(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 368 200**

(51) Int. CI.:
G06F 1/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Número de solicitud europea: **99955547 .7**
(96) Fecha de presentación: **09.06.1999**
(97) Número de publicación de la solicitud: **1086413**
(97) Fecha de publicación de la solicitud: **28.03.2001**

(54) Título: **PROCEDIMIENTO Y SISTEMA PARA EJECUCIÓN SEGURA DE CONTENIDO DE POCA CONFIANZA.**

(30) Prioridad:
12.06.1998 US 97218

(45) Fecha de publicación de la mención BOPI:
15.11.2011

(45) Fecha de la publicación del folleto de la patente:
15.11.2011

(73) Titular/es:
MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WA 98052, US

(72) Inventor/es:
CHAN, Shannon, J.; JENSENWORTH, Gregory; GOERTZEL, Mario, C.; SHAH, Bharat; SWIFT, Michael, M. y WARD, Richard, B.

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

DESCRIPCIÓN

Procedimiento y sistema para ejecución segura de contenido de poca confianza

Campo de la invención

La invención se refiere en general a sistemas informáticos, y más particularmente a las mejoras en seguridad para sistemas informáticos.

Antecedentes de la invención

Históricamente, el contenido ejecutable podía instalarse únicamente en un sistema informático trayendo físicamente medios magnéticos al ordenador y haciendo que alguien con privilegios administrativos lo instalara. En la actualidad, sin embargo, Internet ha hecho muy sencillo y popular para los usuarios informáticos ordinarios la descarga de contenido ejecutable tal como programas, páginas de HTML y controles. En muchos casos, el contenido ejecutable puede descargarse y ejecutarse a través de Internet sin que el usuario se dé cuenta siquiera de que tal acontecimiento ha tenido lugar. De forma similar, los usuarios informáticos pueden recibir correo electrónico o noticias que contienen unos archivos que incluyen un contenido ejecutable, tal como programas ejecutables y/o documentos que contienen macros, y además, el correo o noticias en sí mismos pueden ser una página de HTML. La apertura de un mensaje de este tipo o unos datos adjuntos en el mismo expone el sistema del destinatario a cualquier contenido ejecutable que esté presente.

Desafortunadamente, tal contenido ejecutable a menudo no sigue las reglas, por ejemplo, puede ser malicioso y destruir datos de forma intencionada en la máquina del cliente, ser propenso a errores y dar lugar a que la máquina del cliente se bloquee, o ser bienintencionado aunque descuidado y divulgar una información confidencial acerca del cliente. A pesar de que estos tipos de problemas informáticos han existido con anterioridad en la forma de “virus” y “troyanos”, la omnipresente presencia de la World Wide Web ha hecho que estos problemas se extiendan, y en algunos casos fuera de control.

En el lado de servidor, los servidores web lanzan unos programas de servidor tales como unas secuencias de comandos de CGI en nombre de los clientes y devuelven unos datos a partir de los programas de vuelta a los clientes. La fuente de tales secuencias de comandos no se controla necesariamente de forma cuidadosa, ni están bien escritas todas de tales secuencias de comandos. Como resultado, las secuencias de comandos de CGI pobremente escritas han dado lugar a que máquinas de servidor web se bloqueen o se ralenticen usando demasiados recursos informáticos. Además, un cliente malicioso puede confundir unos programas de servidor pobremente escritos para realizar acciones que no debería realizar, tales como ejecutar otras aplicaciones o escribir o leer datos a o desde un almacenamiento. En último lugar, algunos servidores web incluso permiten a un programa de cliente que envíe secuencias de comandos al servidor para que se ejecuten en nombre del cliente, lo que supone muchos peligros.

En general, los entornos operativos de cliente y de servidor no están protegidos de forma adecuada frente a un contenido ejecutable que no sigue las reglas. Al mismo tiempo, debido a que tanto contenido ejecutable es valioso, la necesidad de ser capaz de recibir y ejecutar un contenido ejecutable continúa creciendo a pesar de los riesgos intrínsecos de contenido de poca confianza.

El documento EP–A–0 588 415 describe un autorizador de conexión punto a punto que implica tres entidades diferentes: un mecanismo autorizador de sistema, un administrador de conexión de cliente, y un administrador de conexión de servidor. El autorizador de sistema se encuentra en la CPU principal o primaria mientras que el cliente y los administradores de conexión de servidor se encuentran en unos procesadores de entrada/salida individuales. Para obtener una información que precisa un usuario y/o un programa de aplicación, el administrador de conexión de cliente emite una petición para el autorizador de sistema. Cuando el autorizador de sistema recibe la petición, en primer lugar verifica que el dispositivo de cliente es quien pretende ser. Si el autorizador de sistema determina que se debe permitir al dispositivo de cliente obtener acceso a la información solicitada, entonces envía un testigo al dispositivo de servidor y una copia del mismo testigo al dispositivo de cliente. Tras la recepción de la copia de testigo a partir del autorizador de sistema, el administrador de conexión de cliente empaqueta la copia de testigo en el interior de un mensaje que envía al dispositivo de servidor. Cuando el administrador de conexión de servidor recibe el mensaje a partir del dispositivo de cliente, compara la copia de testigo con el testigo que ha recibido a partir del autorizador de sistema. Si los testigos se corresponden, el administrador de conexión de servidor responde al dispositivo de cliente y la conexión se establece.

El testigo de acuerdo con el documento EP–A–0 588 415 comprende una dirección de IOP de cliente, un recurso de IOP de cliente, el momento del día y un valor aleatorio. La dirección de IOP de cliente es un campo que contiene la ubicación del IOP de cliente. El recurso de IOP de cliente es un campo que contiene la identificación del recurso que está solicitando la información. El campo de momento del día y el campo de valor aleatorio se usan para fines de unicidad y de cifrado en caso de que se precisara de una seguridad adicional.

Sin embargo, un testigo regular de este tipo de correspondencia sencilla puede usarse sólo para identificar a un usuario específico. Si se concede el acceso al sistema a dicho usuario, el acceso de unos procesos de dicho usuario

a unos recursos se permitirá siempre basándose en los derechos de acceso asociados a dicho usuario.

Sumario de la invención

Es, por lo tanto, el objeto de la presente invención es proporcionar un contexto de ejecución restringida para restringir los recursos a los que puede obtener acceso un contenido determinado.

El objeto precedente se solventa mediante el contenido de las reivindicaciones independientes.

Las realizaciones preferidas son el contenido de las reivindicaciones dependientes.

Brevemente, la presente invención proporciona contextos de ejecución restringida para un contenido de poca confianza (tal como código ejecutable, HTML dinámico, controles de Java o de Active–X) que restringe los recursos a los que puede obtener acceso el contenido. Un proceso restringido se configura para un contenido de poca confianza, y cualquier acción que intenta el contenido se somete a las restricciones del proceso, que se basan en diversos criterios. Siempre que un proceso intenta obtener acceso a un recurso, un testigo restringido asociado con cada proceso se compara frente a una información de seguridad del recurso para determinar si se permite el tipo de acceso. La información de seguridad del recurso por lo tanto determina si un proceso, y por lo tanto el contenido de poca confianza, puede obtener acceso al recurso, y si es así, el tipo de acceso que se permite.

El contenido de poca confianza incluye unos datos descargados a partir de sitios web, y cada uno de tales sitios tiene un proceso restringido configurado para el mismo basándose en la identidad del sitio y en la zona en la que se clasifica el sitio. Las API y los ayudantes de proceso habilitan al sitio web para obtener acceso su propio sitio, archivos y claves de registro, mientras que las ACL en otros recursos no relacionados con el sitio restringen el acceso de ese sitio tal como se desee, basándose en la identidad del sitio, en la zona o en otros criterios. Otro contenido de poca confianza incluye unos mensajes de correo electrónico o noticias junto con cualesquiera conjuntos de datos adjuntos al mismo. Tal contenido se ejecuta de forma similar en un contexto de ejecución restringida en el que las restricciones se basan en unos criterios tales como la identidad del remitente. Los servidores también pueden ejecutar un contenido de poca confianza tal como unas secuencias de comandos y procesos de cliente en el contexto de una ejecución restringida, mediante lo cual las restricciones pueden basarse en unos criterios tales como el autor de la secuencia de comandos, el procedimiento de autenticación de cliente usado, y/o cualquier otra información disponible para el servidor indicativa de cómo puede ser el contenido respecto a si es de confianza o de poca confianza.

Otras ventajas se harán evidentes a partir de la siguiente descripción detallada cuando se toma junto con los dibujos, en los que:

Breve descripción de los dibujos

la figura 1 es un diagrama de bloques que representa un sistema informático en el que puede incorporarse la presente invención;
la figura 2 es un diagrama de bloques que representa en general la creación de un testigo restringido a partir de un testigo existente;
la figura 3 es un diagrama de bloques que representa en general los diversos componentes para determinar si un proceso puede obtener acceso a un recurso;
las figuras 4 y 5 comprenden un diagrama de flujo que representa las etapas generales tomadas para crear un testigo restringido a partir de un testigo existente;
la figura 6 es un diagrama de bloques que representa en general un proceso que tiene un testigo restringido asociado con el mismo que intenta obtener acceso a un recurso;
la figura 7 es un diagrama de bloques que representa en general la lógica para determinar el acceso a un objeto de un proceso que tiene un testigo restringido asociado con el mismo;
la figura 8 es un diagrama de flujo que representa las etapas generales tomadas al determinar si se concede un acceso de proceso a un recurso;
la figura 9 es una representación de un objeto de trabajo que tiene múltiples procesos en el mismo que tienen restricciones comunes;
la figura 10 es un diagrama de bloques que representa en general un contenido de poca confianza en un proceso configurado para el mismo y que está restringido con respecto al acceso a unos recursos de acuerdo con un aspecto de la presente invención;
las figuras 11 a 12 son unos diagramas de bloques que representan unos componentes de acuerdo con otro aspecto de la presente invención para restringir unos procesos de acuerdo con unos criterios para un sitio de poca confianza y un contenido de correo electrónico, respectivamente;
la figura 13 es un diagrama de bloques que representa en general un proceso ayudante para obtener acceso a un recurso en nombre de un proceso restringido de acuerdo con otro aspecto de la presente invención;
las figuras 14 y 15 comprenden un diagrama de flujo que muestra cómo una interfaz de programación de aplicación (API, *Application Programming Interface*) usa un proceso ayudante para devolver una información a partir de un recurso a un proceso al que se restringe la obtención de acceso al recurso de

acuerdo con otro aspecto de la presente invención;
la figura 16 es una representación de los archivos de un sitio aislados respecto de los archivos de otros sitios en un sistema de archivos de acuerdo con otro aspecto de la presente invención;
las figuras 17 y 18 son unos diagramas de bloques que representan cómo se redirigen un sistema de archivos y unas peticiones de registro de un contenido de poca confianza, respectivamente, de acuerdo con otro aspecto de la presente invención;
la figura 19 es un diagrama de flujo que representa un ejemplo de cómo un contenido de correo electrónico puede restringirse de acuerdo con unos criterios incluyendo la identidad del remitente, de acuerdo con otro aspecto de la presente invención;
la figura 20 es un diagrama de bloques que representa en general secuencias de comandos de poca confianza en un procesos configurado para el mismo y restringido con respecto a su acceso a unos recursos de acuerdo con otro aspecto de la presente invención; y
la figura 21 es un diagrama de flujo que representa un ejemplo de cómo pueden restringirse unos procesos de cliente en un servidor de acuerdo con unos criterios, incluyendo cómo se autenticó el cliente, de acuerdo con otro aspecto de la presente invención.

Descripción detallada

Entorno operativo a modo de ejemplo

La figura 1 y la siguiente discusión tienen como objetivo proporcionar una breve descripción general de un entorno de cálculo adecuado en el que puede implementarse la invención. A pesar de que no se requiere, la invención se describirá en el contexto general de unas instrucciones ejecutables por ordenador, tales como unos módulos de programa, que se están ejecutando por un ordenador personal. Generalmente, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos y similares que realizan tareas particulares o que implementan unos tipos de datos abstractos particulares. Además, los expertos en la técnica observarán que la invención puede ponerse en práctica con otras configuraciones de sistema informático, incluyendo dispositivos que pueden sostenerse con la mano, sistemas multiprocesador, electrónica de consumo basada en microprocesador o programable, PC de red, miniordenadores, ordenadores centrales y similares. La invención puede también ponerse en práctica en entornos de cálculo distribuido en los que las tareas se realizan mediante dispositivos de procesamiento remoto que se enlazan a través de una red de comunicaciones. En un entorno de cálculo distribuido, los módulos de programa pueden estar ubicados tanto en los dispositivos de almacenamiento de memoria locales como en los remotos.

Con referencia a la figura 1, un sistema a modo de ejemplo para implementar la invención incluye un dispositivo de cálculo de propósito general en la forma de un ordenador 20 personal convencional o similares, que incluye una unidad 21 de procesamiento, una memoria 22 de sistema, y un bus 23 de sistema que acopla diversos componentes de sistema que incluyen la memoria de sistema para la unidad 21 de procesamiento. El bus 23 de sistema puede ser cualquiera de diversos tipos de estructuras de bus que incluyen un bus de memoria o un controlador de memoria, un bus de periféricos, y un bus local que usa cualquiera de una diversidad de arquitecturas de bus. La memoria de sistema incluye una memoria de sólo lectura (ROM, *read only memory*) 24 y una memoria de acceso aleatorio (RAM, *random access memory*) 25. Un sistema 26 de entrada/salida básico (BIOS, b*asic input/output system*), que contiene las rutinas básicas que ayudan a transferir información entre los elementos en el interior del ordenador 20 personal, tal como durante el inicio, se almacena en la ROM 24. El ordenador 20 personal puede incluir además una unidad 27 de disco duro para leer de y escribir en un disco duro, que no se muestra, una unidad 28 de disco magnético para leer de o escribir en un disco 29 magnético que puede retirarse, y una unidad 30 de disco óptico para leer de o escribir en un disco 31 óptico que puede retirarse tal como un CD–ROM u otros medios ópticos. La unidad 27 de disco duro, la unidad 28 de disco magnético, y la unidad 30 de disco óptico se conectan al bus 23 de sistema mediante una interfaz 32 de unidad de disco duro, una interfaz 33 de unidad de disco magnético, y una interfaz 34 de unidad óptica, respectivamente. Las unidades y sus medios legibles por ordenador asociada proporcionan un almacenamiento no volátil de instrucciones legibles por ordenador, estructuras de datos, módulos de programa y otros datos para el ordenador 20 personal. A pesar de que el entorno a modo de ejemplo que se describe en el presente documento emplea un disco duro, un disco 29 magnético que puede retirarse y un disco 31 óptico que puede retirarse, debe observarse por los expertos en la técnica que otros tipos de medios legibles por ordenador que pueden almacenar datos que son accesibles por un ordenador, tales como casetes magnéticos, tarjetas de memoria flash, discos de vídeo digital, cartuchos de Bernoulli, memorias de acceso aleatorio (RAM), memorias de sólo lectura (ROM) y similares pueden también usarse en el entorno operativo a modo de ejemplo.

Un número de módulos de programa puede almacenarse en el disco duro, en el disco 29 magnético, en el disco 31 óptico, en la ROM 24 o en la RAM 25, incluyendo un sistema 35 operativo (preferiblemente Windows NT), uno o más programas 36 de aplicación, otros módulos 37 de programa y datos 38 de programa. Un usuario puede introducir órdenes e información en el ordenador 20 personal a través de unos dispositivos de entrada tales como un teclado 40 y un dispositivo 42 señalador. Otros dispositivos de entrada (que no se muestran) pueden incluir un micrófono, un *joystick*, un controlador para juegos, una antena parabólica, un escáner o similares. Estos y otros dispositivos de entrada se conectan a menudo a la unidad 21 de procesamiento a través de una interfaz 46 de puerto serie que está acoplada al bus de sistema, pero que puede estar conectada mediante otras interfaces, tales como un puerto paralelo, puerto para juegos o un bus serie universal (USB, *universal serial bus*). Un monitor 47 u otro tipo de

dispositivo de visualización se conecta también al bus 23 de sistema a través de una interfaz, tal como un adaptador 48 de vídeo. Además del monitor 47, los ordenadores personales normalmente incluyen otros dispositivos de salida periféricos (que no se muestran), tales como altavoces e impresoras.

El ordenador 20 personal puede funcionar en un entorno de red que usa conexiones lógicas a uno o más ordenadores remotos, tales como un ordenador 49 remoto. El ordenador 49 remoto puede ser otro ordenador personal, un servidor, un enrutador, un PC de red, un dispositivo del mismo nivel u otro nodo de red común, y normalmente incluye muchos o todos los elementos que se describen anteriormente en relación con el ordenador 20 personal, a pesar de que sólo se ha ilustrado un dispositivo 50 de almacenamiento de memoria en la figura 1. Las conexiones lógicas que se representan en la figura 1 incluyen una red 51 de área local (LAN) y una red 52 de área amplia (WAN). Tales entornos de red son muy comunes en oficinas, redes de ordenadores por toda una empresa, Intranets e Internet.

Cuando se usa en un entorno de red de LAN, el ordenador 20 personal se conecta a la red 51 local a través de una interfaz de red o adaptador 53. Cuando se usa en un entorno de red de WAN, el ordenador 20 personal normalmente incluye un módem 54 u otros medios para establecer unas comunicaciones por la red 52 de área amplia, tal como Internet. El módem 54, que puede ser interno o externo, se conecta al bus 23 de sistema a través de la interfaz 46 de puerto serie. En un entorno de red, los módulos de programa que se representan en relación con el ordenador 20 personal, o partes de los mismos, pueden almacenarse en el dispositivo de almacenamiento de memoria remoto. Se observará que las conexiones de red que se muestran son a modo de ejemplo y pueden usarse otros medios de establecimiento de un enlace de comunicaciones entre los ordenadores.

El modelo de seguridad general

El modelo de seguridad preferente de la presente invención se describe en el presente documento con referencia al modelo de seguridad de Windows NT. No obstante, no existe intención de limitar la presente invención al sistema operativo de Windows NT, sino que por el contrario, la presente invención tiene como objetivo funcionar con y proporcionar beneficios a cualquier mecanismo que realice unas comprobaciones de seguridad en el nivel de sistema operativo. Además, la presente invención puede también usarse con un aislamiento de defecto de software subproceso a subproceso, o con una máquina virtual en la que las restricciones se determinan a partir de la pila de clases actualmente en ejecución. Además, la presente invención no depende necesariamente de un funcionamiento en modo de núcleo funcionamiento, como con el aislamiento de defecto de software o puede implementarse una máquina virtual en modo de usuario.

En general, en el sistema operativo de Windows NT, un usuario realiza tareas obteniendo acceso a los recursos del sistema a través de los procesos (y de sus subprocesos). Con fines de simplicidad en el presente documento, un proceso y sus subprocesos se considerarán conceptualmente equivalentes, y por lo tanto a continuación en el presente documento simplemente se hará referencia a los mismos como un proceso. También, se hará referencia a los recursos del sistema, incluyendo los archivos, la memoria compartida y los dispositivos físicos, que en Windows NT se representan mediante objetos, normalmente como o bien recursos o bien objetos en el presente documento.

Cuando un usuario inicia sesión en el sistema operativo de Windows NT y se autentica, se configura un contexto de seguridad para ese usuario, que incluye construir un testigo 60 de acceso. Tal como se muestra en la parte izquierda de la figura 2, un testigo 60 de acceso basado en el usuario convencional incluye un campo 62 UserAndGroups que incluye un identificador 64 de seguridad (ID de seguridad, o SID) basándose en las credenciales del usuario y uno o más ID 66 de grupo que identifican los grupos (por ejemplo, dentro de una organización) a los que pertenece ese usuario. El testigo 60 también incluye un campo 68 de privilegios que enumera cualquier privilegio asignado al usuario. Por ejemplo, un privilegio de este tipo puede dar a un usuario nivel administrativo la capacidad de ajustar el reloj de sistema a través de una interfaz de programación de aplicación (API) particular. Obsérvese que los privilegios invalidan las comprobaciones de control de acceso, que se describen a continuación, que se realizan por lo demás antes de conceder acceso a un objeto.

Como se describirá en más detalle a continuación y tal como se representa generalmente en la figura 3, un proceso 70 que desea obtener acceso a un objeto 72 especifica el tipo de acceso que desea (por ejemplo, obtener acceso de lectura/escritura a un objeto de archivo) y en el nivel de sistema operativo (por ejemplo, núcleo) proporciona su testigo 60 asociado a un administrador 74 de objeto. El objeto 72 tiene un descriptor 76 de seguridad asociado con el mismo, y el administrador 74 de objeto proporciona el descriptor 76 de seguridad y el testigo 60 a un mecanismo 78 de seguridad. Los contenidos del descriptor 76 de seguridad se determinan normalmente por el propietario (por ejemplo, el creador) del objeto, y generalmente comprenden una lista 80 de control de acceso (ACL), (a discreción) de las entradas de control de acceso, y para cada entrada, uno o más derechos de acceso (acciones permitidas o denegadas) que se corresponden con esa entrada. Cada entrada comprende un indicador de tipo (denegar o permitir), unas banderas, un identificador de seguridad (SID) y unos derechos de acceso en la forma de una máscara de bits en el que cada bit se corresponde con un permiso (por ejemplo, un bit para el acceso de lectura, uno para escritura y así sucesivamente). El mecanismo 78 de seguridad compara los ID de seguridad en el testigo 60 junto con el tipo de acción o acciones solicitadas por el proceso 70 frente a las entradas en la ACL 80. Si se encuentra una correspondencia con un usuario o grupo permitido, y el tipo de acceso deseado está disponible para el usuario o grupo, se devuelve un manejador para el objeto 72 al proceso 70, por lo demás se deniega el acceso.

A modo de ejemplo, un usuario con un testigo que identifica al usuario como un elemento del grupo de "Contabilidad" puede desear obtener acceso a un objeto de archivo particular con acceso de lectura y de escritura. Si el objeto de archivo tiene el identificador de grupo de "Contabilidad" de tipo permitir en una entrada de su ACL 80, y el grupo tiene derechos que habilitan el acceso de lectura y de escritura, se devuelve un manejador que concede el acceso de lectura y de escritura, por lo demás se deniega el acceso. Obsérvese que por razones de eficiencia, la comprobación de seguridad se realiza sólo cuando el proceso 70 intenta en primer lugar obtener acceso al objeto 72 (crear o abrir), y por lo tanto el manejador para el objeto almacena el tipo de información de acceso con el fin de limitar las acciones que pueden realizarse a través del mismo.

El descriptor 76 de seguridad también incluye un sistema ACL, o SACL 81, que comprende unas entradas de tipo auditoría que se corresponde con acciones de cliente que van a auditarse. Unas banderas en cada entrada indican si la auditoría está supervisando unas operaciones con éxito o fallidas, y una máscara de bits en la entrada indica el tipo de operaciones que van a auditarse. Un ID de seguridad en la entrada indica el usuario o grupo que se está auditando. Por ejemplo, considérese una situación en la que un grupo particular se está auditando con el fin de determinar siempre que un elemento de ese grupo que no tiene acceso de escritura a un objeto de archivo intenta escribir en ese archivo. La SACL 81 para ese objeto de archivo incluye una entrada de auditoría que tiene el identificador de seguridad de grupo en el mismo junto con una bandera de fallo y un bit de acceso de escritura ajustados adecuadamente. Siempre que un cliente que pertenece a ese grupo particular intenta escribir en el objeto de archivo y falla, la operación se registra.

Obsérvese que la ACL 80 puede contener uno o más identificadores que se marcan para la denegación a los usuarios de los grupos el acceso (como para todos los derechos o derechos seleccionados) en lugar de conceder acceso al mismo. Por ejemplo, una entrada enumerada en la ACL 80 puede por lo demás permitir a unos elementos de "$grupo_3$" el acceso al objeto 72, pero otra entrada en la ACL 80 puede específicamente denegar todo acceso al "$grupo_{24}$". Si el testigo 60 incluye el ID de seguridad del "$grupo_{24}$", se denegará el acceso con independencia de la presencia del ID de seguridad del "$grupo_3$". Naturalmente, para funcionar adecuadamente, la comprobación de seguridad se dispone con el fin de no permitir el acceso a través de la entrada del "$grupo_3$" antes de la comprobación del estado de "DENEGAR TODO" del entrada del $grupo_{24}$, tal como ubicando las entradas de todo DENEGAR en la parte delantera de la ACL 80. Tal como puede observarse, esta disposición proporciona una eficiencia mejorada, puesto que uno o más elementos aislados de un grupo pueden excluirse de forma separada en la ACL 80 en lugar de tener que enumerar individualmente cada uno de los elementos restantes de un grupo para permitir su acceso.

Obsérvese que en lugar de especificar un tipo de acceso, un autor de llamada puede solicitar un acceso MAXIMUM_ALLOWED, mediante el que un algoritmo determina el máximo tipo de acceso que se permite, basándose en la lista UserAndGroups normal frente a cada una de las entradas en la ACL 80. Más particularmente, el algoritmo recorre hacia abajo la lista de identificadores acumulando los derechos para un usuario dado (es decir, poniendo a nivel bajo las diversas máscaras de bits). Once los derechos se han acumulado, se dan al usuario los derechos acumulados. Sin embargo, si durante el recorrido se encuentra un denegar entrada que hace corresponder un identificador de usuario o de grupo y los derechos solicitados, se deniega el acceso.

Testigos restringidos

Un testigo restringido se crea a partir de un testigo de acceso existente (o bien restringido o bien no restringido) tal como se describe a continuación. Tal como también se describe a continuación, si el testigo restringido incluye cualesquiera ID de seguridad restringida, el testigo se somete a una comprobación de acceso adicional en la que los ID de seguridad restringida se comparan frente a las entradas en la ACL del objeto.

El uso primario de un testigo restringido es que un proceso cree un nuevo proceso con una versión restringida de su propio testigo. El proceso restringido se limita entonces en las acciones que puede realizar sobre los recursos. Por ejemplo, un recurso de objeto de archivo puede tener en su ACL un único SID restringido que identifica el programa de aplicación Microsoft Word, tal que sólo unos procesos restringidos que tienen el mismo SID restringido de Microsoft Word en su testigo asociado restringido pueden obtener acceso al objeto de archivo. Obsérvese que el usuario original todavía necesita tener acceso al objeto, de modo que para obtener acceso a éste, la ACL también necesita contener una entrada de control de acceso que concede el acceso para el usuario, así como al programa Microsoft Word. Entonces, por ejemplo, un código de poca confianza tal como uno descargado a través de un navegador podría ejecutarse en un proceso restringido que no tuviera la ID de seguridad restringida de Microsoft Word en su testigo restringido, evitando el acceso de ese código al objeto de archivo.

Por razones de seguridad, crear un proceso con un testigo diferente normalmente requiere un privilegio que se conoce como el privilegio SeAssignPrimaryToken. Sin embargo, para permitir que los procesos se asocien con testigos restringidos, la administración del proceso permite que un proceso con suficiente acceso a otro proceso modifique su testigo primario en un testigo restringido, si el testigo restringido se deduce a partir del testigo primario. Comparando el ParentTokenId del nuevo testigo del proceso con el TokenId del testigo del proceso existente, el sistema 35 operativo puede asegurar que el proceso sólo está creando una versión restringida de sí mismo.

Un testigo 84 restringido tiene menos acceso que su testigo de progenitor, y puede, por ejemplo, evitar el acceso a un objeto basándose en el tipo de proceso (así como el usuario o grupo) que está intentando obtener acceso al

objeto, en lugar de simplemente permitir o denegar el acceso basándose únicamente en la información de grupo o de usuario. Un testigo restringido puede también no permitir el acceso a través de uno o más ID de seguridad de usuario o de grupo especialmente marcados como “USE_FOR_DENY_ONLY”, incluso a pesar de que el testigo de progenitor permite el acceso a través de esos SID, y/o puede tener privilegios eliminados que están presentes en el testigo de progenitor.

Por lo tanto, una forma en la que reducir el acceso es cambiar un atributo de uno o más usuarios y/o identificadores de seguridad de grupo en un testigo restringido con el fin de ser incapaces de permitir obtener acceso, en lugar de conceder acceso con el mismo. Los ID de seguridad marcados USE_FOR_DENY_ONLY se ignoran efectivamente con fines de conceder acceso, sin embargo, una ACL que tiene una entrada de “DENEGAR” para esa ID de seguridad dará todavía lugar a que se deniegue el acceso. A modo de ejemplo, si el ID de seguridad del $grupo_2$ en el testigo 84 restringido (figura 3) está marcado USE_FOR_DENY_ONLY, cuando el proceso del usuario intenta obtener acceso a un objeto 72 que tiene la ACL 80 que enumera el $grupo_2$ como permitido, esa entrada se ignora efectivamente y el proceso tendrá que ganar acceso mediante algún otro ID de seguridad. Sin embargo, si la ACL 80 incluye una entrada que enumera el $grupo_2$ como DENEGAR con respecto al tipo de acción solicitado, entonces una vez que se comprueba, no se concederá un acceso con independencia de otros ID de seguridad.

Obsérvese que el acceso a objetos no puede reducirse de forma segura eliminando simplemente un ID de seguridad a partir de un testigo del usuario, debido a que ese ID de seguridad puede estar marcado como “DENEGAR” en la ACL de algunos objetos, mediante lo cual eliminar ese identificador concedería en lugar de denegar acceso a esos objetos. Por lo tanto, unos atributos del SID pueden modificarse para USE_FOR_DENY_ONLY en un testigo restringido. Además, no se proporciona un mecanismo para desactivar esta comprobación de seguridad USE_FOR_DENY_ONLY.

Otra forma de reducir el acceso en un testigo restringido es eliminar uno o más privilegios en relación con el testigo de progenitor. Por ejemplo, un usuario que tiene un testigo normal con privilegios administrativos puede configurar un sistema tal que a menos que el usuario específicamente informe al sistema de otro modo, los procesos del usuario se ejecutarán con un testigo restringido que no tiene privilegios. Tal como puede observarse, esto evita unos errores inadvertidos que pueden tener lugar cuando el usuario no está actuando de forma intencionada en una capacidad administrativa. De forma similar, pueden desarrollarse programas para ejecutarse en diferentes modos que dependen de los privilegios de un usuario, mediante lo cual un usuario de nivel administrativo tiene que ejecutar el programa con privilegios administrativos para realizar algunas operaciones, pero opera con unos privilegios reducidos para realizar unas operaciones más básicas. De nuevo, esto ayuda a evitar errores graves que pueden por lo demás tener lugar cuando un usuario de este tipo está intentando simplemente realizar unas operaciones normales pero está realizando la ejecución con unos privilegios elevados.

Otra forma más de reducir el acceso de un testigo es añadir unos ID de seguridad restringida al mismo. Los ID de seguridad restringida son números que representan procesos, operaciones de recurso y similares, hechos únicos tales como añadir un prefijo a los GUID o a los números generados a través de una función hash criptográfica o similares, y pueden incluir información para distinguir estos ID de seguridad con respecto a otros ID de seguridad. A pesar de que no es necesario para la invención, por conveniencia, se proporcionan diversas interfaces de programación de aplicación (API) para interconectar las aplicaciones y los usuarios con unas ID de seguridad, tales como para lograr un GUID para una conversión de ID de seguridad, para representar los ID de seguridad en una forma legible por un ser humano, y así sucesivamente.

Además de restringir el acceso a un recurso basándose en la aplicación (proceso) que solicita el acceso, unas ID de seguridad específicas pueden desarrollarse basándose en unos usos probablemente restringidos de un recurso. A modo de ejemplo, un ID de seguridad tal como “USE_WINDOWS” se ubicaría en las ACL por defecto de estaciones de Windows y el escritorio para permitir obtener acceso al mismo sólo por un proceso que tiene un SID correspondiente en su testigo restringido. De forma similar, la ACL por defecto de un objeto de impresora puede incluir un SID de USE_PRINTING en su ACL por defecto, de modo que un proceso de ese tipo podría crear un proceso restringido con sólo este ID de seguridad enumerado en su testigo restringido, mediante lo cual el proceso restringido sería capaz de obtener acceso a la impresora pero no a otro recurso. Tal como puede observarse, pueden implementarse otros numerosos ID de seguridad para obtener acceso a otros recursos.

Tal como se muestra en la figura 3, se ubican unos ID de seguridad restringida en un campo 82 especial de un testigo 84 restringido, tal como para identificar un proceso que está solicitando una acción. Tal como se describe en más detalle a continuación, requiriendo que se conceda acceso a ambos de al menos un ID de seguridad de usuario (o grupo) y al menos un ID de seguridad restringida a un objeto, un objeto puede conceder de forma selectiva acceso basándose en un proceso que solicita (así como un usuario o grupo). Por ejemplo, un objeto tal como un objeto de archivo puede permitir que unos procesos de Microsoft Word, Microsoft Excel o Windows Explorer obtengan acceso al mismo, pero denegar el acceso a cualquier otro proceso. Además, pueden concederse unos derechos de acceso diferentes a cada uno de los procesos permitidos.

El diseño proporciona una flexibilidad y granularidad significativas en el contexto de un usuario para controlar lo que se permite hacer a procesos diferentes. Un modelo de uso para estas características, que se describe en detalle a continuación, incluye una distinción entre aplicaciones de confianza y aplicaciones de poca confianza. Obsérvese

que el término “aplicación” se usa en un sentido genérico para describir cualquier fragmento de código que puede ejecutarse en “modo de usuario” bajo un contexto de seguridad dado. Por ejemplo, una aplicación tal como Microsoft Word puede ejecutarse como un proceso a partir de un Control ActiveX, que puede cargarse en el interior de un proceso existente y ejecutarse. Las aplicaciones que lanzan otras aplicaciones, tales como Internet Explorer de Microsoft, pueden introduce un “modelo de confianza” que usa esta infraestructura.

A modo de ejemplo, y tal como se describe en más detalle a continuación, una aplicación tal como Internet Explorer puede usar testigos restringidos para ejecutar un código de poca confianza ejecutable bajo procesos diferentes, y controlar qué pueden hacer los procesos dentro de los derechos y privilegios de acceso globales del usuario. Con este fin, la aplicación de Internet Explorer crea un testigo restringido a partir de su propio testigo, y determina qué ID de seguridad restringida se ubicarán en el testigo restringido. Entonces, se restringe el acceso del código de poca confianza ejecutable sólo a esos objetos a los que el contexto restringido puede obtener acceso.

Además, las entradas que se corresponden con SID restringidos y otras restricciones pueden ubicarse en un campo de la SACL 81 para fines de auditoría. Por ejemplo, la SACL de un recurso puede configurarse para una auditoría cada vez que el programa Internet Explorer intenta un acceso de lectura o de escritura de ese recurso, y/o puede auditarse el uso de los SID marcados USE_FOR_DENY_ONLY. Con fines de simplicidad, la auditoría no se describe en detalle a continuación en el presente documento, sin embargo puede observarse fácilmente que los conceptos que se describen con respecto a obtener un control de acceso a través de unos SID restringidos pueden aplicarse a operaciones de auditoría.

Para crear un testigo restringido a partir de un testigo existente, se proporciona una interfaz de programación de aplicación (API), denominada NtFilterToken, tal como se expone a continuación:

```
NTSTATU5
NtFilterToken (
        IN HANDLE ExistingTokenHandle,
        IN ULONG Flags,
        IN PTOKEN_GROUPS SidsToDisable OPTIONAL,
        IN PTOKEN_PRIVILEGES PrivilegesToDelete OPTIONAL,
        IN PTOKEN_GROUPS RestrictingSids OPTIONAL,
        OUT PHANDLE NewTokenHandle
        );
```

La API NtFilterToken se encapsula bajo una API de Win32 denominada CreateRestrictedToken, que se expone adicionalmente a continuación:

```
WINADVAPI
BOOL
APIENTRY
CreateRestrictedToken (
        IN HANDLE ExistingTokenHandle,
        IN DWORD Flags,
        IN DWORD DisableSidCount,
        IN PSID_AND_ATTRIBUTES SidsToDisable OPTIONAL,
        IN DWORD DeletePrivilegeCount,
        IN PLUID_AND_ATTRIBUTES PrivilegesToDelete OPTIONAL,
        IN DWORD RestrictedSidCount,
        IN PSID_AND_ATTRIBUTES SidsToRestrict OPTIONAL,
        OUT PHANDLE NewTokenHandle
        );
```

Tal como se representa en las figuras 2 y 4 a 5, estas API 86 funcionan en conjunción para tomar un testigo 60 existente, o bien restringido o bien no restringido, y para crear un testigo 84 modificado (restringido) a partir del mismo. La estructura de un testigo restringido, que contiene la información de identificación acerca de una instancia de un usuario que ha iniciado sesión, incluye tres campos nuevos, ParentTokenId, RestrictedSidCount y RestrictedSIDs (que se muestran en negrita a continuación):

```
Typedef struct _TOKEN {
TOKEN_SOURCE TokenSource;                    // Ro: 16–Bytes
        LUID TokenId;                        // Ro: 8–Bytes
        LUID AuthenticationId;               // Ro: 8–Bytes
        LUID ParentTokenId;                  // Ro: 8–Bytes
        LARGE_INTEGER ExpirationTime;        // Ro: 8–Bytes
        LUID ModifiedId;                     // Wr: 8–Bytes
```

```
ULONG UserAndGroupCount;                                    // Ro: 4–Bytes
ULONG RestrictedSidCount;                                   // Ro: 4–Bytes
ULONG PrivilegeCount;                                       // Ro: 4–Bytes
ULONG VariableLength;                                       // Ro: 4–Bytes
ULONG DynamicCharged;                                       // Ro: 4–Bytes
ULONG DynamicAvalaible;                                     // Wr: 4–Bytes (Mod)
ULONG DefaultOwnerIndex;                                    // Wr: 4–Bytes (Mod)
PSID_AND_ATTRIBUTES UserAndGroups;                          // Wr: 4–Bytes (Mod)
PSID_AND ATTRIBUTES RestrictedSIDs;                         // Ro: 4–Bytes
PSID PrimaryGroup;                                          // Wr: 4–Bytes (Mod)
PLUID_AND_ATTRIBUTES privileges;                            // Wr: 4–Bytes (Mod)
PULONG DynamicPart;                                         // Wr: 4–Bytes (Mod)
PACL DefaultDacl;                                           // Wr: 4–Bytes (Mod)
TOKEN_TYPE TokenType;                                       // Ro: 1–Byte
SECURITY_IMPERSONATION_LEVEL ImpersonationLevel;            // Ro: 1–Byte
UCHAR TokenFlags;                                           // Ro: 4–Bytes
BOOLEAN TokenInUse;                                         // Wr: 1–Byte
PSECURITY_TOKEN_PROXY_DATA ProxyData;                       // Ro: 4–Bytes
PSECURITY_TOKEN_AUDIT_DATA AuditData;                       // Ro: 4–Bytes
ULONG VariablePart;                                         // Wr: 4–Bytes (Mod)
} TOKEN, * PTOKEN;
```

Obsérvese que cuando un testigo normal (no restringido) se crea ahora, a través de una API de CreateToken, el campo RestrictedSIDs está vacío, como es el campo ParentTokenId.

Para crear un testigo 84 restringido, un proceso llama a la API CreateRestrictedToken con ajustes de bandera de y/o la información en el campo de entradas apropiados, que a su vez invoca a la API NtFilterToken. Tal como se representa comenzando en la etapa 400 de la figura 4, la API NtFilterToken realiza una comprobación para ver si una bandera denominada DISABLE_MAX_SID está a nivel alto, que indica que todos los ID de seguridad para grupos en el nuevo testigo 84 restringido deben estar marcados como USE_FOR_DENY_ONLY. La bandera proporciona una forma conveniente para restringir los (posiblemente muchos) grupos en un testigo sin necesidad de identificar individualmente cada uno de los grupos. Si la bandera está a nivel alto, la etapa 400 se ramifica a la etapa 402 que ajusta un bit que indica USE_FOR_DENY_ONLY en cada uno de los ID de seguridad de grupo en el nuevo testigo 84.

Si la bandera DISABLE_MAX_SID no está a nivel alto, entonces la etapa 400 se ramifica a la etapa 404 para comprobar si cualesquiera ID de seguridad se enumeran individualmente en un campo SidsToDisable de la API NtFilterToken. Tal como se muestra en la etapa 404 de la figura 4, cuando el campo SidsToDisable de entrada opcional está presente, en la etapa 406, cualesquiera ID de seguridad enumerados en el mismo que están presentes también en el campo 62 UserAndGroups del testigo 60 de progenitor se marcan individualmente como USE_FOR_DENY_ONLY en el campo UserAndGroups 88 del nuevo testigo 84 restringido. Tal como se describe anteriormente, tales ID de seguridad pueden usarse sólo para denegar el acceso y no pueden usarse para conceder acceso, y además, no pueden posteriormente eliminarse o habilitarse. Por lo tanto, en el ejemplo que se muestra en la figura 2, el ID de seguridad del grupo2 está marcado como USE_FOR_DENY_ONLY en el testigo 84 restringido habiendo especificado el ID de seguridad del grupo2 en el campo SidsToDisable de entrada de la API NtFilterToken 86.

EL proceso de filtro continúa entonces hasta la etapa 410 de la figura 4, en la que se comprueba una bandera denominada DISABLE_MAX_PRIVILEGES. Esta bandera puede ajustarse de forma similar como un método abreviado conveniente para indicar que todos los privilegios en el nuevo testigo 84 restringido deben eliminarse. Si está a nivel alto, la etapa 410 se ramifica a la etapa 412 que borra todos los privilegios del nuevo testigo 84.

Si la bandera no está a nivel alto, la etapa 410 se ramifica a la etapa 414 en la que se examina el campo PrivilegesToDelete opcional. Si está presente cuando la API NtFilterToken 86 se llama, entonces en la etapa 416, cualquier privilegio enumerado en este campo de entrada que está presente también en el campo 68 de privilegios del testigo 60 existente se elimina individualmente del campo 90 de privilegios del nuevo testigo 84. En el ejemplo que se muestra en la figura 2, los privilegios que se muestran como “$Privilegio_s$” a “$Privilegio_m$” se han eliminado del campo 90 de privilegios del nuevo testigo 84 habiendo especificado esos privilegios en el campo PrivilegesToDelete de entrada de la API NtFilterToken 86. Esto proporciona la capacidad de reducir los privilegios disponibles en un testigo. El proceso continúa hasta la etapa 420 de la figura 5.

Al crear un testigo 84 restringido, si los SID están presentes en el campo RestrictingSids de entrada en la etapa 420, entonces se hace una determinación acerca de si el testigo de progenitor es un testigo normal o es en sí mismo un testigo restringido que tiene SID restringidos. Una API, IsToken restringida se llama en la etapa 422, y soluciona este asunto consultando (a través de la API NtQueryInformationToken) el campo RestrictingSids del testigo de progenitor para ver si no es NULL, mediante lo cual si no es NULL, el testigo de progenitor es un testigo restringido y la API devuelve un VERDADERO. Si la comprobación no se satisface, el testigo de progenitor es un testigo normal y la API

devuelve un FALSO. Obsérvese que con fines de las etapas 426 o 428 posteriores, un testigo de progenitor que está restringido pero que no tiene SID restringidos (es decir, que tiene privilegios eliminados y/o SID USE_FOR_DENY_ONLY) puede tratarse como que no está restringido.

En la etapa 424, si el testigo de progenitor está restringido, la etapa 424 se ramifica a la etapa 426 en el que cualesquiera ID de seguridad que está tanto en el campo de ID de seguridad restringida del testigo de progenitor como en la lista de entrada de ID de seguridad restringida de la API se colocan en el interior del campo 92 de ID de seguridad restringida del nuevo testigo 84. Requiriendo que los ID de seguridad restringida sean comunes a ambas listas evita que un contexto de ejecución restringida añada más ID de seguridad al campo 92 de ID de seguridad restringida, un evento que efectivamente aumentaría en lugar de disminuir el acceso. De forma similar, si ninguno es común en la etapa 426, cualquier testigo creado todavía tiene que restringirse sin aumentar el acceso del mismo, tal como dejando al menos un SID restringido a partir del testigo original en el nuevo testigo. Por lo demás, un campo de SID restringidos vacío en el nuevo testigo puede indicar que el testigo no está restringido, un evento que efectivamente aumentaría en lugar de disminuir el acceso.

Alternativamente, si en la etapa 424 se determina que el testigo de progenitor es un testigo normal, entonces en la etapa 428 el campo RestrictingSids 92 del nuevo testigo 84 se ajusta a lo enumerado en el campo de entrada. Obsérvese que a pesar de que esto añade unos ID de seguridad, el acceso se disminuye realmente debido a que un testigo que tiene SID restringidos se somete a una comprobación de acceso secundaria, tal como se describe en más detalle a continuación.

En último lugar, la etapa 430 se ejecuta también, mediante lo cual el ParentTokenId 93 en el nuevo testigo 84 se ajusta al TokenId del testigo existente (progenitor). Esto proporciona al sistema operativo la opción de permitir posteriormente que un proceso use una versión restringida de su testigo en lugares que en los normalmente no se le permitiría excepto para el testigo de progenitor.

Volviendo a la explicación del funcionamiento de testigos restringidos con referencia particular a las figuras 6 a 8, tal como se representa en la figura 6, un proceso 94 restringido se ha creado y está intentando abrir un objeto 70 de archivo con acceso de lectura/escritura. En el descriptor de seguridad del objeto 72, la ACL 80 tiene un número de ID de seguridad enumeradas en el mismo junto con el tipo de acceso que se permite para cada ID, en el que “RO” indica que se permite acceso de sólo lectura, “WR” indica acceso de lectura/escritura y “SINC” indica que se permite acceso de sincronización. Obsérvese que se deniega específicamente el acceso a “XJones” al objeto 72, incluso si a “XJones” se le permite por lo demás el acceso a través de la pertenencia a un grupo permitido. Además, no se permite al proceso 94 que tiene este testigo 84 asociado con el mismo obtener acceso a ningún objeto a través del ID de seguridad “Baloncesto” en el testigo 84, debido a que este identificador está marcado “DENEGAR” (es decir, USE_FOR_DENY_ONLY).

Con fines de seguridad, los contextos restringidos de seguridad se implementan principalmente en el núcleo de Windows NT. Para intentar obtener acceso al objeto 72, el proceso 94 dota al administrador 74 de objeto de una información que identifica el objeto al que se desea el acceso junto con el tipo de acceso deseado, (figura 8, la etapa 800). En respuesta, tal como se representa en la etapa 802, el administrador 74 de objeto funciona junto con el mecanismo 78 de seguridad para comparar el usuario y las ID de seguridad de grupo enumeradas en el testigo 84 (asociada con el proceso 94) frente a las entradas en la ACL 80, para determinar si el acceso deseado debe concederse o denegarse.

Tal como se representa generalmente en la etapa 804, si no se permite el acceso al usuario o los grupos enumerados, la comprobación de seguridad deniega el acceso en la etapa 814. Sin embargo, si el resultado de la parte de grupo y usuario de la comprobación de acceso indica acceso disponible en la etapa 804, el proceso de seguridad se ramifica a la etapa 806 para determinar si el testigo 84 restringido tiene cualesquiera ID de seguridad restringida. Si no, no existen restricciones adicionales, mediante lo cual la comprobación de acceso se ha completado y se concede el acceso en la etapa 812 (se devuelve un manejador para el objeto) basándose únicamente en acceso de grupo y usuario. De esta forma, un testigo normal se comprueba esencialmente como antes. Sin embargo, si el testigo incluye unos ID de seguridad restringida tal como se determinó mediante la etapa 806, entonces se realiza una comprobación de acceso secundaria en la etapa 808 comparando los ID de seguridad restringida frente a las entradas en la ACL 80. Si esta comprobación de acceso secundaria permite el acceso en la etapa 810, se concede el acceso al objeto en la etapa 812. Si no, se deniega el acceso en la etapa 814.

Tal como se representa lógicamente en la figura 7, se realiza por lo tanto una comprobación en dos partes siempre que unas ID de seguridad restringida están presentes en el testigo 84. Considerando los ID de seguridad en el testigo 84 y los bits 96 de acceso deseado frente al descriptor de seguridad del objeto 72, tanto la comprobación de acceso normal como (AND Bit a bit) la comprobación de acceso de los ID de seguridad restringida deben conceder acceso con el fin de que se conceda acceso al proceso al objeto. A pesar de que no es necesario para la invención, tal como se describe anteriormente, la comprobación de acceso normal tiene lugar en primer lugar, y si se deniega el acceso, no es necesaria una comprobación adicional. Obsérvese que puede denegarse el acceso o bien debido a que ningún ID de seguridad en el testigo se corresponda con un identificador en la ACL, o debido a que específicamente se deniega el acceso a una entrada de ACL al testigo basándose en un identificador de seguridad en el mismo.

Por lo tanto, en el ejemplo que se muestra en la figura 6, no se concede acceso al objeto 72 al proceso 94 debido a que el único SID restringido en el testigo 84 (campo 92) identifica “Internet Explorer”, mientras que no existe un SID restringido homólogo en la ACL del objeto 80. A pesar de que el usuario tenía el derecho a obtener acceso al objeto a través de una ejecución de proceso con un testigo normal, el proceso 94 se restringió con el fin de ser capaz sólo de obtener acceso a objetos que tienen un SID de “Internet Explorer” (no DENY) en sus ACL.

Obsérvese que en lugar de especificar un tipo de acceso, el autor de llamada puede haber especificado un acceso MAXIMUM_ALLOWED, mediante lo cual tal como se describe anteriormente, un algoritmo pasa a través de la ACL 80 determinando el acceso máximo. Con testigos restringidos, si se concede algún tipo de acceso de usuario o grupo en absoluto, el tipo o tipos de derechos de acceso disponibles siguiendo la ejecución del usuario y grupos se especifica como el acceso deseado para la segundo ejecución, que comprueba la lista RestrictedSIDs. De esta forma, es seguro que va a concederse a un testigo restringido menos o igual acceso que a un testigo normal.

En último lugar, debe observarse que el modelo de seguridad de lo que se describe en el presente documento puede usarse junto con otros modelos de seguridad. Por ejemplo, los modelos de seguridad basados en capacidad que residen en la parte superior de un sistema operativo pueden usarse antes que el modelo de seguridad a nivel de sistema operativo de la presente invención.

Trabajos

Un Trabajo es un objeto de núcleo que tiene un conjunto de procesos organizados en el mismo, en el que cada trabajo puede tener un tipo de restricción diferente asociada con el mismo. De acuerdo con la presente invención, los testigos restringidos pueden integrarse con unos objetos de trabajo de Windows NT para permitir la administración de múltiples procesos que se ejecutan bajo las mismas restricciones. Una restricción de objeto de trabajo se expone a continuación:

```
Typedef struct _JOBOBJECT_SECURITY_LIMIT_INFORMATION {
        ULONG SecurityLimitFlags ;
        MANEJADOR JobToken ;
        PTOKEN_GROUPS SidsToDisable ;
        PTOKEN_PRIVILEGES PrivilegesToDelete ;
        PTOKEN_GROUPS RestrictedSIDs ;
} JOBOBJECT_SECURITY_LIMIT_INFORMATION,
*PJOBOBJECT_SECURITY_LIMIT_INFORMATION;
```

en la que la SecurityLimitFlags relevante puede ser:

```
#define WORK_OBJECT_SECURITY_FILTER_TOKENS    0x00000008
```

Estos diversos fragmentos de información pueden ajustarse en un objeto de trabajo que usa una API de NtSetInformationJobObject, mientras que se asigna un proceso a un trabajo que usa la API NtAssignProcessToJobObject. El límite de seguridad ajustado en el trabajo tiene lugar cuando se asigna un proceso. Para restringir un trabajo, la bandera límite WORK_OBJECT_SECURITY_FILTER_TOKENS está a nivel alto, mediante lo cual el testigo primario del proceso al que se asigna el trabajo se filtra usando la información de SidsToDisable, PrivilegesToDelete y RestrictedSIDs proporcionada en la información límite de seguridad, de una forma similar a cómo los testigos asociados con procesos se filtran tal como se describe anteriormente.

Tal como se muestra en la figura 9, un trabajo 110 tiene un número de procesos (por ejemplo, los procesos 112 a 114) asignados al mismo a través de la API de trabajo NtAssignProcessToJobObject. Cada proceso 112 a 114 tiene un testigo 116_a a 116_c respectivo asociado con el mismo que es el mismo con respecto a sus restricciones, que se muestran como “Restricciones R”. Por ejemplo, el objeto 110 de trabajo restringe los procesos 112 a 114 en el mismo a realizar sólo determinadas operaciones, debido a que los testigos 116_a a 116_c bajo los que se ejecutan tienen determinados ID de seguridad (por ejemplo, SID de Administrador) deshabilitados, determinados privilegios eliminados y/o un conjunto de ID de seguridad restringida añadido. Obsérvese que los testigos pueden ser el mismo con respecto a sus otros derechos de acceso también, en cuyo caso todos los testigos son esencialmente idénticos, si bien esto no se requiere. Si un proceso (por ejemplo, el proceso 114) produce otro proceso 118, este proceso 118 también se ejecuta en el trabajo 110 con las mismas restricciones R, puesto que el objeto 110 de trabajo asegura que las mismas restricciones R son asociada con el nuevo proceso 118 a través de su testigo 116_d.

Ejecución de contenido de poca confianza

De acuerdo con un aspecto de la presente invención, se testigos restringidos usan para configurar contextos restringidos de seguridad para ejecución de contenido de poca confianza. Un tipo de contenido que normalmente no es de confianza es el contenido descargado a partir de un sitio de Internet. Para restringir tal contenido, un proceso restringido puede configurarse para la ejecución de cada sitio por el que se navega, asociando el proceso con (al menos) un testigo restringido genéricamente junto con otras restricciones. Obsérvese que el proceso puede también

estar en el interior de un objeto de trabajo, mediante lo cual cualesquiera procesos creados por el proceso del sitio se dan automáticamente las mismas restricciones. Otra ventaja para los objetos de trabajo es que las operaciones basadas en ventanas pueden restringirse, de modo que, por ejemplo, un proceso no puede apagar la máquina u obtener acceso a los datos de portapapeles. En todo caso, cualquier acción que realiza el contenido del sitio, tal como a través de HTML dinámico, controles de Java o de Active–X, tiene lugar en el interior del proceso, mediante lo cual se somete a las restricciones del proceso. Obsérvese que diferentes tramas se tratan como sitios diferentes, con independencia de su sitio fuente actual, incluso a pesar de que múltiples tramas pueden visualizarse simultáneamente en ventanas en la pantalla.

A modo de ejemplo, tal como se muestra en la figura 10, cualquier contenido al que se obtiene acceso a través de un navegador 130, tal como Internet Explorer, puede ejecutarse en un proceso 132 que tiene un testigo 134 restringido incluyendo un SID de “Internet Explorer” restringido en su campo 136 de SID restringidos. Tal como se describe anteriormente, esto evita automáticamente el acceso a cualquier recurso que no tiene un permitir entrada que se corresponde con el SID de Internet Explorer, (a menos que el acceso sea a través de otro SID restringido correspondiente). Por lo tanto, por ejemplo, el proceso 132 puede presentar páginas de HTML en una ventana de Internet Explorer, pero tiene restringido el hacer mucho más a menos que una ACL de un recurso específicamente incluya una entrada con un SID restringido correspondiente que permita la acción.

Otra forma en la que restringir el contenido descargado de Internet es restringir el acceso a unos recursos basándose en la identidad del sitio. Por ejemplo, cada sitio tiene un URL (Localizador Uniforme de Recursos) único, y puede tener un ID de certificado binario. Para una restricción basándose en el URL, tal como se muestra en la figura 11, el navegador 130 actúa como un mecanismo de discriminación que pasa el sitio URL 138 a un convertidor 140 que convierte la cadena de URL en un SID restringido. Un mecanismo de este tipo usa una función hash criptográfica unidireccional tal como MD5 para convertir la cadena en un valor binario de 128 bits, que entonces se hace un SID restringido añadiendo una pequeña cantidad de información tal como una cabecera o similar al mismo que indica que el número es un SID y cómo se generó el número. Un SID de este tipo es, con una probabilidad extremadamente alta, único con respecto a los otros SID. Obsérvese que si un ID de certificado binario está disponible para el sitio, entonces ese valor puede usarse para generar el SID restringido. En todo caso, tal como se muestra en la figura 10, el SID restringido se ubica en el campo 136 de SID restringidos del testigo 134 restringido asociado con el proceso 132 configurado para ese sitio.

Además, la restricción puede basarse en zonas de Internet (una colección de sitios). Por ejemplo, tal como también se muestra en la figura 11, el usuario puede configurar una zona que comprende una lista de sitios específicamente de confianza, y otra para sitios de específicamente de poca confianza. La zona de confianza se corresponderá con un SID restringido mientras que la zona de poca confianza se corresponderá con otro. Las zonas pueden también configurarse para distinguir entre sitios de Intranet y sitios de Internet. Una distinción adicional puede hacerse para sitios de Internet que tienen un tipo de certificado y así sucesivamente, para cualquier granularidad deseada en la que puede hacerse una distinción. En todo caso, el navegador 130 tiene acceso a una información 131 que agrupa los sitios en zonas, mediante lo cual puede generarse un SID restringido que se corresponde con la zona del sitio, y ubicarse en el testigo 136 restringido para identificar la zona para unos recursos de sistema para unas comprobaciones de seguridad basadas en ACL.

Tal como se describe anteriormente, una vez que un SID restringido está presente en un testigo, el mecanismo 78 de seguridad realiza una evaluación de acceso con una comprobación de usuario y de grupo seguida por (si la comprobación de usuario y grupo se pasa) una comprobación de los SID restringidos. Como resultado, la ACL para cada recurso determina cómo se maneja un código de poca confianza por la presencia o ausencia de unas entradas correspondientes en el mismo. Por ejemplo, un objeto de archivo puede permitir el acceso de lectura a un sitio o zona específicamente de confianza, pero no a cualquier otro tipo de acceso o cualquier acceso que sea a cualquier otro sitio o zona. Un archivo altamente confidencial puede específicamente denegar el acceso a cualquier testigo restringido que tiene el SID de “Internet Explorer” en el mismo con el fin de evitar cualquier acceso a través de algún otro SID restringido. Por lo tanto, por ejemplo, tal como se muestra en la figura 10, el recurso A (142) permite el acceso de lectura al proceso 132 debido a que su ACL 144 contiene una entrada de “Sitel.com” que se corresponde con el SID restringido de “Sitel.com” en el testigo 134 restringido del proceso 132. Sin embargo, el Recurso B (146) no permitirá el acceso debido a que no está presente una entrada correspondiente.

Si bien el mecanismo anterior funciona para restringir el acceso, surgen determinadas dificultades en las que los recursos pueden carecer de la granularidad para discriminar basándose en las diversas distinciones tales como los sitios o zonas. Por ejemplo, el proceso configurado para un sitio de Internet particular puede necesitar obtener acceso a ese sitio. Sin embargo, el controlador que maneja los sockets de Windows (el controlador de dispositivo AFD) tiene su ACL configurada para o bien permitir o bien denegar la conexión a redes, pero no puede en la práctica distinguir entre cada sitio o zona. Denegar la conexión a redes completamente evita que un proceso obtenga acceso a su propio sitio correspondiente, pero permitir que un proceso obtenga acceso a cualquier sitio permite el acceso del proceso a otros sitios, lo que puede potencialmente divulgar una información confidencial acerca de un cliente. Las razones para restringir el acceso de red incluyen evitar el acceso a archivos no seguros en servidores de archivo que normalmente no son accesibles desde Internet, evitar el acceso a no seguros bases de datos internas, evitar ataques a los recursos de red (por ejemplo a través de paquetes o respuestas de DHCP defectuosas) y evitar fugas de información acerca de otras formas posibles de alcanzar la red.

Para resolver este dilema restringiendo de forma selectiva un proceso a determinados sitios, la presente invención generalmente restringe la conexión a redes del proceso, pero emplea un ayudante de proceso de confianza que puede realizar la conexión a redes para el sitio. El ayudante de proceso restringe el proceso en un grado que permite que el proceso restringido obtenga acceso sólo a sitios seleccionados (por ejemplo, todos los sitios que terminan en “Microsoft.com”) tal como se determinó por el ayudante de proceso. Más particularmente, tal como se muestra en la figura 13, se deniega al proceso 132 restringido un acceso directo al controlador 150 de AFD de sockets de Windows, (tal como se representa por la “X”), mediante lo cual el proceso 132 restringido no puede directamente conseguir el manejador de socket para ningún sitio, incluyendo el propio. Obsérvese que denegando el acceso a procesos restringidos a través de la ACL 152 del controlador 150 de dispositivo, puede no obtenerse acceso al controlador 150 de dispositivo por el proceso 132 restringido a través de alguna otra forma, tal como llamando directamente al núcleo. Sin embargo, un ayudante 154 de proceso, que es de confianza, tiene acceso al controlador 150 a través de su propio testigo 156 no restringido y puede recuperar un manejador de socket. Modificando el API 158 Winsock connect() para invocar el ayudante 154 de proceso siempre que un proceso 132 restringido intenta recuperar un manejador de socket, el manejador para uno de los sockets verificados o permitidos de un sitio se devuelve cuando se usa el mecanismo apropiado (es decir, la API 158).

Más particularmente, tal como se muestra en la figura 14, cuando las API Winsock socket() y connect() 158 se llaman en la etapa 1400 para devolver un manejador para un socket, la API connect() 158 en primer lugar comprueba en la etapa 1502 para ver si el proceso 132 al que llama está restringido a través de un SID restringido apropiado en su testigo 134. Si no está restringido, la etapa 1402 se ramifica a la etapa 1404, en el que la API connect() 158 funciona como antes para devolver un manejador de socket o un código de error apropiado (por ejemplo, no socket disponible, host inalcanzable, acceso denegado y así sucesivamente).

Sin embargo, cuando un proceso 132 restringido hace la llamada connect()tal como se determinó mediante la etapa 1402, la API connect() 158 llama al ayudante 154 de proceso (figuras 13 y 18) en la etapa 1406. Tal como se representa mediante la etapa 1510 de la figura 15, el ayudante 154 de proceso extrae la información de sitio del sitio a partir del testigo 134 restringido, y en la etapa 1512, compara las direcciones permitidas (o denegadas) del sitio con la dirección solicitada. Si no están permitidas, la etapa 1512 se ramifica a la etapa 1514 en el que la petición de socket se deniega efectivamente tal como ajustando un código de error apropiado. Si son la misma, la etapa 1512 se ramifica a la etapa 1516, en el que el ayudante 154 de proceso accede al controlador 150 de dispositivo para obtener el manejador de socket al sitio que se corresponde con el proceso 132 (suponiendo que un manejador está disponible y que no existen otros errores). El ayudante 154 de proceso (figura 15) duplica el manejador en la etapa 1518, entonces vuelve a la API connect() 158 (figura 14) con o bien el manejador duplicado o bien alguna indicación de un error, después de que la API connect() 158 devuelva el manejador de socket o un código de error al proceso 132 restringido en la etapa 1420 (figura 14). De esta forma, si se devuelve un manejador, el proceso 132 restringido puede después de eso acceder al sitio verificado/ permitido a través del manejador, pero no a otros sitios. Obsérvese que el ayudante 154 de proceso es de confianza, y se escribe de forma cuidadosa para permitir sólo el manejador de socket de ese sitio.

De forma similar, el Wininet.dll que (entre otras funciones) descarga datos identificados de URL al nivel de archivo emplea un ayudante de proceso cuando un proceso restringido descarga un archivo. El ayudante de proceso extrae y valida el URL que un sitio está solicitando, y si se valida, realiza la descarga y otro Wininet funciona mientras que proporciona notificaciones al proceso restringido. Tal como puede observarse fácilmente, permitir que el ayudante de proceso ajuste las ACL en los archivos descargados asegura esos archivos respecto de otros sitios. De nuevo, cuando un proceso restringido llama la API relacionada con wininet, la API invoca al ayudante de proceso apropiado.

Además de restringir el acceso a otros sitios, se restringe a un proceso restringido configurado para un sitio el acceso a archivos diferentes del propio. Con este fin, cualquier archivo creado por un sitio en el sistema del usuario se ubica en una carpeta que tiene una entrada de ACL que se corresponde específicamente con ese sitio. Por ejemplo, tal como se muestra en la figura 16, (en la que se representan carpetas como triángulos para indicar la estructura jerárquica de las carpetas), cada sitio (Site$_1$.com a sitio$_n$.com) tiene su propia subcarpeta 160$_1$ a 160$_n$ en el sistema de archivos, tal como bajo una carpeta 162 de archivos de sitio de Internet. Debido a que sólo el proceso de ese sitio (por ejemplo, el sitio$_1$.com) tendrá un SID restringido en su testigo que se corresponde con la entrada en su ACL 164$_1$, ningún otro sitio (por ejemplo, el sitio$_2$.com) será capaz de obtener acceso a sus archivos, y viceversa. Tal aislamiento posibilita que los sitios obtengan acceso (por ejemplo, crear, abrir, lectura, escritura y borrado) a sus propios archivos, pero evita que el contenido de otro sitio obtenga acceso a esos archivos.

Tal como se muestra en la figura 17, el mecanismo para llevar a cabo el aislamiento de archivos basándose en un sitio puede construirse en el interior de las API 170 para la apertura y la creación de archivos. Las API para la creación y la apertura de archivos se configuran para reconocer (examinando los SID restringidos en el testigo 134 restringido) cuando se llama un proceso 132 configurado para un sitio de Internet, y ajustar la ruta para ubicar y/o localizar adecuadamente cada archivo basándose en el testigo 134 restringido. Obsérvese que debido a que otras operaciones de archivo (por ejemplo, lectura, escritura y borrado) usan el manejador de archivos devuelto por estas API, las otras API no necesitan modificarse para la redirección de ruta.

De forma similar, tal como se muestra en la figura 18, un proceso 132 restringido puede sólo indirectamente obtener acceso al registro 174 de sistema, pero de acuerdo con la invención, sólo a través de una vista virtualizada del

registro, por ejemplo, bajo su propio nombre clave basado en el sitio (con independencia del nombre clave proporcionado por el proceso). Por ejemplo, un sitio puede incluir un control Active–X que intenta escribir en el registro 174 de sistema. Para proporcionar una vista virtualizada del registro para un proceso restringido, las API 176 de WIN32 de registro para obtener acceso al registro (por ejemplo, las API RegSaveKey() y RegRestoreKey()) redirigen unos procesos restringidos a y desde sus subárboles 178 específicos de sitio en el registro (o copia del mismo). De esta forma, el contenido de poca confianza cree que puede registrar información en el registro 174 en la ubicación especificada, pero realmente se restringe el acceso al subárbol o registro virtual 178, por lo tanto aislando efectivamente la información de registro existente.

Otra forma en la que el contenido de poca confianza puede descargarse a una máquina es a través de correo electrónico (correo electrónico). Con el correo electrónico, el nivel de confianza depende de quien envíe el mensaje, y, si hay cualesquiera conjuntos de datos adjuntos, el tipo (y si se conoce, el origen) de los conjuntos de datos adjuntos. La figura 12 muestra a modo de ejemplo unos componentes para determinar la confianza basándose en la identidad del remitente, en la que una aplicación 133 de correo pasa la identidad 135 a un mecanismo 137 de discriminación. El mecanismo 137 de discriminación determina un nivel de confianza del usuario con el fin de convertir la identidad del usuario (y otros criterios tales como alguna autenticación que verifique que la identidad es verdadera) en un SID 139 restringido, tal como consultando el SID 139 restringido en un nivel de confianza en la tabla 141 de SID o similar. El proceso 143 en el que el contenido de correo se ejecuta se asocia con un testigo 145 restringido que incluye el SID 139 restringido.

La figura 19 representa la lógica general usada por el mecanismo 137 de discriminación para una forma a modo de ejemplo de restringir el acceso basándose en el remitente de un mensaje de correo electrónico. Obsérvese que el campo “De” del mensaje se usa para determinar la supuesta identidad del remitente, mientras que un mecanismo de autenticación (por ejemplo, en el interior de la aplicación 133 de correo electrónico) tal como a través de una firma digital puede usarse para verificar que el remitente es de hecho la fuente del mensaje. Debe además observarse que las restricciones basadas en fuente pueden configurarse de forma similar para las noticias en línea, mediante lo cual el contenido de noticias descargado se ejecuta en un contexto restringido basándose en la identidad del remitente.

A modo de ejemplo, considérese la siguiente política simplificada que puede configurarse en una máquina, en la que un remitente de mensaje de correo electrónico es o bien de mucha confianza, de confianza o de poca confianza, y el remitente puede o puede no estar autenticado. Por ejemplo, el supervisor de un individuo y sus compañeros de trabajo inmediatos pueden ser remitentes de mucha confianza, empleados y amigos cercanos remitentes de confianza, y todos los demás de poca confianza. En la máquina, los ACL de recurso se configuran con entradas que se corresponden con SID restringidos tales que un proceso con un testigo restringido que tiene un SID restringido que indica un nivel de confianza de uno puede leer y escribir archivos, mientras que un proceso de nivel de confianza dos puede sólo leer archivos. Se deniega el acceso a un proceso con un testigo restringido que tiene un SID restringido que indica un nivel de confianza de tres a todos los recursos. Un nivel de confianza de cero significa que el testigo normal (no restringido) será asociado con el proceso.

Tal como se muestra en la figura 19 en la etapa 1900, si el usuario es de mucha confianza (por ejemplo, comparando el nombre en el campo “De” con una lista de nombres), entonces la etapa 1902 se ejecuta para determinar si el remitente está autenticado. Por ejemplo, el mensaje puede estar firmado digitalmente de modo que se garantice al destinatario la verdadera identidad del remitente. Si está autenticado, entonces la etapa 1904 ajusta el nivel de confianza a cero y la etapa 1906 asocia el testigo normal con el proceso configurado para el mensaje. Si no está autenticado, la etapa 1908 ajusta el nivel de confianza a dos, después de lo cual la etapa 1920 crea un testigo restringido para el proceso que configura para el mensaje basándose en el nivel de confianza de dos y asocia el testigo restringido con la configuración de proceso para el mensaje. Obsérvese que los SID restringidos para niveles de confianza de uno, dos y tres están predeterminados y están almacenados en la máquina tal como en la tabla 141 o similares.

Si el usuario no es de mucha confianza entonces la etapa 1900 se ramifica a la etapa 1910 para determinar si el usuario es de confianza o de poca confianza. De acuerdo con la política de seguridad, si es de confianza, la etapa 1910 se ramifica a la etapa 1912 para determinar si el remitente está autenticado. Si es así, el proceso de mensaje se configura con el fin de ser nivel uno de confianza (acceso de lectura y de escritura) en las etapas 1914 y 1920. Si se autentica en la etapa 1912, entonces el nivel de confianza se ajusta a dos (sólo lectura) en la etapa 1908 y el proceso restringido por consiguiente en la etapa 1920, de nuevo buscando e insertando el SID restringido apropiado en el interior del testigo restringido. En último lugar, si el usuario es de poca confianza, la etapa 1910 se ramifica a la etapa 1916 en la que el nivel de confianza se ajusta a tres y en la etapa 1920 el proceso se restringe de manera correspondiente con el fin de denegarse el acceso a los recursos del sistema. Obsérvese que la política de seguridad anterior sugiere la implementación a través de un sistema de puntos, es decir, diversos puntos acumulados que dependen de la identidad del remitente y diversos puntos si se autentican, y determinar un nivel de acceso basándose en los puntos totales.

Tal como puede observarse fácilmente, una política similar puede configurarse para restringir conjuntos de datos adjuntos. Sin embargo, Además de restringir basándose en el remitente de unos datos adjuntos, puede comprobarse el tipo de los datos adjuntos. Por ejemplo, archivos ejecutables (*.exe, *.bat, *.com y así sucesivamente) pueden diferenciarse de documentos de sólo texto, que a su vez se tratan de manera diferente con respecto a restricciones

que los documentos de Microsoft Word y Microsoft Excel (que puede contener virus u otro código que no sigue las reglas a través de macros). De hecho, puede emplearse y/o combinarse cualquier número de criterios con otros criterios para configurar contextos restringidos de seguridad de acuerdo con una política de seguridad deseada.

Volviendo a una consideración de contenido de poca confianza en servidores web, procesos de cliente, secuencias de comandos y otros programas ayudantes tales pueden limitarse a través de contextos de ejecución restringida como según lo que pueda hacer tal contenido. De acuerdo con otro aspecto de la presente invención, un primer modo para que un servidor web use contextos es lanzar todo este contenido en contextos restringidos que les impiden dañar dato sistema operativo de núcleo e interferir con los datos de otros programas ayudantes. Tal como se describe anteriormente, esto puede lograrse configurando un proceso asociado con un testigo restringido adecuadamente para cada fragmento de contenido de poca confianza, denegando así el acceso a archivos de datos del sistema a través de sus ACL y aislando cualquiera de los archivos del contenido de poca confianza tal como se desee. Obsérvese que la protección de memoria está ya presente a través de límites de proceso.

Además, las restricciones pueden basarse en la secuencia de comandos del servidor que se ejecuta, tal como discriminando de acuerdo con el autor de la secuencia de comandos y/o determinando si la secuencia de comandos necesita obtener acceso a cualquiera de los archivos o necesita compartir archivos con otros usuarios o archivos del sistema. Por ejemplo, tal como se muestra en la figura 20, mediante la ejecución de una secuencia 200 de comandos de servidor web ayudante en un proceso 202 que está restringido creando un testigo 204 restringido incluyendo a SID 206 restringido especialmente para el mismo, una secuencia de comandos pueden estar limitada sólo a aquéllos recursos que necesita y ningún otro, limitándola efectivamente de para no realice nada distinto de lo que se pretende que haga. Por lo tanto, tal como se muestra en la figura 20, la secuencia de comandos$_1$ 200 puede obtener acceso al recursoX (218) y el recursoZ (226) debido a que cada uno de sus respectivos ACL 220, 228 incluye una entrada de “secuencia de comandos$_1$”. De forma similar, la secuencia de comandos$_2$ 210 puede obtener acceso al recursoY (222) y el recursoZ (226) debido a que cada uno de sus respectivos ACL 224, 228 incluye una entrada de “secuencia de comandos$_2$”. Por lo tanto, si los recursos que se muestran son archivos de datos, la secuencia de comandos$_1$ no puede interferir con los datos de secuencia de comandos$_2$ y viceversa, excepto posiblemente en el recursoZ, sin embargo el recursoZ está diseñado para ser un archivo compartido. Por ejemplo, a través de sus entradas de ACL, el recurso Z (226) puede ser uno archivo de datos de sólo lectura con respecto a estos procesos 202, 212, mediante lo cual ambas secuencias de comandos 200, 210 pueden usar la información del archivo sin cambiarla.

Otra opción proporcionada por la presente invención es crear un objeto de trabajo, que tal como se describe anteriormente, es un objeto de nivel de núcleo que contiene las restricciones para todos los procesos en el mismo. Además, un objeto de trabajo puede contener otras limitaciones tales como cuánto de un procesamiento de CPU puede ser consumido. Tal como se describe anteriormente, cualquier proceso añadido ese trabajo recibe las mismas restricciones, mediante lo cual el servidor web puede crear procesos en nombre del cliente y entonces añadir el proceso a un trabajo existente. Las restricciones se aplican entonces automáticamente al proceso.

De acuerdo con la presente invención, el servidor web puede además elegir qué restricciones aplicar a los procesos de cliente y similares basándose en cualquier número de criterios, incluyendo la identidad del cliente (por ejemplo, ya sean un empleado de una empresa o no), el procedimiento de autenticación usado por el cliente (por ejemplo, presentando una contraseña, un certificado, usando una tarjeta inteligente, o incluso una huella dactilar/exploración de retina), y también la ubicación del cliente (por ejemplo, basándose en si el cliente está en la misma red, marcando, o conectándose a través de Internet). La figura 21 muestra un diagrama de flujo a modo de ejemplo que muestra la lógica para configurar los niveles de confianza que dependen del tipo de autenticación, y que dependen de si la autenticación tuvo lugar a través de alguna conexión remota (y por lo tanto teóricamente menos segura), en contraposición a a través de la máquina local. Con fines de simplicidad, las etapas individuales y la lógica de la figura 21 no se describen en detalle, debido a que los niveles de confianza y su uso en recuperar unos SID restringidos correspondientes para configurar testigos restringidos se describen de forma similar anteriormente. Obsérvese, sin embargo, que en general, cuanto de mayor confianza es la autenticación, menor será el nivel de confianza asignado (que en el presente documento se corresponde con derechos de acceso aumentados).

En último lugar, un Proveedor de Servicios de Internet puede usar contextos de ejecución restringida para hospedar de forma segura múltiples sitios en un único servidor web. Con este fin, cada sitio web se ejecuta en un contexto de ejecución restringida separado compuesto por un objeto de trabajo y un testigo restringido. El objeto de trabajo proporciona unas cuotas, de modo que una parte de los recursos del servidor web se asigna a cada sitio web. El testigo restringido proporciona aislamiento entre los sitios web, protegiendo datos privados tales como listas de clientes, frente a un acceso no autorizado.

Tal como puede observarse a partir de la precedente descripción detallada, los contextos de ejecución restringida de la presente invención restringen el acceso de contenido de poca confianza a los recursos del sistema. Pueden aplicarse restricciones recurso a recurso, dependiendo de cualquier criterio disponible para el sistema.

Si bien la invención es susceptible de diversas modificaciones y construcciones alternativas, determinadas realizaciones ilustradas de la misma se muestran en los dibujos y se han descrito anteriormente en detalle.

REIVINDICACIONES

1. Un procedimiento de restricción de acceso de contenido a unos recursos para un sistema que tiene un mecanismo (78) de seguridad proporcionado por un sistema (35) operativo que determina el acceso de unos procesos (94) a unos recursos (142, 146) basándose en una información en un testigo de acceso asociada con cada uno de los procesos (94) frente a una información (144, 148) de seguridad asociada con cada uno de los recursos (142, 146), comprendiendo el procedimiento las etapas de, configurar un proceso (132) para el contenido, deducir un testigo (134) de acceso restringido a partir de un testigo de acceso, determinar una información (136) de restricción basándose en una información que se corresponde con el contenido, modificar dicho testigo (134) de acceso restringido con una información (136) de restricción, y usar el testigo (134) de acceso restringido como el testigo de acceso del proceso (132) del contenido.

2. El procedimiento según la reivindicación 1 en el que el contenido comprende unos datos que se obtienen a partir de una fuente de poca confianza.

3. El procedimiento según la reivindicación 2 en el que la fuente de poca confianza es un disco flexible.

4. El procedimiento según la reivindicación 2 en el que el contenido escribe un archivo en el sistema, y que además comprende las etapas de generar un identificador de seguridad restringida que se corresponde con el sitio, añadir el identificador de seguridad a la información de seguridad del archivo, y almacenar el archivo en el sistema.

5. El procedimiento según la reivindicación 2 en el que el contenido escribe un archivo en el sistema, y que además comprende la etapa de redirigir una ruta proporcionada por el contenido a una ruta asociada con el sitio.

6. El procedimiento según la reivindicación 2 en el que los datos comprenden unos datos de red y la fuente de poca confianza es un sitio de red.

7. El procedimiento según la reivindicación 6 en el que el sitio es un sitio de Internet, y la etapa de determinar una información de restricción incluye la etapa de generar un identificador de seguridad a partir de una información única del sitio de Internet.

8. El procedimiento según la reivindicación 7 en el que la información única comprende un identificador de certificado binario del sitio de Internet.

9. El procedimiento según la reivindicación 7 en el que la información única comprende un Localizador Uniforme de Recursos (URL) del sitio de Internet, y en el que la etapa de generar un identificador de seguridad incluye la etapa de convertir el URL en el identificador de seguridad restringida.

10. El procedimiento según la reivindicación 9 en el que la etapa de convertir el URL en el identificador de seguridad restringida incluye la etapa de ejecución de una función *hash* sobre el URL con una función hash criptográfica.

11. El procedimiento según la reivindicación 1 en el que el contenido comprende un mensaje de correo electrónico.

12. El procedimiento según la reivindicación 11 en el que la etapa de determinar una información de restricción incluye las etapas de determinar el remitente del mensaje.

13. El procedimiento según la reivindicación 12 en el que la etapa de determinar el remitente del mensaje incluye la etapa de autenticar el remitente.

14. El procedimiento según la reivindicación 1 en el que el contenido comprende una secuencia de comandos.

15. El procedimiento según la reivindicación 14 en el que el sistema es un servidor, y en el que la etapa de determinar una información de restricción incluye la etapa de determinar cómo se autenticó el cliente para el servidor.

16. El procedimiento según la reivindicación 1 en el que el contenido comprende unos datos descargados a partir de un sitio, que además comprende las etapas de determinar una zona que se corresponde con el sitio, y en el que la etapa de añadir la información de restricción a un testigo de acceso restringido incluye la etapa de añadir un identificador de seguridad restringida que se corresponde con la zona al testigo de acceso restringido.

17. El procedimiento según la reivindicación 1 que además comprende la etapa de obtener acceso al recurso a través de un ayudante de proceso.

18. El procedimiento según la reivindicación 17 en el que el ayudante de proceso extrae una información asociada con el contenido a partir del testigo restringido.

19. El procedimiento según la reivindicación 17 que además comprende las etapas de detectar en una interfaz de programación de aplicación un intento de obtener acceso a un recurso a partir del proceso que tiene al testigo restringido como su testigo de acceso, y en respuesta, llamar al ayudante de proceso para obtener acceso al

recurso, y devolver una información a partir del ayudante de proceso al proceso que tiene al testigo restringido como su testigo de acceso.

20. El procedimiento según la reivindicación 17 que además comprende la etapa de ajustar la información de seguridad del recurso para permitir obtener acceso al ayudante de proceso y denegar el acceso al proceso que tiene al testigo restringido como su testigo de acceso.

21. El procedimiento según la reivindicación 1 que además comprende la etapa de colocar el proceso en el interior de un objeto de trabajo.

22. El procedimiento según la reivindicación 1 en el que el contenido intenta obtener acceso a un registro de sistema, y que además comprende la etapa de redirigir una ubicación de registro proporcionada por el contenido a una ubicación de registro asociada con el contenido.

23. Un sistema para restringir la obtención de acceso de contenido a unos recursos (142, 146) para un sistema informático, que comprende, un proceso (132) configurado para el contenido, un mecanismo de discriminación para determinar al menos un identificador (136) de seguridad restringida basándose en una información que se corresponde con el contenido, un mecanismo para crear un testigo (134) de acceso restringido para el proceso (132) añadiendo el al menos un identificador (136) de seguridad restringida al testigo (134) de acceso restringido, y un mecanismo de seguridad para determinar el acceso del proceso (132) de contenidos a un recurso (142, 146) comparando una información en el testigo (134) de acceso restringido con la información de seguridad (144, 148) asociada con el recurso (142,146).

24. El sistema según la reivindicación 23 en el que el contenido comprende unos datos descargados a partir de un sitio.

25. El sistema según la reivindicación 24 en el que el sitio es un sitio de Internet, y en el que el mecanismo de discriminación genera un identificador de seguridad restringida a partir de una información del sitio de Internet.

26. El sistema según la reivindicación 25 en el que la información del sitio de Internet comprende un identificador de certificado binario del sitio de Internet.

27. El sistema según la reivindicación 25 en el que la información del sitio de Internet comprende un Localizador Uniforme de Recursos (URL), y que además comprende un convertidor para convertir el URL en el identificador de seguridad restringida.

28. El sistema según la reivindicación 27 en el que el convertidor incluye una función hash criptográfica unidireccional.

29. El sistema según la reivindicación 23 en el que el contenido comprende un mensaje de correo electrónico.

30. El sistema según la reivindicación 29 en el que el mecanismo de discriminación determina un identificador de seguridad restringida basándose en el remitente del mensaje.

31. El sistema según la reivindicación 23 en el que el contenido comprende una secuencia de comandos.

32. El sistema según la reivindicación 23 en el que el sistema es un servidor, y en el que un identificador de seguridad restringida se genera según cómo se autenticó el cliente para conectarse con el servidor.

33. El sistema según la reivindicación 23 que además comprende un ayudante de proceso para extraer una información asociada con el contenido a partir del testigo restringido y para obtener acceso al recurso en nombre del proceso.

34. El sistema según la reivindicación 23 que además comprende un objeto de trabajo, en el que el proceso se ejecuta en el objeto de trabajo.

35. El sistema según la reivindicación 23, en el que el contenido comprende una pluralidad de contenido dispuesto en sitios web diferentes, teniendo el contenido de cada sitio web un proceso (132) configurado para el mismo.

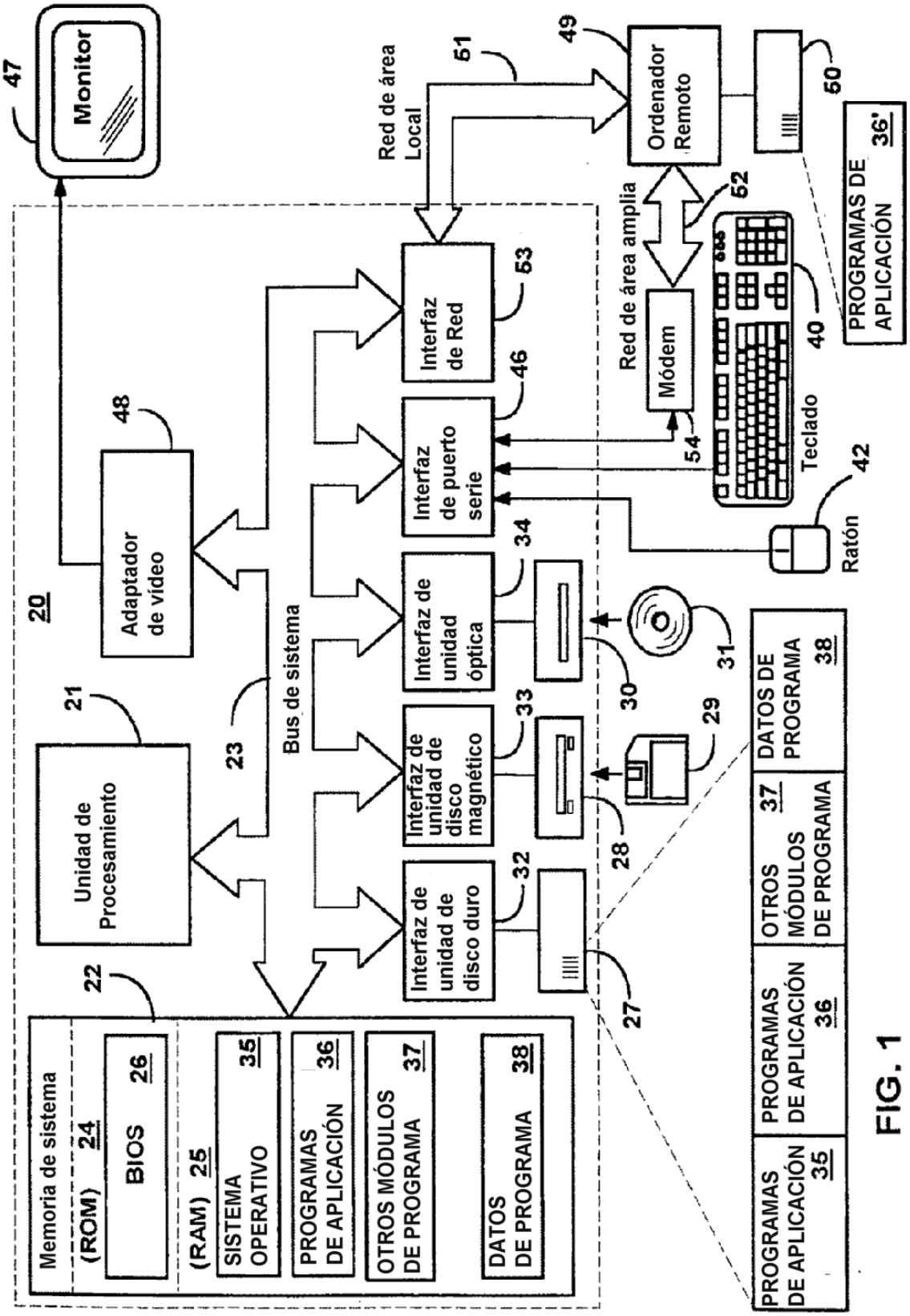

Monitor
47
Red de área Local
51
49
Ordenador Remoto
50
PROGRAMAS DE APLICACIÓN
36'
Red de área amplia
62
Módem
54
Teclado
40
Ratón
42
Interfaz de Red
53
Interfaz de puerto serie
46
Interfaz de unidad óptica
34
Interfaz de unidad de disco magnético
33
Interfaz de unidad de disco duro
32
Adaptador de vídeo
48
20
Unidad de Procesamiento
21
23
Bus de sistema
22
Memoria de sistema
(ROM) 24
BIOS 26
(RAM) 25
SISTEMA OPERATIVO 35
PROGRAMAS DE APLICACIÓN 36
OTROS MÓDULOS DE PROGRAMA 37
DATOS DE PROGRAMA 38
30
31
29
28
27
PROGRAMAS DE APLICACIÓN 35
PROGRAMAS DE APLICACIÓN 36
OTROS MÓDULOS DE PROGRAMA 37
DATOS DE PROGRAMA 38

FIG. 1

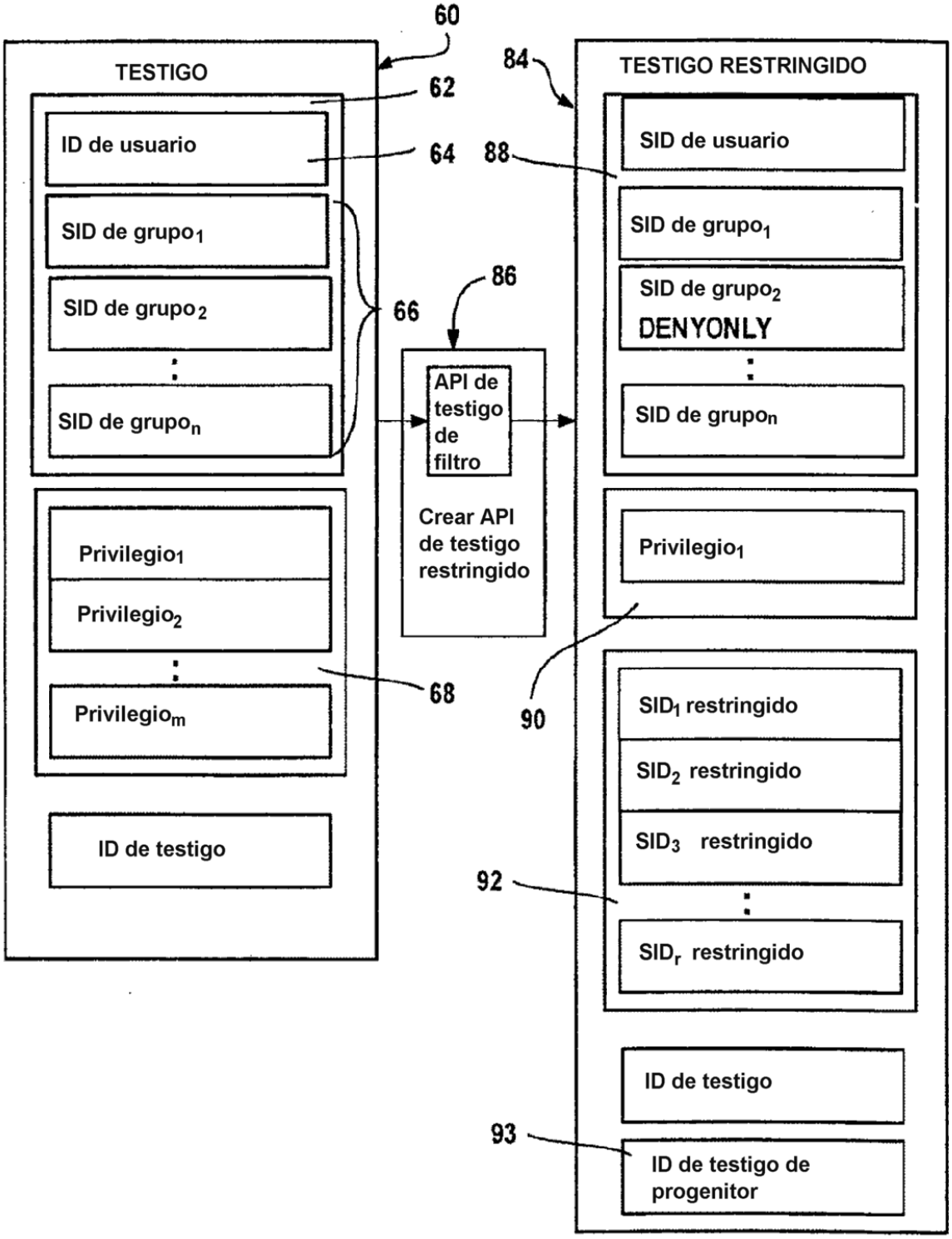

60
TESTIGO
62
ID de usuario
64
SID de grupo1
SID de grupo2
SID de grupon
66
86
API de testigo de filtro
Crear API de testigo restringido
Privilegio1
Privilegio2
Privilegiom
68
ID de testigo
84
TESTIGO RESTRINGIDO
88
SID de usuario
SID de grupo1
SID de grupo2
DENYONLY
SID de grupon
Privilegio1
90
SID1 restringido
SID2 restringido
SID3 restringido
92
SIDr restringido
ID de testigo
93
ID de testigo de progenitor

FIG. 2

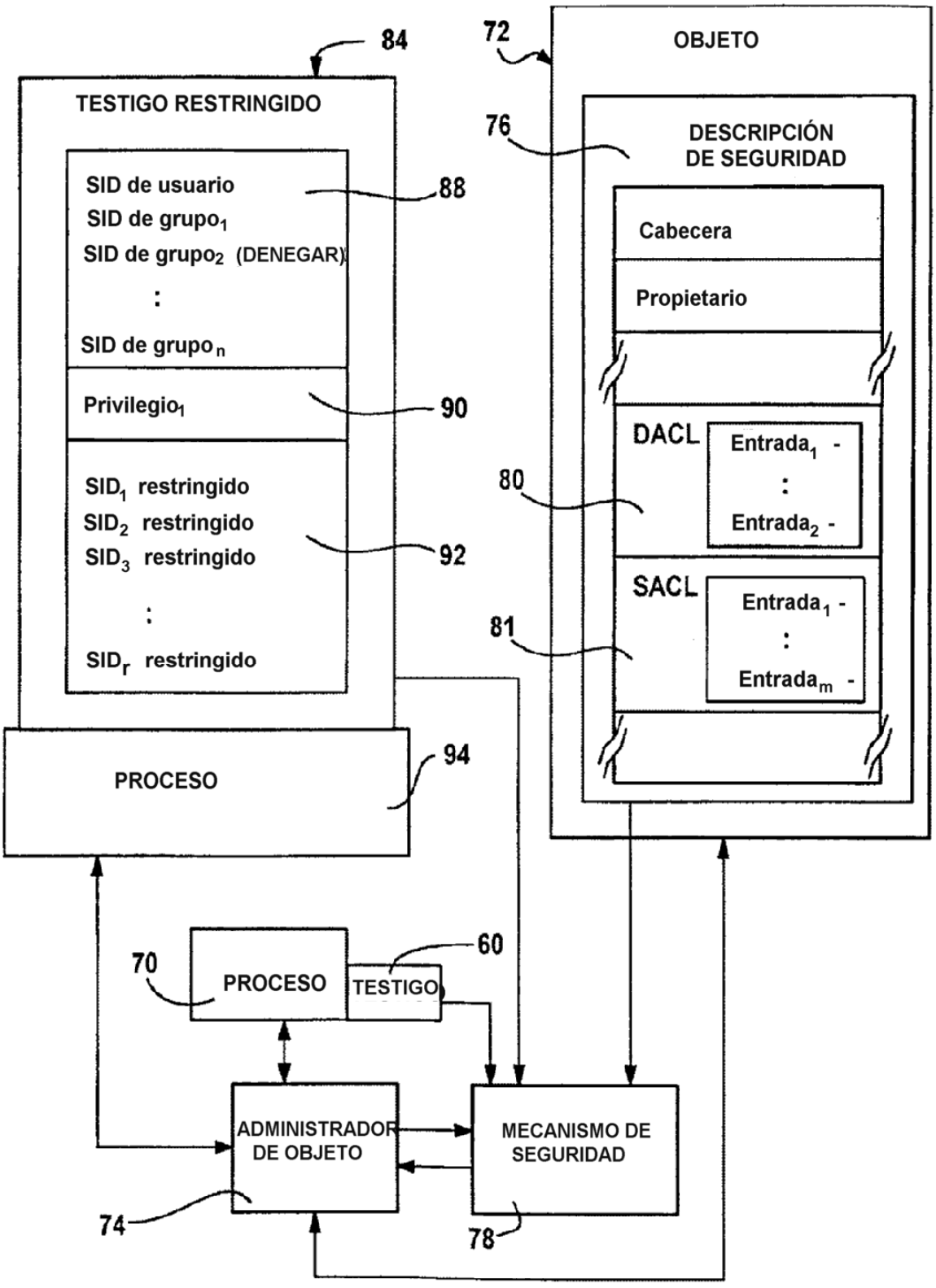

84
TESTIGO RESTRINGIDO
SID de usuario
SID de grupo1
SID de grupo2 (DENEGAR)
:
SID de grupo n
88
Privilegio1
90
SID1 restringido
SID2 restringido
SID3 restringido
:
SIDr restringido
92
PROCESO
94
72
OBJETO
76
DESCRIPCIÓN DE SEGURIDAD
Cabecera
Propietario
DACL
Entrada1 -
:
Entrada2 -
80
SACL
Entrada1 -
:
Entradam -
81
70
PROCESO
TESTIGO
60
ADMINISTRADOR DE OBJETO
MECANISMO DE SEGURIDAD
74
78

FIG. 3

FIG. 4

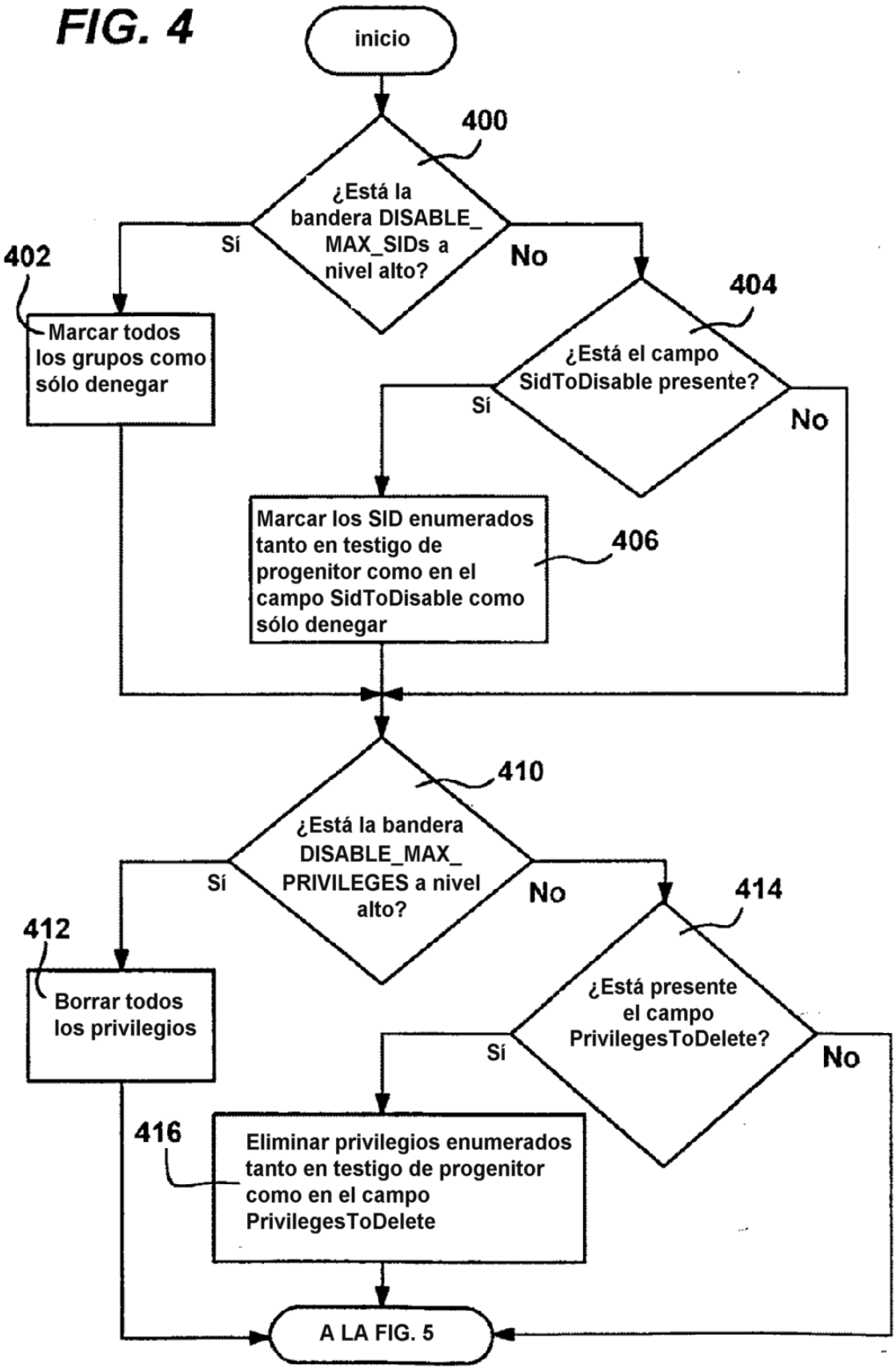
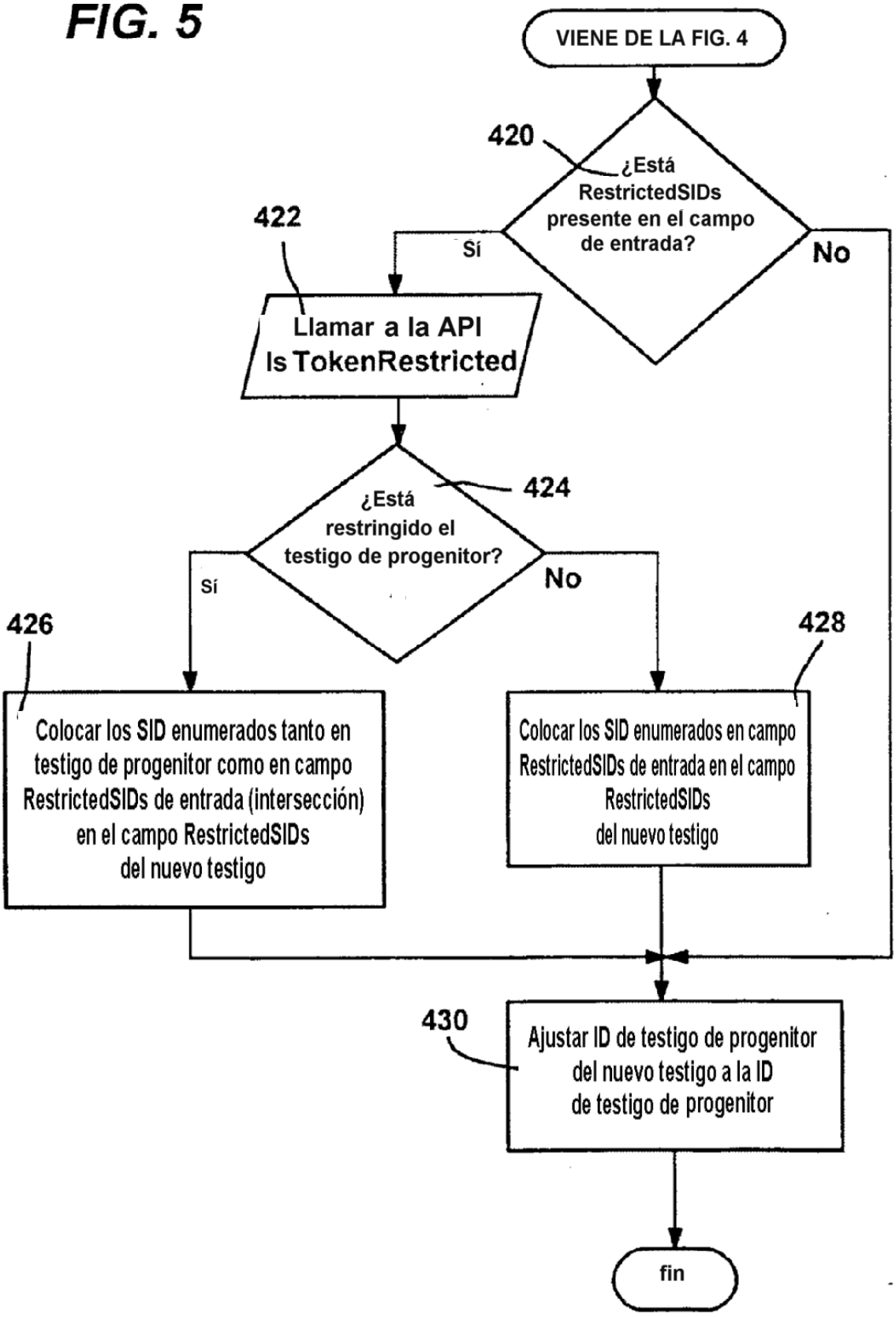

inicio
400
¿Está la bandera DISABLE_MAX_SIDs a nivel alto?
Sí
No
402
Marcar todos los grupos como sólo denegar
404
¿Está el campo SidToDisable presente?
Sí
No
Marcar los SID enumerados tanto en testigo de progenitor como en el campo SidToDisable como sólo denegar
406
410
¿Está la bandera DISABLE_MAX_PRIVILEGES a nivel alto?
Sí
No
412
Borrar todos los privilegios
414
¿Está presente el campo PrivilegesToDelete?
Sí
No
416
Eliminar privilegios enumerados tanto en testigo de progenitor como en el campo PrivilegesToDelete
A LA FIG. 5

FIG. 5


VIENE DE LA FIG. 4
420
¿Está RestrictedSIDs presente en el campo de entrada?
Sí
No
422
Llamar a la API IsTokenRestricted
¿Está restringido el testigo de progenitor?
424
Sí
No
426
Colocar los SID enumerados tanto en testigo de progenitor como en campo RestrictedSIDs de entrada (intersección) en el campo RestrictedSIDs del nuevo testigo
428
Colocar los SID enumerados en campo RestrictedSIDs de entrada en el campo RestrictedSIDs del nuevo testigo
430
Ajustar ID de testigo de progenitor del nuevo testigo a la ID de testigo de progenitor
fin

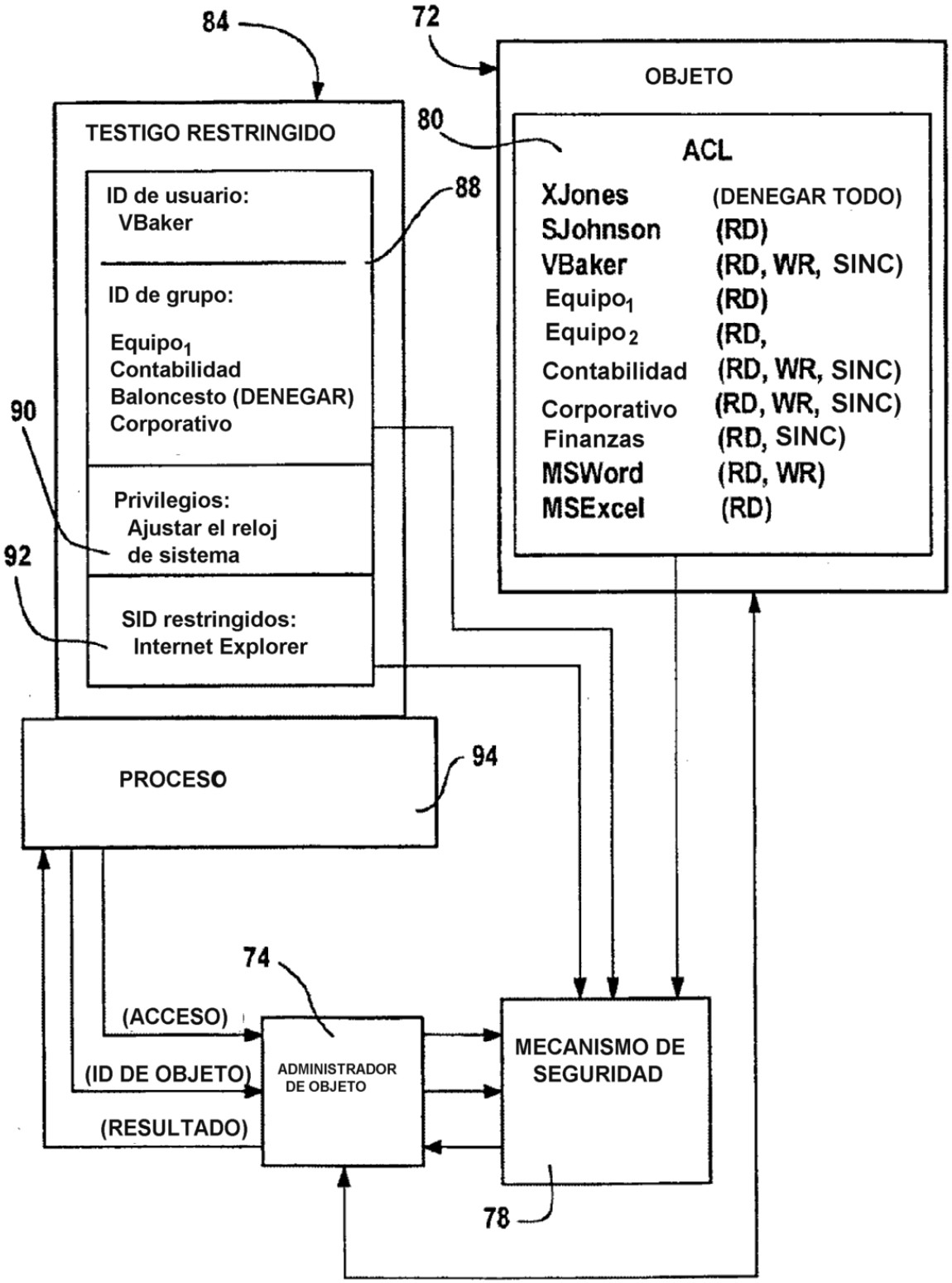

84
72
OBJETO
TESTIGO RESTRINGIDO
80
ACL
ID de usuario:
VBaker
88
XJones (DENEGAR TODO)
SJohnson (RD)
VBaker (RD, WR, SINC)
Equipo1 (RD)
Equipo2 (RD,
Contabilidad (RD, WR, SINC)
Corporativo (RD, WR, SINC)
Finanzas (RD, SINC)
MSWord (RD, WR)
MSExcel (RD)
ID de grupo:
Equipo1
Contabilidad
Baloncesto (DENEGAR)
Corporativo
90
Privilegios:
Ajustar el reloj
de sistema
92
SID restringidos:
Internet Explorer
PROCESO
94
74
(ACCESO)
(ID DE OBJETO)
(RESULTADO)
ADMINISTRADOR
DE OBJETO
MECANISMO DE
SEGURIDAD
78

FIG. 6

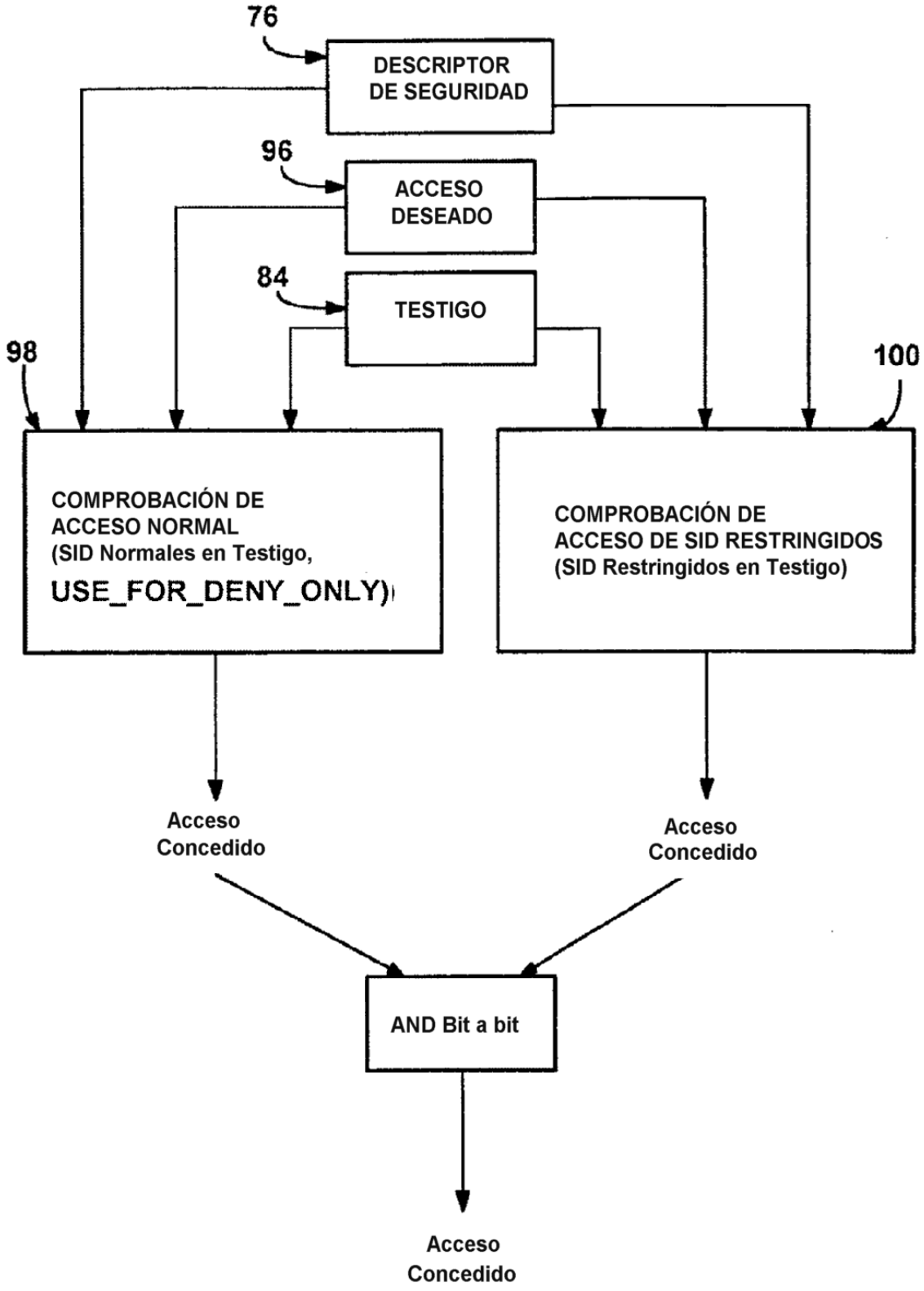
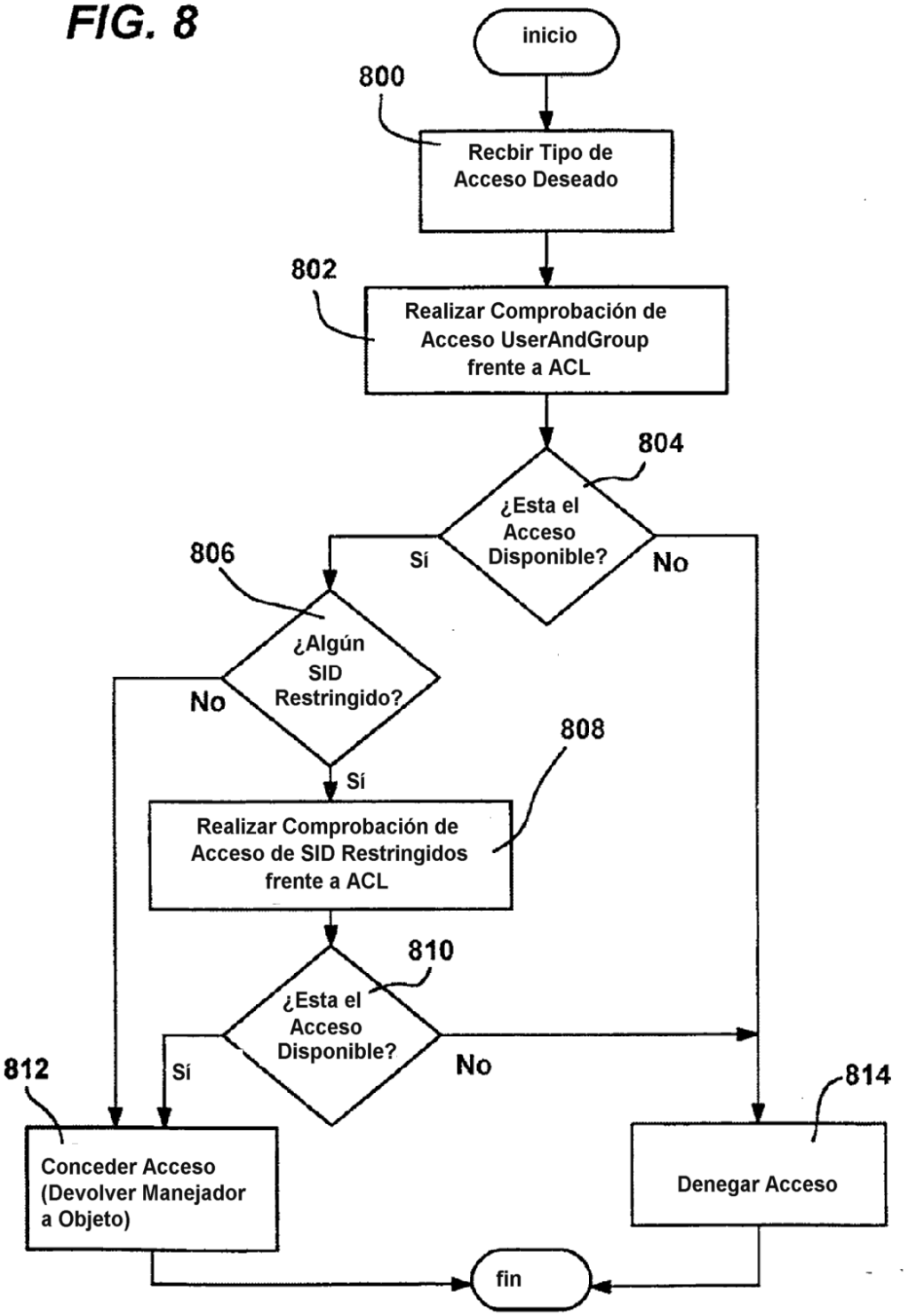

76
DESCRIPTOR DE SEGURIDAD
96
ACCESO DESEADO
84
TESTIGO
98
100
COMPROBACIÓN DE ACCESO NORMAL (SID Normales en Testigo, USE_FOR_DENY_ONLY)
COMPROBACIÓN DE ACCESO DE SID RESTRINGIDOS (SID Restringidos en Testigo)
Acceso Concedido
Acceso Concedido
AND Bit a bit
Acceso Concedido

FIG. 7

FIG. 8


inicio
800
Recbir Tipo de Acceso Deseado
802
Realizar Comprobación de Acceso UserAndGroup frente a ACL
804
¿Esta el Acceso Disponible?
Sí
No
806
¿Algún SID Restringido?
No
Sí
808
Realizar Comprobación de Acceso de SID Restringidos frente a ACL
810
¿Esta el Acceso Disponible?
Sí
No
812
Conceder Acceso (Devolver Manejador a Objeto)
814
Denegar Acceso
fin

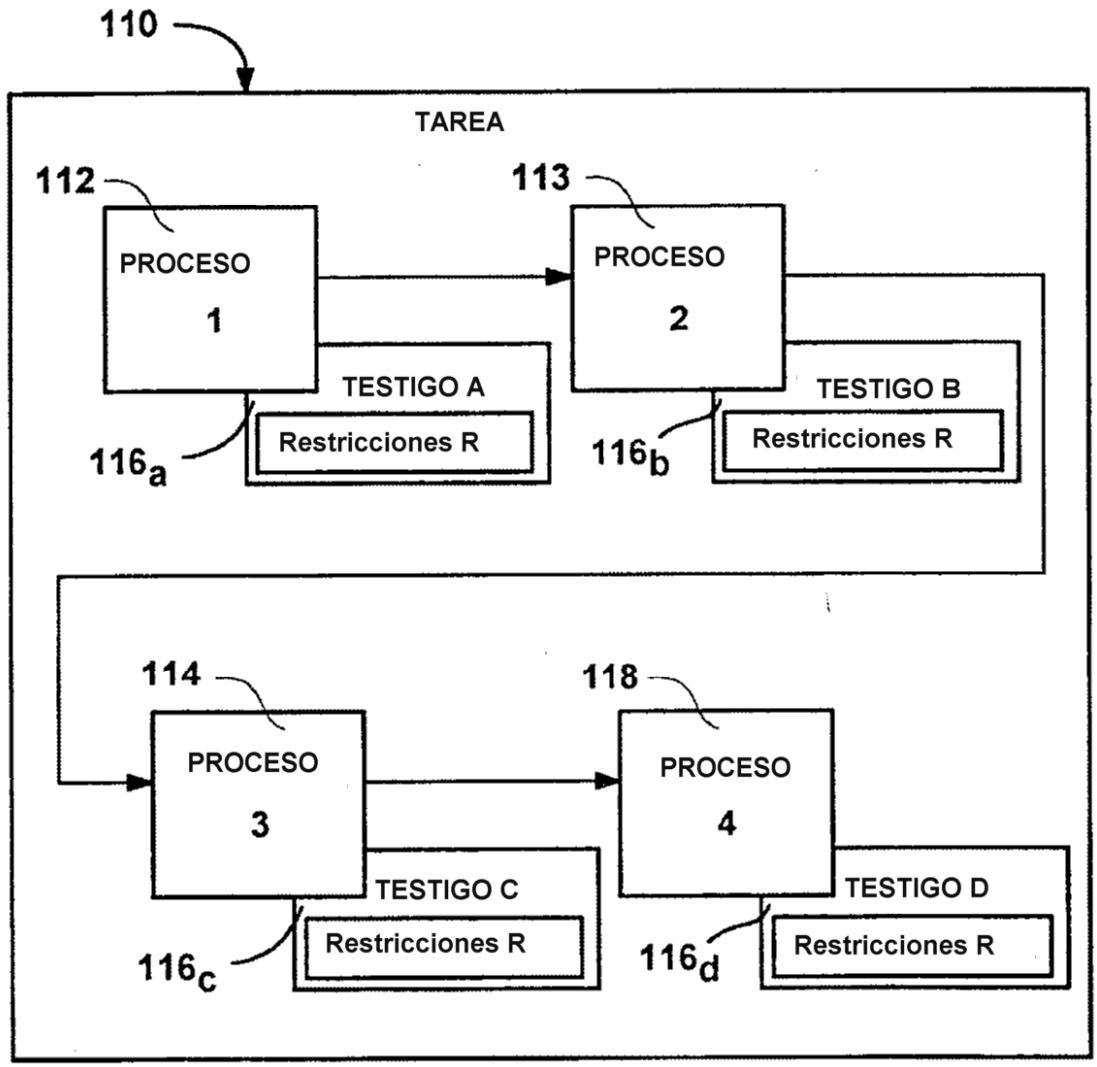

110
TAREA
112
PROCESO
1
TESTIGO A
Restricciones R
116a
113
PROCESO
2
TESTIGO B
Restricciones R
116b
114
PROCESO
3
TESTIGO C
Restricciones R
116c
118
PROCESO
4
TESTIGO D
Restricciones R
116d

FIG. 9

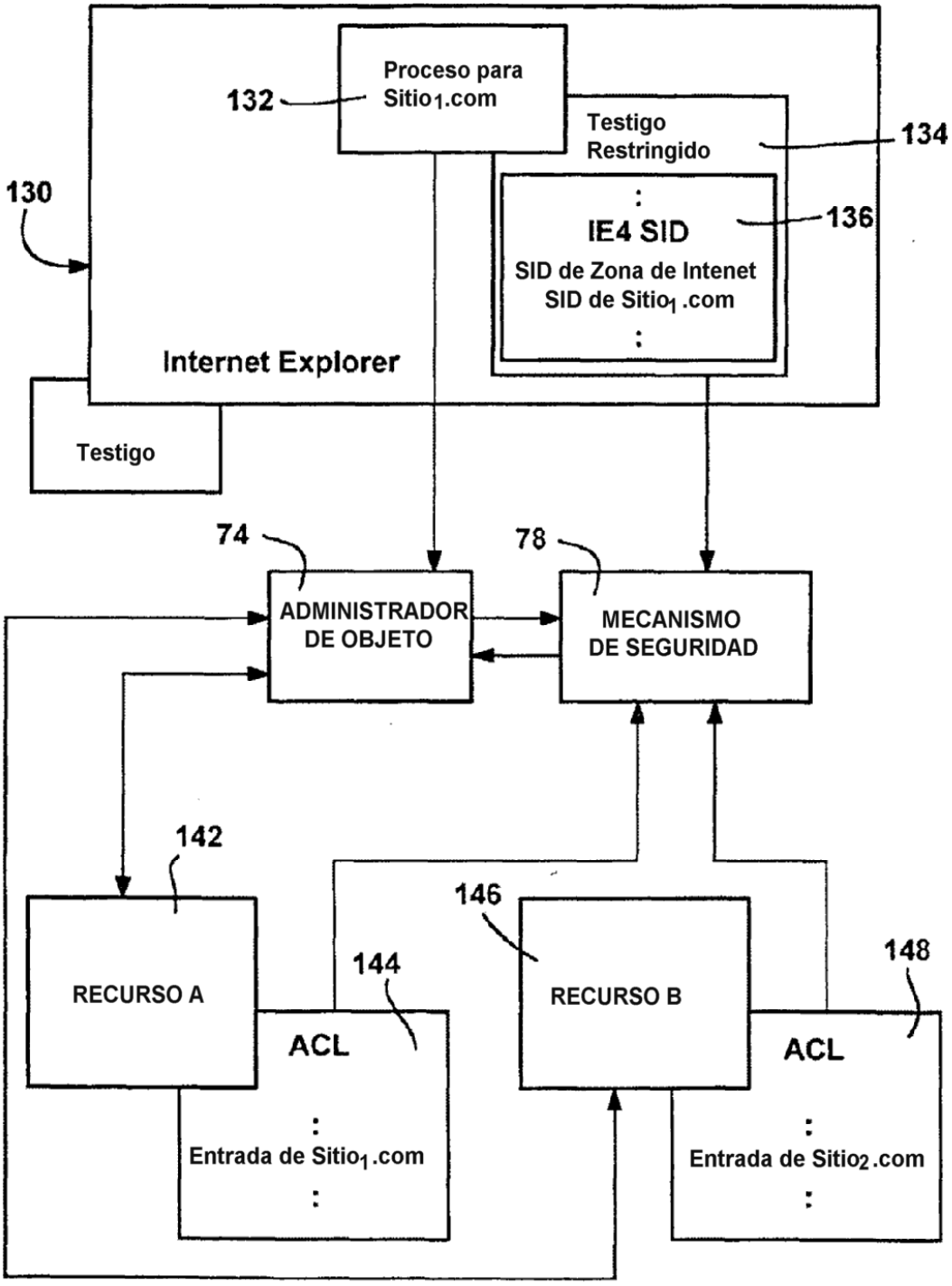

Proceso para Sitio1.com
132
Testigo Restringido
134
130
IE4 SID
136
SID de Zona de Intenet
SID de Sitio1 .com
Internet Explorer
Testigo
74
78
ADMINISTRADOR DE OBJETO
MECANISMO DE SEGURIDAD
142
146
148
144
RECURSO A
RECURSO B
ACL
ACL
Entrada de Sitio1 .com
Entrada de Sitio2 .com

FIG. 10

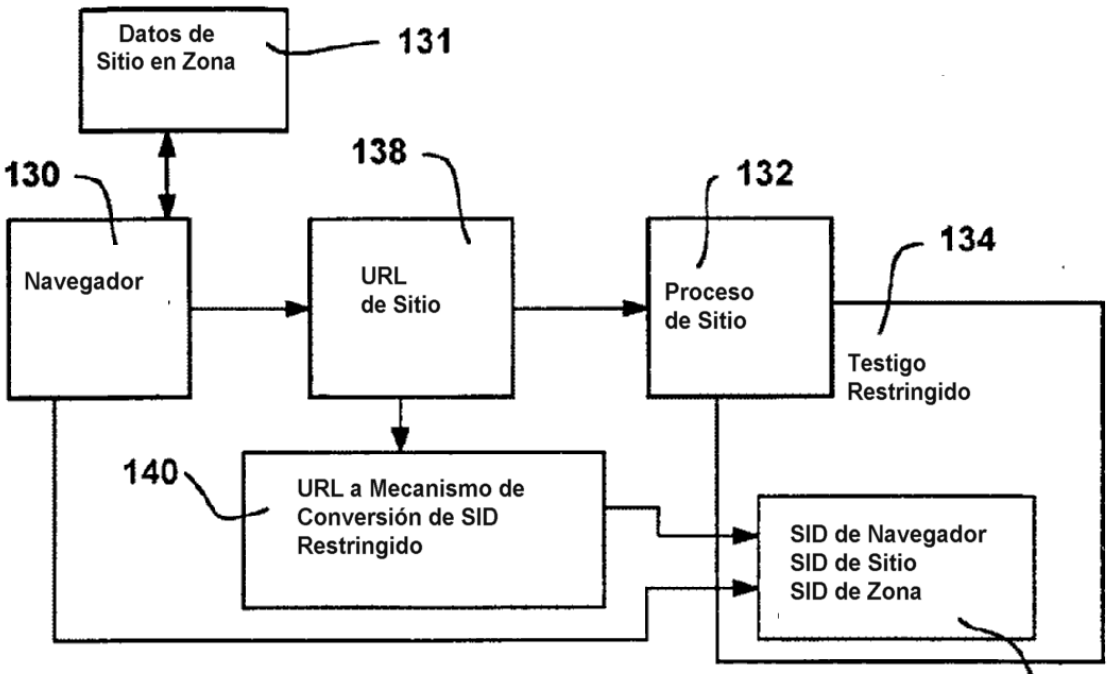

Datos de Sitio en Zona
131
130
Navegador
138
URL de Sitio
132
Proceso de Sitio
134
Testigo Restringido
140
URL a Mecanismo de Conversión de SID Restringido
SID de Navegador
SID de Sitio
SID de Zona
136

FIG. 11

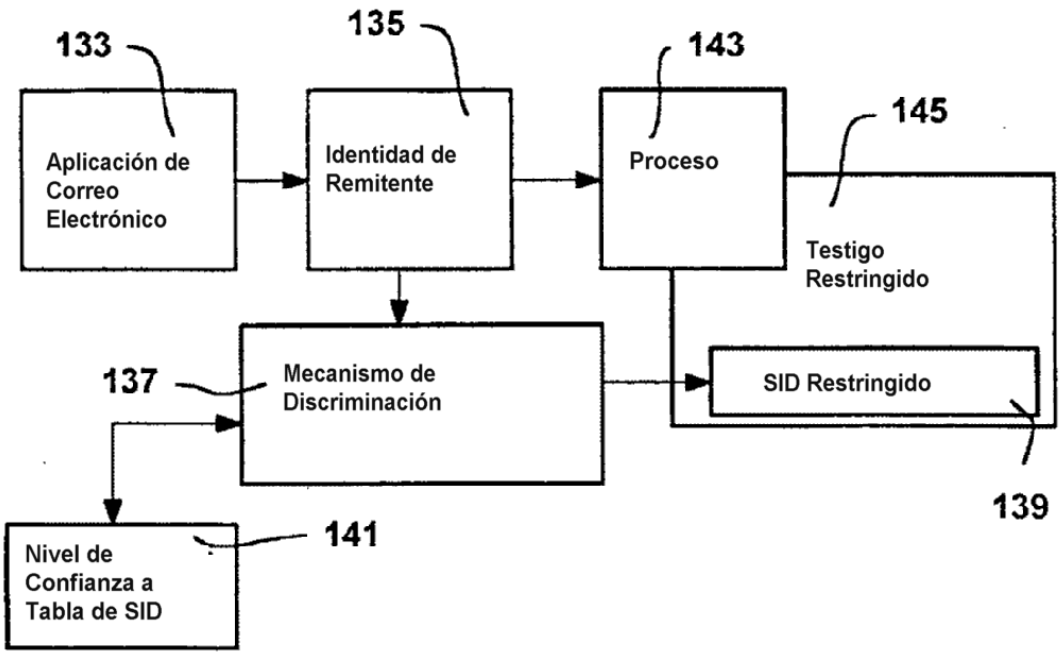

133
Aplicación de Correo Electrónico
135
Identidad de Remitente
143
Proceso
145
Testigo Restringido
137
Mecanismo de Discriminación
SID Restringido
139
141
Nivel de Confianza a Tabla de SID

FIG. 12

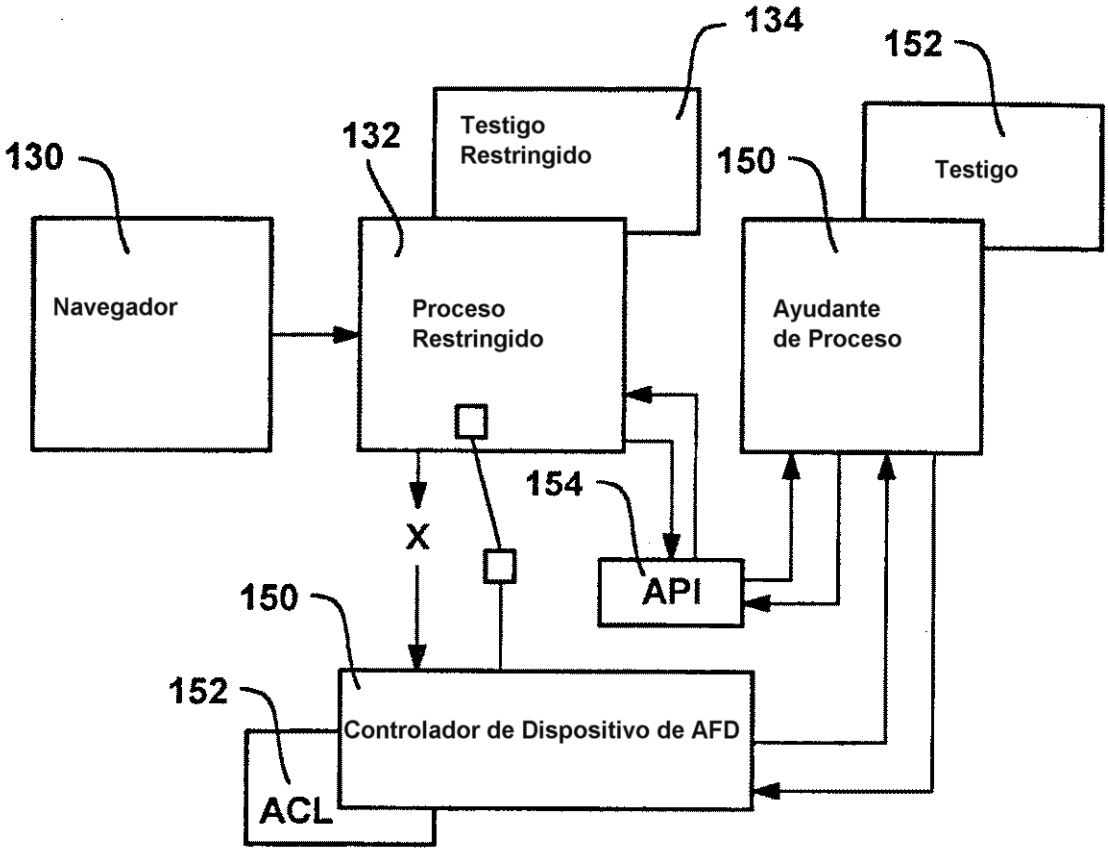

134
152
Testigo
Restringido
132
130
150
Testigo
Navegador
Proceso
Restringido
Ayudante
de Proceso
154
X
API
150
152
Controlador de Dispositivo de AFD
ACL

FIG. 13

FIG. 14

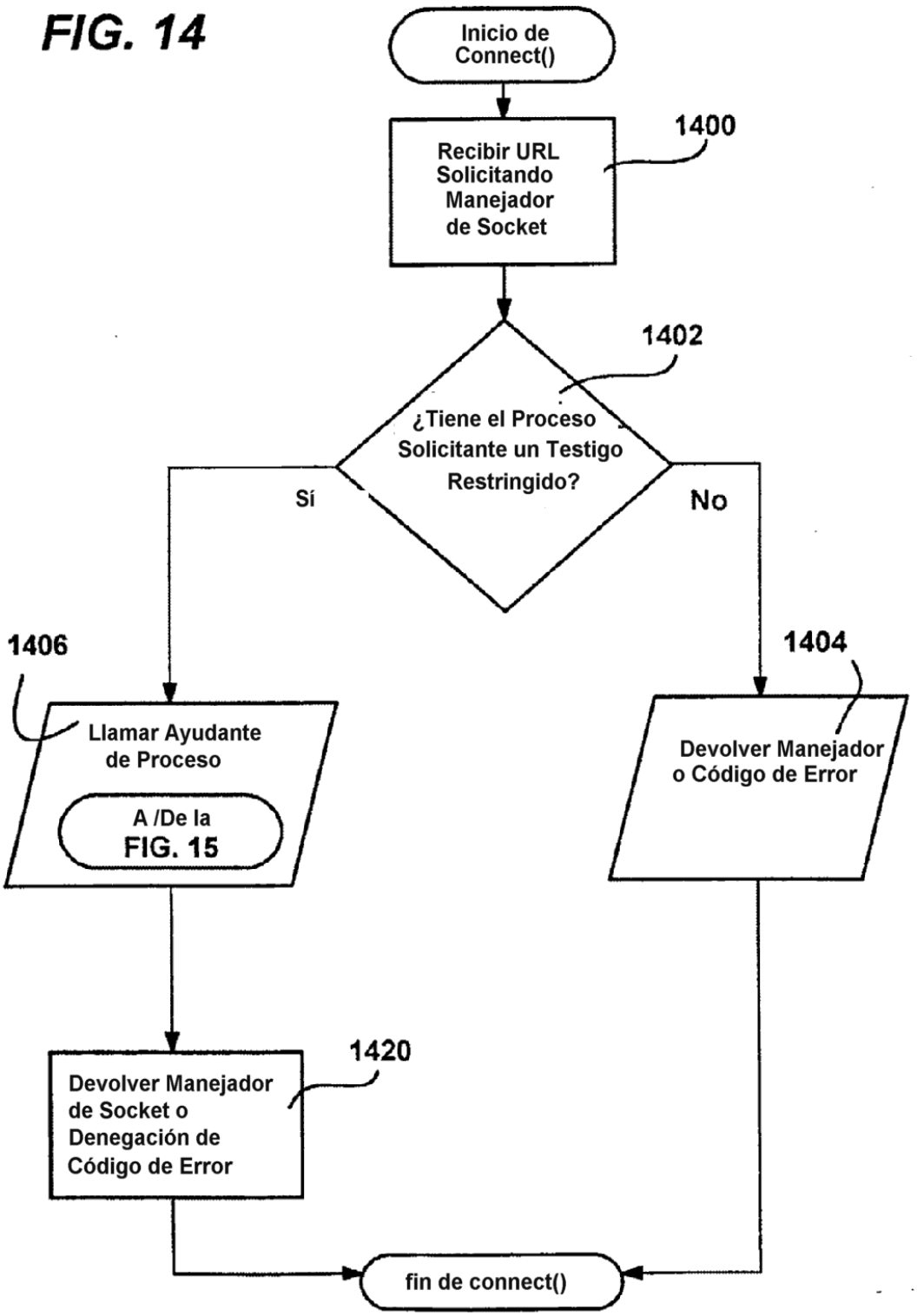
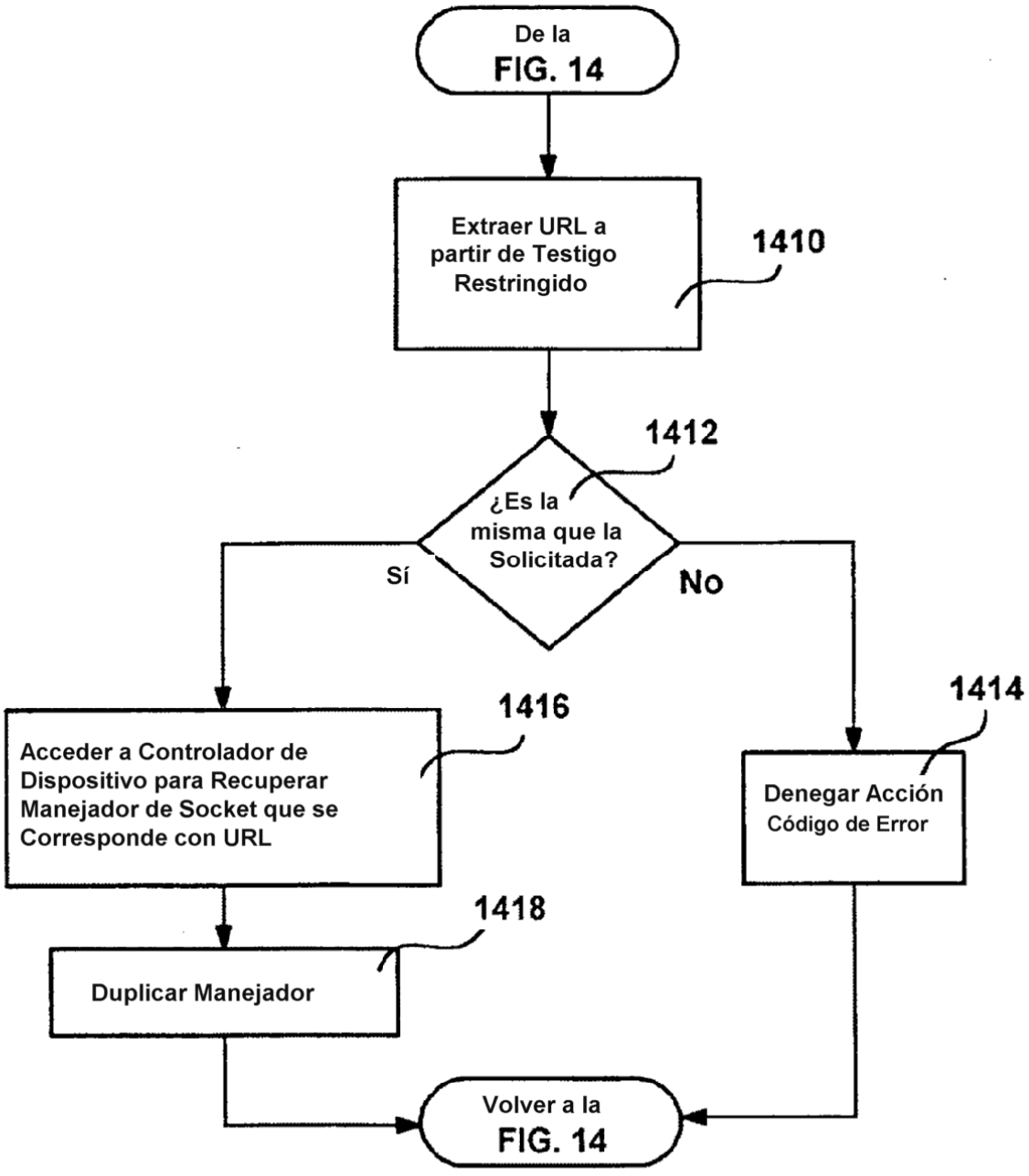

Inicio de Connect()
1400
Recibir URL Solicitando Manejador de Socket
1402
¿Tiene el Proceso Solicitante un Testigo Restringido?
Sí
No
1406
Llamar Ayudante de Proceso
A /De la FIG. 15
1404
Devolver Manejador o Código de Error
1420
Devolver Manejador de Socket o Denegación de Código de Error
fin de connect()

FIG. 15


De la
FIG. 14
Extraer URL a partir de Testigo Restringido
1410
1412
¿Es la misma que la Solicitada?
Sí
No
1414
Acceder a Controlador de Dispositivo para Recuperar Manejador de Socket que se Corresponde con URL
1416
Denegar Acción
Código de Error
1418
Duplicar Manejador
Volver a la
FIG. 14

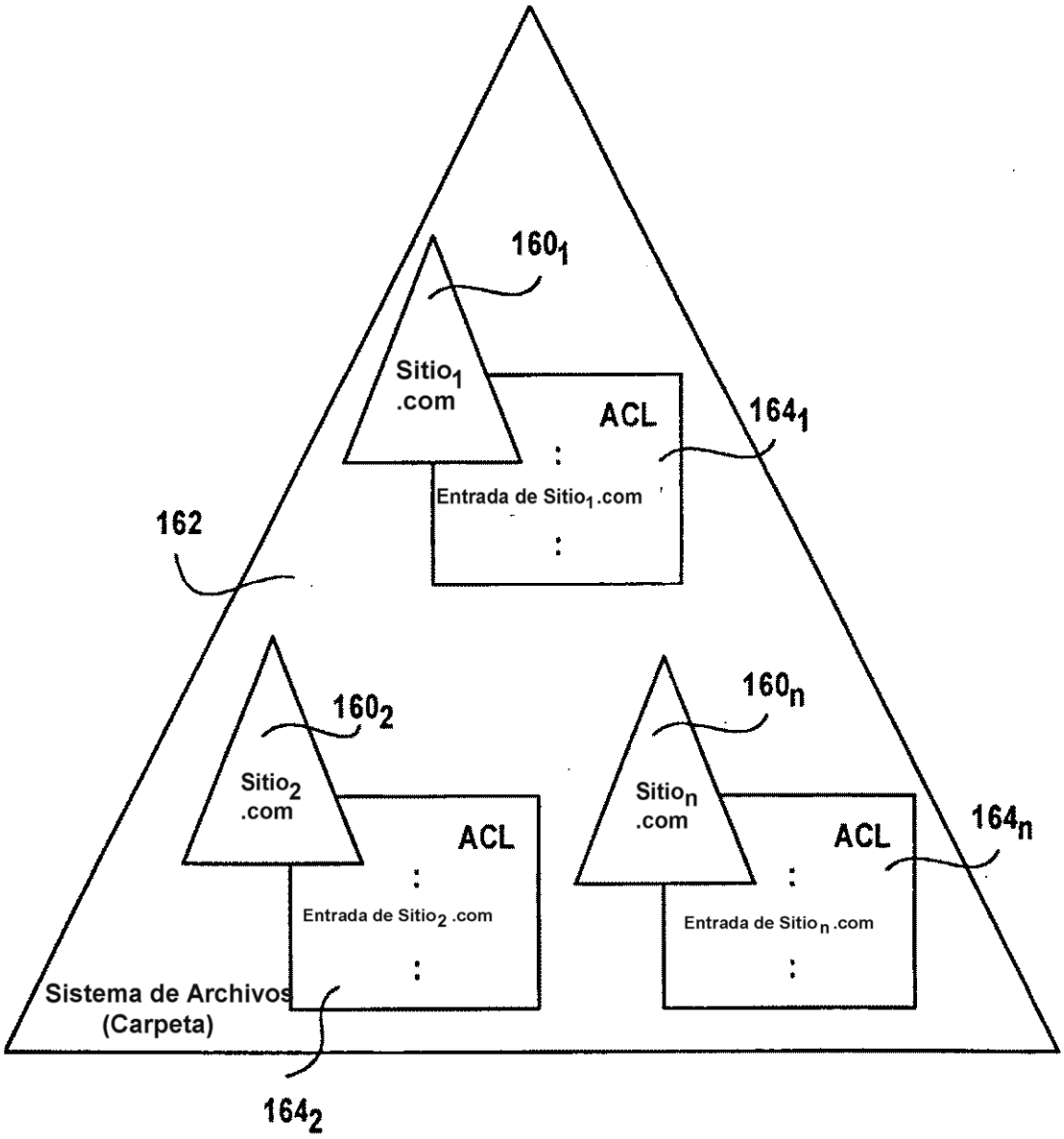

160$_1$
Sitio$_1$ .com
ACL
164$_1$
Entrada de Sitio$_1$.com
162
160$_2$
Sitio$_2$ .com
ACL
Entrada de Sitio$_2$ .com
160$_n$
Sitio$_n$ .com
ACL
164$_n$
Entrada de Sitio$_n$ .com
Sistema de Archivos (Carpeta)
164$_2$

FIG. 16

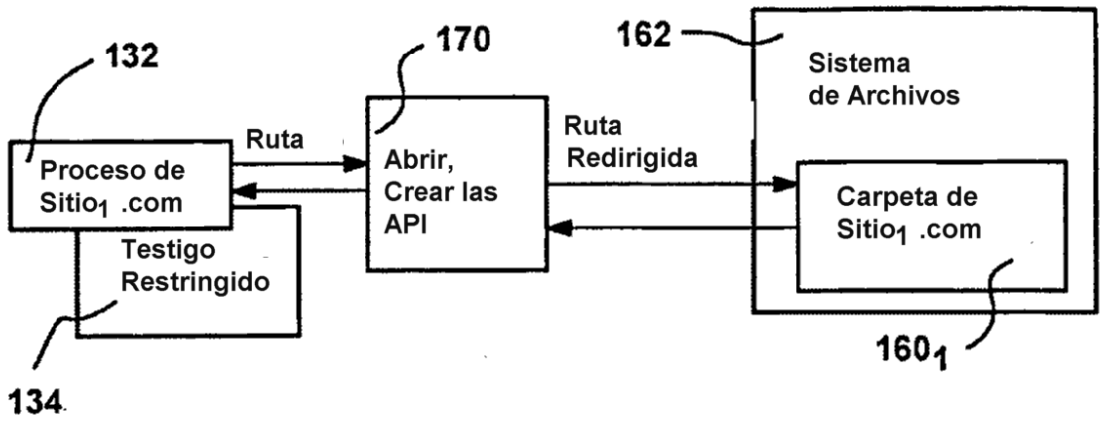

132
170
162
Sistema
de Archivos
Ruta
Proceso de
Sitio1 .com
Abrir,
Crear las
API
Ruta
Redirigida
Carpeta de
Sitio1 .com
Testigo
Restringido
1601
134

FIG. 17

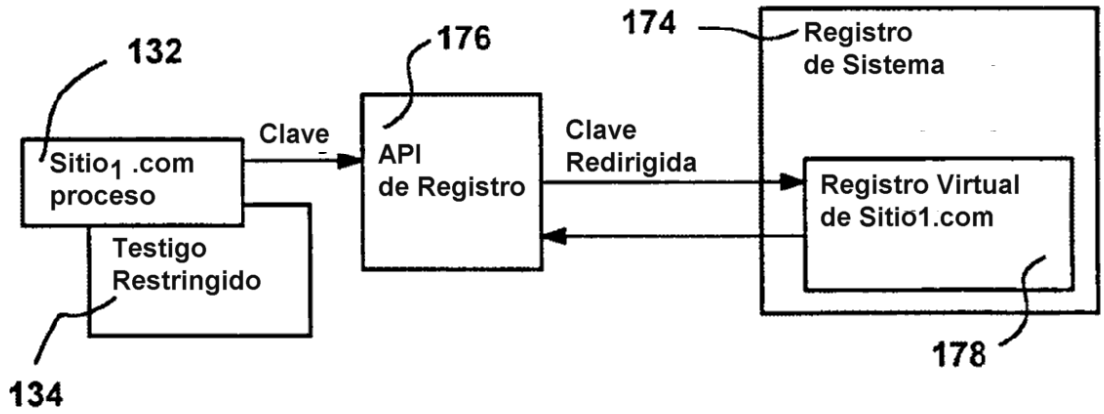

174
Registro
de Sistema
132
176
Clave
Sitio1 .com
proceso
API
de Registro
Clave
Redirigida
Registro Virtual
de Sitio1.com
Testigo
Restringido
178
134

FIG. 18

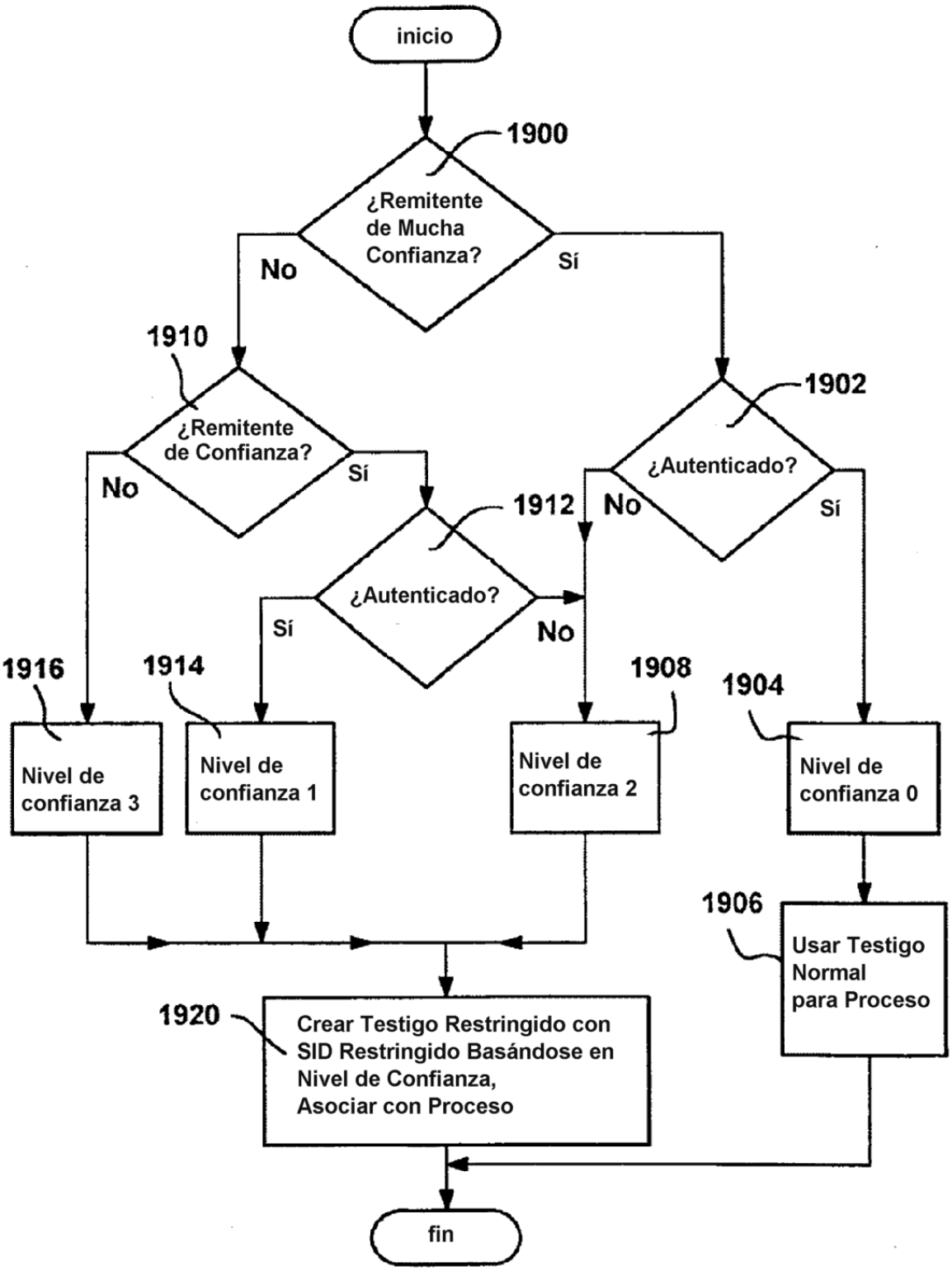

inicio
1900
¿Remitente de Mucha Confianza?
No
Sí
1910
¿Remitente de Confianza?
No
Sí
1902
¿Autenticado?
No
Sí
1912
¿Autenticado?
Sí
No
1916
Nivel de confianza 3
1914
Nivel de confianza 1
1908
Nivel de confianza 2
1904
Nivel de confianza 0
1906
Usar Testigo Normal para Proceso
1920
Crear Testigo Restringido con SID Restringido Basándose en Nivel de Confianza, Asociar con Proceso
fin

FIG. 19

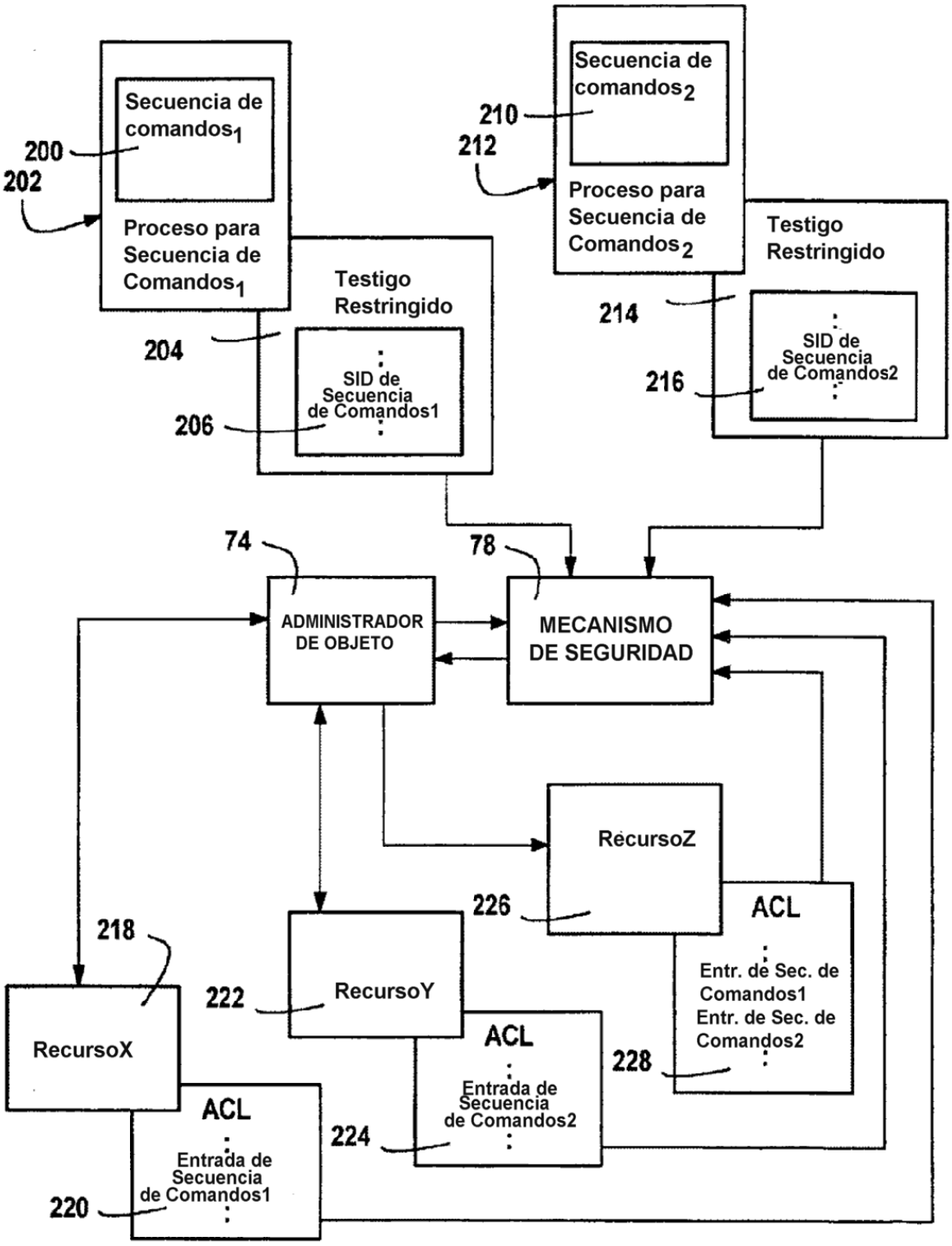

Secuencia de comandos1
200
202
Proceso para Secuencia de Comandos1
Testigo Restringido
204
SID de Secuencia de Comandos1
206
Secuencia de comandos2
210
212
Proceso para Secuencia de Comandos2
Testigo Restringido
214
SID de Secuencia de Comandos2
216
74
78
ADMINISTRADOR DE OBJETO
MECANISMO DE SEGURIDAD
RecursoZ
226
ACL
Entr. de Sec. de Comandos1
Entr. de Sec. de Comandos2
228
218
RecursoX
222
RecursoY
ACL
Entrada de Secuencia de Comandos2
224
ACL
Entrada de Secuencia de Comandos1
220

FIG. 20

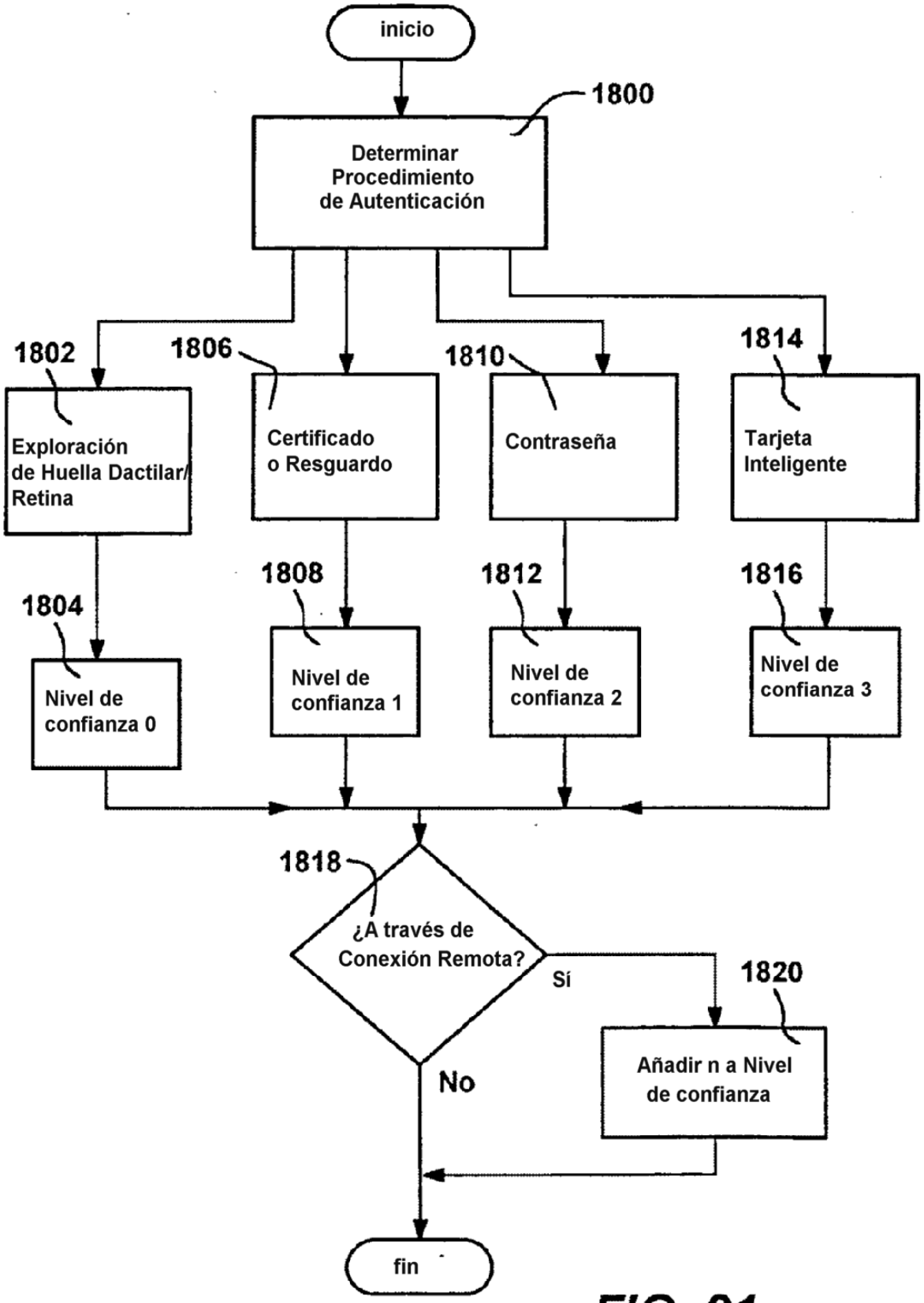

inicio
1800
Determinar Procedimiento de Autenticación
1802
Exploración de Huella Dactilar/ Retina
1806
Certificado o Resguardo
1810
Contraseña
1814
Tarjeta Inteligente
1804
Nivel de confianza 0
1808
Nivel de confianza 1
1812
Nivel de confianza 2
1816
Nivel de confianza 3
1818
¿A través de Conexión Remota?
Sí
No
1820
Añadir n a Nivel de confianza
fin

FIG. 21