema operativo seguro. Después de que se ha realizado el cifrado, la clave de cifrado, es decir, la clave de cifrado de la imagen, se envuelve con una segunda clave pública de un par de claves pública-privada, o con una clave simétrica derivada de un par de claves pública-privada, del sistema informático anfitrión. En una o más realizaciones, el cifrado página por página puede ser a través de un algoritmo de clave simétrica que utiliza, por ejemplo, un vector de inicialización, referido en el presente documento como un valor de ajuste, siendo el valor de ajuste único para cada página de la imagen.

La FIG. 2 representa una realización de procesamiento de descompresión, de acuerdo con uno o más aspectos. En una o más realizaciones, el ultravisor o control de interfaz segura ofrece una funcionalidad, accesible para el hipervisor, a través de llamadas de ultravisor, que incluyen una llamada de operación de descompresión de inicio, una llamada de operación de descompresión de realización página por página y una llamada de operación de descompresión completa. Antes de poder llamar al ultravisor, la imagen del sistema operativo seguro, cifrada, se carga en la memoria de la máquina virtual (VM), que sigue siendo insegura, para permitir que el hipervisor realice la carga. Además, el hipervisor debe conocer las direcciones de todas las páginas que constituyen la imagen del sistema operativo, y también los ajustes utilizados para el cifrado de cada página. No es importante para los aspectos de la presente invención cómo esta información se hace disponible para el hipervisor. Por simplicidad, se puede cargar un componente de arranque no cifrado en la memoria de la máquina virtual, junto con la imagen cifrada. Este componente se puede utilizar para llamar al hipervisor con los siguientes parámetros: encabezado SE, una lista de direcciones de

páginas de memoria y una lista de ajustes (por ejemplo, una dirección de ajuste por página).

En lugar de proporcionar listas de páginas completas al hipervisor, también es posible especificar una lista de rangos de páginas para ahorrar memoria. De modo similar, en lugar de proporcionar una lista de ajustes, es posible especificar rangos de ajustes en los que los ajustes posteriores difieran en un valor positivo fijo. Alternativamente, los valores de ajuste podrían derivarse de las direcciones de página. Ninguna de las opciones anteriores tiene una influencia en la interacción entre el hipervisor y el ultravisor. El hipervisor inicia la operación de descompresión por una llamada al ultravisor, por ejemplo, "comienza a descomprimir", con el encabezado SE como un parámetro 200. El encabezado SE contiene la clave de cifrado de imagen del propietario envuelta por la clave de anfitrión pública y los valores hash de integridad de la imagen para el contenido de la página de la imagen y las direcciones de página. El ultravisor extrae la clave de cifrado de la imagen y los valores de integridad del encabezado SE 202, y determina si los datos del encabezado SE son válidos 204. Si "no", entonces la operación de descompresión ha fallado y se ingresa en un estado de espera deshabilitada 206. El estado de espera deshabilitada se ingresa de modo que el sistema operativo invitado no puede ejecutarse, es decir, allí el invitado es incapaz de salir del estado de espera deshabilitada. Por ejemplo, se puede devolver un error al hipervisor. En una o más realizaciones, el sistema invitado se puede restablecer, en cuyo caso comenzaría de nuevo con la carga de la imagen cifrada e iniciaría el proceso de descompresión descrito anteriormente. Alternativamente, el sistema invitado podría apagarse, lo que eliminaría eficazmente la máquina virtual, es decir, liberaría los recursos de anfitrión utilizados por esa máquina virtual.

El hipervisor recorre todas las direcciones de página de las regiones de memoria especificadas, ordenadas de direcciones de página de menor a mayor (en una o más realizaciones) 208, y llama a la operación "descomprimir página" del ultravisor con una dirección de página y un ajuste utilizados para el cifrado de esta página 210. A continuación, el ultravisor realiza las siguientes operaciones. El ultravisor establecerá la página de memoria como segura y descifrará el contenido de la página de memoria utilizando la clave de cifrado de la imagen del encabezado SE y el valor de ajuste que se utilizó para cifrar la página. Para evitar que un valor de ajuste se utilice más de una vez, cada nuevo ajuste debe ser representativo por un número mayor que un ajuste utilizado previamente, y cada dirección de página debe ser mayor que una dirección de página 212 utilizada previamente. Si "no", entonces el descompresión ha fallado y se ingresa en un estado de espera deshabilitada 206. De lo contrario, el ultravisor marca la página como segura, descifra la página utilizando la clave de cifrado de la imagen y la ajusta, y determina un hash de contenido acumulativo para el contenido de la página y un hash de dirección acumulativo para las direcciones de página 214. (Por ejemplo, el cálculo del hash acumulativo puede ser: cum_hash: = cum_hash + hash parcial (página i) sobre todas las páginas, donde 1<= i <= *n*, y + es una operación complicada.) El ultravisor determina, en una realización, un valor de hash en ejecución a partir del contenido de la página de memoria y un valor resumido de todas las páginas previamente descifradas, y otro valor de hash a partir de la dirección de página de la página de memoria y un valor resumido de las direcciones de página de todas las páginas previamente descifradas. Se podría lograr una mejora con respecto al cálculo hash de dirección página por página, tal como se indicó anteriormente, concatenando múltiples direcciones de página en una página de memoria y luego calculando el valor de hash sobre esa página de memoria.

Después de que se hayan procesado 208 todas las páginas de imagen del sistema operativo, el hipervisor llama al ultravisor con una operación 216 de "finalizar la descompresión", en donde el ultravisor compara el contenido calculado y los valores hash de dirección para el contenido y las direcciones de página con los respectivos valores hash de integridad contenidos en el encabezado SE 218. Si no coinciden, entonces la operación de descompresión falló y se devuelve una indicación de error al hipervisor 206. De lo contrario, la operación de descompresión ha sido exitosa y el ultravisor puede iniciar la ejecución de invitado seguro con una palabra de estado del programa (program status word (PSW)) del encabezado SE, que inicia la ejecución de la imagen del sistema operativo en modo seguro.

Obsérvese que con el procesamiento descrito en el presente documento, las páginas de memoria no necesitan ser contiguas, ya que la imagen puede tener "huecos", pero el proceso de descifrado debería (en una o más realizaciones) tener lugar en un orden de menor a mayor, para permitir tanto la validez del ajuste como la verificación de la integridad descritas en el presente documento. Las páginas que pertenezcan al invitado pero que no se desempaquetan durante la inicialización del invitado serán seguras y pondrán a cero en el acceso del invitado seguro.

A modo de ejemplo adicional, la FIG. 3A representa una realización del contenido de la memoria después de la carga de la imagen del sistema operativo seguro. Como se observa, en una o más realizaciones, se supone que la imagen del sistema operativo seguro está cifrada con una clave de cifrado de imagen por un usuario o propietario de la imagen del sistema operativo seguro. Esta clave de cifrado de imagen es una clave secreta elegida por el usuario o propietario. Después de que se completa el cifrado, la clave de cifrado se envuelve con una clave pública de, o con una clave simétrica derivada de, un par de claves pública-privada del sistema informático de anfitrión y se almacena en una estructura de metadatos (es decir, encabezado ejecución segura (SE)) 301, que se proporciona dentro del espacio de dirección de invitado 300 como parte de los metadatos 303 del disco de invitado 310. El disco de invitado 310 y el espacio de dirección de invitado 300 son (en una o más realizaciones) memoria proporcionada por el hipervisor a la máquina virtual para ejecutar la imagen del sistema operativo seguro. Junto con el encabezado SE 301, se puede proporcionar un componente de arranque 302 para iniciar la ejecución de la imagen del sistema operativo seguro. En una o más implementaciones, estos metadatos 303 no están cifrados, pero están protegidos contra la manipulación. La imagen del sistema operativo seguro 305 incluye páginas de dirección 304 que, como se señaló anteriormente, se han cifrado individualmente utilizando la clave de cifrado de imagen del propietario para asegurar que la imagen del sistema operativo esté protegida. La imagen del sistema operativo seguro se puede cargar en la memoria (como se

representa en la FIG. 3A) a través del hipervisor.

La FIG. 3B representa además los metadatos utilizados, en una o más realizaciones, para descomprimir la imagen del sistema operativo seguro. Como se ilustra, el encabezado SE 301 dentro del espacio de dirección de invitado 300 puede incluir la clave de cifrado de imagen (envuelta en una clave pública) 306, así como un hash de integridad del contenido de la imagen 307 y un hash de integridad de las direcciones de imagen 308. El componente de arranque 302 puede incluir, en una o más realizaciones, el código de arranque 309, que puede ser, o incluir, un fragmento de código ejecutable por el hipervisor, así como información por página 311, que incluye direcciones de página y valores de ajuste. Como se ha señalado, en una o más realizaciones, el hipervisor puede enviar las direcciones de página al ultravisor, que están ordenadas, tal como en orden ascendente o descendente, como un enfoque eficiente para ser capaz de evaluar que la dirección de página y el valor de ajuste son diferentes de las direcciones de página y valores de ajuste anteriores de la imagen del sistema operativo seguro. Solo a modo de ejemplo, las direcciones de página se describen en una o más realizaciones del presente documento como estando dispuestas en orden ascendente, tal como en el ejemplo de las FIGS. 4A-4C siguientes. Como se describe en el presente documento, los metadatos que contienen la página y las listas de ajustes pueden ser producidos por una entidad, tal como una herramienta de software, y esta entidad y el hipervisor tendrán un entendimiento común del orden de las direcciones y ajustes dentro de la lista. Más particularmente, en una o más implementaciones, el orden que importa es el orden en donde las páginas se cifran y codifican, y el orden en donde el hipervisor presenta las páginas al ultravisor, coincidiendo estos órdenes. El orden puede ser, como se ha señalado, ascendente o descendente, pero el ultravisor necesita saber qué dirección se utilizó al generar la imagen. Por lo tanto, la dirección normalmente la prescribirá la arquitectura de control de interfaz segura (o ultravisor).

Las FIGS. 4A-4C representan una realización del proceso de descompresión, que muestra estructuras de datos y/u operaciones del hipervisor y el ultravisor, conforme a un proceso de descifrado incremental y verificación de la integridad, de acuerdo con uno o más aspectos de la presente invención.

En la FIG. 4A, se representan las estructuras de datos relevantes del espacio de dirección de invitado 300, el hipervisor 400 y el ultravisor 410. Como se muestra, se hace una llamada al hipervisor 400 para ejecutar una imagen del sistema operativo seguro. El hipervisor 400 recibe el encabezado SE 301 y almacena temporalmente el encabezado para reenviarlo al ultravisor 410. El hipervisor llama al ultravisor 410 para iniciar el proceso de descompresión, y el ultravisor extrae y coloca dentro de su memoria protegida (por ejemplo, registra) la clave 306 de cifrado de imagen, el hash 307 del contenido de integridad y el hash 308 de la dirección de integridad del encabezado SE. Como se ilustra, el ultravisor 410 también mantiene, en una o más realizaciones, registros con un valor de ajuste actual 411, una dirección actual de la página que se va a descifrar 412, un hash de contenido actual 413 y un hash de dirección actual 414, así como un hash de contenido acumulativo 415, un hash de dirección acumulativo 416, un último valor de ajuste 417 y una última dirección de página 418.

A modo de ejemplo, el hash de contenido acumulativo 415 incluye un valor de hash sobre el contenido de las páginas de memoria que han sido descifradas por el ultravisor hasta el momento, el hash de dirección acumulativo 416 contiene un valor de hash sobre las direcciones de las páginas de memoria que se han descifrado hasta el momento, el último valor de ajuste 417 y la última dirección 418 contendrán los valores de ajuste y de dirección vistos por última vez por el ultravisor, que en una o más realizaciones descritas en el presente documento, permiten al ultravisor asegurar que las direcciones de página y los valores de ajuste no se reutilicen. El valor de ajuste, la dirección actual, el hash del contenido actual y el hash de dirección actual son registros de trabajo y se refieren a la página que el ultravisor 410 está procesando actualmente.

El hipervisor 400 recupera el control y construye una estructura de datos 401 que incluye, en una o más realizaciones, un par de dirección de página y valor de ajuste 402 para cada dirección de página de la imagen del sistema operativo seguro. El hipervisor 400 proporciona un índice 403 que se utiliza para iterar sobre las páginas de memoria de la imagen.

La FIG. 4B representa el espacio de dirección de invitado, las estructuras del hipervisor y ultravisor y el procesamiento de la FIG. 4A, con una operación de descompresión en proceso, donde el hipervisor 400 ha proporcionado el par dirección de página y valor de ajuste 402 al ultravisor 410. Este par actual se ha colocado en los registros de dirección actual y ajuste actual, y se muestra que se está realizando una comparación para asegurar que la dirección actual sea mayor que la última dirección y que el ajuste actual sea mayor que el último ajuste. Además, como parte del descifrado del contenido de la página particular en el espacio de dirección de invitado, el hash del contenido actual y el hash de la dirección actual se pueden añadir al hash de contenido acumulativo y al hash de dirección acumulativo que mantiene el ultravisor 410. Como se ilustra en la FIG. 4B, ciertas páginas se han descifrado 304'. Después del descifrado, estas páginas están, en una o más realizaciones, protegidas por el milicódigo del ultravisor 410, de modo que el hipervisor 400 no puede acceder a las páginas protegidas una vez descifradas. El ultravisor 410 podría utilizar, por ejemplo, una instalación de clave de almacenamiento para asegurar que el hipervisor no tenga acceso a las páginas no cifradas de la imagen del sistema operativo seguro.

La FIG. 4C representa las estructuras de datos y los procesos de las FIGS. 4A y 4B después de que el ultravisor haya descifrado todas las páginas de memoria de la imagen del sistema operativo seguro (así como protegidas por el ultravisor). El hipervisor 400 realiza una llamada final de descompresión al ultravisor para verificar la integridad de la

imagen del sistema operativo seguro empleando el hash de contenido acumulativo 415 y el hash de dirección acumulativo 416. Como se ilustra, se llama al ultravisor para comparar el hash de contenido de integridad recibido con el encabezado SE con el hash de contenido acumulativo comprobado, así como para comparar el hash de dirección de integridad del encabezado SE y el hash de dirección acumulativo obtenido. Suponiendo que las comparaciones respectivas son verdaderas, es decir, que el hash de contenido acumulativo es igual al hash del contenido de integridad y que el hash de dirección acumulativo es igual al hash de la dirección de integridad, se confirma la integridad de la imagen del sistema operativo seguro y el ultravisor puede iniciar la ejecución de la imagen descifrada del sistema operativo seguro utilizando, por ejemplo, una palabra de estado del programa (PSW) del encabezado SE.

Con referencia a las FIGS. 5A y 5B, se describen detalles adicionales de una o más realizaciones, en lo que se refiere a uno o más aspectos de la presente invención.

Haciendo referencia a la FIG. 5A, en una realización, una imagen de sistema operativo seguro se descifra de manera incremental (500), lo que incluye para una página de una pluralidad de páginas de la imagen de sistema operativo seguro: recibir una dirección de página de la página y un valor de ajuste utilizado durante el cifrado de la página (502); determinar que el valor de ajuste no se ha utilizado previamente durante el descifrado de otra página de la pluralidad de páginas de la imagen del sistema operativo seguro (504); y descifrar el contenido de la página de memoria en la dirección de la página utilizando una clave de cifrado de imagen utilizada en el cifrado de la página y el valor de ajuste para facilitar la obtención de una imagen descifrada del sistema operativo seguro (506). Además, se verifica la integridad de la imagen del sistema operativo seguro (508) y, en base a la verificación de la integridad de la imagen del sistema operativo seguro, se inicia la ejecución de la imagen descifrada del sistema operativo seguro (510).

En una o más realizaciones, el descifrado incremental y la verificación de la integridad se realizan mediante un control de interfaz segura del sistema informático, donde el control de interfaz segura es una entidad confiable del sistema informático, y la dirección de la página y el valor de ajuste se reciben por el control de interfaz segura desde un hipervisor (512). Además, en una realización, las direcciones de página de la pluralidad de direcciones de página están en un orden fijo (por ejemplo, de menor a mayor), y el descifrado incremental de la imagen del sistema operativo seguro incluye determinar que la dirección de la página es diferente a cualquier dirección de página anterior de páginas previamente descifradas de la pluralidad de páginas de la imagen del sistema operativo seguro (514).

En una o más implementaciones, el descifrado incremental de la imagen del sistema operativo seguro incluye además determinar que el valor de ajuste es diferente de cualquiera de los valores de ajuste previos utilizados durante el descifrado de las páginas previamente descifradas de la pluralidad de páginas de la imagen del sistema operativo seguro (516).

A modo de ejemplo adicional, la verificación de la integridad puede incluir emplear, en parte, un hash de contenido acumulativo obtenido utilizando el contenido de la página de memoria de la pluralidad de páginas y un hash de dirección acumulativo obtenido utilizando las direcciones de página de la pluralidad de páginas (518). Por ejemplo, como se muestra en la FIG. 5B, la verificación de la integridad de la imagen del sistema operativo seguro puede incluir comparar el hash de contenido acumulativo con un hash de contenido de integridad recibido por el control de interfaz segura desde el hipervisor, y comparar el hash de dirección acumulativo con un hash de dirección de integridad recibido por el control de interfaz segura desde el hipervisor (520).

En una o más realizaciones, el proceso incluye extraer, mediante el control de interfaz segura, la clave de cifrado de la imagen y el hash del contenido de integridad y el hash de la dirección de integridad de los metadatos recibidos desde el hipervisor (522). En un ejemplo específico, el control de interfaz segura incluye elementos de firmware del sistema informático, y el sistema operativo seguro debe ejecutarse como un sistema operativo invitado seguro dentro del entorno informático (524). A modo de ejemplo, se puede asociar un par de claves pública-privada al sistema informático, siendo la clave privada conocida solo por el control de interfaz segura y no accesible por el software que se ejecuta en el sistema informático, y donde el propietario de la imagen del sistema operativo seguro utiliza la clave pública para envolver la clave de cifrado de la imagen utilizada para cifrar la imagen del sistema operativo seguro y almacenar la clave de cifrado de la imagen envuelta en una estructura de metadatos reenviada al control de interfaz segura (526). Además, el control de interfaz segura puede recibir la estructura de metadatos con una llamada desde el hipervisor, incluyendo además la estructura de metadatos valores de hash de integridad utilizados en la verificación de la integridad de la imagen del sistema operativo seguro (528).

Son posibles otras variaciones y realizaciones.

Los aspectos de la presente invención pueden utilizarse por muchos tipos de entornos informáticos. Con referencia a la FIG. 6A se describe otra realización de un entorno informático para incorporar y utilizar uno o más aspectos de la presente invención. En este ejemplo, un entorno informático 10 incluye, por ejemplo, una unidad central de procesamiento (CPU) 12, una memoria 14, y uno o más dispositivos de entrada/salida y/o interfaces 16 acoplados entre sí mediante, por ejemplo, uno o más buses 18 y/u otras conexiones. Como ejemplos, el entorno informático 10 puede incluir un procesador PowerPC® ofrecido por International Business Machines Corporation, Armonk, Nueva York; un HP Superdome con procesadores Intel Itanium II ofrecidos por Hewlett Packard Co., Palo Alto, California; y/u otras máquinas basadas en arquitecturas ofrecidas por International Business Machines Corporation, Hewlett Packard, Intel Corporation, Oracle u otros. IBM, z/Architecture, IBM Z, z/OS, PR/SM y PowerPC son marcas comerciales o

marcas comerciales registradas de International Business Machines Corporation en al menos una jurisdicción. Intel e Itanium son marcas comerciales o marcas comerciales registradas de Intel Corporation o sus subsidiarias en los Estados Unidos y otros países.

La unidad central de procesamiento nativa 12 incluye uno o más registros nativos 20, tales como uno o más registros de propósito general y/o uno o más registros de propósito especial utilizados durante el procesamiento dentro del entorno. Estos registros incluyen información que representa el estado del entorno en cualquier instante particular de tiempo.

Además, la unidad central de procesamiento nativa 12 ejecuta instrucciones y código que están almacenados en la memoria 14. En un ejemplo particular, la unidad central de procesamiento ejecuta el código de emulador 22 almacenado en la memoria 14. Este código permite que el entorno informático configurado en una arquitectura emule otra arquitectura. Por ejemplo, el código de emulador 22 permite que máquinas basadas en arquitecturas diferentes de la arquitectura de hardware z/Architecture, tales como los procesadores PowerPC, los servidores HP Superdome u otros, emulen la arquitectura de hardware z/Architecture y ejecuten el software y las instrucciones desarrolladas en base a la arquitectura de hardware z/Architecture.

Con referencia a la FIG. 6B se describen detalles adicionales relacionados con el código de emulador 22. Las instrucciones de invitado 30 almacenadas en la memoria 14 comprenden instrucciones de software (por ejemplo, que se correlacionan con instrucciones de máquina) que fueron desarrolladas para ejecutarse en una arquitectura distinta a la de la CPU nativa 12. Por ejemplo, las instrucciones de invitado 30 pueden haber sido diseñadas para ejecutarse en un procesador basado en la arquitectura de hardware z/Architecture, pero en cambio, se están emulando en una CPU 12 nativa, que puede ser, por ejemplo, un procesador Intel Itanium II. En un ejemplo, el código de emulador 22 incluye una rutina de búsqueda de instrucciones 32 para obtener una o más instrucciones de invitado 30 de la memoria 14 y, opcionalmente, proporcionar almacenamiento en memoria intermedia local para las instrucciones obtenidas. También incluye una rutina de traducción de instrucciones 34 para determinar el tipo de instrucción de invitado que se ha obtenido y para traducir la instrucción de invitado en una o más instrucciones nativas 36 correspondientes. Esta traducción incluye, por ejemplo, identificar la función que realizará la instrucción de invitado y elegir la instrucción o instrucciones nativas para realizar esa función.

Además, el código de emulador 22 incluye una rutina de control de emulación 40 para provocar que se ejecuten las instrucciones nativas. La rutina de control de emulación 40 puede hacer que la CPU 12 nativa ejecute una rutina de instrucciones nativas que emulen una o más instrucciones de invitado obtenidas previamente y, al concluir dicha ejecución, devuelva el control a la rutina de búsqueda de instrucciones para emular la obtención de la siguiente instrucción de invitado o un grupo de instrucciones de invitado. La ejecución de las instrucciones nativas 36 puede incluir cargar datos en un registro desde la memoria 14; almacenar datos de vuelta a la memoria desde un registro; o realizar algún tipo de operación aritmética o lógica, según lo determinado por la rutina de traducción.

Cada rutina se implementa, por ejemplo, en un software, que se almacena en la memoria y se ejecuta mediante la unidad central de procesamiento nativa 12. En otros ejemplos, una o más de las rutinas u operaciones se implementan en firmware, hardware, software o alguna combinación de los mismos. Los registros del procesador emulado pueden emularse utilizando los registros 20 de la CPU nativa o utilizando ubicaciones en la memoria 14. En realizaciones, las instrucciones de invitado 30, las instrucciones nativas 36 y el código de emulador 22 pueden residir en la misma memoria o pueden distribuirse entre diferentes dispositivos de memoria.

Los entornos informáticos descritos anteriormente son solo ejemplos de entornos informáticos que se pueden utilizar. Se pueden utilizar otros entornos, incluidos, pero sin limitarse a, entornos no particionados, entornos particionados y/o entornos emulados; las realizaciones no se limitan a ningún entorno.

Cada entorno informático puede configurarse para incluir uno o más aspectos de la presente invención. Por ejemplo, cada uno puede configurarse para proporcionar un procesamiento de desbordamiento, de acuerdo con uno o más aspectos de la presente invención.

Uno o más aspectos pueden estar relacionados con la informática en la nube.

Debe entenderse que, aunque esta divulgación incluye una descripción detallada de la informática en la nube, la implementación de las enseñanzas mencionadas en el presente documento no se limita a un entorno informático en la nube. Más bien, las realizaciones de la presente invención son capaces de ser implementadas junto con cualquier otro tipo de entorno informático conocido ahora o desarrollado posteriormente.

La informática en la nube es un modelo de prestación de servicios para permitir un acceso de red conveniente bajo demanda a un conjunto compartido de recursos informáticos configurables (por ejemplo, redes, ancho de banda de red, servidores, procesamiento, memoria, almacenamiento, aplicaciones, máquinas virtuales y servicios) que se pueden aprovisionar y liberar rápidamente con un mínimo esfuerzo de gestión o interacción con un proveedor del servicio. Este modelo en la nube puede incluir al menos cinco características, al menos tres modelos de servicio y al menos cuatro modelos de implementación.

Las características son como sigue:

Autoservicio bajo demanda: un consumidor en la nube puede aprovisionar de forma unilateral capacidades informáticas, tales como la hora del servidor y el almacenamiento de red, según sea necesario de forma automática, sin requerir interacción humana con el proveedor del servicio.

Amplio acceso a la red: las capacidades están disponibles sobre una red y se accede a ellas a través de mecanismos estándar que promueven el uso por plataformas heterogéneas de clientes ligeros o pesados (por ejemplo, teléfonos móviles, ordenadores portátiles y PDA).

Agrupación de recursos: los recursos informáticos del proveedor se agrupan para servir a múltiples consumidores usando un modelo de múltiples inquilinos, con diferentes recursos físicos y virtuales asignados dinámicamente y reasignados según la demanda. Existe un sentido de independencia de ubicación en el sentido de que el consumidor generalmente no tiene control ni conocimiento sobre la ubicación exacta de los recursos proporcionados, pero puede ser capaz de especificar la ubicación en un nivel de abstracción más alto (por ejemplo, país, estado o centro de datos).

Elasticidad rápida: las capacidades se pueden aprovisionar rápida y elásticamente, en algunos casos de manera automática, para poner a escala rápidamente y liberar rápidamente para escalar rápidamente. Para el consumidor, las capacidades disponibles para el aprovisionamiento suelen parecer ilimitadas y pueden adquirirse en cualquier cantidad en cualquier momento.

Servicio medido: los sistemas en la nube controlan y optimizan automáticamente el uso de los recursos al aprovechar una capacidad de medición en algún nivel de abstracción apropiado para el tipo de servicio (por ejemplo, almacenamiento, procesamiento, ancho de banda y cuentas de usuario activas). El uso de recursos se puede monitorizar, controlar e informar, proporcionando transparencia tanto para el proveedor como para el consumidor del servicio utilizado.

Los modelos de servicio son como sigue:

Software como servicio (SaaS): la capacidad que se proporciona al consumidor es para utilizar las aplicaciones del proveedor que se ejecutan en una infraestructura en la nube. Las aplicaciones son accesibles desde varios dispositivos cliente a través de una interfaz de cliente ligero, tal como un navegador web (por ejemplo, correo electrónico basado en web). El consumidor no gestiona ni controla la infraestructura en la nube subyacente, que incluye la red, los servidores, los sistemas operativos, el almacenamiento o incluso las capacidades de aplicaciones individuales, con la posible excepción de ajustes limitados de configuración de aplicación específicos del usuario.

Plataforma como servicio (PaaS): la capacidad proporcionada al consumidor es para implementar en la infraestructura en la nube aplicaciones creadas o adquiridas por el consumidor creadas usando lenguajes de programación y herramientas soportadas por el proveedor. El consumidor no gestiona ni controla la infraestructura en la nube subyacente, incluyendo las redes, los servidores, los sistemas operativos o el almacenamiento, pero tiene control sobre las aplicaciones implementadas y, posiblemente, las configuraciones del entorno del alojamiento de aplicaciones.

Infraestructura como Servicio (IaaS): la capacidad proporcionada al consumidor es para aprovisionar procesamiento, almacenamiento, redes y otros recursos informáticos fundamentales donde el consumidor es capaz de implementar y ejecutar software arbitrario, que puede incluir sistemas operativos y aplicaciones. El consumidor no gestiona ni controla la infraestructura en la nube subyacente, pero tiene control sobre los sistemas operativos, el almacenamiento, las aplicaciones implementadas y, posiblemente, un control limitado de los componentes de interconexión de redes seleccionados (por ejemplo, los firewalls del anfitrión).

Los modelos de implementación son como sigue:

Nube privada: la infraestructura en la nube se opera únicamente para una organización. Se puede gestionar por la organización o por un tercero y puede existir en las instalaciones o fuera de las instalaciones.

Nube comunitaria: la infraestructura en la nube es compartida por varias organizaciones y soporta una comunidad específica que tiene intereses compartidos (por ejemplo, misión, requisitos de seguridad, políticas y consideraciones de cumplimiento). Se puede gestionar por las organizaciones o un tercero y puede existir en las instalaciones o fuera de las instalaciones.

Nube pública: la infraestructura en la nube se pone a disposición del público en general o de un gran grupo industrial y es propiedad de una organización que vende servicios en la nube.

Nube híbrida: la infraestructura en la nube es una composición de dos o más nubes (privada, comunitaria o pública) que siguen siendo entidades únicas, pero están unidas entre sí por una tecnología estandarizada o propietaria que permite la portabilidad de datos y aplicaciones (por ejemplo, la ráfaga en la nube para equilibrar la carga entre nubes).

Un entorno informático en la nube está orientado a servicio con un enfoque en ausencia de estado, bajo acoplamiento, modularidad e interoperabilidad semántica. En el corazón de la informática en la nube hay una infraestructura que incluye una red de nodos interconectados.

Haciendo referencia ahora a la FIG. 7, se representa un entorno informático en la nube 50 ilustrativo. Como se muestra, el entorno informático en la nube 50 incluye uno o más nodos informáticos en la nube 52 con los que pueden comunicarse los dispositivos informáticos locales utilizados por los consumidores en la nube, tales como, por ejemplo, un asistente digital personal (PDA) o un teléfono móvil 54A, un ordenador de sobremesa 54B, un ordenador portátil 54C y/o un sistema informático de automóvil 54N. Los nodos 52 pueden comunicarse unos con otros. Se pueden agrupar (no mostrado) física o virtualmente, en una o más redes, tales como nubes privadas, comunitarias, públicas o híbridas, como se ha descrito anteriormente, o una combinación de las mismas. Esto permite que el entorno informático en la nube 50 ofrezca infraestructura, plataformas y/o software como servicios para los que un consumidor en la nube no necesita mantener recursos en un dispositivo informático local. Se entiende que los tipos de dispositivos informáticos 54A-N mostrados en la FIG. 7 se pretende que sean únicamente ilustrativos y que los nodos informáticos 52 y el entorno informático en la nube 50 pueden comunicarse con cualquier tipo de dispositivo informatizado sobre cualquier tipo de red y/o conexión direccionable de red (por ejemplo, utilizando un navegador web).

Haciendo referencia ahora a la FIG. 8, se muestra un conjunto de capas de abstracción funcionales proporcionadas por el entorno informático en la nube 50 (FIG. 7). Se debe entender de antemano que los componentes, capas y funciones mostrados en la FIG. 8 se pretende que sean únicamente ilustrativos y las realizaciones de la invención no se limitan a los mismos. Como se representa, se proporcionan las siguientes capas y funciones correspondientes:

La capa de hardware y software 60 incluye componentes de hardware y software. Ejemplos de componentes de hardware incluyen: ordenadores centrales 61; servidores basados en la arquitectura RISC (por las siglas en inglés de Reduced Instruction Set Computer - Ordenador de Conjunto de Instrucciones Reducido) 62; servidores 63; servidores blade 64; dispositivos de almacenamiento 65; y redes y componentes de interconexión de redes 66. En algunas realizaciones, los componentes de software incluyen software de servidor de aplicaciones de red 67 y el software de base de datos 68.

La capa de virtualización 70 proporciona una capa de abstracción a partir de la que se pueden proporcionar los siguientes ejemplos de entidades virtuales: servidores virtuales 71; almacenamiento virtual 72; redes virtuales 73, incluyendo redes privadas virtuales; aplicaciones virtuales y sistemas operativos 74; y clientes virtuales 75.

En un ejemplo, la capa de gestión 80 puede proporcionar las funciones descritas a continuación. El aprovisionamiento de recursos 81 proporciona una adquisición dinámica de recursos informáticos y otros recursos que se utilizan para realizar tareas dentro del entorno informático en la nube. La Medición y Fijación de precios 82 proporcionan un seguimiento de los costes a medida que se utilizan los recursos dentro del entorno informático en la nube, y el cobro o facturación por el consumo de estos recursos. En un ejemplo, estos recursos pueden incluir licencias de software de aplicaciones. La seguridad proporciona verificación de identidad para los consumidores y las tareas en la nube, así como protección para los datos y otros recursos. El portal de usuario 83 proporciona acceso al entorno informático en la nube para consumidores y administradores de sistemas. La gestión de nivel de servicio 84 proporciona la asignación y gestión de recursos informáticos en la nube de manera que se cumplan los niveles de servicio requeridos. La planificación y el cumplimiento del Acuerdo de Nivel de Servicio (SLA) 85 proporcionan un acuerdo previo para, y la adquisición de, recursos informáticos en la nube para los que se anticipa un requisito futuro de acuerdo con un SLA.

La capa de cargas de trabajo 90 proporciona ejemplos de funcionalidad para lo cual se puede utilizar el entorno informático en la nube. Ejemplos de cargas de trabajo y funciones que se pueden proporcionar desde esta capa incluyen: mapeo y navegación 91; desarrollo de software y gestión del ciclo de vida 92; distribución de educación en el aula virtual 93; procesamiento de analíticas de datos 94; procesamiento de transacciones 95; y procesamiento de control de interfaz segura (ultravisor) 96.

Aspectos de la presente invención puede ser un sistema, un método y/o un producto de programa informático en cualquier posible nivel de detalle técnico de integración. El producto de programa informático puede incluir un medio (o medios) de almacenamiento legible por ordenador que tiene instrucciones de programa legibles por ordenador en el mismo para hacer que un procesador lleve a cabo aspectos de la presente invención.

El medio de almacenamiento legible por ordenador puede ser un dispositivo tangible que puede retener y almacenar instrucciones para su uso por un dispositivo de ejecución de instrucciones. El medio de almacenamiento legible por ordenador puede ser, por ejemplo, pero no se limita a, un dispositivo de almacenamiento electrónico, un dispositivo de almacenamiento magnético, un dispositivo de almacenamiento óptico, un dispositivo de almacenamiento electromagnético, un dispositivo de almacenamiento de semiconductores o cualquier combinación adecuada de los anteriores. Una lista no exhaustiva de ejemplos más específicos del medio de almacenamiento legible por ordenador incluye lo siguiente: un disquete de ordenador portátil, un disco duro, una memoria de acceso aleatorio (RAM), una memoria de solo lectura (ROM), una memoria de solo lectura programable y borrable (EPROM o memoria Flash), una memoria estática de acceso aleatorio (SRAM), una memoria portátil de solo lectura de disco compacto (CD-ROM), un disco versátil digital (DVD), una memoria USB, un disco flexible, un dispositivo codificado mecánicamente, tal como tarjetas perforadas o estructuras elevadas en una ranura que tiene instrucciones grabadas en las mismas, y cualquier combinación adecuada de los anteriores. Un medio de almacenamiento legible por ordenador, como se usa en el presente documento, no debe interpretarse como señales transitorias *en sí*, tales como ondas de radio u otras ondas electromagnéticas que se propagan libremente, ondas electromagnéticas que se propagan a través de una guía de ondas u otros medios de transmisión (por ejemplo, pulsos de luz que pasan a través de un cable de fibra óptica) o

señales eléctricas transmitidas a través de un cable.

Las instrucciones del programa legibles por ordenador descritas en el presente documento pueden descargarse a los respectivos dispositivos informáticos/de procesamiento desde un medio de almacenamiento legible por ordenador o a un ordenador externo o dispositivo de almacenamiento externo a través de una red, por ejemplo, Internet, una red de área local, una red de área amplia y/o una red inalámbrica. La red puede comprender cables de transmisión de cobre, fibras de transmisión óptica, transmisión inalámbrica, enrutadores, firewalls, conmutadores, ordenadores de puerta de enlace y/o servidores periféricos. Una tarjeta adaptadora de red o interfaz de red en cada dispositivo informático/de procesamiento recibe instrucciones de programa legibles por ordenador desde la red y envía las instrucciones de programa legibles por ordenador para su almacenamiento en un medio de almacenamiento legible por ordenador dentro del dispositivo informático/de procesamiento respectivo.

Las instrucciones de programa legibles por ordenador para llevar a cabo las operaciones de la presente invención pueden ser instrucciones de ensamblador, instrucciones de arquitectura de conjunto de instrucciones (ISA), instrucciones de máquina, instrucciones dependientes de la máquina, microcódigo, instrucciones de firmware, datos de ajuste de estado, datos de configuración para circuitería integrada o código fuente o código objeto escritos en cualquier combinación de uno o más lenguajes de programación, incluido un lenguaje de programación orientado a objetos tales como Smalltalk, C++ o similares, y lenguajes de programación procedimentales, tal como el lenguaje de programación "C" o lenguajes de programación similares. Las instrucciones de programa legibles por ordenador pueden ejecutarse completamente en el ordenador del usuario, parcialmente en el ordenador del usuario, como un paquete de software independiente, parcialmente en el ordenador del usuario y parcialmente en un ordenador remoto o completamente en el ordenador o servidor remoto. En este último escenario, el ordenador remoto puede conectarse al ordenador del usuario a través de cualquier tipo de red, incluyendo una red de área local (LAN) o una red de área amplia (WAN), o la conexión puede realizarse a un ordenador externo (por ejemplo, a través de Internet utilizando un Proveedor de Servicios de Internet). En algunas realizaciones, la circuitería electrónica que incluye, por ejemplo, circuitería lógica programable, matrices de puertas programables en campo (FPGA) o matrices lógicas programables (PLA) pueden ejecutar las instrucciones de programa legibles por ordenador mediante el uso de la información de estado de las instrucciones de programa legibles por ordenador para personalizar la circuitería electrónica, con el fin de realizar aspectos de la presente invención.

Aspectos de la presente invención se describen en el presente documento con referencia a ilustraciones de diagrama de flujo y/o diagramas de bloques de métodos, aparatos (sistemas) y productos de programas informáticos de acuerdo con las realizaciones de la invención. Se entenderá que cada bloque de las ilustraciones del diagrama de flujo y/o los diagramas de bloques, y las combinaciones de bloques en las ilustraciones del diagrama de flujo y/o los diagramas de bloques, pueden implementarse mediante instrucciones de programa legibles por ordenador.

Estas instrucciones de programa legibles por ordenador pueden proporcionarse a un procesador de un ordenador de propósito general, ordenador de propósito especial u otro aparato de procesamiento de datos programable para producir una máquina, de modo que las instrucciones, que se ejecutan a través del procesador del ordenador u otro aparato de procesamiento de datos programable, crear medios para implementar las funciones/actos especificados en el diagrama de flujo y/o bloque o bloques del diagrama de bloques. Estas instrucciones de programa legibles por ordenador también pueden almacenarse en un medio de almacenamiento legible por ordenador que puede dirigir un ordenador, un aparato de procesamiento de datos programable y/u otros dispositivos para que funcionen de una manera particular, de modo que el medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo comprende un artículo de fabricación que incluye instrucciones que implementan aspectos de la función/acto especificados en el diagrama de flujo y/o bloque o bloques del diagrama de bloques.

Las instrucciones del programa legibles por ordenador también pueden cargarse en un ordenador, otro aparato de procesamiento de datos programable u otro dispositivo para provocar que se realicen una serie de etapas operativas en el ordenador, otro aparato programable u otro dispositivo para producir un proceso implementado por ordenador, de modo que las instrucciones que se ejecutan en el ordenador, otro aparato programable u otro dispositivo implementan las funciones/actos especificados en el diagrama de flujo y/o bloque o bloques del diagrama de bloques.

El diagrama de flujo y los diagramas de bloques de las Figuras ilustran la arquitectura, la funcionalidad y la operación de posibles implementaciones de sistemas, métodos y productos de programas informáticos de acuerdo con diversas realizaciones de la presente invención. A este respecto, cada bloque en el diagrama de flujo o diagramas de bloques puede representar un módulo, segmento o porción de instrucciones, que comprende una o más instrucciones ejecutables para implementar la función o funciones lógicas especificadas. En algunas implementaciones alternativas, las funciones indicadas en los bloques pueden ocurrir fuera del orden indicado en las Figuras. Por ejemplo, dos bloques mostrados en sucesión pueden, de hecho, ejecutarse sustancialmente de manera simultánea, o los bloques pueden ejecutarse en ocasiones en el orden inverso, dependiendo de la funcionalidad implicada. También se observará que cada bloque de los diagramas de bloques y/o la ilustración del diagrama de flujo, y las combinaciones de bloques en los diagramas de bloques y/o la ilustración del diagrama de flujo, pueden implementarse mediante sistemas basados en hardware de propósito especial que realizan las funciones o actos especificados o llevan a cabo combinaciones de hardware de propósito especial e instrucciones de ordenador.

Además de lo anterior, un proveedor de servicios que ofrece la gestión de los entornos de cliente puede proporcionar,

ofrecer, implementar, gestionar, atender, etc., uno o más aspectos. Por ejemplo, el proveedor de servicios puede crear, mantener, soportar, etc. un código informático y/o una infraestructura informática que realice uno o más aspectos para uno o más clientes. A cambio, el proveedor de servicios puede recibir el pago del cliente bajo una suscripción y/o un acuerdo de tarifa, como ejemplos. Además, o alternativamente, el proveedor de servicios puede recibir el pago de la venta de contenido publicitario a uno o más terceros.

En un aspecto, se puede implementar una aplicación para realizar una o más realizaciones. Como ejemplo, la implementación de una aplicación comprende proporcionar una infraestructura informática operable para realizar una o más realizaciones.

Como aspecto adicional, se puede implementar una infraestructura informática que comprenda integrar un código legible por ordenador en un sistema informático, en donde el código en combinación con el sistema informático es capaz de realizar una o más realizaciones.

Como un aspecto adicional más, se puede proporcionar un proceso para integrar una infraestructura informática que comprende integrar un código legible por ordenador en un sistema informático. El sistema informático comprende un medio legible por ordenador, en donde el medio informático comprende una o más realizaciones. El código en combinación con el sistema informático es capaz de realizar una o más realizaciones.

Aunque anteriormente se han descrito varias realizaciones, estas son solamente ejemplos. Por ejemplo, se pueden usar entornos informáticos de otras arquitecturas para incorporar y usar una o más realizaciones. Además, se pueden usar diferentes instrucciones u operaciones. Adicionalmente, se pueden especificar diferentes tipos de indicadores. Son posibles muchas variaciones.

Además, otros tipos de entornos informáticos pueden beneficiarse y utilizarse. Como ejemplo, se puede utilizar un sistema de procesamiento de datos adecuado para almacenar y/o ejecutar código de programa que incluya al menos dos procesadores acoplados directa o indirectamente a elementos de memoria a través de un bus de sistema. Los elementos de memoria incluyen, por ejemplo, una memoria local empleada durante la ejecución real del código del programa, un almacenamiento masivo y una memoria caché que proporciona almacenamiento temporal de al menos parte del código del programa con el fin de reducir el número de veces que el código debe recuperarse del almacenamiento masivo durante la ejecución.

Los dispositivos de Entrada/Salida o E/S (incluyendo, pero sin limitarse a, teclados, pantallas, dispositivos señaladores, DASD, cintas, CD, DVD, memorias USB y otros medios de memoria, etc.) se pueden acoplar al sistema o bien directamente o a través de controladores de E/S intervinientes. Los adaptadores de red también se pueden acoplar al sistema para permitir que el sistema de procesamiento de datos llegue a acoplarse a otros sistemas de procesamiento de datos o impresoras remotas o dispositivos de almacenamiento a través de redes privadas o públicas intervinientes. Los módems, módems de cable y tarjetas Ethernet son solo algunos de los tipos de adaptadores de red disponibles.

La terminología utilizada en el presente documento tiene el propósito de describir las realizaciones particulares solamente y no se pretende que sea limitante. Como se usa en el presente documento, las formas singulares “un”, “una” y “el”, “la” se pretende que incluyan también las formas plurales, a menos que el contexto lo indique claramente de otro modo. Se entenderá además que los términos “comprende” y/o “que comprende”, cuando se usan en esta memoria descriptiva, especifican la presencia de características, números enteros, etapas, operaciones, elementos y/o componentes indicados, pero no excluyen la presencia o adición de una o más características, números enteros, etapas, operaciones, elementos, componentes y/o grupos de los mismos.

Las correspondientes estructuras, materiales, actos y equivalentes de todos los medios o etapas más elementos de función en las reivindicaciones siguientes, si los hay, se pretende que incluyan cualquier estructura, material o acto para realizar la función en combinación con otros elementos reivindicados, como se reivindica específicamente. La descripción de una o más realizaciones se ha presentado con propósitos de ilustración y descripción, pero no se pretende que sea exhaustiva ni esté limitada a la forma divulgada. Muchas modificaciones y variaciones resultarán evidentes para los expertos en la técnica. La realización se eligió y describió con el fin de explicar mejor diversos aspectos y la aplicación práctica, y para permitir que otros expertos en la técnica entiendas diversas realizaciones con diversas modificaciones que sean adecuadas para el uso particular contemplado.

REIVINDICACIONES

1. Un sistema informático (102) para facilitar el procesamiento seguro dentro de un entorno informático (100), comprendiendo el sistema informático:

una memoria (104);

un procesador (110) acoplado a la memoria (104), en donde el sistema informático está configurado para realizar un método que comprende:

descifrar de manera incremental una imagen de sistema operativo seguro, mediante un control de interfaz segura (115, 410) del sistema informático (102), siendo el control de interfaz segura (115, 410) una entidad confiable del sistema informático (102), incluyendo el descifrado incremental para una página de una pluralidad de páginas de la imagen del sistema operativo seguro:

recibir, mediante el control de interfaz segura (115, 410) desde un hipervisor (114, 400), una dirección de página de la página y un valor de ajuste utilizado durante el cifrado de la página, en donde las páginas y direcciones de página de la pluralidad de páginas se proporcionan en un orden fijo para facilitar el descifrado incremental;

colocar la dirección de página de la página en un registro de dirección actual (412) y el valor de ajuste utilizado durante el cifrado de la página en un registro de ajuste actual (411);

determinar que la dirección de la página es diferente de cualquier dirección de página anterior de las páginas previamente descifradas de la pluralidad de páginas de la imagen del sistema operativo seguro, la determinación de que la dirección de la página es diferente, incluyendo la comparación del registro de dirección actual (412) con un último registro de dirección (418) que contiene una dirección de página de una última página descifrada de manera incremental;

determinar que el valor de ajuste no se ha utilizado previamente durante el descifrado de otra página de la pluralidad de páginas de la imagen del sistema operativo seguro, la determinación de que el valor de ajuste no se ha utilizado previamente durante el descifrado de otra página comprende comparar el registro de ajuste actual (411) con un último registro de ajuste (417) que contiene un valor de ajuste de una última página de la pluralidad de páginas descifradas de manera incremental; y

descifrar el contenido de la página de memoria en la dirección de la página utilizando una clave de cifrado de imagen utilizada para cifrar la página y el valor de ajuste para facilitar la obtención de una imagen descifrada del sistema operativo seguro;

verificar, mediante el control de interfaz segura (115, 410), la integridad de la imagen del sistema operativo seguro; y

en base a la verificación de la integridad de la imagen del sistema operativo seguro, iniciar la ejecución de la imagen descifrada del sistema operativo seguro.

2. El sistema informático de la reivindicación 1, en donde las direcciones de página de la pluralidad de páginas no son contiguas y están en el orden fijo.

3. El sistema informático de una cualquiera de las reivindicaciones anteriores, en donde la verificación de la integridad incluye emplear, en parte, un hash de contenido acumulativo (415) obtenido utilizando el contenido de la página de memoria de la pluralidad de páginas y un hash de dirección acumulativo (416) obtenido utilizando las direcciones de página de la pluralidad de páginas.

4. El sistema informático de la reivindicación 3, en donde la verificación de la integridad de la imagen del sistema operativo seguro incluye comparar el hash de contenido acumulativo (415) con un hash de contenido de integridad (307) recibido por el control de interfaz segura (115, 410) y comparar el hash de dirección acumulativo (416) con un hash de dirección de integridad (308) recibido por el control de interfaz segura.

5. El sistema informático de la reivindicación 4, que comprende además extraer, mediante el control de interfaz segura (114, 410), la clave de cifrado de imagen (306) y el hash de contenido de integridad (308) y el hash de dirección de integridad de los metadatos recibidos del hipervisor (114, 400).

6. El sistema informático de una cualquiera de las reivindicaciones anteriores, en donde el control de interfaz segura (115, 410) comprende uno o más elementos del sistema informático (102), seleccionándose el uno o más elementos del grupo que consiste en elementos de hardware y elementos de firmware, y la imagen del sistema operativo seguro debe ejecutarse como un sistema operativo invitado seguro dentro del entorno informático (102).

7. El sistema informático de una cualquiera de las reivindicaciones anteriores, en donde un par de claves pública-privada está asociado al sistema informático (102), siendo la clave privada conocida solamente por el control de interfaz segura (115, 410) y no accesible por el software que se ejecuta en el sistema informático (102), y en donde el

propietario de la imagen del sistema operativo seguro utiliza la clave pública para generar datos del acuerdo de claves para comunicar de forma segura la clave de cifrado de imagen (306) utilizada para cifrar la imagen del sistema operativo seguro con el control de interfaz segura (115, 410) y almacenar los datos del acuerdo de claves en una estructura de metadatos (301).

8. El sistema informático de la reivindicación 7, que comprende además recibir mediante el control de interfaz segura (115, 410) la estructura de metadatos (301) con una llamada desde el hipervisor (114, 400), comprendiendo además la estructura de metadatos (301) valores de hash de integridad utilizados en la verificación de la integridad de la imagen del sistema operativo seguro.

9. Un producto de programa informático que comprende:

un medio de almacenamiento legible por ordenador legible por un circuito de procesamiento y que almacena instrucciones para realizar un método que comprende:

descifrar de manera incremental una imagen de sistema operativo seguro, mediante un control de interfaz segura (115, 410) del sistema informático (102), siendo el control de interfaz segura (115, 410) una entidad confiable del sistema informático (102), incluyendo el descifrado incremental para una página de una pluralidad de páginas de la imagen del sistema operativo seguro:

recibir, mediante el control de interfaz segura (115, 410) desde un hipervisor (114, 400), una dirección de página de la página y un valor de ajuste utilizado durante el cifrado de la página, en donde las páginas y direcciones de página de la pluralidad de páginas se proporcionan en un orden fijo para facilitar el descifrado incremental;

colocar la dirección de página de la página en un registro de dirección actual (412) y el valor de ajuste utilizado durante el cifrado de la página en un registro de ajuste actual (411);

determinar que la dirección de la página es diferente de cualquier dirección de página anterior de las páginas previamente descifradas de la pluralidad de páginas de la imagen del sistema operativo seguro, la determinación de que la dirección de la página es diferente, incluyendo comparar el registro de dirección actual (412) con un último registro de dirección (418) que contiene una dirección de página de una última página descifrada de manera incremental;

determinar que el valor de ajuste no se ha utilizado previamente durante el descifrado de otra página de la pluralidad de páginas de la imagen del sistema operativo seguro, la determinación de que el valor de ajuste no se ha utilizado previamente durante el descifrado de otra página comprende comparar el registro de ajuste actual (411) con un último registro de ajuste (417) que contiene un valor de ajuste de una última página de la pluralidad de páginas descifradas de manera incremental; y

descifrar el contenido de la página de memoria en la dirección de la página utilizando una clave de cifrado de imagen utilizada para cifrar la página y el valor de ajuste para facilitar la obtención de una imagen descifrada del sistema operativo seguro;

verificar, mediante el control de interfaz segura (115, 410), la integridad de la imagen del sistema operativo seguro; y

en base a la verificación de la integridad de la imagen del sistema operativo seguro, iniciar la ejecución de la imagen descifrada del sistema operativo seguro.

10. El producto de programa informático de la reivindicación 9, en donde las direcciones de página de la pluralidad de páginas no son contiguas y están en el orden fijo.

11. El producto de programa informático de la reivindicación 10, en donde la verificación de la integridad incluye emplear, en parte, un hash de contenido acumulativo obtenido utilizando el contenido de la página de memoria de la pluralidad de páginas y un hash de dirección acumulativo obtenido utilizando las direcciones de página de la pluralidad de páginas.

12. El producto de programa informático de la reivindicación 11, en donde la verificación de la integridad de la imagen del sistema operativo seguro incluye comparar el hash de contenido acumulativo (415) con un hash de contenido de integridad (307) recibido por el control de interfaz segura (115, 410).

13. El producto de programa informático de una cualquiera de las reivindicaciones 11 a 12, en donde la verificación de la integridad de la imagen del sistema operativo seguro incluye además comparar el hash de dirección acumulativo (416) con un hash de dirección de integridad (308) recibido por el control de interfaz segura (115, 410).

14. El producto de programa informático de la reivindicación 13, que comprende además extraer la clave de cifrado de imagen (306) y el hash de contenido de integridad (307) y el hash de dirección de integridad (308) de los metadatos recibidos desde un hipervisor.

15. Un método implementado por ordenador, que comprende:

descifrar de manera incremental una imagen de sistema operativo seguro, mediante un control de interfaz segura (115, 410) del sistema informático (102), siendo el control de interfaz segura (115, 410) una entidad confiable del sistema informático (102), incluyendo el descifrado incremental para una página de una pluralidad de páginas de la imagen del sistema operativo seguro:

recibir, mediante el control de interfaz segura (115, 410) desde un hipervisor (114, 400), una dirección de página de la página y un valor de ajuste utilizado durante el cifrado de la página, en donde las páginas y direcciones de página de la pluralidad de páginas se proporcionan en un orden fijo para facilitar el descifrado incremental;

colocar la dirección de página de la página en un registro de dirección actual (412) y el valor de ajuste utilizado durante el cifrado de la página en un registro de ajuste actual (411);

determinar que la dirección de la página es diferente de cualquier dirección de página anterior de páginas previamente descifradas de la pluralidad de páginas de la imagen del sistema operativo seguro, la determinación de que la dirección de la página es diferente, incluyendo comparar

el registro de dirección actual (412) a un último registro de direcciones (418) que contiene una dirección de página de una última página descifrada de manera incremental;

determinar que el valor de ajuste no se ha utilizado previamente durante el descifrado de otra página de la pluralidad de páginas de la imagen del sistema operativo seguro, la determinación de que el valor de ajuste no se ha utilizado previamente durante el descifrado de otra página comprende comparar el registro de ajuste actual (411) con un último registro de ajuste (417) que contiene un valor de ajuste de una última página de la pluralidad de páginas descifradas de manera incremental; y

descifrar el contenido de la página de memoria en la dirección de la página utilizando una clave de cifrado de imagen utilizada para cifrar la página y el valor de ajuste para facilitar la obtención de una imagen descifrada del sistema operativo seguro;

verificar, mediante el control de interfaz segura (115, 410), la integridad de la imagen del sistema operativo seguro; y

en base a la verificación de la integridad de la imagen del sistema operativo seguro, iniciar la ejecución de la imagen descifrada del sistema operativo seguro.

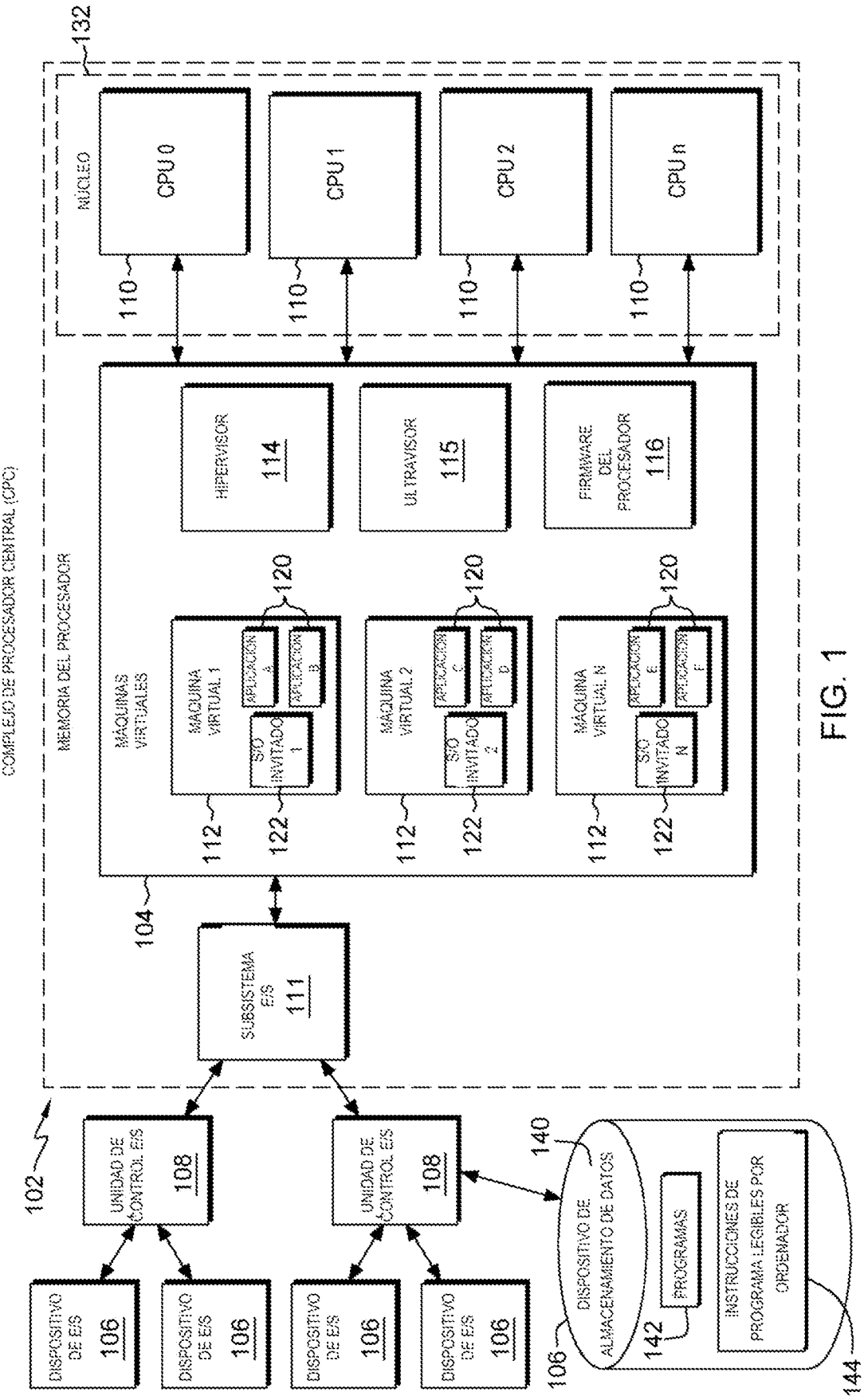
100
COMPLEJO DE PROCESADOR CENTRAL (CPC)
MEMORIA DEL PROCESADOR
132
NÚCLEO
CPU 0
CPU 1
CPU 2
CPU n
110
MÁQUINAS VIRTUALES
HIPERVISOR 114
ULTRAVISOR 115
FIRMWARE DEL PROCESADOR 116
MÁQUINA VIRTUAL 1
MÁQUINA VIRTUAL 2
MÁQUINA VIRTUAL N
APLICACIÓN A
APLICACIÓN B
APLICACIÓN C
APLICACIÓN D
APLICACIÓN E
APLICACIÓN F
S/O INVITADO 1
S/O INVITADO 2
S/O INVITADO N
120
112
122
104
SUBSISTEMA E/S 111
102
UNIDAD DE CONTROL E/S 108
DISPOSITIVO DE E/S 106
140
DISPOSITIVO DE ALMACENAMIENTO DE DATOS
PROGRAMAS
142
INSTRUCCIONES DE PROGRAMA LEGIBLES POR ORDENADOR
144

FIG. 1

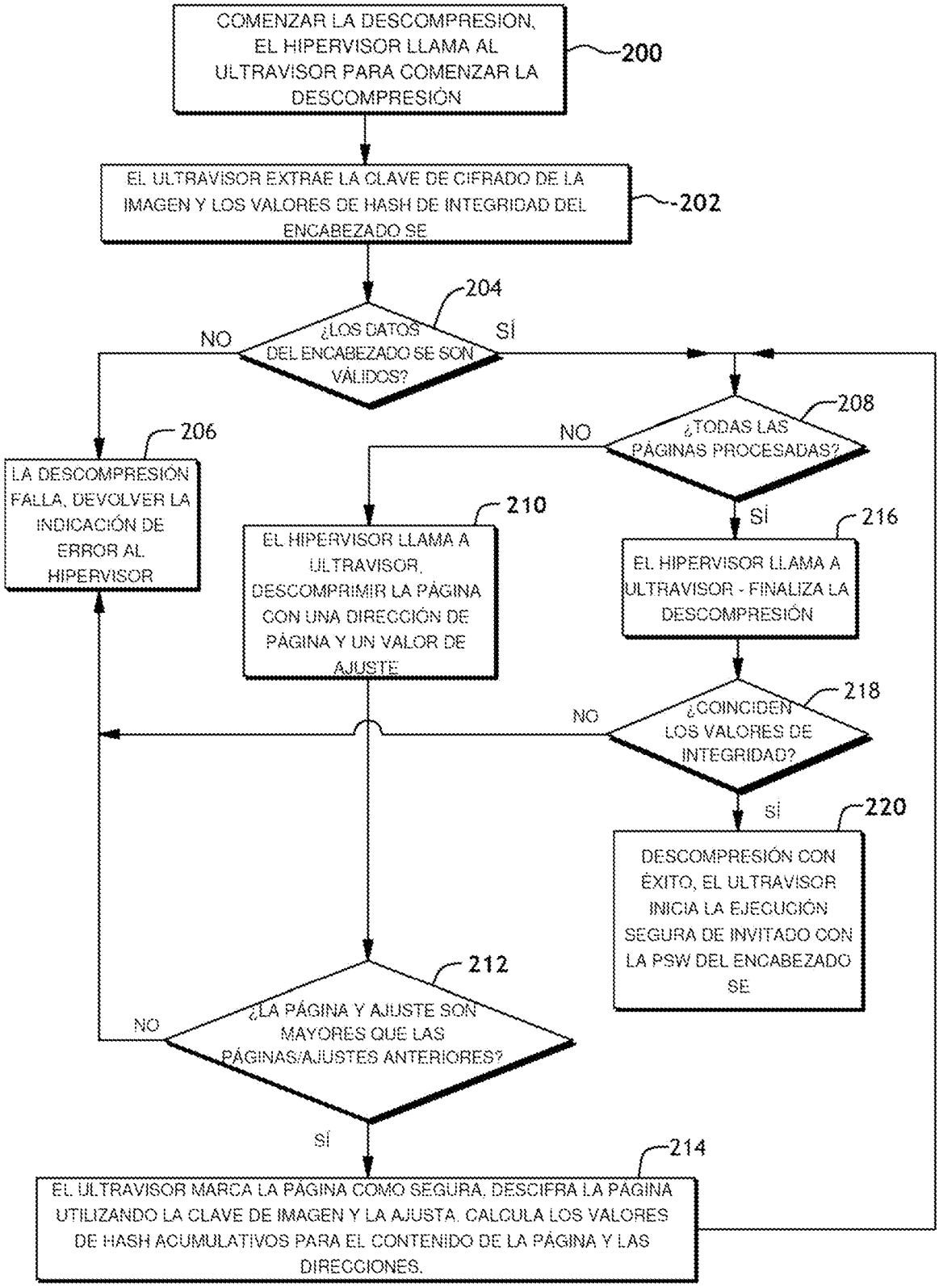
COMENZAR LA DESCOMPRESIÓN, EL HIPERVISOR LLAMA AL ULTRAVISOR PARA COMENZAR LA DESCOMPRESIÓN
200
EL ULTRAVISOR EXTRAE LA CLAVE DE CIFRADO DE LA IMAGEN Y LOS VALORES DE HASH DE INTEGRIDAD DEL ENCABEZADO SE
202
204
¿LOS DATOS DEL ENCABEZADO SE SON VÁLIDOS?
NO
SÍ
206
LA DESCOMPRESIÓN FALLA, DEVOLVER LA INDICACIÓN DE ERROR AL HIPERVISOR
208
¿TODAS LAS PÁGINAS PROCESADAS?
NO
SÍ
210
EL HIPERVISOR LLAMA A ULTRAVISOR, DESCOMPRIMIR LA PÁGINA CON UNA DIRECCIÓN DE PÁGINA Y UN VALOR DE AJUSTE
216
EL HIPERVISOR LLAMA A ULTRAVISOR - FINALIZA LA DESCOMPRESIÓN
218
¿COINCIDEN LOS VALORES DE INTEGRIDAD?
NO
SÍ
220
DESCOMPRESIÓN CON ÉXITO, EL ULTRAVISOR INICIA LA EJECUCIÓN SEGURA DE INVITADO CON LA PSW DEL ENCABEZADO SE
212
¿LA PÁGINA Y AJUSTE SON MAYORES QUE LAS PÁGINAS/AJUSTES ANTERIORES?
NO
SÍ
214
EL ULTRAVISOR MARCA LA PÁGINA COMO SEGURA, DESCIFRA LA PÁGINA UTILIZANDO LA CLAVE DE IMAGEN Y LA AJUSTA, CALCULA LOS VALORES DE HASH ACUMULATIVOS PARA EL CONTENIDO DE LA PÁGINA Y LAS DIRECCIONES.

FIG. 2

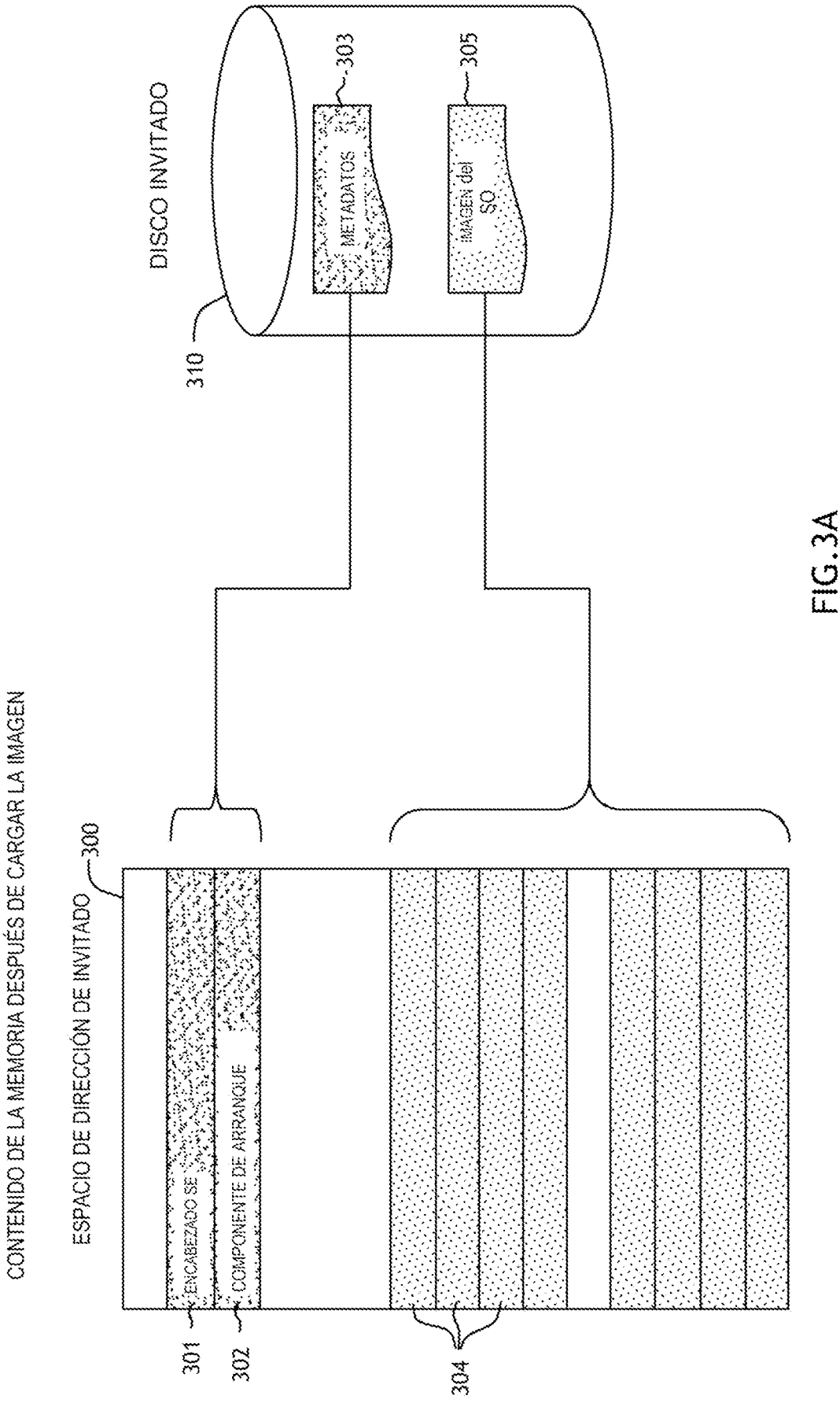
CONTENIDO DE LA MEMORIA DESPUÉS DE CARGAR LA IMAGEN
ESPACIO DE DIRECCIÓN DE INVITADO
300
ENCABEZADO SE
COMPONENTE DE ARRANQUE
301
302
304
DISCO INVITADO
310
303
METADATOS
305
IMAGEN del SO

FIG. 3A

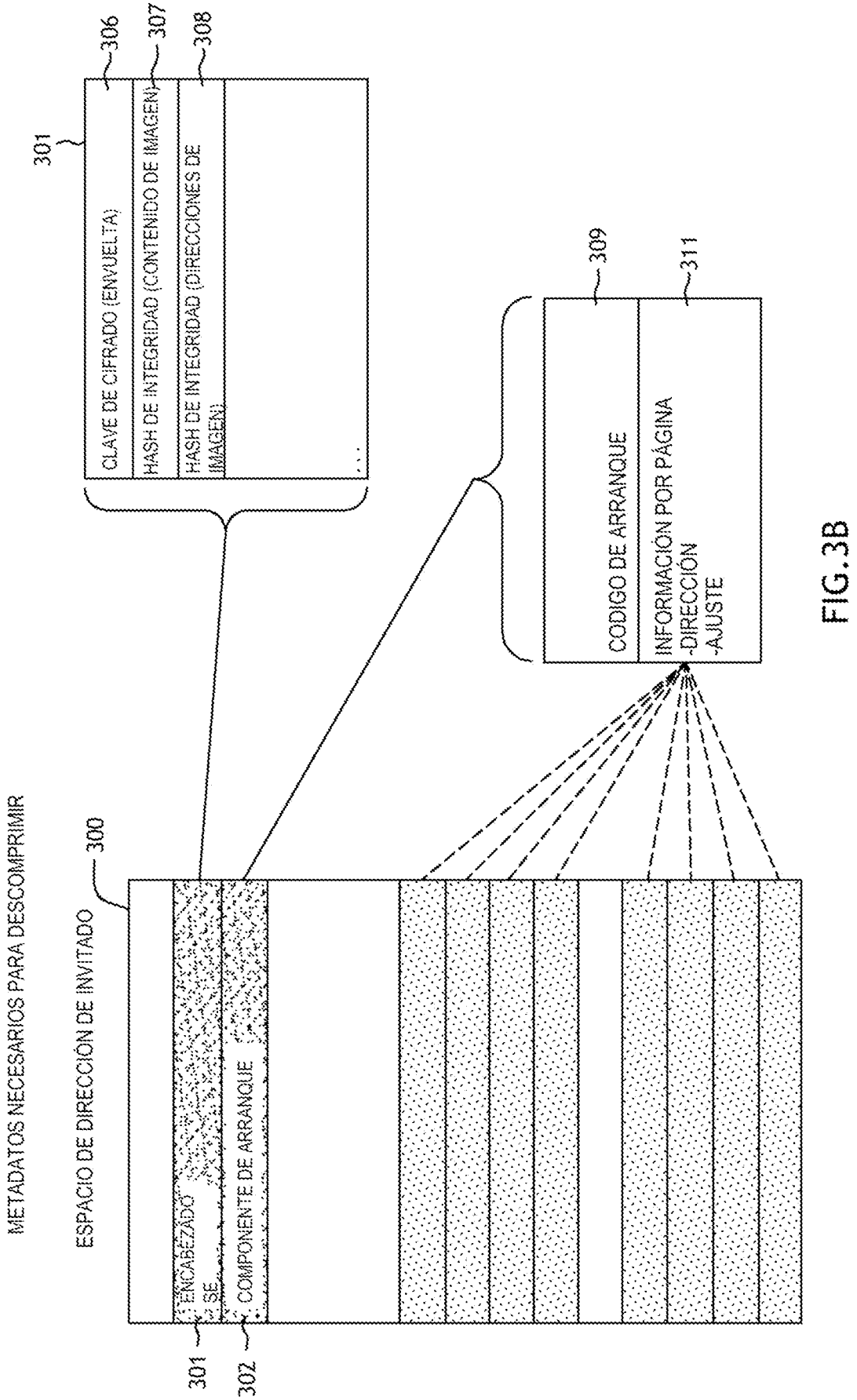
METADATOS NECESARIOS PARA DESCOMPRIMIR
ESPACIO DE DIRECCIÓN DE INVITADO
300
ENCABEZADO SE
COMPONENTE DE ARRANQUE
301
302
301
306
307
308
CLAVE DE CIFRADO (ENVUELTA)
HASH DE INTEGRIDAD (CONTENIDO DE IMAGEN)
HASH DE INTEGRIDAD (DIRECCIONES DE IMAGEN)
...
309
311
CODIGO DE ARRANQUE
INFORMACIÓN POR PÁGINA
-DIRECCIÓN
-AJUSTE

FIG. 3B

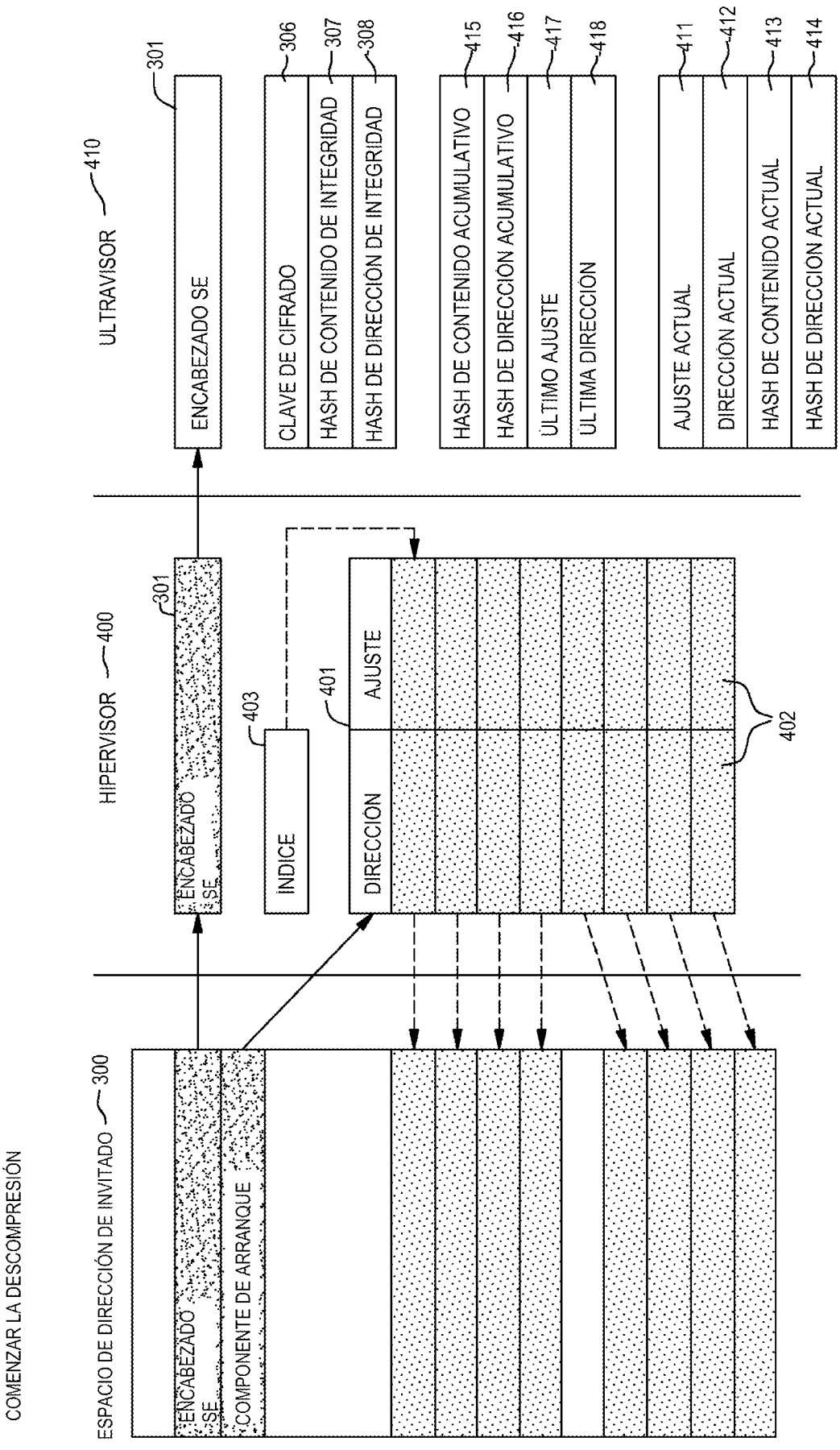
COMENZAR LA DESCOMPRESIÓN
ESPACIO DE DIRECCIÓN DE INVITADO
300
ENCABEZADO SE
COMPONENTE DE ARRANQUE
HIPERVISOR
400
301
ENCABEZADO SE
403
ÍNDICE
401
AJUSTE
DIRECCIÓN
402
ULTRAVISOR
410
301
ENCABEZADO SE
306
CLAVE DE CIFRADO
307
HASH DE CONTENIDO DE INTEGRIDAD
308
HASH DE DIRECCIÓN DE INTEGRIDAD
415
HASH DE CONTENIDO ACUMULATIVO
416
HASH DE DIRECCIÓN ACUMULATIVO
417
ÚLTIMO AJUSTE
418
ÚLTIMA DIRECCIÓN
411
AJUSTE ACTUAL
412
DIRECCIÓN ACTUAL
413
HASH DE CONTENIDO ACTUAL
414
HASH DE DIRECCIÓN ACTUAL

FIG. 4A

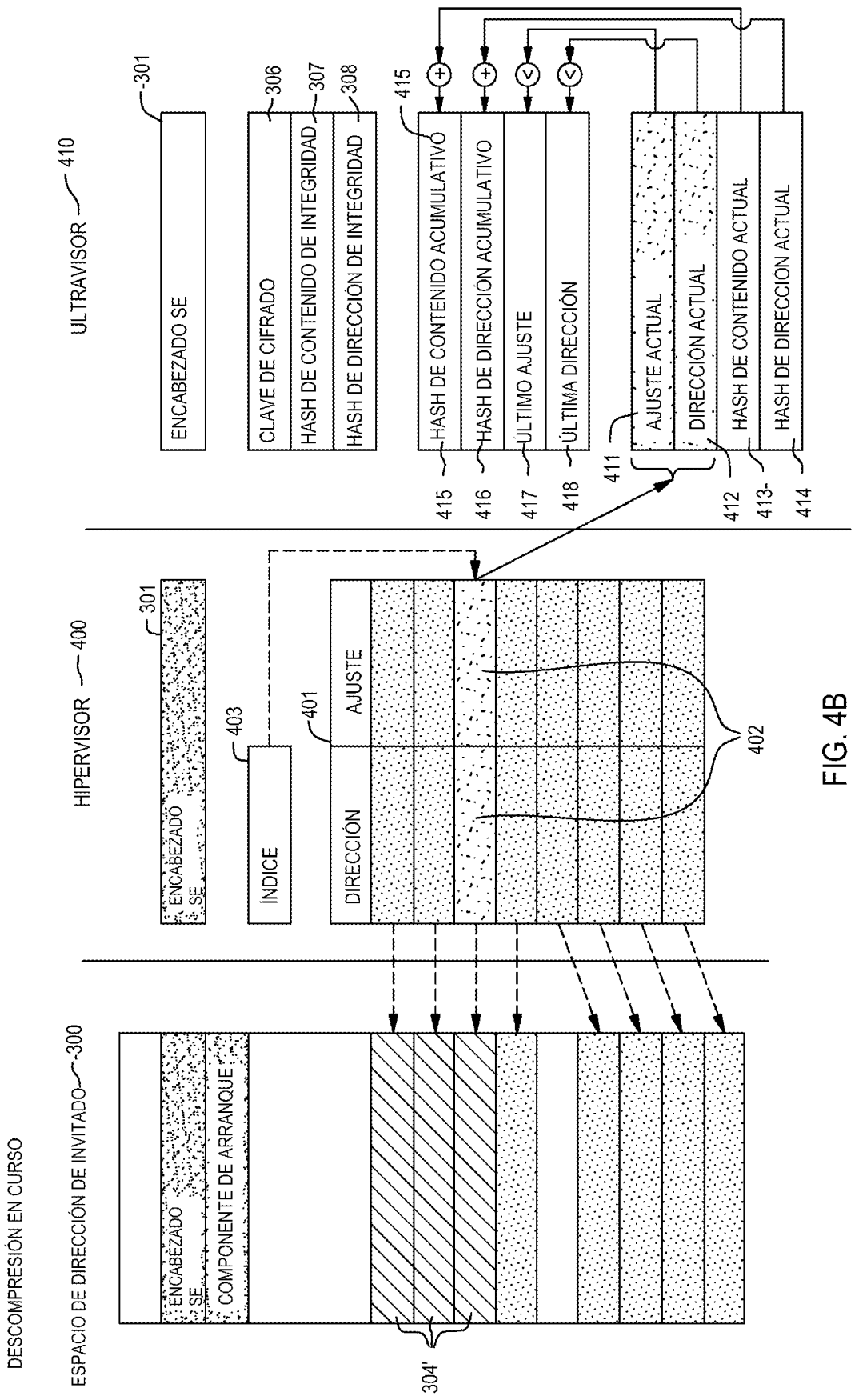
DESCOMPRESIÓN EN CURSO
ESPACIO DE DIRECCIÓN DE INVITADO
300
ENCABEZADO SE
COMPONENTE DE ARRANQUE
304'
HIPERVISOR
400
301
ENCABEZADO SE
403
ÍNDICE
401
AJUSTE
DIRECCIÓN
402
ULTRAVISOR
410
301
ENCABEZADO SE
306
CLAVE DE CIFRADO
307
HASH DE CONTENIDO DE INTEGRIDAD
308
HASH DE DIRECCIÓN DE INTEGRIDAD
415
HASH DE CONTENIDO ACUMULATIVO
HASH DE DIRECCIÓN ACUMULATIVO
ÚLTIMO AJUSTE
ÚLTIMA DIRECCIÓN
AJUSTE ACTUAL
DIRECCIÓN ACTUAL
HASH DE CONTENIDO ACTUAL
HASH DE DIRECCIÓN ACTUAL
415
416
417
418
411
412
413
414

FIG. 4B

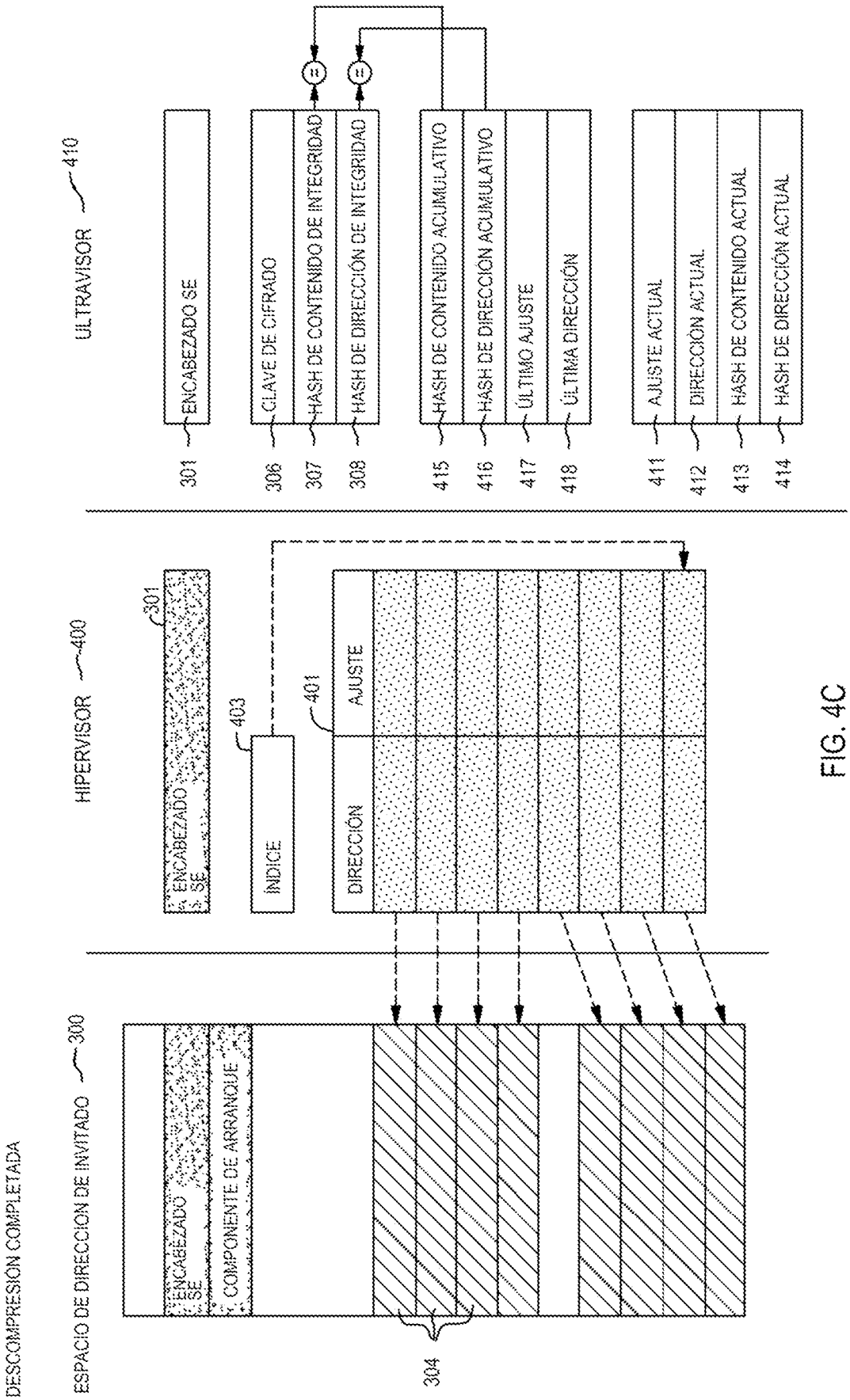
DESCOMPRESIÓN COMPLETADA
ESPACIO DE DIRECCIÓN DE INVITADO
300
ENCABEZADO SE
COMPONENTE DE ARRANQUE
304
HIPERVISOR
400
301
ENCABEZADO SE
403
ÍNDICE
401
DIRECCIÓN
AJUSTE
ULTRAVISOR
410
301
ENCABEZADO SE
306
CLAVE DE CIFRADO
307
HASH DE CONTENIDO DE INTEGRIDAD
308
HASH DE DIRECCIÓN DE INTEGRIDAD
415
HASH DE CONTENIDO ACUMULATIVO
416
HASH DE DIRECCIÓN ACUMULATIVO
417
ÚLTIMO AJUSTE
418
ÚLTIMA DIRECCIÓN
411
AJUSTE ACTUAL
412
DIRECCIÓN ACTUAL
413
HASH DE CONTENIDO ACTUAL
414
HASH DE DIRECCIÓN ACTUAL

FIG. 4C

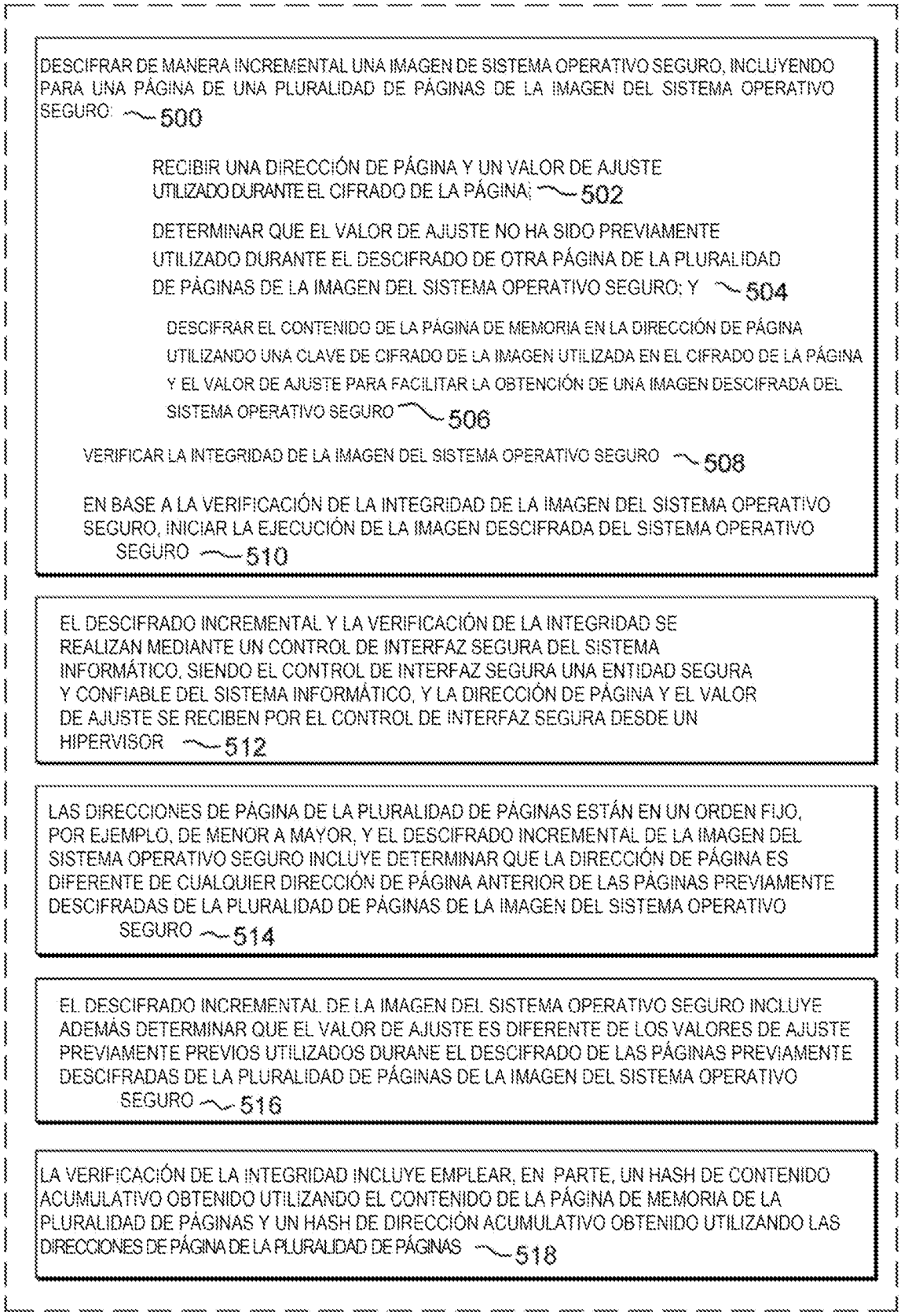
DESCIFRAR DE MANERA INCREMENTAL UNA IMAGEN DE SISTEMA OPERATIVO SEGURO, INCLUYENDO PARA UNA PÁGINA DE UNA PLURALIDAD DE PÁGINAS DE LA IMAGEN DEL SISTEMA OPERATIVO SEGURO: 500
RECIBIR UNA DIRECCIÓN DE PÁGINA Y UN VALOR DE AJUSTE UTILIZADO DURANTE EL CIFRADO DE LA PÁGINA; 502
DETERMINAR QUE EL VALOR DE AJUSTE NO HA SIDO PREVIAMENTE UTILIZADO DURANTE EL DESCIFRADO DE OTRA PÁGINA DE LA PLURALIDAD DE PÁGINAS DE LA IMAGEN DEL SISTEMA OPERATIVO SEGURO; Y 504
DESCIFRAR EL CONTENIDO DE LA PÁGINA DE MEMORIA EN LA DIRECCIÓN DE PÁGINA UTILIZANDO UNA CLAVE DE CIFRADO DE LA IMAGEN UTILIZADA EN EL CIFRADO DE LA PÁGINA Y EL VALOR DE AJUSTE PARA FACILITAR LA OBTENCIÓN DE UNA IMAGEN DESCIFRADA DEL SISTEMA OPERATIVO SEGURO 506
VERIFICAR LA INTEGRIDAD DE LA IMAGEN DEL SISTEMA OPERATIVO SEGURO 508
EN BASE A LA VERIFICACIÓN DE LA INTEGRIDAD DE LA IMAGEN DEL SISTEMA OPERATIVO SEGURO, INICIAR LA EJECUCIÓN DE LA IMAGEN DESCIFRADA DEL SISTEMA OPERATIVO SEGURO 510
EL DESCIFRADO INCREMENTAL Y LA VERIFICACIÓN DE LA INTEGRIDAD SE REALIZAN MEDIANTE UN CONTROL DE INTERFAZ SEGURA DEL SISTEMA INFORMÁTICO, SIENDO EL CONTROL DE INTERFAZ SEGURA UNA ENTIDAD SEGURA Y CONFIABLE DEL SISTEMA INFORMÁTICO, Y LA DIRECCIÓN DE PÁGINA Y EL VALOR DE AJUSTE SE RECIBEN POR EL CONTROL DE INTERFAZ SEGURA DESDE UN HIPERVISOR 512
LAS DIRECCIONES DE PÁGINA DE LA PLURALIDAD DE PÁGINAS ESTÁN EN UN ORDEN FIJO, POR EJEMPLO, DE MENOR A MAYOR, Y EL DESCIFRADO INCREMENTAL DE LA IMAGEN DEL SISTEMA OPERATIVO SEGURO INCLUYE DETERMINAR QUE LA DIRECCIÓN DE PÁGINA ES DIFERENTE DE CUALQUIER DIRECCIÓN DE PÁGINA ANTERIOR DE LAS PÁGINAS PREVIAMENTE DESCIFRADAS DE LA PLURALIDAD DE PÁGINAS DE LA IMAGEN DEL SISTEMA OPERATIVO SEGURO 514
EL DESCIFRADO INCREMENTAL DE LA IMAGEN DEL SISTEMA OPERATIVO SEGURO INCLUYE ADEMÁS DETERMINAR QUE EL VALOR DE AJUSTE ES DIFERENTE DE LOS VALORES DE AJUSTE PREVIAMENTE PREVIOS UTILIZADOS DURANE EL DESCIFRADO DE LAS PÁGINAS PREVIAMENTE DESCIFRADAS DE LA PLURALIDAD DE PÁGINAS DE LA IMAGEN DEL SISTEMA OPERATIVO SEGURO 516
LA VERIFICACIÓN DE LA INTEGRIDAD INCLUYE EMPLEAR, EN PARTE, UN HASH DE CONTENIDO ACUMULATIVO OBTENIDO UTILIZANDO EL CONTENIDO DE LA PÁGINA DE MEMORIA DE LA PLURALIDAD DE PÁGINAS Y UN HASH DE DIRECCIÓN ACUMULATIVO OBTENIDO UTILIZANDO LAS DIRECCIONES DE PÁGINA DE LA PLURALIDAD DE PÁGINAS 518

FIG. 5A

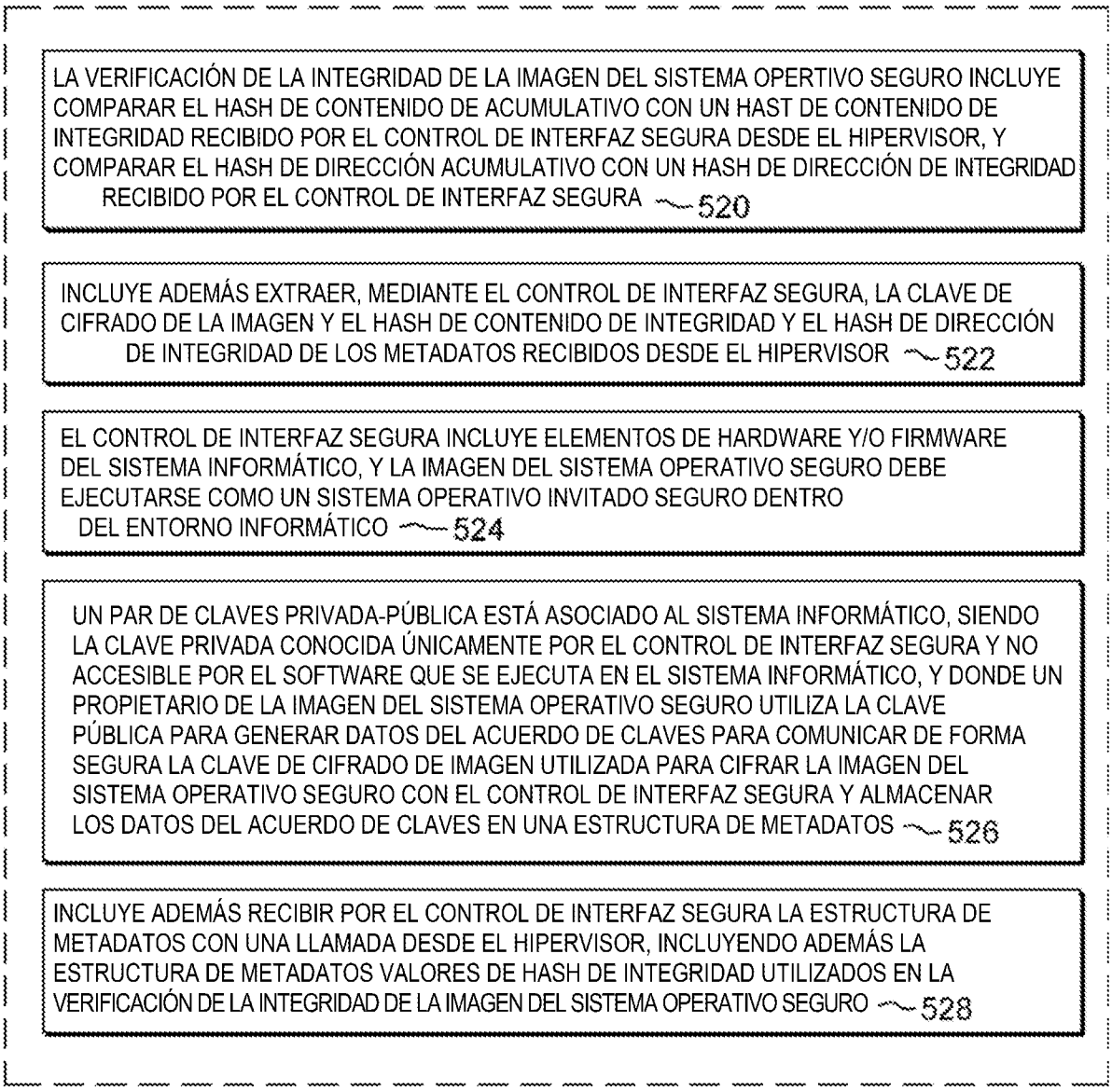
LA VERIFICACIÓN DE LA INTEGRIDAD DE LA IMAGEN DEL SISTEMA OPERTIVO SEGURO INCLUYE COMPARAR EL HASH DE CONTENIDO DE ACUMULATIVO CON UN HAST DE CONTENIDO DE INTEGRIDAD RECIBIDO POR EL CONTROL DE INTERFAZ SEGURA DESDE EL HIPERVISOR, Y COMPARAR EL HASH DE DIRECCIÓN ACUMULATIVO CON UN HASH DE DIRECCIÓN DE INTEGRIDAD RECIBIDO POR EL CONTROL DE INTERFAZ SEGURA 520
INCLUYE ADEMÁS EXTRAER, MEDIANTE EL CONTROL DE INTERFAZ SEGURA, LA CLAVE DE CIFRADO DE LA IMAGEN Y EL HASH DE CONTENIDO DE INTEGRIDAD Y EL HASH DE DIRECCIÓN DE INTEGRIDAD DE LOS METADATOS RECIBIDOS DESDE EL HIPERVISOR 522
EL CONTROL DE INTERFAZ SEGURA INCLUYE ELEMENTOS DE HARDWARE Y/O FIRMWARE DEL SISTEMA INFORMÁTICO, Y LA IMAGEN DEL SISTEMA OPERATIVO SEGURO DEBE EJECUTARSE COMO UN SISTEMA OPERATIVO INVITADO SEGURO DENTRO DEL ENTORNO INFORMÁTICO 524
UN PAR DE CLAVES PRIVADA-PÚBLICA ESTÁ ASOCIADO AL SISTEMA INFORMÁTICO, SIENDO LA CLAVE PRIVADA CONOCIDA ÚNICAMENTE POR EL CONTROL DE INTERFAZ SEGURA Y NO ACCESIBLE POR EL SOFTWARE QUE SE EJECUTA EN EL SISTEMA INFORMÁTICO, Y DONDE UN PROPIETARIO DE LA IMAGEN DEL SISTEMA OPERATIVO SEGURO UTILIZA LA CLAVE PÚBLICA PARA GENERAR DATOS DEL ACUERDO DE CLAVES PARA COMUNICAR DE FORMA SEGURA LA CLAVE DE CIFRADO DE IMAGEN UTILIZADA PARA CIFRAR LA IMAGEN DEL SISTEMA OPERATIVO SEGURO CON EL CONTROL DE INTERFAZ SEGURA Y ALMACENAR LOS DATOS DEL ACUERDO DE CLAVES EN UNA ESTRUCTURA DE METADATOS 526
INCLUYE ADEMÁS RECIBIR POR EL CONTROL DE INTERFAZ SEGURA LA ESTRUCTURA DE METADATOS CON UNA LLAMADA DESDE EL HIPERVISOR, INCLUYENDO ADEMÁS LA ESTRUCTURA DE METADATOS VALORES DE HASH DE INTEGRIDAD UTILIZADOS EN LA VERIFICACIÓN DE LA INTEGRIDAD DE LA IMAGEN DEL SISTEMA OPERATIVO SEGURO 528

FIG. 5B

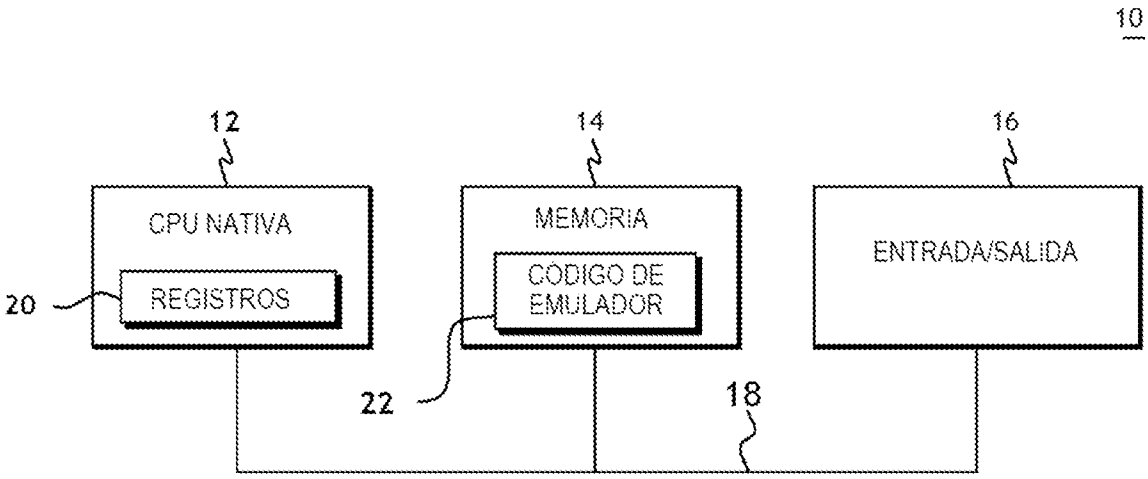
10
12
14
16
CPU NATIVA
REGISTROS
20
MEMORIA
CÓDIGO DE EMULADOR
22
ENTRADA/SALIDA
18

FIG. 6A

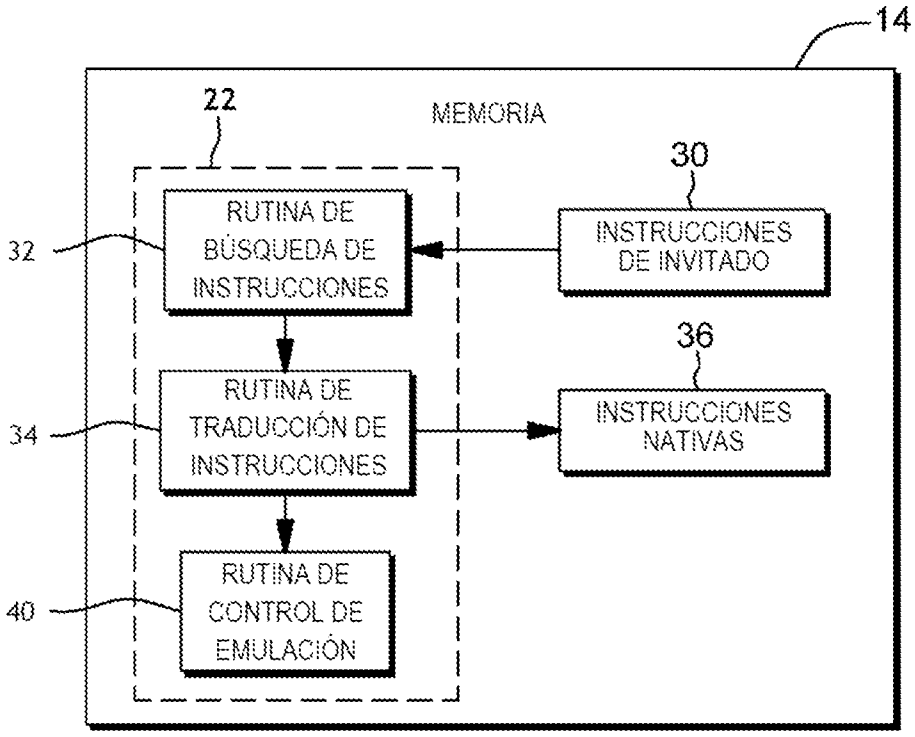
14
22
MEMORIA
30
RUTINA DE BÚSQUEDA DE INSTRUCCIONES
32
INSTRUCCIONES DE INVITADO
36
RUTINA DE TRADUCCIÓN DE INSTRUCCIONES
34
INSTRUCCIONES NATIVAS
RUTINA DE CONTROL DE EMULACIÓN
40

FIG. 6B

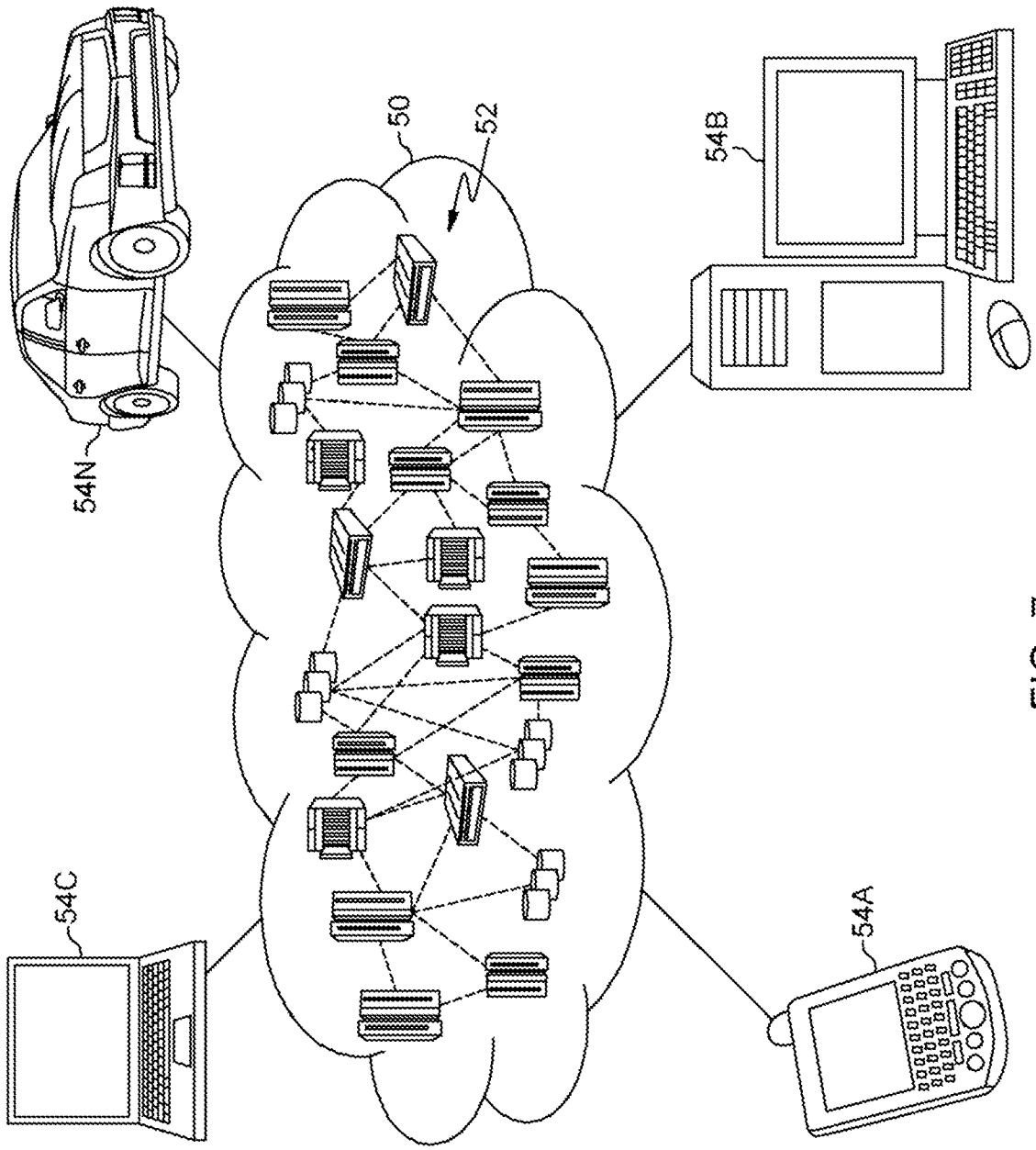

FIG. 7

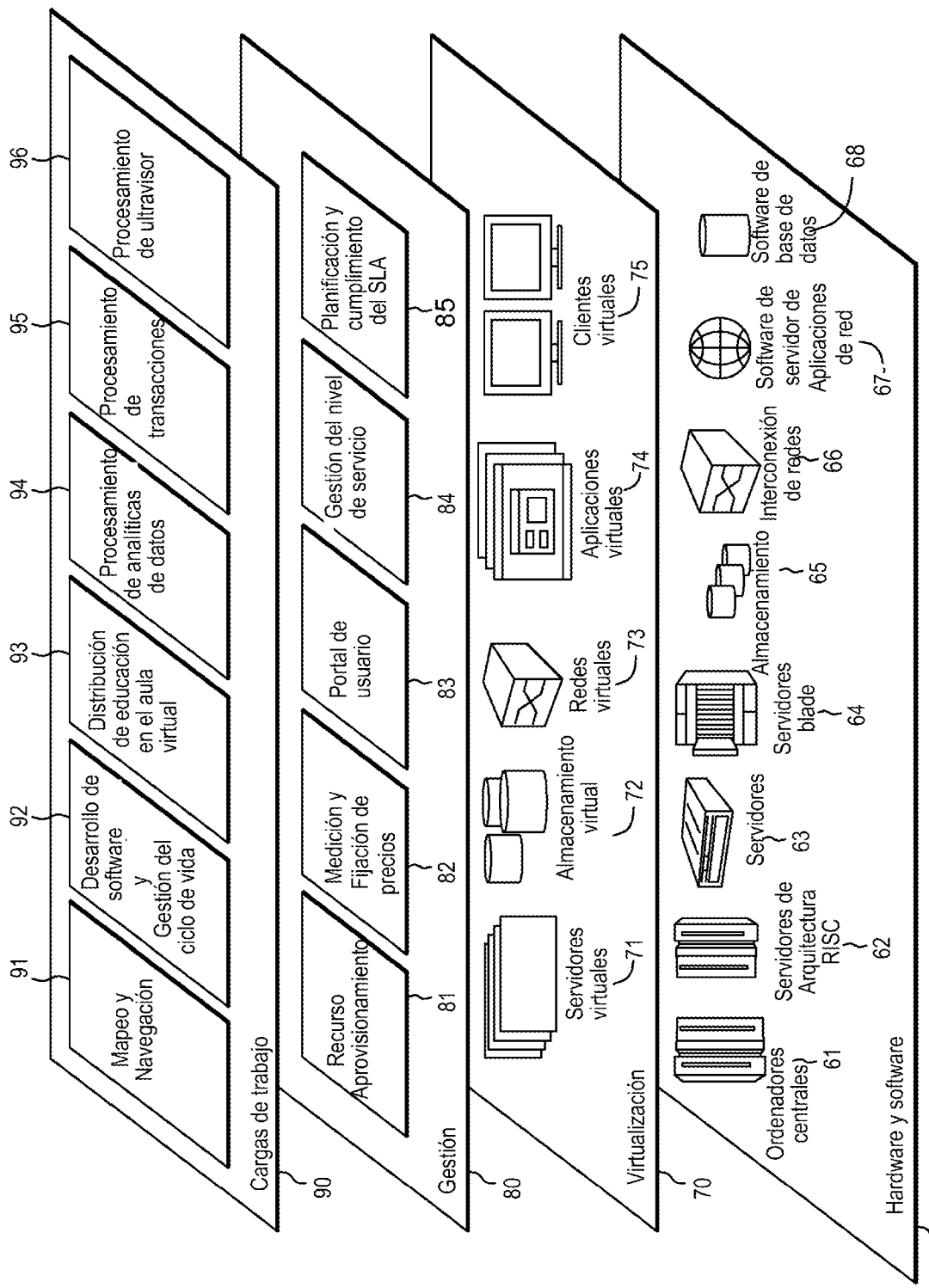
Cargas de trabajo
90
Mapeo y Navegación
91
Desarrollo de software y Gestión del ciclo de vida
92
Distribución de educación en el aula virtual
93
Procesamiento de analíticas de datos
94
Procesamiento de transacciones
95
Procesamiento de ultravisor
96
Gestión
80
Recurso Aprovisionamiento
81
Medición y Fijación de precios
82
Portal de usuario
83
Gestión del nivel de servicio
84
Planificación y cumplimiento del SLA
85
Virtualización
70
Servidores virtuales
71
Almacenamiento virtual
72
Redes virtuales
73
Aplicaciones virtuales
74
Clientes virtuales
75
Hardware y software
60
Ordenadores centrales
61
Servidores de Arquitectura RISC
62
Servidores
63
Servidores blade
64
Almacenamiento
65
Interconexión de redes
66
Software de servidor de Aplicaciones de red
67
Software de base de datos
68

FIG. 8