(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **3 010 152**

(51) Int. Cl.:

G06F 21/74 (2013.01)
H04L 9/40 (2012.01)
G06F 21/51 (2013.01)
G06F 21/53 (2013.01)
G06F 21/54 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **20.04.2013 E 20212125 (7)**

(97) Fecha y número de publicación de la concesión europea: **13.11.2024 EP 3805966**

(54) Título: **Zona segura para compras seguras**

(30) Prioridad:

20.04.2012 US 201261636201 P
19.04.2013 US 201313866687

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
01.04.2025

(73) Titular/es:

FINGON LLC (100.00%)
57 Pond Brook Road
Newton CT 06470, US

(72) Inventor/es:

IGNATCHENKO, SERGEY y
IVANCHYKHIN, DMYTRO

(74) Agente/Representante:

LEHMANN NOVO, María Isabel

DESCRIPCIÓN

Zona segura para compras seguras

Solicitudes relacionadas

La presente solicitud reivindica la prioridad de la solicitud provisional US Nº 61/636.201, presentada el 20 de Abril de 2012, y la solicitud no provisional US Nº 13/866.687, presentada el 19 de Abril de 2013, ambas tituladas "Secure Zone for Secure Purchases".

Campo de la divulgación

Los sistemas, métodos y aparatos descritos en el presente documento se refieren a la seguridad de las transacciones comerciales y de otros datos sensibles basadas en redes informáticas.

Antecedentes

Las compras por Internet, la banca en línea y otras formas de transmisión de datos confidenciales basadas en la red son muy populares, pero pueden ser susceptibles a una diversidad de violaciones de seguridad resultantes de virus informáticos, puertas traseras, registradores de teclado y otras formas de ataques a un ordenador u otro dispositivo del usuario. Estos ataques están relacionados generalmente con vulnerabilidades en el sistema operativo del dispositivo usado para acceder a la red. Lo que se necesita es una plataforma de hardware adecuada para implementar soluciones de seguridad que no sean susceptibles a ataques basados en software.

El documento EP 2 045 753 A1 se refiere a la asociación de aplicaciones basada en la identificación criptográfica. Cuando cada aplicación se carga para su uso por parte de un sistema operativo, una aplicación de gestión de mensajes en el interior del sistema operativo asocia las aplicaciones con métricas de relatividad para su uso posterior en la gestión de mensajes entre procesos. Se verifica un identificador criptográfico asociado a cada aplicación y, en base a la verificación, cada aplicación se asocia a una métrica de relatividad. La aplicación de gestión de mensajes recibe un mensaje desde un proceso de origen. A continuación, la aplicación de gestión de mensajes determina un proceso de destino para el mensaje, una métrica de relatividad para el proceso de origen y una métrica de relatividad para el proceso de destino. En base a un análisis de las métricas de relatividad del proceso de origen y del proceso de destino, la aplicación de gestión de mensajes determina si pasar o no el mensaje al proceso de destino.

El documento EP 1 612 670 A2 se refiere a la gestión segura de tareas en un ordenador. Un sistema de gestión de tareas en un sistema operativo ejecuta múltiples tareas en paralelo. El sistema de gestión de tareas incluye un modificador de estado de espera de ejecución configurado para generar un verificador de una tarea a partir de datos almacenados en un espacio de direcciones de tareas y para almacenar el verificador generado en un área de almacenamiento de verificadores, cuando la tarea se modifica desde un estado ejecutado a un estado de espera de ejecución; y un modificador de estado ejecutado configurado para generar un verificador de la tarea a partir de los datos almacenados en el espacio de direcciones de tareas y para verificar la coincidencia del verificador generado con el verificador de la tarea almacenada en el área de almacenamiento de verificadores, cuando la tarea se modifica desde el estado de espera de ejecución al estado ejecutado.

El documento WO 2009/111409 A1 divulga un sistema y un método para autorizar la ejecución de código de software en base a derechos accesibles. Las realizaciones incluyen sistemas y métodos para autorizar la ejecución de código de software o capacidades de acceso en entornos operativos seguros. Los perfiles pueden ser emitidos por entidades de confianza para extender la confianza a otras entidades para permitir que esas otras entidades proporcionen o controlen la ejecución de aplicaciones en un entorno operativo seguro, tal como en dispositivos informáticos particulares. Puede recibirse una solicitud en un primer programa desde un segundo programa. A continuación, se identifica un perfil. El perfil incluye al menos un derecho asociado con el segundo programa. El perfil se autentica en base a un primer resumen indicativo del perfil y el segundo programa se autentica en base a un segundo resumen indicativo del segundo programa. A continuación, la solicitud se ejecuta en base al derecho.

El documento WO 2011/051757 A1 describe un transactor para su uso en conexión con transacciones que implican información segura y no segura que está configurado para presentar avisos en pantallas de interfaz de usuario asociadas con un modo operativo no seguro para advertir a los usuarios que no introduzcan información segura. Cuando el transactor opera en un modo seguro, no se muestran dichos avisos. Además, cuando está en el modo seguro, el transactor puede estar configurado para aceptar entradas de datos solo desde áreas designadas de una interfaz de usuario, tal como una pantalla táctil.

Sumario

La presente invención se refiere a un aparato según la reivindicación 1 independiente, y a un método según la

reivindicación 11 independiente. Las realizaciones preferidas se reivindican en las reivindicaciones dependientes respectivas.

Descripción de los dibujos

La Figura 1 es un diagrama de bloques de un sistema ejemplar según la presente divulgación.

La Figura 2 es un diagrama de flujo que ilustra un método ejemplar mediante el cual un sistema según la presente divulgación puede aceptar una tarea para su ejecución; organizar el proceso de ejecución de la tarea; y realizar un borrado después de la ejecución de la tarea.

La Figura 3A representa una estructura de datos ejemplar que incorpora dos fragmentos de código separados en una relación integrada.

La Figura 3B es una implementación ejemplar de una partición lógica de la memoria de datos en el interior de la zona segura.

La Figura 4 es un diagrama de flujo de un método ejemplar mediante el cual la zona segura puede conmutar tareas.

La Figura 5 es un diagrama de flujo de un método ejemplar mediante el cual puede realizarse una transacción segura con tarjeta de crédito en el contexto de la presente divulgación.

La Figura 6 es un diagrama de flujo de un método ejemplar mediante el cual un comerciante puede procesar una solicitud de transacción en el contexto de la presente divulgación.

Descripción detallada

Ciertos aspectos ilustrativos de los sistemas, aparatos y métodos según la presente invención se describen en el presente documento con relación a la siguiente descripción y a las figuras adjuntas. Sin embargo, estos aspectos son indicativos de solo algunas de las diversas maneras en las que pueden emplearse los principios de la invención y se pretende que la presente invención incluya la totalidad de dichos aspectos y sus equivalentes. Otras ventajas y características novedosas de la invención pueden resultar evidentes a partir de la siguiente descripción detallada cuando se considera junto con las figuras.

En la siguiente descripción detallada, se exponen numerosos detalles específicos para proporcionar una comprensión completa de la invención. En otros casos, las estructuras, las interfaces y los procesos bien conocidos no se han mostrado en detalle para no oscurecer innecesariamente la invención. Sin embargo, será evidente para una persona experta en la técnica que no es necesario usar esos detalles específicos divulgados en el presente documento para llevar a la práctica la invención y no representan una limitación del alcance de la invención, excepto tal como se indica en las reivindicaciones. Se pretende que ninguna parte de la presente memoria descriptiva se interprete como una negación de cualquier parte del alcance completo de la invención. Aunque se describen determinadas realizaciones de la presente divulgación, estas realizaciones tampoco pretenden limitar el alcance completo de la invención.

La presente divulgación proporciona sistemas, métodos y aparatos para realizar de manera segura acciones o transacciones basadas en ordenador. Por ejemplo, podría ser deseable usar un ordenador para establecer una conexión segura con un servidor remoto, tal como una conexión SSL, con el fin de ver transacciones de cuentas bancarias o para comprar ciertos productos o servicios. En otro ejemplo, podría ser deseable que un televisor equipado de manera apropiada reciba contenido multimedia encriptado desde una tienda de Internet. En cada caso, un individuo experto podría interceptar los datos en el interior de un sistema operativo que ejecuta el ordenador (por ejemplo, un número de tarjeta de crédito enviado a través de la conexión SSL o una película transferida desde la tienda de Internet) por ejemplo, instalando malware (tal como un virus, un registrador de teclado o un troyano) en el sistema operativo del ordenador del usuario. Las invenciones descritas en el presente documento proporcionan una manera de transferir ciertas actividades a una zona segura, que no puede verse comprometida incluso si el sistema operativo está bajo el control completo del atacante, para garantizar que estas actividades basadas en ordenador realmente permanezcan seguras contra ataques. Además, para una mayor seguridad, la zona segura puede hacerse resistente a la manipulación y/o puede usar técnicas de detección de manipulación, por ejemplo, con el borrado de una o más claves criptográficas tras la detección de una manipulación.

La Figura 1 muestra un ejemplo mediante el cual puede implementarse una zona 150 segura según la presente divulgación en un dispositivo 120 más grande, tal como un ordenador, ordenador portátil, teléfono inteligente, televisor, reproductor de música personal, decodificador, etc.

Una zona 150 segura según la presente divulgación puede comprender primero una interfaz 151 a una o más zonas 152 no seguras. La expresión "zona no segura", tal como se usa en el presente documento, se refiere a

cualquier dispositivo, procesador, sistema operativo u otro objeto, o combinación de los mismos, que sea capaz de proporcionar mensajes, códigos, tareas u otra información a una zona 150 segura. La interfaz 151 puede estar configurada para recibir estos mensajes, códigos o tareas desde esas zonas 152 no seguras. Por ejemplo, si se implementa una zona 150 segura en un ordenador portátil, la interfaz 151 puede implementarse como algún tipo de bus (por ejemplo, un bus PCIe) y puede estar configurada para recibir mensajes, código, tareas u otra información desde la unidad central de procesamiento del ordenador portátil. Si la zona 150 segura se implementara en un televisor, la interfaz 151 podría implementarse una vez más, por ejemplo, como algún tipo de bus (por ejemplo, un bus I^2C), y podría configurarse para recibir información desde un decodificador separado o desde la unidad de microcontrolador del televisor.

Una zona 150 segura puede comprender además un supervisor 160 acoplado a la interfaz 151. El supervisor 160 puede usarse para controlar el acceso a los componentes de la zona 150 segura, y puede usarse para hacer cumplir ciertas reglas operativas de la zona 150 segura, proporcionando ciertas garantías de seguridad al usuario final. Por ejemplo, en una realización, el supervisor 160 puede ser capaz de: (1) recibir código ejecutable que puede ejecutarse en uno o más procesadores 162 seguros en el interior de la zona 150 segura a través de la interfaz 151; (2) verificar que se cumplan ciertos requisitos (tal como se describe más detalladamente a continuación) para este código; (3) si se cumplen los requisitos, cargar este código en una o más memorias 164 de instrucciones situadas en el interior de la zona 150 segura; (4) borrar y/o llenar previamente una o más memorias 165 de datos situadas en el interior de la zona 150 segura; (5) indicar al procesador 162 seguro que ejecute el código cargado en la memoria 164 de instrucciones; (6) controlar uno o más indicadores 193, que pueden usarse para señalar a un usuario ciertos modos de seguridad del dispositivo 120 informático; (7) controlar uno o más periféricos en el interior del dispositivo 120 informático; (8) proporcionar una retroalimentación visual al usuario final relacionada con el origen del código cargado; y (9) realizar un borrado (en la medida requerida) después de ejecutarse el código. Cada una de estas funciones se describe más detalladamente en el presente documento. En una realización, el supervisor 160 puede comprender además un almacenamiento 170 temporal (por ejemplo, implementado como RAM o memoria flash). En una realización, el supervisor 160 puede implementarse en hardware en el interior de la zona 151 segura, de manera que el supervisor 160 no pueda verse afectado o modificado.

Tal como se ha indicado anteriormente, la zona 150 segura puede comprender también un procesador 162 seguro, que puede estar configurado para ejecutar código cargado en la memoria 164 de instrucciones e intercambiar datos con la interfaz 151. El procesador 162 seguro puede ser un procesador de propósito general o cualquier forma adecuada de procesador de propósito especial. En algunas realizaciones, el procesador 162 seguro puede implementarse como un hardware separado del supervisor 160; en algunas otras realizaciones, el supervisor 160 y el procesador 162 seguro podrían implementarse usando el mismo hardware, siempre y cuando se tengan en cuenta los requisitos funcionales especificados a continuación. Además, se entenderá que, aunque la Figura 1 muestra que el procesador 162 seguro tiene una denominada "arquitectura Harvard" (con memoria 164 de instrucciones y memoria 165 de datos separadas), pueden usarse otras arquitecturas (tales como la omnipresente arquitectura de von Neumann) siempre y cuando el supervisor 160 aplique instrucciones equivalentes y restricciones de datos (por ejemplo, el bit XN puede usarse en procesadores ARM® para proporcionar cierta separación de la memoria de datos respecto de la memoria de instrucciones, siempre y cuando el supervisor 160 aplique el bit XN en las áreas de memoria apropiadas y no pueda ser alterado por un código cargable que se ejecuta en el procesador 162 seguro).

En ciertas realizaciones, la zona 150 segura puede comprender además uno o más motores 121 criptográficos. Estos motores 121 criptográficos pueden estar configurados para implementar uno o más algoritmos criptográficos, tales como AES o RSA. El motor 121 criptográfico puede recibir datos desde el supervisor 160 para la encriptación o la desencriptación, y puede proporcionar el texto encriptado resultante (o texto no encriptado, según corresponda) al supervisor 160. En algunas realizaciones, el motor 121 criptográfico también puede ser usado por el procesador 162 seguro; en este caso, puede ser deseable tener una separación clara entre cualquier tarea relacionada con la criptografía que provenga del supervisor 160 al motor 121 criptográfico y cualquier tarea relacionada con la criptografía que provenga del procesador 162 seguro al motor 121 criptográfico, para evitar cualquier fuga de información asociada con un componente al otro. La zona 150 segura puede comprender también un generador 124 de números aleatorios para proporcionar soporte a los procesos criptográficos.

En otras realizaciones, el supervisor 160 puede estar configurado para realizar parte o la totalidad de la funcionalidad del motor 121 criptográfico, y es posible que no se necesite un motor 121 criptográfico separado.

Si se espera que la zona 150 segura realice un procesamiento de imagen y/o vídeo, esta puede comprender además un decodificador 122. Por ejemplo, si la zona 150 segura recibe contenido multimedia encriptado desde la zona 152 no segura (tal como desde una aplicación 112 de reproductor de vídeo que se ejecuta en el interior del sistema 111 operativo), el código que se ejecuta en el procesador 162 seguro (con o sin la ayuda del motor 121 criptográfico, dependiendo de la realización) podría ser responsable de desencriptar el contenido, y, a continuación, el decodificador 122 puede ser responsable de decodificar el contenido. Este decodificador 122 puede comprender,

por ejemplo, implementaciones de algoritmos tales como H.264, VC-1, PNG, JPEG, etc. En algunos casos, el decodificador 122 puede incluir también ciertas capacidades de procesamiento de texto.

En algunas realizaciones, el decodificador 122 puede implementarse en hardware (por ejemplo, como un procesador DSP especializado). Tal como se muestra en la Figura 1, el decodificador 122 puede estar acoplado al procesador 162 seguro, de manera que los datos desencriptados puedan pasar desde el motor 121 criptográfico al decodificador 122.

En algunas otras realizaciones, el procesador 162 seguro puede estar configurado para realizar parte o la totalidad de la funcionalidad del decodificador 122, y es posible que no se necesite un decodificador separado. En todavía otras realizaciones, la zona 150 segura puede no proporcionar soporte nativo para la decodificación de imágenes y/o vídeo, pero puede ser capaz de recibir y ejecutar código (en el procesador 162 seguro) diseñado para implementar este tipo de procesamiento de contenido multimedia.

Tal como se ha indicado anteriormente, la zona 150 segura puede comprender además una o más memorias 164 de instrucciones y memorias 165 de datos, que pueden implementarse como algún tipo de memoria volátil, tal como, por ejemplo, RAM. La ausencia de almacenamiento grabable persistente para el código ejecutable puede garantizar que no puedan instalarse virus, puertas traseras u otro código malicioso en el interior de la zona 150 segura. Además, la zona 150 segura puede contener uno o más almacenamientos 166 de certificados dedicados, que pueden implementarse como memoria no volátil de solo lectura, y uno o más almacenamientos 167 de claves dedicados, que pueden implementarse como memoria no volátil. El almacenamiento 167 de claves puede usarse, por ejemplo, para el almacenamiento de una o más claves privadas (que pueden ser generadas, por ejemplo, por el supervisor 160 usando RNG 124), una o más claves públicas correspondientes o certificados digitales asociados, y/o un identificador de dispositivo único. Esta información puede usarse para identificar y/o autenticar el dispositivo 120 basado en ordenador en cuyo interior está situada la zona 150 segura.

Tal como se ha indicado anteriormente, una zona 150 segura está destinada a ser usada en el contexto de un dispositivo 120 basado en ordenador más grande, tal como un televisor o un ordenador portátil. De esta manera, se entenderá que el dispositivo 120 basado en ordenador puede comprender una serie de componentes que están fuera de la zona 150 segura, pero que, sin embargo, pueden ayudar en el funcionamiento de la zona 150 segura. Por ejemplo, el dispositivo 120 puede comprender dispositivos de entrada/salida tradicionales, tales como un teclado 192 o una pantalla 123; en otras realizaciones, el dispositivo 120 puede comprender además otros dispositivos de E/S (tales como un ratón, transceptores de control remoto, altavoces o cámaras). Estos dispositivos de E/S pueden ser beneficiosos para el funcionamiento de la zona 150 segura cuando, por ejemplo, un usuario desea introducir datos seguros (por ejemplo, un PIN de tarjeta) sin el riesgo de que el sistema 111 operativo los intercepte o modifique. El dispositivo 120 puede comprender además un puerto 118 de comunicaciones, que permite que el dispositivo se comunique con otros dispositivos. En el ejemplo anterior, el puerto 118 de comunicaciones puede ser útil para crear una conexión entre el dispositivo 120 y un ordenador remoto a través de una conexión de red. Además, dicho dispositivo 120 basado en ordenador puede ejecutar un sistema 111 operativo y una o más aplicaciones 112.

Finalmente, tal como se muestra en la Figura 1, el dispositivo 120 puede comprender también unos medios para indicar cuándo el dispositivo 120 está funcionando en un modo seguro, mostrados en la Figura 1 como "indicador" 193. Dicho indicador 193 puede ser, por ejemplo, un LED verde que está colocado en una carcasa exterior del dispositivo 120 y que es fácilmente visible para un usuario. Si el LED está encendido, el dispositivo 120 puede estar funcionando en un modo seguro (más específicamente, en "modo seguro de pantalla parcial", tal como se describe a continuación).

Como resultado, un dispositivo 120 según la presente divulgación puede comprender además hardware adicional que permita al mismo tomar el control de estos componentes periféricos del dispositivo 120, por ejemplo, desde el sistema 111 operativo. Por ejemplo, el dispositivo 120 seguro puede comprender un mezclador 181, que permite que la zona 150 segura controle la pantalla 123. El dispositivo 120 puede comprender también un interruptor 194 de teclado, que permite que la zona 150 segura controle el teclado 192. De esta manera, pueden usarse los mismos dispositivos de entrada/salida (por ejemplo, el teclado 192 y la pantalla 123) para soportar zonas tanto no seguras como seguras. Se entenderá que, aunque la Figura 1 muestra componentes tales como el mezclador 181 y el interruptor 194 de teclado implementados fuera de la zona 150 segura, en algunas realizaciones estos componentes pueden colocarse en el interior de la zona 150 segura.

Finalmente, la zona 150 segura puede estar opcionalmente asegurada físicamente, de manera que sea resistente a la manipulación. La zona 150 segura puede (de manera alternativa, o además de ser resistente a la manipulación) incorporar también una o más técnicas de detección de manipulación. Por ejemplo, ya se conocen y se han descrito en la técnica diversos métodos resistentes a la manipulación para proteger procesadores criptográficos; véase http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-641.pdf. En algunas realizaciones, puede ser deseable, por ejemplo, fabricar la zona 150 segura en el interior de un único chip. En otra realización, la zona 150 segura podría

tener una carcasa segura. En algunas de estas realizaciones, la zona 150 segura puede estar configurada para ejecutar una o más respuestas posibles si detecta que la integridad del chip se ha visto comprometida y/o si detecta una penetración a la carcasa segura. Estas respuestas pueden variar desde el borrado de cualquier clave de encriptado almacenada en el interior del almacenamiento 167 de claves hasta la destrucción física de la totalidad o parte de la zona 150 segura.

La Figura 2 muestra un método ejemplar mediante el cual una zona 150 segura según la presente divulgación puede aceptar una tarea para su ejecución; organizar el proceso de ejecución de la tarea; y realizar un borrado después de la ejecución de la tarea.

En la etapa 205, la interfaz 151 puede recibir el código desde la zona 152 no segura, y puede pasar este código al supervisor 160 para su ejecución por parte del procesador 162 seguro. Debe entenderse que, cada vez que se transfiere código en la etapa 205, el código puede incluir además datos de aplicación relacionados.

En la etapa 210, antes de ejecutar cualquier código recibido, el supervisor 160 puede borrar todos los datos almacenados en el interior de la memoria 164 de instrucciones y la memoria 165 de datos. Por ejemplo, el supervisor 160 podría poner a cero toda la memoria 164 de instrucciones y la memoria 165 de datos. Esto puede realizarse para evitar que el código, los datos antiguos, o ambos, afecten al código que se está cargando actualmente, y para evitar fugas de información entre diferentes fragmentos de código.

En algunas realizaciones, el proveedor de código puede haber encriptado el código (y cualquier dato de aplicación relacionado) antes de enviar el mismo a la zona 150 segura. Por ejemplo, el proveedor de código puede haber usado una clave pública correspondiente a una clave privada del supervisor 160 (que puede haber sido almacenada previamente en el almacenamiento 167 de claves, y que puede ser usada por el supervisor 160 para desencriptar el código) para encriptar el código. De esta manera, en la etapa 215, si el código se ha encriptado usando una clave pública del supervisor 160, el supervisor 160 puede extraer una copia de la clave privada correspondiente desde el almacenamiento 167 de claves e indicar al motor 121 criptográfico que desencripte el código (y cualquier dato asociado, si corresponde) usando esta clave privada.

Además, el código (y cualquier dato relacionado) puede haber sido firmado también digitalmente usando la clave privada del proveedor de código, garantizando la autenticidad del código. Para permitir la validación de la firma digital y del código firmado, puede proporcionarse con el código un certificado digital capaz de autenticar al proveedor de código. Por ejemplo, el proveedor de código puede tener una clave privada y un certificado digital correspondiente que ha sido firmado por un "certificado raíz" de una autoridad de certificación. En dicha implementación, el certificado raíz puede haber sido almacenado previamente en el almacenamiento 166 de certificados. En algunas realizaciones, en lugar de un único certificado, pueden incluirse "cadenas de certificados" completas con el código. En otras realizaciones, pueden usarse formas alternativas de obtener certificados intermedios (por ejemplo, emitiendo una solicitud a un servidor (no mostrado) a través del sistema 111 operativo OS y el puerto 118 de comunicaciones).

En la etapa 220, el supervisor 160 puede indicar al motor 121 criptográfico que valide la firma digital del proveedor de código. Esta validación de la firma digital normalmente incluirá la validación del certificado recibido con el código. Por ejemplo, si el certificado del proveedor de código fue firmado por una autoridad de certificación, tal como VeriSign®, el supervisor 160 puede tomar una copia del certificado raíz de VeriSign apropiado desde el almacenamiento 166 de certificados y puede verificar que este certificado raíz se usó para firmar el certificado del proveedor de código, realizando una validación de firma de infraestructura de clave pública (PKI) típica; en algunos casos, puede implementarse una validación más elaborada (por ejemplo, incluyendo "cadenas de certificados").

En algunas realizaciones, pueden usarse otros esquemas de validación de firmas (por ejemplo, los usados en la infraestructura de clave pública simple (SPKI)/infraestructura de seguridad distribuida simple (SDSI) o la "red de confianza" usada en privacidad bastante buena (PGP)).

En algunas realizaciones, el supervisor 160 puede realizar además una validación de la lista de revocación de certificados (CRL) para garantizar que todos los certificados implicados en la validación de la firma todavía son válidos. Puede obtenerse una CRL, por ejemplo, mediante una solicitud a un servidor que aloja CRLs. Esta solicitud puede realizarse, por ejemplo, mediante el sistema 111 operativo y el puerto 118 de comunicaciones de la zona 152 no segura.

En algunas realizaciones, puede usarse el protocolo de estado de certificado en línea (OCSP) para verificar la validez del certificado (en lugar de o además de la validación de la CRL).

En ciertas realizaciones, el certificado digital del proveedor de código puede diferir ligeramente de un certificado tradicional, de manera que contenga no solo una entrada de texto capaz de identificar al propietario del certificado (normalmente el campo "CN" de un certificado digital X.509), que indica el nombre del proveedor de código asociado con el certificado, sino que puede contener además una imagen (por ejemplo, PNG o JPEG) con una

representación visual de la identidad del proveedor de código. Esta imagen puede ser una parte del certificado digital en el sentido de que puede estar cubierta por la firma del emisor del certificado de la misma manera en la que deben cubrirse los otros campos del certificado; por ejemplo, en un certificado X.509, dicha "imagen de identidad" puede incluirse como una extensión en el campo "Extensiones". Tal como se describirá más detalladamente a continuación, en algunas realizaciones, puede ser deseable también mostrar esta "imagen de identidad" en una parte indicada previamente de la pantalla 123 mientras se ejecuta el código.

En la etapa 225, el supervisor 160 puede tomar el control de uno o más periféricos del dispositivo 120 informático que necesita para ejecutar el código recibido. Por ejemplo, el supervisor 160 puede tomar el control del teclado 192 y la pantalla 123 del dispositivo 120. En dicho caso, el supervisor 160 puede indicar al interruptor 194 de teclado que desconecte efectivamente el teclado 192 de los componentes no seguros (tal como el sistema 111 operativo) y que enrute toda la entrada de teclado a la zona 150 segura. El supervisor 160 puede indicar también al mezclador 181 que combine la salida desde el procesador 171 de imágenes y el decodificador 122 para formar una imagen en la pantalla 123, desconectando efectivamente la zona no segura con respecto a la pantalla 123.

En algunas realizaciones, en la etapa 230, puede determinarse si la tarea debe ejecutarse o no en modo de pantalla parcial. Si se determina que la tarea debe ejecutarse en modo seguro de pantalla parcial, puede ser deseable proporcionar una o más confirmaciones afirmativas al usuario de que el dispositivo 120 está funcionando ahora en el modo seguro de pantalla parcial. De esta manera, en la etapa 235, el supervisor 160 puede proporcionar la "imagen de identidad" a partir del certificado del proveedor de código (cuyo certificado ha sido validado en la etapa 220) al procesador 171 de imágenes, y puede indicar al mezclador 181 que muestre información desde el procesador 171 de imágenes en un área designada de la pantalla 123. En la etapa 240, el supervisor 160 puede encender el indicador 193.

En dichas realizaciones, el usuario puede confirmar que la tarea se está ejecutando en la zona 150 segura comprobando que el indicador 193 está encendido, y puede confirmar que la tarea se recibió desde un proveedor de código legítimo verificando que la información mostrada en el área designada de la pantalla 123 (por ejemplo, la imagen de identidad del certificado del proveedor de código) corresponde a las expectativas del usuario para esta tarea.

Si, por ejemplo, la información mostrada en la pantalla 123 no coincide con las expectativas del usuario (por ejemplo, el nombre del proveedor de código es incorrecto o se muestra una imagen de identidad incorrecta), el usuario puede realizar una acción apropiada para detener la tarea. Por ejemplo, el usuario podría presionar una combinación de teclas especial en el teclado 192 para indicar al supervisor 160 que finalice la sesión segura. De manera alternativa, si la información mostrada en la pantalla 123 coincide con las expectativas del usuario, pero el indicador 193 está apagado (lo que puede suceder, por ejemplo, si el sistema 111 operativo está comprometido y un atacante que controla el sistema 111 operativo simula la salida de pantalla sin relegar el control a la zona 150 segura), el usuario puede realizar de manera similar cualquier acción apropiada para detener la tarea. De esta manera, con el fin de que el usuario tenga la seguridad de que está trabajando en un entorno completamente seguro, (i) la imagen de identidad debe mostrarse en el área designada de la pantalla 123 y (ii) el indicador 193 debe estar encendido.

En ciertas realizaciones, el proveedor de código puede decidir que la tarea no requiere la provisión de un entorno completamente seguro para el usuario, sino que requiere acceso al área completa de la pantalla 123 (es decir, "modo seguro de pantalla completa"). Esto puede implementarse, por ejemplo, estableciendo un indicador booleano, que indique si se debe usar (es decir, mostrar la imagen de identidad en) el modo de pantalla completa o de pantalla parcial; para garantizar la seguridad, el supervisor 160 puede garantizar que el indicador 193 esté encendido solo en el modo seguro de pantalla parcial (es decir, cuando se muestra la imagen de identidad). Si, en la etapa 230, se determina que la tarea debe ejecutarse en el modo seguro de pantalla completa, el supervisor 160 puede conceder acceso al procesador 162 seguro a toda la pantalla 123 y proceder a la etapa 245. El modo de pantalla completa podría ser útil, por ejemplo, si el usuario simplemente desea desencriptar y mostrar contenido multimedia protegido que el usuario ya posee (la zona 150 segura proporciona capacidades técnicas útiles (tales como el motor 121 criptográfico y el decodificador 122)), pero no requiere el entorno completamente seguro que podría usar en situaciones tales como las comunicaciones seguras.

En la etapa 245, el supervisor 160 puede cargar el código recibido en la memoria 164 de instrucciones, puede almacenar cualquier dato de aplicación recibido en la memoria 165 de datos y puede indicar al procesador 162 seguro que empiece a ejecutar el código recibido.

En la etapa 250, el supervisor 160 puede empezar a esperar uno o más eventos relacionados con la ejecución del código. Por ejemplo, en la transición 252, el código que se ejecuta en el procesador 162 seguro puede solicitar al supervisor 160 que conmute al modo seguro de pantalla completa y obtenga acceso a toda la pantalla 123 (es decir, sin que se muestre la "imagen de identidad"). En dicho caso, tal como se ha descrito anteriormente, en la etapa 254, el supervisor 160 puede apagar el indicador 193 para mostrar que el supervisor 160 ya no controla la

salida a la pantalla 123 (y, por lo tanto, que una parte designada de la pantalla no puede ser usada para identificar al proveedor de código). El supervisor 160 puede indicar también al mezclador 181 que muestre solo información desde el decodificador 122 en la pantalla 123, concediendo efectivamente toda la pantalla 123 al código que se ejecuta en el procesador 162 seguro.

En la transición 255, el código que se ejecuta en el procesador 162 seguro puede solicitar al supervisor 160 que conmute a un modo seguro de pantalla parcial y vuelva a mostrar la imagen de identidad del proveedor de tareas. Esto puede ocurrir, por ejemplo, si un usuario desea confirmar que todavía se está ejecutando el código del mismo proveedor. En este caso, en la etapa 256, el supervisor 160 puede indicar al mezclador 181 que muestre información desde el decodificador 122 solo en la parte designada de la pantalla 123, mientras que en la otra parte el supervisor 160 empezará a mostrar de nuevo la imagen de identidad. El supervisor 160 puede encender también el indicador 193 para asegurar al usuario que la imagen mostrada es una imagen de identidad legítima.

Si, en la transición 257, la ejecución del código ha terminado, el código que se ejecuta en el procesador 162 seguro puede enviar una notificación de vuelta al supervisor 160 notificando al mismo que la ejecución del código ha terminado, y el supervisor 160 puede realizar ciertas etapas para devolver el control a la zona 152 no segura.

En algunas realizaciones puede ocurrir que, tal como se muestra en la transición 260, el código que se ejecuta en el procesador 162 seguro termine de manera anómala (por ejemplo, a través de una excepción del procesador 162 seguro).

En este caso, en la etapa 270, el supervisor 160 puede mostrar un mensaje de notificación al usuario que indica que una tarea segura se ha terminado de manera anómala y que el sistema está a punto de cambiar al modo operativo no seguro. El método puede esperar en la etapa 270 hasta que el usuario confirme que ha visto este mensaje de notificación (por ejemplo, presionando un botón en el teclado). Esta confirmación puede ser deseable, ya que, de lo contrario, el usuario puede tener la percepción errónea de que la tarea segura todavía se está ejecutando después de que haya terminado de manera anómala. En algunas realizaciones, este mensaje de notificación puede mostrarse solo si la tarea ha conmutado su modo desde el modo de pantalla parcial al modo de pantalla completa al menos una vez durante el tiempo de ejecución de la tarea.

En la etapa 275, el supervisor 160 puede empezar una rutina de "borrado" y puede borrar todas las memorias 164 y 165 de instrucciones y de datos (por ejemplo, poniendo las mismas a cero). En la etapa 280, el supervisor 160 puede apagar el indicador 193. Finalmente, en la etapa 285, el supervisor 160 puede transferir el control de cualquier dispositivo de E/S de vuelta a la zona 152 no segura; por ejemplo, podría indicar al interruptor 194 de teclado que procese la entrada del teclado 192 a través del sistema 111 operativo del dispositivo 120 informático, así como indicar al mezclador 181 que muestre la información que proviene desde el sistema 111 operativo, en la pantalla 123.

En ciertas realizaciones, puede ser deseable que una tarea incluya más de un fragmento de código. Esto puede permitir un procesamiento seguro en entornos sustancialmente más complicados, tal como en el caso del procesamiento seguro de tarjetas de crédito.

La Figura 3A ilustra una estructura de datos ejemplar para implementar tareas con dos fragmentos de código ejecutable (y datos asociados con cada fragmento de código ejecutable). Tal como se muestra en la Figura 3A, una "subtarea" 325 puede estar formada por el código2 321, los datos2 322 y una firma2 323 digital asociada. La tarea 305 puede estar formada por el código1 311 y los datos1 312 junto con la subtarea 325, todos los cuales pueden estar incluidos en la firma1 313.

Sin embargo, debería entenderse que una tarea 305 puede contener más de una subtarea 325 (teniendo potencialmente cada subtarea su propio conjunto de certificados y permisos digitales). Esto puede usarse, por ejemplo, de manera que la tarea pueda conmutar la ejecución a una de sus subtareas, esperar la finalización de la subtarea y, a continuación, conmutar a otra subtarea. Es posible además que una o más de las subtareas puedan contener subtareas adicionales y así sucesivamente.

Tanto la tarea 305 como la subtarea 325 pueden tener ciertos permisos, que describen el acceso que su código respectivo puede tener a diversas partes de la zona 150 segura y/o a cualquier dispositivo periférico (tal como el teclado 192 y/o la pantalla 123). Por ejemplo, puede permitirse que el código2 321 (en el interior de la subtarea 325) acceda a partes de la zona 150 segura, tal como se describe en los permisos2 320, mientras que los permisos1 310 pueden describir a qué partes de la zona 150 segura pueden ser accedidas por el código1 311. Estos permisos pueden almacenarse en certificados digitales firmados, por ejemplo, por una o más autoridades de certificación. Tal como se muestra en la Figura 3A, un primer certificado 314 digital puede contener permisos1 310, mientras que un segundo certificado digital 324 puede contener permisos2 320. Estos permisos pueden implementarse, por ejemplo, en el interior del campo "Uso extendido de claves" en un certificado X.509. En algunas realizaciones, es posible que los certificados no se incluyan en la tarea, pero pueden obtenerse por separado sin afectar a la

seguridad.

En algunas realizaciones, puede ser deseable que un desarrollador de código pueda asignar permisos al código que desarrolla. Por ejemplo, para una mayor seguridad, el desarrollador de código puede desear reducir los permisos asociados a una tarea o subtarea particular. En estas realizaciones, puede incluirse otro conjunto de permisos en el interior de la tarea (o la subtarea, según corresponda). Sin embargo, en la medida en que dichos permisos secundarios se incluyen en el interior de una tarea o subtarea, puede ser deseable que el supervisor 160 interprete estos permisos teniendo en cuenta cualquier permiso existente ya firmado por una autoridad de certificación. Por ejemplo, si el desarrollador de código desea añadir un conjunto de permisos adicional al código1 311, entonces estos permisos adicionales solo pueden modificar los permisos1 310. Puede ser deseable además exigir que cualesquiera de dichos permisos secundarios no puedan exceder sus respectivos permisos subyacentes. Por ejemplo, en el caso del código1 311, es posible que no se permita que los permisos adicionales amplíen el alcance de los permisos1 310 según lo dispuesto por la autoridad de certificación.

Cuando el supervisor 160 recibe una tarea (tal como la tarea 305 que contiene la subtarea 325, tal como se muestra en la Figura 3A), el supervisor 160 puede cargar el código1 311 y los datos1 312 en la memoria 164 de instrucciones y la memoria 165 de datos, según corresponda, para su posterior ejecución (por ejemplo, tal como se ha descrito más detalladamente anteriormente con respecto a la Figura 2 en la etapa 245). Durante la ejecución del código1, el supervisor 160 puede hacer cumplir las restricciones especificadas en los permisos1 310. Tras la recepción de la tarea, el supervisor 160 puede almacenar también los permisos2 320, el código2 321 y los datos2 322 en algún lugar en el interior de la zona 150 segura (por ejemplo, el código2 321 puede almacenarse en la memoria 164 de instrucciones y los datos2 322 pueden almacenarse en la memoria 165 de datos, potencialmente en forma encriptada para prevenir un uso indebido). En este punto, sin embargo, ni el código2 (ni sus permisos2 320 ni datos2 322 asociados) participa activamente en la ejecución del código1.

Tal como se ha descrito más detalladamente anteriormente, con respecto a la Figura 2, en la etapa 250, el supervisor 160 espera uno o más eventos relacionados con la tarea. Dichos eventos pueden incluir ciertos tipos de solicitudes desde el código actualmente en ejecución al supervisor 160. En realizaciones que soportan tareas complejas, por ejemplo, tal como se describe con respecto a la Figura 3A, el supervisor 160 puede soportar las solicitudes desde el código que se está ejecutando actualmente para ejecutar una o más subtareas. Por ejemplo, el supervisor 160 puede soportar una solicitud desde el código1 311 para ejecutar el código2 321. Dicha solicitud puede contener, por ejemplo, el inicio y el final de una región en el interior de la memoria 165 de datos que el código2 321 tiene permiso para usar para sus propios fines, y el inicio y el final de un área en el interior de la memoria 165 de datos que será accesible tanto para el código1 311 como para el código2 321 con el fin de intercambiar datos entre el código1 311 y el código2 321.

La Figura 3B es una división lógica ejemplar de la memoria 165 de datos en tres áreas, que pueden ser usadas por dos fragmentos de código, código1 311 y código2 321. Sin embargo, se entenderá que la ubicación y el tamaño relativos de las tres áreas son meramente ejemplares y pueden depender de muchos factores, incluyendo las preferencias de los desarrolladores del código1 y cualquier directriz para añadir una subtarea 325 a una tarea 305 (que puede ser creada, por ejemplo, por los desarrolladores de la subtarea 325). Tal como se muestra en la Figura 3B, el bloque 371 de memoria de datos es "privado" para el código1; el bloque 372 de memoria de datos es "privado" para el código2; y el bloque 370 de datos es un área compartida que puede ser accesible tanto para el código1 como para el código2. Por ejemplo, si se usa el bloque 370 de datos compartido, el código1 puede almacenar algunos datos en el interior del área 370 de memoria compartida a los que puede acceder el código2 cuando el código1 se suspende y el código2 se carga para su ejecución. De manera similar, el código2 puede almacenar datos en el interior del área 370 de memoria compartida a los que puede acceder el código1 cuando se termina el código2 y se reanuda el código1.

La Figura 4 ilustra un método ejemplar mediante el cual el supervisor 160 puede gestionar una solicitud desde una tarea 305 que se están ejecutando actualmente en el procesador 162 seguro para llamar a una subtarea 325.

En la etapa 410, el supervisor 160 puede indicar al procesador 162 seguro que suspenda la ejecución del código1 311. En la etapa 420, el supervisor 160 puede almacenar el estado actual de la tarea 305. Por ejemplo, el supervisor 160 puede almacenar el estado actual del código1 311. En ciertas realizaciones, esto podría requerir que el supervisor 160 almacene un valor actual de un registro de contador de programa y/o cualquier otro registro del procesador 162 seguro en el interior del almacenamiento 170 temporal del supervisor 160. El supervisor 160 puede preservar también el estado actual de cualquier memoria 165 de datos asociada con el código1 311. Esto puede incluir, por ejemplo, indicar al procesador 162 seguro (y/o la memoria 165 de datos) que restrinja el acceso del código que se ejecuta en el procesador 162 seguro (es decir, el código1 311) a las áreas 370 y 371 de memoria de datos. Además de, o en lugar de, dicha restricción, el supervisor 160 puede encriptar el área 371 de memoria de datos, y/o puede calcular un hash seguro (tal como SHA-256) del área 371 de memoria de datos y puede almacenar el valor de este hash en el interior del almacenamiento 170 temporal del supervisor 160. El supervisor 160 puede almacenar además el estado actual de cualquier periférico (tal como la pantalla 123 y/o el teclado 192).

Por ejemplo, el supervisor 160 puede leer el estado actual de cualquier LED en el teclado 192 y almacenar los mismos en el interior del almacenamiento 170 temporal. De manera similar, el supervisor 160 puede leer el estado de la pantalla 123 y almacenar el mismo (por ejemplo, como una matriz de píxeles) en el interior del almacenamiento 170 temporal.

En la etapa 430, el supervisor 160 puede conmutar el control de cualquier periférico según los permisos2 320 de la subtarea 325. Por ejemplo, en ciertas realizaciones, los permisos1 310 de la tarea 305 pueden permitir que el código1 311 acceda a ciertos periféricos (tales como el teclado 192), pero los permisos2 320 de la subtarea 325 pueden prohibir que el código2 321 acceda a algunos de los periféricos permitidos en los permisos1 310. Además, la pantalla 123 puede ser también borrada en esta etapa 430.

En la etapa 435, el supervisor 160 puede ejecutar una rutina de borrado para garantizar que el código2 321 de subtarea que está a punto de ejecutarse no se vea afectado por ningún dato que quede en la memoria 165 de datos por la ejecución del código1 311. Por ejemplo, el supervisor 160 puede poner a cero el área 372 de memoria de datos.

En la etapa 440, el supervisor 160 puede indicar al procesador 162 seguro que empiece a ejecutar el código2 321. Por ejemplo, el supervisor 160 puede ordenar al procesador 162 seguro que empiece la ejecución en un punto predefinido en el interior del código2. De manera alternativa, el punto de inicio del código2 puede incluirse en la tarea 305. El supervisor 160 puede proporcionar también una referencia al procesador 162 seguro que permita al mismo localizar y acceder a las áreas 370 y 372 de memoria de datos destinadas a ser usadas por el código2 321. Por ejemplo, en ciertas realizaciones, el supervisor 160 puede pasar un puntero al procesador 162 seguro que hace referencia a estas ubicaciones de memoria mediante uno o más registros situados en el interior del supervisor 160.

Durante la ejecución del código2 321 (tal como se muestra en la etapa 450), el supervisor 160 puede hacer cumplir cualquier permiso2 320 asociado con el código2 321. Por ejemplo, si en la etapa 450, el supervisor 160 recibe una solicitud desde el código2 321 para el control de pantalla completa (por ejemplo, correspondiente a la etapa 252, tal como se muestra en la Figura 2), el supervisor 160 puede verificar si los permisos2 320 permiten o no que el código2 321 asuma dicho control de pantalla completa y proceda con la etapa 252 solo si esos permisos2 320 permiten el control de pantalla completa.

En la etapa 460, el código2 321 puede haber terminado su ejecución, es decir, la subtarea 325 puede haberse completado. En la etapa 465, el supervisor 160 puede realizar una o más actividades de borrado en preparación para la transición de nuevo a la ejecución del código1; por ejemplo, el supervisor 160 puede poner a cero el área 372 de memoria. En la etapa 470, el supervisor 160 puede conmutar el control de cualquier periférico (tal como la pantalla 123 y/o el teclado 192) de nuevo al estado en el que se encontraba antes del inicio de la ejecución del código2 (o según los permisos1 310, si el estado de los periféricos no se almacenó en el momento en el que comenzó la subtarea 325).

En la etapa 475, el supervisor 160 puede restaurar el estado de la tarea 305, que se almacenó en la etapa 420. Por ejemplo, el supervisor 160 puede restaurar el estado del código1, de manera que empiece a ejecutarse donde lo dejó en el momento en el que se llamó a la subtarea 325. Esto puede conseguirse, por ejemplo, actualizando un contador de programa y/o cualquier otro registro del procesador 162 seguro a los valores almacenados en el almacenamiento 170 temporal, por ejemplo, durante la etapa 420. Si el área 371 de memoria se encriptó en la etapa 420, el supervisor 160 puede asegurarse de que sea desencriptado. Si se calculó un hash seguro en la etapa 420, el hash puede volverse a calcular y puede compararse con el valor de hash original. Si el hash calculado en esta etapa 475 no coincide con el valor de hash almacenado en la etapa 420, puede deducirse que el código2 ha conseguido violar la integridad del bloque 371 de memoria de datos del código1, y la ejecución del código1 no debería reanudarse (posiblemente mostrando un mensaje apropiado al usuario). Además, si en la etapa 420, el procesador 162 seguro y/o la memoria 165 de datos recibieron indicaciones para restringir el acceso solo a los bloques 370 y 372 de datos, en esta etapa 475 el supervisor 160 puede levantar esta restricción, y el código1 311 (y el procesador 162 seguro) puede recibir acceso a toda la memoria 165 de datos. Finalmente, puede restaurarse el estado de cualquier periférico (tal como el teclado 192) almacenado, por ejemplo, en la etapa 420. Si se almacenó el estado de la pantalla 123, el estado de la pantalla 123 puede restaurarse al valor almacenado; de lo contrario, puede mostrarse una pantalla 123 en blanco.

En la etapa 480, el supervisor 160 puede indicar al procesador 162 seguro que reanude la ejecución del código1 311.

Las realizaciones descritas hasta ahora han detallado tres modos operativos del dispositivo 120: "modo no seguro", "modo seguro de pantalla completa" y "modo seguro de pantalla parcial". Para indicar que el dispositivo 120 está funcionando en "modo seguro de pantalla parcial", tal como se ha descrito anteriormente, el indicador 193 puede estar encendido. En otra realización según la presente divulgación, el dispositivo 120 puede funcionar en un cuarto modo operativo "súper seguro" o "extra seguro", tal como se describirá más detalladamente a continuación. En

dicha realización, el indicador 193 puede tener otro estado "super seguro" (además de los estados "apagado" y "encendido" descritos anteriormente); este estado super seguro del indicador 193 puede indicar que el dispositivo 120 está funcionando actualmente en modo super seguro. En dicha realización, por ejemplo, el indicador 193 puede implementarse como dos LEDs separados (cada uno fácilmente visible para el usuario). Si un LED está encendido, puede indicar que el dispositivo 120 está funcionando en el modo seguro de pantalla parcial (descrito más detalladamente anteriormente); si dos LEDs están encendidos, el dispositivo 120 puede estar funcionando en un modo súper seguro o extra seguro. Puede especificarse si cualquier fragmento de código (tal como el código1 311 o el código2 321) puede conmutar o no al modo super seguro en el interior de sus campos de permisos respectivos (por ejemplo, permisos1 310 o permisos2 320, respectivamente).

La Figura 5 es una realización ejemplar que demuestra cómo puede realizarse una tarea particular, una transacción segura con tarjeta de crédito, en el interior de la presente divulgación. Además de su aplicabilidad a la presente divulgación, el método mostrado en la Figura 5 puede usarse para implementar la funcionalidad descrita en la solicitud de patente provisional US Nº 61/623.702, presentada el 13 de Abril de 2012, titulada "Apparatuses, Methods And Systems For Computer-Based Secure Transactions", y que tiene a Sergey Ignatchenko y Dmytro Ivanchykhin como inventores.

En la etapa 500, un usuario puede navegar por Internet usando un navegador web (que se ejecuta como una aplicación 112 en un sistema 111 operativo) en su ordenador 120. Usando su navegador, el usuario puede visitar un sitio de comercio electrónico del comerciante, puede seleccionar algunos artículos a comprar y puede proceder a pagar seleccionando un enlace web correspondiente.

En la etapa 505, el navegador del usuario puede realizar una solicitud al servidor definido en el enlace web, puede recibir la respuesta y puede determinar que los datos recibidos son una tarea que debe realizarse en un modo seguro. El navegador puede determinar esto, por ejemplo, mediante una forma especial de la URL en el enlace web, o mediante la estructura de los datos recibidos.

En la etapa 510, el navegador web puede enviar la tarea a la interfaz 151 de una zona 150 segura que se está ejecutando en el dispositivo 120 informático del usuario. Esto puede realizarse, por ejemplo, mediante una llamada al sistema 111 operativo, que a su vez puede pasar la información a la interfaz 151.

La tarea enviada a la zona 150 segura puede estructurarse tal como se muestra en la Figura 3A, en el que la tarea 305 general (es decir, el código1 311 y los datos1 312) puede ser proporcionada por el comerciante (siendo la firma1 313 la firma digital del comerciante en este código y datos), y en el que la subtarea 325 (es decir, el código2 321 y los datos2 322) puede ser proporcionada, por ejemplo, por el banco del comerciante (al que se hace referencia normalmente como banco adquirente), o cualquier otra entidad en la que confíe el banco del comerciante, con la firma2 323 como la firma digital del banco/entidad de confianza.

Para los propósitos del presente ejemplo, puede suponerse que puede ejercerse un control significativo sobre los bancos o entidades de confianza, de manera que la probabilidad de que el código2 o los datos 2 contengan códigos o datos maliciosos sea muy baja o se elimine. Por el contrario, se supone que los comerciantes no son tan seguros y que existe la posibilidad de que el código o los datos proporcionados por un comerciante puedan ser maliciosos. De esta manera, en este ejemplo, ¡puede suponerse que una autoridad de certificación no emitirá ningún certificado! 314 para una tarea 305 (del comerciante) de manera que la tarea tendría permisos1 310 para ejecutar el código1 311 en el modo "super seguro". Sin embargo, las autoridades de certificación pueden emitir un certificado2 324 con permisos2 320 que permiten que una subtarea 325 (del banco) entre al modo "súper seguro". Este enfoque puede usarse para garantizar que un comerciante malintencionado no pueda acceder a los datos de una tarjeta de crédito (incluyendo un PIN) y puede usarse además para proporcionar un registro probatorio suficiente para la resolución de futuras disputas.

En la etapa 515, el supervisor 160 puede recibir la tarea 305, verificar su integridad y cargar la misma en la memoria 165 de instrucciones de la zona 150 segura (por ejemplo, según las etapas 205 - 240, tal como se analiza más detalladamente con respecto a la Figura 2). En particular, el supervisor 160 puede mostrar al usuario una "imagen de identidad" del certificado1 314, lo que le dará la oportunidad al usuario de asegurarse de que la tarea 305 proviene de una fuente esperada, es decir, el proveedor de comercio electrónico al que pretende comprar un artículo.

En la etapa 520, el código1 311 (que pertenece al comerciante) puede ejecutarse, en un modo seguro. Este código puede ejecutarse con el supervisor 160 mostrando la "imagen de identidad" del certificado1 314 en el área designada de la pantalla 123, y con el indicador 193 en un estado "seguro". Este código1 311 puede implementar cierta interacción (opcional) con el usuario y/o comunicaciones seguras con un servidor remoto. Por ejemplo, puede solicitarse al usuario que seleccione un producto u opciones de productos (tales como color, tamaño, etc.). Esta interacción puede continuar hasta que el usuario decida salir (en cuyo caso el código1 puede terminar), o hasta que el usuario decida seguir adelante con una compra, especificando todos los términos de la compra (incluyendo el

importe total a cobrar, producto, dirección de entrega, etc.)

En los casos en los que el código1 inicia una conexión segura con el servidor de un comerciante, el código1 311 puede garantizar que el sistema 111 operativo no pueda interceptar o modificar la comunicación. Para conseguir esto, en una realización, el código1 311 puede contener una implementación de una pila TCP/IP completa, y el procesador 162 seguro puede acceder directamente al puerto 118 de comunicaciones del dispositivo 120 informático. En otra realización, una pila TCP/IP (que incluye opcionalmente soporte SSL) puede ser implementada por el supervisor 160. En todavía otra realización, el código1 311 que se está ejecutando en el procesador 162 seguro puede usar las capacidades de transporte de red (por ejemplo, la pila TCP/IP) del sistema 111 operativo no seguro, de manera que el sistema 111 operativo se use meramente como un intermediario para transmitir paquetes o mensajes seguros (por ejemplo, mensajes SSL) sin la capacidad para determinar y/o alterar el contenido de los paquetes o mensajes seguros. En este último caso, el procesador 162 seguro puede comunicarse, a través de la interfaz 151, con el sistema 111 operativo no seguro. En particular, el código1 311 puede enviar solicitudes para realizar operaciones TCP/IP y, a continuación, puede enviar y recibir mensajes SSL a través de una conexión TCP/IP establecida en su nombre por el sistema 111 operativo. En este ejemplo, el sistema 111 operativo proporciona solo capacidades TCP/IP, en el que toda la criptografía SSL reside en el interior de la zona 150 segura.

En la etapa 525, si se han finalizado todos los términos de la compra y el usuario desea continuar con la compra, el código1 311 puede recopilar datos de finalización de la transacción. Esta información puede incluir detalles de la transacción (que pueden estar en forma de una descripción textual o una o más imágenes, que pueden incluir, por ejemplo, una descripción de los productos implicados y/o la dirección de entrega) y la cantidad de la transacción (que puede incluir la moneda de la transacción).

En la etapa 527, el código1 311 puede designar alguna área de la memoria 165 de datos para ser usada como área 370 de memoria compartida y alguna área para que el código2 321 la use como su propia área 372 de memoria de datos; almacenar los datos de finalización de la transacción (por ejemplo, tal como se generan en la etapa 525) en el área 370 de memoria compartida; y, a continuación, solicitar al supervisor 160 que empiece a ejecutar el código2. A continuación, el supervisor 160 puede pasar el control al código2 321, por ejemplo, según las etapas 410-450, tal como se ha descrito anteriormente con respecto a la Figura 4.

Si los permisos2 320 indican que el código2 321 puede ejecutarse en modo "súper seguro" o "extra seguro", entonces el supervisor 160 puede conmutar el indicador 193 a un estado "súper seguro" (por ejemplo, encendiendo dos luces LED del indicador 193), lo que indica al usuario que la información que el usuario pueda introducir no será accesible para el comerciante, sino solo para el banco (o una entidad de confianza equivalente). Además, los permisos2 pueden especificar que el supervisor 160 puede permitir que el procesador 162 seguro acceda a un lector de tarjetas de crédito (no mostrado).

En la etapa 530, el código2 321 puede mostrar los datos de finalización de la transacción (recopilados en la etapa 525 y pasados al código2 a través del área 370 de memoria compartida en la etapa 527), y puede solicitar al usuario (por ejemplo, mostrando instrucciones en la pantalla 123) que confirme que el usuario va a completar la transacción y/o que proporcione información relevante. Por ejemplo, puede solicitarse al usuario que inserte una tarjeta de crédito en un lector de tarjetas (no mostrado), introduzca un número PIN usando el teclado y/o confirme que está de acuerdo con los detalles de la transacción (por ejemplo, artículos, cantidades, totales, etc.) tal como se muestra en la pantalla 123. El código2 321 puede procesar cualquier entrada disponible (por ejemplo, el código2 321 puede leer el PIN cuando se introduce en el teclado 192 y puede leer los datos de la tarjeta de crédito desde el lector de tarjetas de crédito).

En realizaciones en las que el usuario ha introducido un PIN, en la etapa 535, el código2 puede realizar una verificación del PIN. Esta verificación de PIN puede implementarse como una verificación de "PIN sin conexión" (es decir, usando la tarjeta para validar el PIN) o como una verificación de "PIN en línea" (es decir, emitiendo una solicitud a la red de pagos, a través del servidor, para validar el PIN).

En la etapa 540, si la tarjeta de crédito es una ICC y el lector de tarjetas tiene capacidades de lectura de ICC, el código2 321 puede firmar digitalmente los detalles de transacción mostrados al usuario por la ICC (por ejemplo, usando una MAC de una manera similar a la que se usa en los protocolos EMV, o usando criptografía pública/privada). La firma digital puede incluir los detalles de la transacción, tal como se mostraron al usuario, o, de manera alternativa, puede incluir los valores hash que se calcularon para dichos detalles de la transacción. La firma digital puede tener en cuenta también un PIN, si se usó. Incluso si se tiene en cuenta el PIN en el cálculo de la firma digital, puede ser deseable garantizar que el propio PIN no esté incluido en el mensaje enviado al comerciante para evitar revelar el PIN al comerciante.

En la etapa 542, el código2 321 puede enviar una solicitud de transacción a un servidor del comerciante. Esta solicitud puede incluir datos de finalización de la transacción (por ejemplo, identidad del comerciante, detalles de la transacción y cantidad de la transacción), así como una firma digital, si se calculó una (por ejemplo, tal como en la

etapa 540). La solicitud de transacción puede enviarse al servidor del comerciante, por ejemplo, a través de un canal SSL seguro. Dicho canal SSL puede ser establecido por el código2 de manera independiente (de una manera similar a la descrita en la etapa 520), o puede usarse el contexto SSL de la conexión SSL establecida por el código1. En el último caso, en la etapa 527, el código1 311 puede almacenar el contexto SSL existente en el área 370 de memoria compartida (en el que el contexto SSL es una estructura de datos que representa el contexto de la conexión SSL), y el código2 321 puede usar este contexto para continuar la conversación SSL existente.

En la etapa 545, el código2 321 puede esperar un resultado de la transacción desde el servidor del comerciante. Cuando el código2 recibe los resultados de la transacción desde el servidor del comerciante, puede mostrar los resultados de la transacción al usuario y puede esperar la confirmación del usuario de que ha visto los resultados de la transacción. A continuación, el código2 321 puede escribir los resultados en el área 370 de memoria compartida (para su uso posterior por el código1 311) y puede finalizar.

En algunas realizaciones, en lugar de que el código2 321 realice las etapas 542 y 545, el código2 321 puede preparar la solicitud de transacción (incluyendo la firma digital desde la etapa 540, si es necesario) y, a continuación, puede finalizar, dejando las etapas restantes de enviar la transacción al servidor e interpretar la respuesta al código1 311.

Después de que el código2 321 termina, el supervisor 160 puede realizar cualquier etapa necesaria para conmutar de nuevo al código1 311 que se está ejecutando. Por ejemplo, el supervisor 160 puede realizar etapas similares a las analizadas con respecto a las etapas 460-480 de la Figura 4, y puede conmutar además el indicador 193 de nuevo al estado "seguro" desde el estado "súper seguro", puede desconectar el lector de tarjetas de crédito desde el procesador 162 seguro y reanudar la ejecución del código1 311.

En la etapa 550, el código1 puede analizar los resultados de la transacción que el código2 almacenó en el área 370 de memoria compartida y, dependiendo de los resultados del análisis, puede continuar la interacción con el usuario o finalizar.

La Figura 6 es un diagrama de flujo que ilustra cómo un comerciante puede procesar una solicitud de transacción recibida desde un dispositivo 120 informático. En la etapa 610, el comerciante puede recibir una solicitud de transacción (por ejemplo, formada por el dispositivo informático en la etapa 542) y puede verificar que la moneda y la cantidad de la transacción, y el hash de la ID del comerciante y los detalles de la transacción (que, tal como se ha descrito con respecto a la etapa 540, pueden ser los detalles reales de la transacción o los valores hash respectivos, dependiendo de la realización) se ajustan a las expectativas del comerciante de lo que se mostró y fue aceptado por el usuario. Si, en la etapa 620, el comerciante determina que los valores recibidos no son los mismos que los esperados por el comerciante, el comerciante puede optar por rechazar la transacción en la etapa 622 como potencialmente basada en información fraudulenta o maliciosa, y puede proceder a la etapa 650. Si, por otra parte, en la etapa 620, los valores recibidos son los mismos que los esperados por el comerciante, en la etapa 625, el comerciante puede guardar una copia del mensaje para sus propios registros y con propósitos de resolución de posibles futuras disputas y puede transmitir el mensaje al banco para un posterior procesamiento de la transacción.

Debe entenderse que, aunque el comerciante puede verificar si los detalles de la transacción recibida se ajustan o no a las expectativas del comerciante, es posible que el comerciante no tenga acceso a información confidencial del usuario (tal como, por ejemplo, el número de tarjeta de crédito del usuario, el PIN u otra información que esté destinada a ser recibida por el banco y no por el comerciante). De manera similar, es posible que el comerciante no pueda verificar la firma del mensaje como firmada por el código2, mientras que el banco puede tener esa capacidad.

En la etapa 630, el banco puede recibir el mensaje, verificar su firma y realizar otros procedimientos de aprobación (por ejemplo, verificar si el usuario tiene o no fondos suficientes para realizar la transacción). En particular, si la verificación de la firma falla, el banco puede rechazar la transacción al considerar que está basada en información inconsistente.

En la etapa 640, el banco informa al comerciante acerca de si la transacción ha sido aprobada o denegada. En la etapa 650, el comerciante puede reenviar la información acerca del estado de la transacción (por ejemplo, si se ha aceptado o rechazado, fecha de entrega estimada, etc.) al ordenador 120. Además, en la etapa 660, el comerciante puede almacenar la confirmación desde el banco y la información enviada al usuario para sus propios registros y con propósito de resolución de posibles futuras disputas.

Por separado, si la transacción ha sido aceptada, el comerciante puede proceder en un tiempo apropiado a enviar o entregar los bienes comprados al usuario.

Debe reconocerse que, incluso si el comerciante puede no ser capaz de verificar la firma (que es el caso si se usan firmas de tipo MAC), la seguridad e integridad de la transacción se mantienen por el hecho de que el banco tiene la capacidad de verificar la firma. De esta manera, incluso si los datos de transacción comunicados por el dispositivo

120 al comerciante coinciden con las expectativas del comerciante, el banco todavía puede rechazar la transacción si el banco no puede verificar la firma. Por consiguiente, una transacción se aprueba después de que tanto los datos de la transacción como la firma hayan sido verificados de manera apropiada.

Aunque se han ilustrado y descrito realizaciones y aplicaciones específicas de la presente invención, debe entenderse que la invención no está limitada a la configuración y a los componentes precisos divulgados en el presente documento. Los términos, las descripciones y las figuras usados en el presente documento se establecen solo a modo de ilustración y no pretenden ser limitaciones. Diversas modificaciones, cambios y variaciones que serán evidentes para las personas expertas en la técnica pueden realizarse en la disposición, operación y detalles de los aparatos, métodos y sistemas de la presente invención divulgados en el presente documento. A modo de ejemplo no limitativo, se entenderá que los diagramas de bloques incluidos en el presente documento pretenden mostrar un subconjunto seleccionado de los componentes de cada aparato y sistema, y cada aparato y sistema representado puede incluir otros componentes que no se muestran en los dibujos. Además, las personas expertas en la técnica reconocerán que ciertas etapas y funcionalidades descritas en el presente documento pueden omitirse o reordenarse sin restar importancia al alcance o rendimiento de las realizaciones descritas en el presente documento.

Los diversos bloques lógicos, módulos, circuitos y etapas de algoritmo ilustrativos descritos en conexión con las realizaciones divulgadas en el presente documento pueden implementarse como hardware electrónico, software informático o combinaciones de ambos. Para ilustrar claramente esta intercambiabilidad de hardware y software, diversos componentes ilustrativos, bloques, módulos, circuitos y etapas se han descrito anteriormente de manera general en términos de su funcionalidad. El hecho de que dicha funcionalidad se implemente como hardware o software depende de la aplicación particular y de las restricciones de diseño impuestas sobre el sistema general. La funcionalidad descrita puede implementarse de diversas maneras para cada aplicación particular, tal como usando cualquier combinación de microprocesadores, microcontroladores, matrices de puertas programables en campo (FPGAs), circuitos integrados específicos de la aplicación (ASICs) y/o sistema en un chip (SoC), pero dichas decisiones de implementación no deben interpretarse como una desviación respecto al alcance de la presente invención.

Las etapas de un método o algoritmo descrito en conexión con las realizaciones divulgadas en el presente documento, pueden realizarse directamente en hardware, en un módulo de software ejecutado por un procesador o en una combinación de los dos. Un módulo de software puede residir en memoria RAM, memoria flash, memoria ROM, memoria EPROM, memoria EEPROM, registros, disco duro, un disco extraíble, un CD-ROM o cualquier otra forma de medio de almacenamiento conocido en la técnica.

Los métodos divulgados en el presente documento comprenden una o más etapas o acciones para conseguir el método descrito. Las etapas y/o acciones del método pueden intercambiarse entre sí sin apartarse del alcance de la presente invención. En otras palabras, a menos que se requiera un orden específico de etapas o acciones para el funcionamiento apropiado de la realización, el orden y/o el uso de etapas y/o acciones específicos pueden modificarse sin apartarse del alcance de la presente invención.

REIVINDICACIONES

1. Aparato (120), que comprende:

una memoria configurada para almacenar datos;

una zona (150) segura que comprende una interfaz (151) y configurada para ejecutar una tarea que comprende una subtarea que comunica uno o más paquetes de datos a través de una red, en el que los datos relacionados con la subtarea se borran de la memoria después de ejecutar la subtarea; y

una zona (152) no segura acoplada a la zona (150) segura a través de la interfaz (151), en el que la zona segura está configurada para usar las capacidades de red de la zona no segura para comunicar los uno o más paquetes de datos a través de la red según la subtarea, y en el que las capacidades de red de la zona no segura reciben los paquetes de datos comunicados desde la zona segura a través de la interfaz.

2. Aparato (120) según la reivindicación 1, en el que las capacidades de red de la zona (152) no segura incluyen una pila TCP/IP.

3. Aparato (120) según la reivindicación 1, en el que la tarea y la subtarea tienen un código ejecutable respectivo, y la tarea y la subtarea están firmadas digitalmente por los proveedores de código respectivos, y

en el que la zona (150) segura está configurada para aplicar conjuntos de permisos respectivos mientras se ejecuta el código ejecutable respectivo de la tarea y la subtarea, en el que el conjunto de permisos respectivo para la tarea se basa en al menos una de entre la información asociada con la tarea firmada y la información en un certificado digital del proveedor de código respectivo para la tarea, y en el que el conjunto de permisos respectivo para la subtarea se basa en al menos una de entre la información asociada con la subtarea firmada y la información en un certificado digital del proveedor de código respectivo para la subtarea.

4. Aparato (120) según la reivindicación 3, que comprende además un dispositivo periférico, en el que la zona (150) segura está configurada para asumir el control del dispositivo periférico en un modo seguro y la zona (152) no segura está configurada para recibir el control del dispositivo periférico transferido desde la zona segura en un modo no seguro, y en el que los conjuntos de permisos respectivos de la tarea y la subtarea prescriben derechos de acceso separados al dispositivo periférico para el código ejecutable respectivo.

5. Aparato (120) según la reivindicación 4, que comprende además un dispositivo de conmutación para el dispositivo periférico, en el que el dispositivo periférico es un dispositivo de entrada y el dispositivo de conmutación está configurado para conmutar la entrada recibida por el dispositivo de entrada a la zona (150) segura en el modo seguro y a la zona (152) no segura en el modo no seguro.

6. Aparato (120) según la reivindicación 4, que comprende además un dispositivo de mezclado acoplado al dispositivo periférico, en el que el dispositivo periférico es una pantalla y el dispositivo de mezclado está configurado para mezclar las salidas desde la zona segura y la zona no segura para generar una salida de visualización a la pantalla.

7. Aparato (120) según la reivindicación 3, en el que la zona segura comprende un procesador seguro para ejecutar la tarea y la subtarea, el procesador (162) seguro comprende un almacenamiento temporal configurado para almacenar un estado actual de un área de memoria de datos usada por la tarea cuando se conmuta para ejecutar la subtarea.

8. Aparato (120) según la reivindicación 7, en el que el almacenamiento temporal está configurado además para almacenar un hash seguro del área de memoria de datos usada por la tarea cuando se conmuta para ejecutar la subtarea.

9. Aparato (120) según la reivindicación 7, en el que el almacenamiento temporal está configurado además para almacenar un estado actual de cualquier dispositivo periférico usado por la tarea cuando se conmuta para ejecutar la subtarea.

10. Aparato (120) según la reivindicación 3, en el que la zona segura comprende una memoria para la ejecución de código, y en el que la zona segura está configurada para:

cargar y ejecutar el código ejecutable de la tarea;

suspender la ejecución del código ejecutable de la tarea;

preservar un estado de la tarea;

conceder derechos de acceso a los dispositivos periféricos del aparato según el conjunto de permisos de la subtarea; y

ejecutar la subtarea.

11. Método que comprende:

recibir una tarea en una zona (150) segura de un aparato (120) acoplada a una zona (152) no segura del aparato a través de una interfaz (151);

ejecutar una subtarea de la tarea por parte de la zona segura, en el que la ejecución de la subtarea comprende comunicar uno o más paquetes de datos a través de una red usando capacidades de red proporcionadas por la zona no segura, y en el que las capacidades de red de la zona no segura reciben los paquetes de datos comunicados desde la zona segura a través de la interfaz; y

borrar una memoria de datos relacionados con la subtarea después de ejecutar la subtarea.

12. Método según la reivindicación 11, en el que las capacidades de red de la zona (152) no segura incluyen una pila TCP/IP.

13. Método según la reivindicación 11, en el que el método comprende además aplicar los conjuntos de permisos respectivos mientras se ejecuta el código ejecutable respectivo de la tarea y la subtarea, en el que la tarea y la subtarea tienen un código ejecutable respectivo, y la tarea y la subtarea están firmadas digitalmente por los proveedores de código respectivos, en el que el conjunto de permisos respectivo para la tarea está basado en al menos una de entre la información asociada con la tarea firmada y la información en un certificado digital del proveedor de código respectivo para la tarea, y en el que el conjunto de permisos respectivo para la subtarea está basado en al menos una de entre la información asociada con la subtarea firmada y la información en un certificado digital del proveedor de código respectivo para la subtarea.

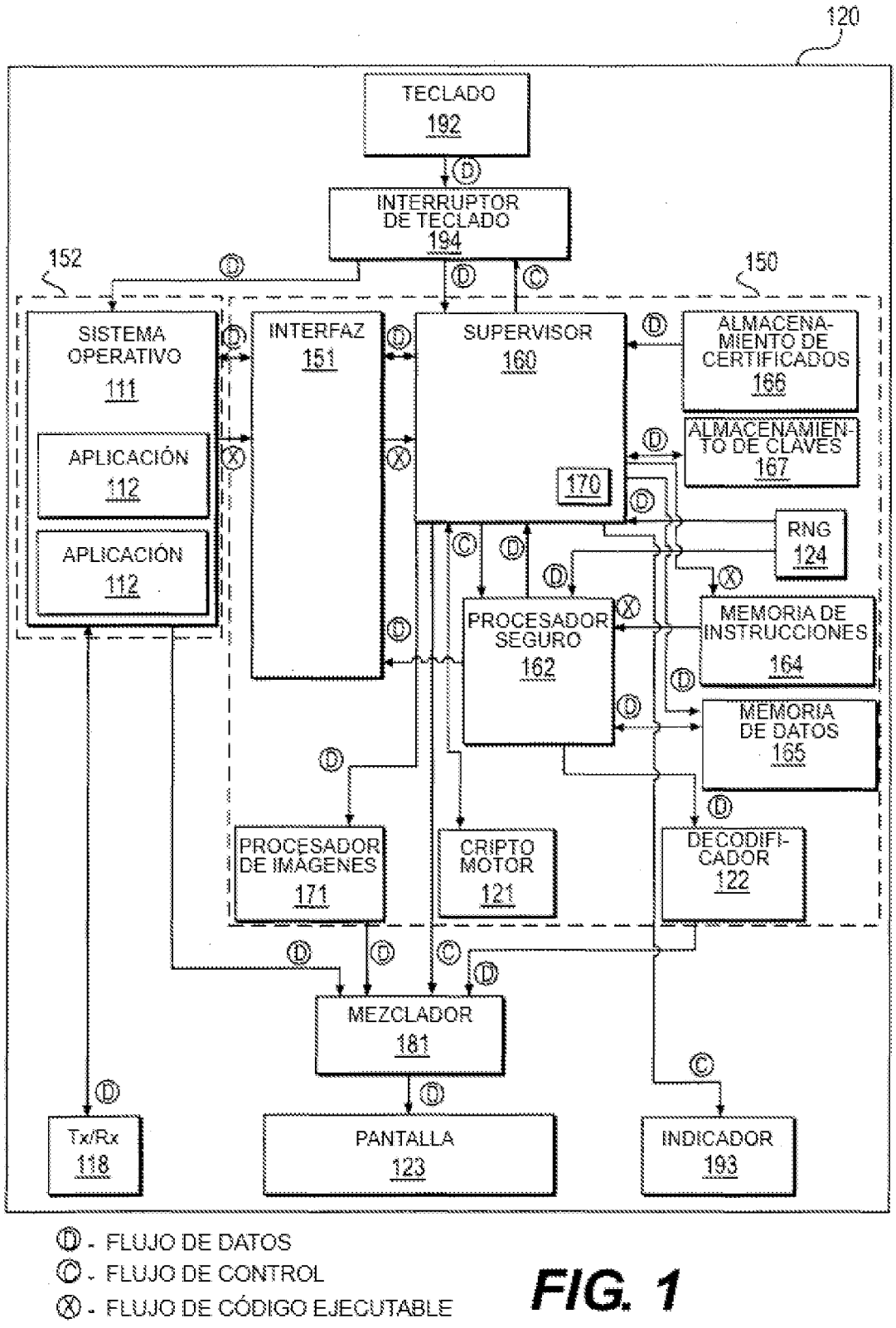
120
TECLADO
192
INTERRUPTOR DE TECLADO
194
152
150
SISTEMA OPERATIVO
111
APLICACIÓN
112
APLICACIÓN
112
INTERFAZ
151
SUPERVISOR
160
ALMACENA-MIENTO DE CERTIFICADOS
166
ALMACENAMIEN-TO DE CLAVES
167
170
RNG
124
PROCESADOR SEGURO
162
MEMORIA DE INSTRUCCIONES
164
MEMORIA DE DATOS
165
PROCESADOR DE IMÁGENES
171
CRIPTO MOTOR
121
DECODIFI-CADOR
122
MEZCLADOR
181
Tx/Rx
118
PANTALLA
123
INDICADOR
193
Ⓓ - FLUJO DE DATOS
Ⓒ - FLUJO DE CONTROL
Ⓧ - FLUJO DE CÓDIGO EJECUTABLE

FIG. 1

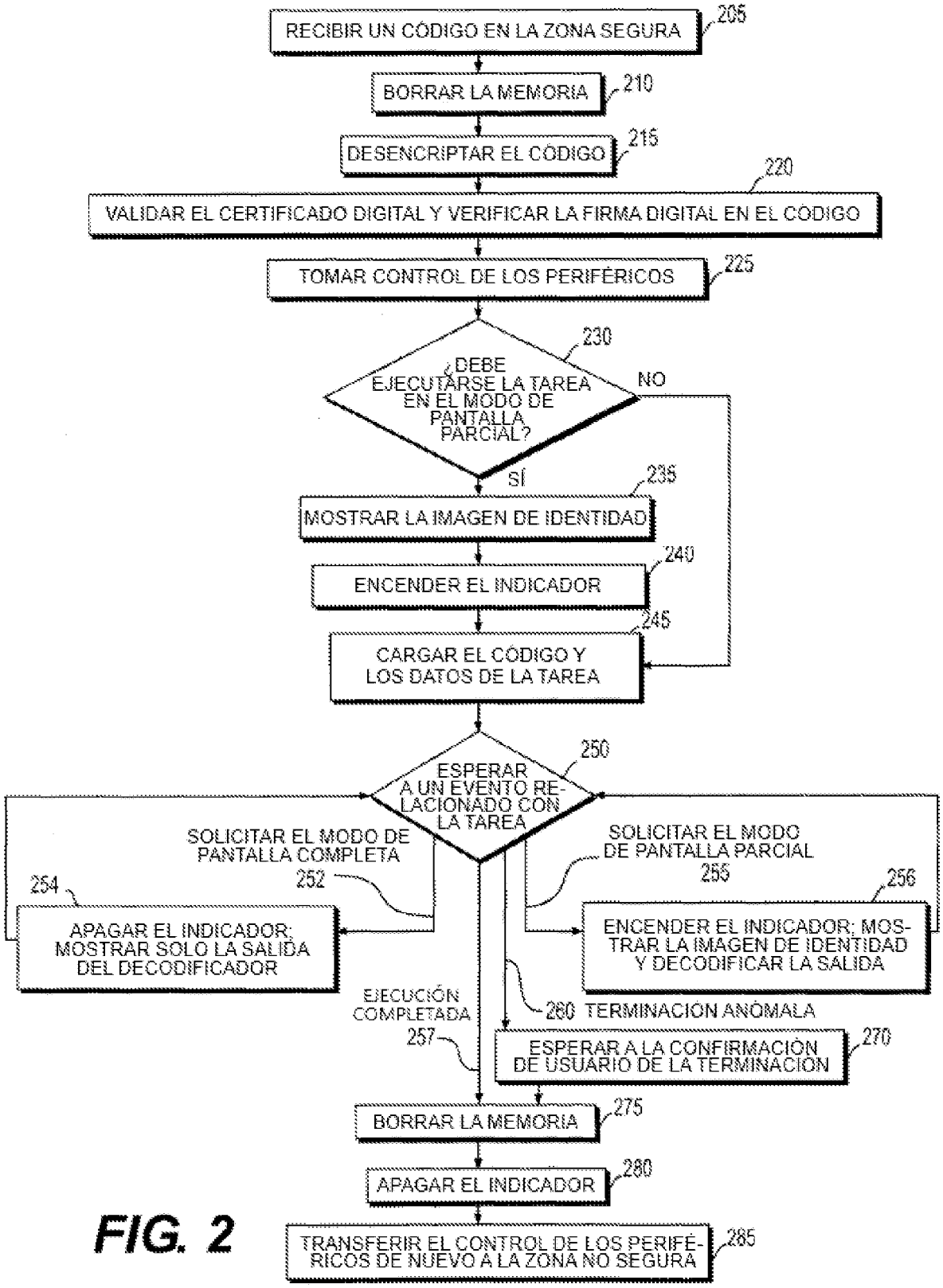
RECIBIR UN CÓDIGO EN LA ZONA SEGURA
205
BORRAR LA MEMORIA
210
DESENCRIPTAR EL CÓDIGO
215
VALIDAR EL CERTIFICADO DIGITAL Y VERIFICAR LA FIRMA DIGITAL EN EL CÓDIGO
220
TOMAR CONTROL DE LOS PERIFÉRICOS
225
¿DEBE EJECUTARSE LA TAREA EN EL MODO DE PANTALLA PARCIAL?
230
NO
SÍ
MOSTRAR LA IMAGEN DE IDENTIDAD
235
ENCENDER EL INDICADOR
240
CARGAR EL CÓDIGO Y LOS DATOS DE LA TAREA
245
ESPERAR A UN EVENTO RELACIONADO CON LA TAREA
250
SOLICITAR EL MODO DE PANTALLA COMPLETA
252
SOLICITAR EL MODO DE PANTALLA PARCIAL
255
254
APAGAR EL INDICADOR; MOSTRAR SOLO LA SALIDA DEL DECODIFICADOR
256
ENCENDER EL INDICADOR; MOSTRAR LA IMAGEN DE IDENTIDAD Y DECODIFICAR LA SALIDA
EJECUCIÓN COMPLETADA
257
260 TERMINACIÓN ANÓMALA
ESPERAR A LA CONFIRMACIÓN DE USUARIO DE LA TERMINACIÓN
270
BORRAR LA MEMORIA
275
APAGAR EL INDICADOR
280
TRANSFERIR EL CONTROL DE LOS PERIFÉRICOS DE NUEVO A LA ZONA NO SEGURA
285

FIG. 2

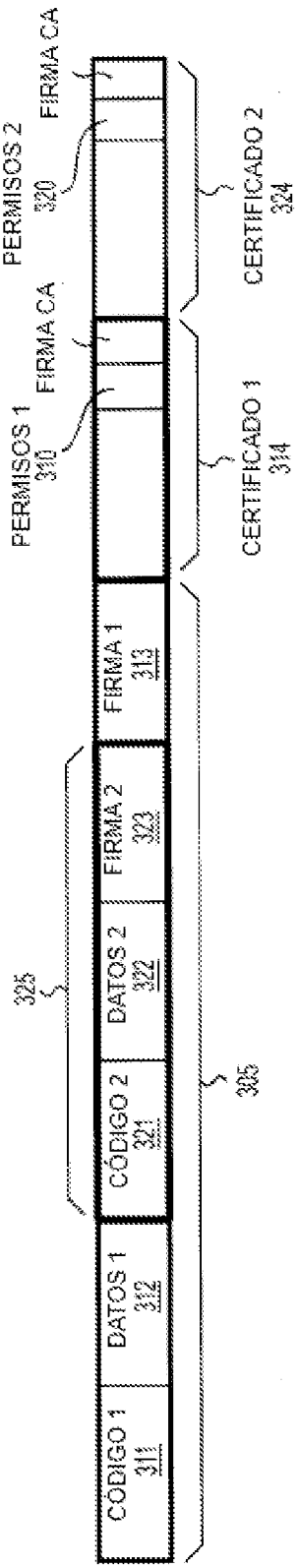
PERMISOS 1
310
FIRMA CA
PERMISOS 2
320
FIRMA CA
CÓDIGO 1
311
DATOS 1
312
CÓDIGO 2
321
DATOS 2
322
FIRMA 2
323
FIRMA 1
313
CERTIFICADO 1
314
CERTIFICADO 2
324
325
305

FIG. 3A

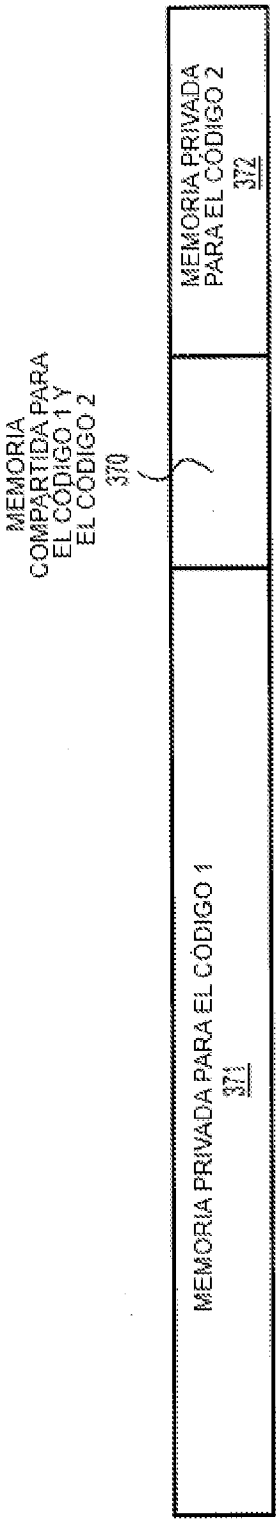
MEMORIA
COMPARTIDA PARA
EL CÓDIGO 1 Y
EL CÓDIGO 2
370
MEMORIA PRIVADA
PARA EL CÓDIGO 2
372
MEMORIA PRIVADA PARA EL CÓDIGO 1
371

FIG. 3B

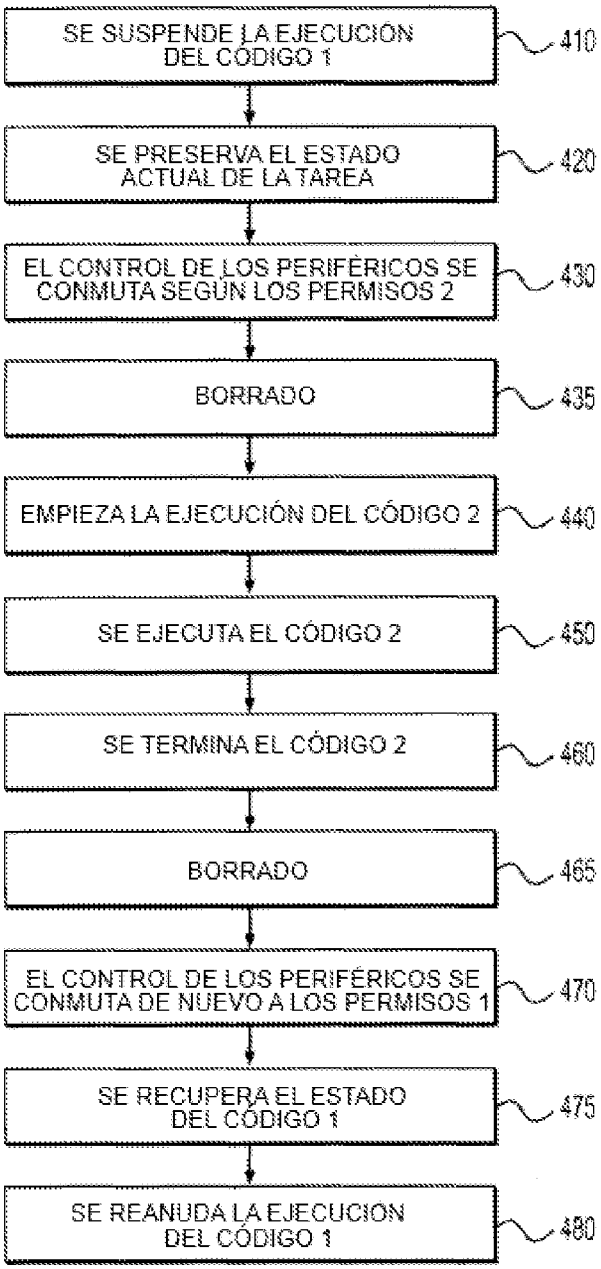
SE SUSPENDE LA EJECUCIÓN DEL CÓDIGO 1
410
SE PRESERVA EL ESTADO ACTUAL DE LA TAREA
420
EL CONTROL DE LOS PERIFÉRICOS SE CONMUTA SEGÚN LOS PERMISOS 2
430
BORRADO
435
EMPIEZA LA EJECUCIÓN DEL CÓDIGO 2
440
SE EJECUTA EL CÓDIGO 2
450
SE TERMINA EL CÓDIGO 2
460
BORRADO
465
EL CONTROL DE LOS PERIFÉRICOS SE CONMUTA DE NUEVO A LOS PERMISOS 1
470
SE RECUPERA EL ESTADO DEL CÓDIGO 1
475
SE REANUDA LA EJECUCIÓN DEL CÓDIGO 1
480

FIG. 4

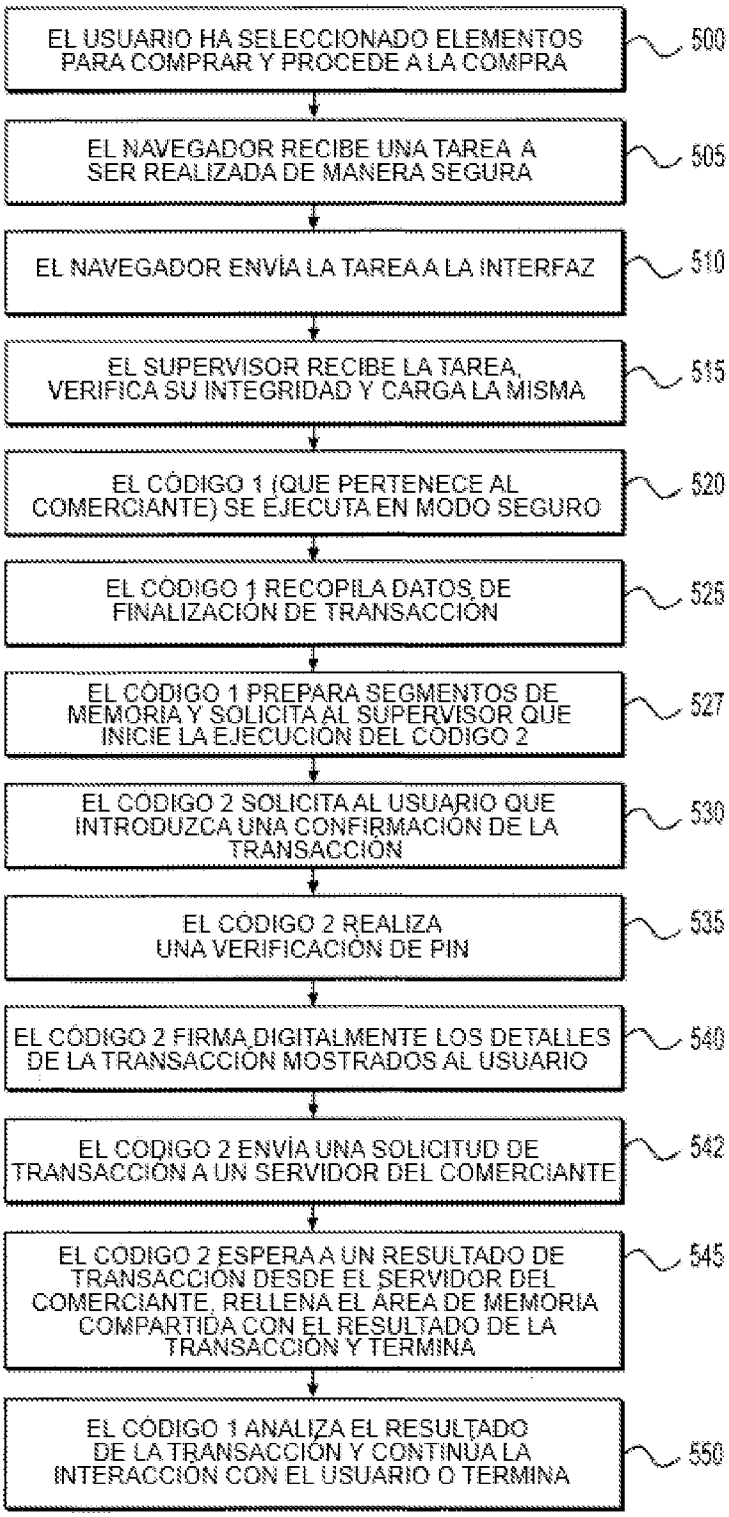
EL USUARIO HA SELECCIONADO ELEMENTOS PARA COMPRAR Y PROCEDE A LA COMPRA
500
EL NAVEGADOR RECIBE UNA TAREA A SER REALIZADA DE MANERA SEGURA
505
EL NAVEGADOR ENVÍA LA TAREA A LA INTERFAZ
510
EL SUPERVISOR RECIBE LA TAREA, VERIFICA SU INTEGRIDAD Y CARGA LA MISMA
515
EL CÓDIGO 1 (QUE PERTENECE AL COMERCIANTE) SE EJECUTA EN MODO SEGURO
520
EL CÓDIGO 1 RECOPILA DATOS DE FINALIZACIÓN DE TRANSACCIÓN
525
EL CÓDIGO 1 PREPARA SEGMENTOS DE MEMORIA Y SOLICITA AL SUPERVISOR QUE INICIE LA EJECUCIÓN DEL CÓDIGO 2
527
EL CÓDIGO 2 SOLICITA AL USUARIO QUE INTRODUZCA UNA CONFIRMACIÓN DE LA TRANSACCIÓN
530
EL CÓDIGO 2 REALIZA UNA VERIFICACIÓN DE PIN
535
EL CÓDIGO 2 FIRMA DIGITALMENTE LOS DETALLES DE LA TRANSACCIÓN MOSTRADOS AL USUARIO
540
EL CÓDIGO 2 ENVÍA UNA SOLICITUD DE TRANSACCIÓN A UN SERVIDOR DEL COMERCIANTE
542
EL CÓDIGO 2 ESPERA A UN RESULTADO DE TRANSACCIÓN DESDE EL SERVIDOR DEL COMERCIANTE, RELLENA EL ÁREA DE MEMORIA COMPARTIDA CON EL RESULTADO DE LA TRANSACCIÓN Y TERMINA
545
EL CÓDIGO 1 ANALIZA EL RESULTADO DE LA TRANSACCIÓN Y CONTINÚA LA INTERACCIÓN CON EL USUARIO O TERMINA
550

FIG. 5

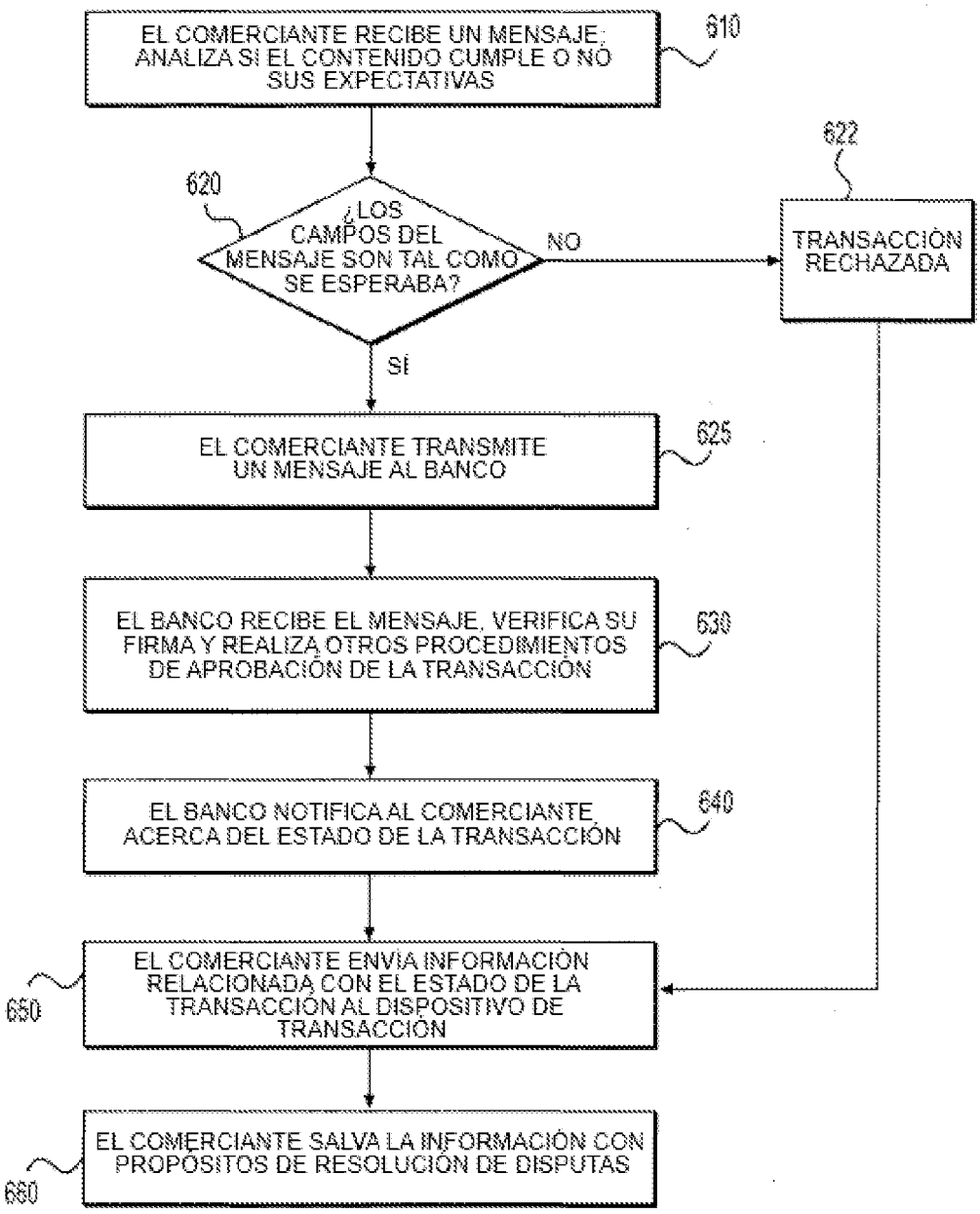
EL COMERCIANTE RECIBE UN MENSAJE, ANALIZA SI EL CONTENIDO CUMPLE O NO SUS EXPECTATIVAS
610
620
¿LOS CAMPOS DEL MENSAJE SON TAL COMO SE ESPERABA?
NO
622
TRANSACCIÓN RECHAZADA
SÍ
EL COMERCIANTE TRANSMITE UN MENSAJE AL BANCO
625
EL BANCO RECIBE EL MENSAJE, VERIFICA SU FIRMA Y REALIZA OTROS PROCEDIMIENTOS DE APROBACIÓN DE LA TRANSACCIÓN
630
EL BANCO NOTIFICA AL COMERCIANTE ACERCA DEL ESTADO DE LA TRANSACCIÓN
640
EL COMERCIANTE ENVIA INFORMACIÓN RELACIONADA CON EL ESTADO DE LA TRANSACCIÓN AL DISPOSITIVO DE TRANSACCIÓN
650
EL COMERCIANTE SALVA LA INFORMACIÓN CON PROPÓSITOS DE RESOLUCIÓN DE DISPUTAS
660

FIG. 6