(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 835 741**

(51) Int. CI.:

***H04L 9/32*** (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **17.10.2014 E 14189436 (0)**

(97) Fecha y número de publicación de la concesión europea: **07.10.2020 EP 3010176**

(54) Título: **Método y entidad receptora para la ejecución segura de software**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
**23.06.2021**

(73) Titular/es:

**QUBALT GMBH (100.0%)**
**Gartenstrasse 10**
**24534 Neumünster, DE**

(72) Inventor/es:

**HARDER, WULF;**
**UNRUH, DOMINIQUE y**
**RESAS, UWE**

(74) Agente/Representante:

**ISERN JARA, Jorge**

## DESCRIPCIÓN

Método y entidad receptora para la ejecución segura de software

Campo de la invención

La presente invención se refiere a un método para la ejecución segura de software en una entidad receptora, en donde el software se ejecuta condicionalmente en la entidad receptora. Además, la presente invención se refiere a una entidad receptora para la ejecución segura de software y a un método para la transmisión y ejecución seguras de software.

Antecedentes

La porción de software usado en automóviles, aeronave y otros medios de transporte (en lo sucesivo: vehículos) está creciendo constantemente. Por lo tanto, también está creciendo el riesgo de mal funcionamiento en vehículos debido a errores de software. En los vehículos actuales, la instalación de software puede realizarse únicamente en talleres con equipo de hardware especial y, por lo tanto, los errores de software graves pueden arreglarse únicamente por medio de caras retiradas del producto. Como resultado, existe un interés en la conexión de los vehículos a la Internet y ejecutar instalaciones de software en línea, en cualquier sitio y momento. Sin embargo, mientras que la conexión de los vehículos a la Internet ofrece muchos beneficios, también crea riesgos de ciberataques a los vehículos.

Por ejemplo, un ataque exitoso a los métodos de instalación de software en línea puede introducir un tipo especial de software malicioso en el software del vehículo, llamado "caballo de troya" o "troyano". Un troyano puede diseñarse para transmitir información sensible de privacidad como la ubicación del vehículo, velocidad del vehículo y contenido de audio/vídeo a un adversario. También pueden programarse para tomar el control del vehículo y, en el peor caso, pueden provocar accidentes cuando tienen acceso a componentes de vehículo críticos para la seguridad, como controles de dirección, gestión del motor o sistemas de frenos. Por lo tanto, el uso de troyanos para ataques a vehículos particulares, sus pasajeros y su carga es un escenario realista.

En otro escenario de ataque más, podrían introducirse troyanos sofisticados en una misión a gran escala en todos los vehículos de un fabricante específico. Con acceso a información de tiempo, o la capacidad de recibir señales externas, por ejemplo, a través de la Internet, puede ejecutarse un ciberataque coordinado a todos los vehículos en movimiento de un fabricante. Esto sería un ciberataque de una magnitud nunca antes sufrida.

Los métodos de instalación de software convencionales usan certificados digitales para garantizar que únicamente se instala software designado. Los certificados digitales se generan y verifican usando primitivas criptográficas normalizadas, como funciones de troceo y, por ejemplo, el algoritmo de encriptación RSA asimétrico, como se describe en la siguiente descripción simplificada. El emisor del software introduce el software en una función de troceo criptográfica para calcular un primer valor de troceo. A continuación, se encripta el primer valor de troceo usando el algoritmo de encriptación RSA asimétrico y la clave secreta del emisor. A continuación, el software y el primer valor de troceo encriptado se transmiten al receptor. El receptor verifica, a continuación, la autenticidad del software recibido aplicando la función de troceo criptográfica al software recibido para calcular un segundo valor de troceo. El receptor desencripta, a continuación, el valor de troceo recibido con la transmisión de software usando el algoritmo de desencriptación RSA asimétrico y la clave pública del emisor para obtener el primer valor de troceo. Si el primer y el segundo valores de troceo coinciden, se acepta el software recibido y puede instalarse.

Existen varios problemas con el uso de primitivas criptográficas normalizadas. El principal problema es que no existe ninguna prueba matemática de que las primitivas criptográficas son realmente seguras. Adicionalmente, las primitivas criptográficas no proporcionan seguridad teórica para la información. "Seguridad teórica para la información" significa que una primitiva criptográfica no puede descifrarse incluso si el adversario tiene potencia de cálculo ilimitada, porque el adversario no tiene suficiente información para descifrar la primitiva criptográfica.

Otro problema es que podrían aplicarse avances potenciales en los campos de investigación como criptoanálisis, solucionadores SAT y teoría de la complejidad para descifrar primitivas criptográficas sin pruebas de seguridad. Tales avances descifraron, por ejemplo, las funciones de troceo MD2, MD4, MD5, SHA-0, SHA-1 y el cifrado de flujo RC4. La historia de la criptografía muestra que las primitivas criptográficas normalizadas generalmente tienen una vida útil limitada.

Adicionalmente, algunas normas para primitivas criptográficas podrían verse influenciadas de tal manera que pueden descifrarse con métodos de criptoanálisis no publicados secretos para conseguir acceso a información secreta transmitida encriptada en la Internet, almacenada encriptada en servidores; para introducir troyanos en sistemas informáticos específicos para conseguir acceso a información secreta; o para influenciar o controlar el sistema informático. Introducir troyanos en sistemas informáticos es una capacidad estratégica importante que puede también usarse para la ciberguerra.

Estos métodos de criptoanálisis no publicados secretos podrían filtrarse al público o a países hostiles. Independiente

de cualquier método filtrado, grupos de piratas informáticos o investigadores en países hostiles podrían desarrollar experiencia en la exploración de las debilidades de las primitivas criptográficas normalizadas. Si esta experiencia se usara para descifrar los métodos de encriptación asimétrica desplegados en los métodos de instalación de software, las claves secretas de fabricantes de vehículos podrían caer en las manos de adversarios. También, el conocimiento acerca de cómo implementar ataques de preimagen en funciones de troceo criptográficas podría usarse para introducir troyanos en vehículos.

Diversos equipos de investigación en universidades y en empresas en todo el mundo están trabajando en el desarrollo de ordenadores cuánticos. Tan pronto como estén disponibles los ordenadores cuánticos, los métodos de encriptación asimétrica desplegados actualmente se descifrarán con los mismos. Ya se han desarrollado prototipos de ordenadores cuánticos con unos pocos bits cuánticos, por ejemplo, se ha producido un dispositivo con 14 bits cuánticos en la Universidad de Innsbruck, Austria (véase la referencia [5]). Sin embargo, para descifrar los métodos de encriptación asimétrica actuales, al menos se requerirán mil o incluso varios miles de bits cuánticos. En 2012, IBM presentó una estimación de tiempo aproximada, indicando que el primer ordenador cuántico utilizable de forma práctica estaría disponible en 10-15 años (véase la referencia [6]).

Es posible que la existencia del primer ordenador cuántico utilizable de forma práctica se mantenga en secreto. Es concebible que el público no conozca ni la existencia de los primeros ordenadores cuánticos, ni cuándo se han descifrado con los mismos los métodos de encriptación asimétrica actuales.

La criptografía poscuántica es la práctica y estudio de métodos criptográficos que se suponen que son seguros contra ataques de ordenadores cuánticos. Sin embargo, no se sabe qué métodos criptográficos pueden llamarse realmente poscuánticos, porque aún existen muchas cuestiones matemáticas abiertas. Por ejemplo, la criptografía basada en rejilla se presenta por la comunidad criptográfica como poscuántica, pero ya existen declaraciones de los principales científicos de informática cuántica que cuestionan esta opinión. Por lo tanto, los métodos criptográficos poscuánticos son más bien hoy un campo de investigación amplio y están alejados de ser seguros de forma demostrable contra ataques informáticos.

Pruebas de seguridad matemática faltantes; normas de criptografía debilitadas; avances en campos de investigación como criptoanálisis, solucionadores SAT y teoría de la complejidad; y el desarrollo de ordenadores cuánticos, son amenazas a largo plazo para los métodos de instalación de software en línea de vehículos y la ejecución segura de software.

Una solución de técnica anterior se divulga en el documento de ABIDIN AYSAJAN *ET AL.*, con el título "Direct proof of security of Wegman-Carter authentication with partially known key", QUANTUM INFORMATION PROCESSING, KLUWER ACADEMIC, DORDRECHT, NL, vol. 13, n.º 10, 20 de septiembre de 2013, páginas 2155-2170, ISSN 1570-0755.

Por consiguiente, es un aspecto de la presente invención proporcionar una ejecución de software segura mejorada en una entidad receptora.

Breve sumario de la invención

El anterior y otros objetos se consiguen por medio del conjunto adjunto de reivindicaciones.

De acuerdo con un primer aspecto, se propone un método para la ejecución segura de software en una entidad receptora, en donde el software se ejecuta condicionalmente en la entidad receptora. El método comprende las siguientes etapas en la entidad receptora: ejecutar un esquema de autenticación con una entidad emisora usando una clave de autenticación (que no se marca como inutilizable) para seleccionar una función de troceo de una familia de funciones de troceo, recibir el software y un primer valor de troceo desde la entidad emisora, en donde el primer valor de troceo se calcula introduciendo el software en la función de troceo seleccionada, almacenar la clave de autenticación y datos de uso acerca del uso de la clave de autenticación almacenada, actualizar los datos de uso con cada uso de la clave de autenticación almacenada, marcar la clave de autenticación almacenada como inutilizable o borrar la clave de autenticación almacenada cuando se alcanza un límite de uso definido de acuerdo con los datos de uso actualizados, introducir el software recibido en la función de troceo seleccionada para calcular un segundo valor de troceo, y ejecutar el software únicamente si el primer valor de troceo es igual al segundo valor de troceo. El presente método mejora la seguridad para ejecutar un software transmitido por una entidad emisora y recibido en una entidad receptora comparando un primer valor de troceo calculado por la entidad emisora y un segundo valor de troceo calculado por la entidad receptora. La entidad emisora calcula el primer valor de troceo introduciendo el software en una función de troceo que se selecciona de una familia de funciones de troceo usando la clave de autenticación del esquema de autenticación ejecutado entre la entidad emisora y la entidad receptora. De una forma análoga, la entidad receptora calcula el segundo valor de troceo introduciendo el software recibido en la misma función de troceo que se selecciona de la misma familia de funciones de troceo, dependiendo de dicha clave de autenticación común.

En particular, seleccionar una función de troceo de una familia de funciones de troceo usando la clave de autenticación y calcular un primer valor de troceo del software puede ser equivalente a introducir una clave de autenticación y el

software en cualquier orden en una función de troceo. Bits de la clave de autenticación incluso pueden mezclarse con bits del software.

Debido a que la clave de autenticación almacenada se invalida, por ejemplo, marcando la misma como inutilizable o borrando la misma, se limita la cantidad de información que es utilizable para criptoanálisis o solucionadores de SAT.

En particular, el software puede transmitirse desde la entidad emisora a la entidad receptora mediante métodos en línea o mediante métodos fuera de línea. Dichos métodos en línea pueden incluir la transmisión del software a través de una red, por ejemplo, la Internet o una red inalámbrica. Dichos métodos fuera de línea pueden incluir la provisión del software a la entidad receptora mediante una portadora de datos, como un CD, un DVD o una memoria USB.

El software transmitido desde la entidad emisora a la entidad receptora puede estar encriptado o puede no estar encriptado. Como alternativa o adicionalmente, dicho software puede estar transformado o no transformado.

En una realización adicional, el esquema de autenticación es teóricamente seguro para la información. En particular, el esquema de autenticación es teóricamente seguro para la información si su seguridad deriva únicamente a partir de la teoría de información. No puede descifrarse incluso cuando un adversario tiene potencia de cálculo ilimitada, porque el adversario no tiene suficiente información para descifrar el esquema de autenticación seguro. Por lo tanto, el esquema de autenticación también es seguro contra adversarios cuánticos.

En una realización adicional, el esquema de autenticación es el esquema de autenticación de Wegman y Carter (véanse las referencias [1] y [2]), que es ventajosamente teóricamente seguro para la información. En la referencia [3] se estudia la seguridad del esquema de autenticación de Wegman y Carter para ajustes en los que la clave de autenticación se conoce parcialmente por el adversario.

En una realización adicional, la entidad receptora es un vehículo. Por ejemplo, el vehículo es un coche, un camión, una aeronave, un vehículo espacial o una embarcación.

Ventajosamente, el presente método evita ataques a vehículos, un vehículo particular o un grupo de vehículos. Tales ataques evitados por el presente método pueden incluir enviar datos privados desde la entidad receptora a un adversario, por ejemplo, mediante un troyano incorporado en el software recibido. Tales datos privados de la entidad receptora pueden incluir datos de ubicación, contenido de audio, contenido de vídeo y datos personales almacenados en el sistema informático del vehículo.

Adicionalmente, tales ataques pueden incluir manipular los sistemas críticos para la seguridad del vehículo, como gestión de motor, sistema de frenos, control de dirección y ayuda a la conducción. Además, tales ataques pueden incluir manipular el vehículo para crear y transmitir datos, como datos de ubicación y velocidad de vehículo, y/o manipular el funcionamiento del vehículo para dañar la reputación del fabricante del vehículo, por ejemplo, aumentando el consumo de combustible, provocando fallos de motor puntuales o desencadenando problemas eléctricos. Adicionalmente, tales ataques pueden incluir tomar el control del vehículo para ejecutar un ataque dirigido a pasajeros y cargas, inhabilitar el vehículo o controlar remotamente el vehículo.

En una realización adicional, la entidad receptora es un dispositivo. Por ejemplo, el dispositivo es un dispositivo móvil o un dispositivo crítico para la seguridad, en particular, un dispositivo médico. Ventajosamente, el presente método evita ataques a dispositivos, un dispositivo particular o un grupo de dispositivos. Tales ataques evitados por el presente método pueden incluir enviar datos privados desde la entidad receptora a un adversario, por ejemplo, mediante un troyano incorporado en el software recibido. Tales datos privados desde la entidad receptora pueden incluir datos de ubicación, contenido de audio, contenido de vídeo y datos personales almacenados en el sistema informático del dispositivo.

Además, tales ataques pueden incluir manipular el dispositivo para crear y transmitir datos falsos. Adicionalmente, tales ataques pueden incluir tomar el control del dispositivo para ejecutar un ataque dirigido al sistema del que el dispositivo es parte, o para inhabilitar el dispositivo.

En una realización adicional, para cada una de una pluralidad de entidades receptoras, la entidad emisora transmite el software y un primer valor de troceo individual distinto a cada entidad receptora. Ventajosamente, en el proceso de transmisión de una pieza particular de software a una pluralidad de entidades receptoras, un adversario no puede usar un primer valor de troceo interceptado enviado a una de las entidades receptoras para atacar a las otras entidades receptoras.

En particular, la entidad emisora comprende al menos dos unidades. Una primera unidad puede calcular el primer valor de troceo del software y una segunda unidad puede transferir el software y el primer valor de troceo calculado a la entidad receptora. Como alternativa, una primera unidad calcula el primer valor de troceo del software y transfiere el primer valor de troceo calculado a la entidad receptora, y una segunda unidad transfiere el software a la entidad receptora.

En una realización adicional, un conjunto de N claves de autenticación se almacena en la entidad receptora en un proceso de producción de la entidad receptora, siendo N ≥ 1. Ventajosamente, la distribución de las claves de autenticación se hace de una forma segura. Por ejemplo, puede usarse la distribución de clave cuántica para transmitir las claves de autenticación.

En particular, el proceso de producción incluye todas las etapas empleadas por un fabricante para producir y configurar una entidad receptora, a través de su entrega al usuario. Ventajosamente, las claves de autenticación pueden almacenarse de forma segura en la entidad receptora durante el proceso de producción, mientras permanece completamente bajo el control del fabricante.

En una realización adicional, el conjunto de N claves de autenticación se almacena en un hardware seguro, siendo N ≥ 1, en donde el hardware seguro se instala en la entidad receptora. El hardware seguro puede ser un chip seguro o un módulo de seguridad de hardware. En particular, el hardware seguro es un dispositivo de almacenamiento resistente a manipulaciones que está configurado para la protección contra manipulación de la clave de autenticación. Esto evita que los adversarios usen claves de autenticación manipuladas para ejecutar software malicioso ventajosamente. Adicionalmente, las claves de autenticación se mantienen en secreto. Como alternativa, o adicionalmente, las claves de autenticación pueden distribuirse de una forma segura. Por ejemplo, las claves de autenticación pueden almacenarse en un chip seguro y resistente a manipulaciones, en una tarjeta inteligente, en una tarjeta SIM resistente a manipulaciones o en un módulo de seguridad de hardware. En particular, las tarjetas SIM y los lectores de tarjetas SIM pueden usarse para instalar claves de autenticación en la entidad receptora de una forma segura en cualquier momento.

En una realización adicional, la clave de autenticación se protege usando criptografía de caja blanca.

La criptografía de caja blanca es un enfoque para diseñar e implementar algoritmos criptográficos de tal manera que su código de programa, datos internos y claves están protegidos contra ataques como depuración, ingeniería inversa y recuperación de claves (véase la referencia [4]). En particular, la clave de autenticación se almacena en un almacenamiento resistente a manipulaciones que está configurado para proteger contra manipulación de la clave de autenticación, para evitar que un adversario use claves de autenticación manipuladas para ejecutar software malicioso ventajosamente. Adicionalmente, las claves de autenticación se mantienen en secreto.

En una realización adicional, la clave de autenticación almacenada en la entidad receptora se modifica por medio de datos de modificación que se comparten entre la entidad emisora y la entidad receptora. En particular, un adversario puede atacar la entidad receptora únicamente si tiene acceso a las claves de autenticación y puede supervisar toda la comunicación entre la entidad emisora y la entidad receptora para interceptar dichos datos de modificación. Por lo tanto, la seguridad se mejora usando dichos datos de modificación.

En una realización adicional, los datos de modificación contienen una clave de modificación, en donde la clave de autenticación almacenada se procesa mediante una función XOR con la clave de modificación, formando una nueva clave de autenticación. Ventajosamente, por medio de dicha clave de modificación, se proporciona una operación rápida para modificar las claves de autenticación almacenadas.

En una realización adicional, los datos de modificación son más pequeños que la clave de autenticación, o más pequeños que N claves de autenticación. En particular, los datos de modificación se introducen en un algoritmo para generar datos de modificación ampliados. El algoritmo puede ser un generador de números pseudoaleatorios que toma los datos de modificación como semilla para inicializar el generador de números pseudoaleatorios. El resultado del generador de números pseudoaleatorios son los datos de modificación ampliados.

Cualquier dato que se ha transmitido entre la entidad emisora y la entidad receptora puede usarse como datos de modificación.

En una realización adicional, los datos de uso se incorporan como una bandera, como un contador o como una lista blanca.

Cualquier realización del primer aspecto puede combinarse con cualquier realización del primer aspecto para obtener otra realización del primer aspecto.

De acuerdo con un segundo aspecto, la invención se refiere a un programa informático que comprende un código de programa para ejecutar el método del primer aspecto o de una realización del primer aspecto para la ejecución segura de software en una entidad receptora cuando se ejecuta en al menos un ordenador.

De acuerdo con un tercer aspecto, se propone una entidad receptora para la ejecución segura de software, en donde el software se ejecuta condicionalmente en la entidad receptora. La entidad receptora incluye una unidad de autenticación, una unidad de recepción, una unidad de almacenamiento y una unidad de procesamiento. La unidad de autenticación está configurada para ejecutar un esquema de autenticación con una entidad emisora usando una clave de autenticación para seleccionar una función de troceo de una familia de funciones de troceo. La unidad de

recepción está configurada para recibir el software y un primer valor de troceo desde la entidad emisora, en donde el primer valor de troceo se calcula introduciendo el software en la función de troceo seleccionada. La unidad de almacenamiento está configurada para almacenar la clave de autenticación y datos de uso acerca del uso de la clave de autenticación almacenada. La unidad de procesamiento está configurada para actualizar los datos de uso con cada uso de la clave de autenticación almacenada, para marcar la clave de autenticación almacenada como inutilizable o para borrar la clave de autenticación almacenada cuando se alcanza un límite de uso definido de acuerdo con los datos de uso actualizados, para introducir el software recibido en la función de troceo seleccionada para calcular un segundo valor de troceo, y para ejecutar el software únicamente si el primer valor de troceo es igual al segundo valor de troceo.

Cualquier realización del primer aspecto puede combinarse con el anterior tercer primer aspecto para obtener una realización adicional del tercer aspecto.

De acuerdo con un cuarto aspecto, se propone un sistema (disposición) para una transmisión y ejecución seguras de software. El sistema incluye una entidad emisora y una entidad receptora. El software se transfiere desde la entidad emisora a la entidad receptora, en donde el software se ejecuta condicionalmente en la entidad receptora. La entidad emisora y la entidad receptora están configuradas para ejecutar un esquema de autenticación usando una clave de autenticación para seleccionar una función de troceo de una familia de funciones de troceo, respectivamente. La entidad emisora transmite el software junto con un primer valor de troceo a la entidad receptora. La entidad emisora calcula el primer valor de troceo introduciendo el software en la función de troceo seleccionada. La entidad receptora almacena la clave de autenticación y datos de uso acerca del uso de la clave de autenticación almacenada. Adicionalmente, la entidad receptora actualiza los datos de uso con cada uso de la clave de autenticación almacenada. Además, la entidad receptora está configurada para marcar la clave de autenticación almacenada como inutilizable o para borrar la clave de autenticación almacenada cuando se alcanza un límite de uso definido de acuerdo con los datos de uso actualizados. La entidad receptora está configurada para introducir el software recibido en la función de troceo seleccionada para calcular un segundo valor de troceo. El software se ejecuta en la entidad receptora únicamente si y únicamente si el primer valor de troceo es igual al segundo valor de troceo.

Cualquier realización del primer aspecto puede combinarse con el anterior cuarto aspecto para obtener una realización adicional del cuarto aspecto.

El siguiente ejemplo de "Instalación de software" puede ilustrar la funcionalidad de los aspectos mencionados anteriormente.

Ejemplo: Instalación de software

Este ejemplo de "Instalación de software" incluye "Inicialización de un módulo de instalación de software" e "Instalación en línea de software".

Inicialización de un módulo de instalación de software:

En este ejemplo, un vehículo está equipado con la entidad receptora anteriormente mencionada, que se incorpora como un módulo de instalación de software que asegura que en el vehículo únicamente se instala y ejecuta software válido. El módulo de instalación de software comprende una unidad de recepción que proporciona una interfaz de red para conexiones por cable e inalámbricas, una unidad de autenticación para la ejecución del esquema de autenticación usando la función de troceo, una unidad de almacenamiento para almacenar claves de autenticación en una memoria resistente a manipulaciones y una unidad de ejecución para operar con datos de uso.

En el fabricante de vehículos, durante el proceso de producción del vehículo, se inicializa el módulo de instalación de software del vehículo usando un sistema de inicialización designado. El sistema de inicialización comprende un módulo de seguridad de hardware y una interfaz de red de cable.

El proceso de inicialización puede consistir en varias etapas. Primero, el sistema de inicialización del fabricante se conecta a través de una red de cable al módulo de instalación de software del vehículo. Después de que se haya establecido un canal de comunicación seguro, comienza la inicialización real. En el módulo de seguridad de hardware del sistema de inicialización se generan un conjunto de claves de autenticación y sus correspondientes identificadores y se almacenan en una base de datos. El conjunto de claves de autenticación es único y se asigna a únicamente este vehículo particular. En una siguiente etapa, las claves de autenticación e identificadores se transmiten a través de la red de cable desde el sistema de inicialización al módulo de instalación de software del vehículo. La unidad receptora recibe las claves de autenticación e identificadores, y la unidad de almacenamiento almacena los mismos en la memoria resistente a manipulaciones. A continuación, para cada clave de autenticación, la unidad de procesamiento genera datos de uso para esa clave y marca las claves de autenticación almacenadas como no usadas en los datos de uso.

En este punto, un conjunto idéntico de claves de autenticación e identificadores se almacenan en el sistema de inicialización del fabricante y en el módulo de instalación de software del vehículo. Después de que se haya inicializado

el módulo de instalación de software del vehículo, está preparado para recibir e instalar software.

Instalación en línea de software:

Después de que se hayan enviado los vehículos a sus usuarios, el fabricante del vehículo tiene la capacidad de realizar mantenimiento de software en línea durante toda la vida útil del vehículo. El vehículo solo necesita estar en línea y el fabricante del vehículo puede instalar remotamente el software.

El fabricante del vehículo despliega un sistema de distribución de software en línea para hacer la instalación de software en línea. Este sistema comprende un módulo de seguridad de hardware que contiene una unidad de autenticación para la ejecución del esquema de autenticación usando la función de troceo, una unidad de almacenamiento que contiene y utiliza una base de datos con el conjunto completo de claves de autenticación y datos de uso de todos los vehículos equipados con el módulo de instalación de software, y una unidad de procesamiento que opera en los datos de uso.

El sistema de distribución de software en línea comprende además una unidad emisora con capacidad de transmisión de un paquete de instalación a través de una conexión de Internet inalámbrica al módulo de instalación de software del vehículo. Adicionalmente, el sistema de distribución de software en línea almacena el software que debería instalarse en los vehículos.

El proceso de instalación de software en línea para un vehículo específico consiste en varias etapas. En un principio, el sistema de distribución de software en línea se conecta a través de una conexión de Internet inalámbrica al módulo de instalación de software del vehículo y solicita instalar el software. Si el módulo de instalación de software acepta la petición, puede comenzar la instalación y el sistema de distribución de software en línea prepara un paquete de instalación especialmente para el vehículo.

La creación del paquete de instalación puede hacerse como se indica a continuación. El software a instalar se pasa al módulo de seguridad de hardware. El módulo de seguridad de hardware solicita a la unidad de almacenamiento para este vehículo específico una clave de autenticación no usada y su correspondiente identificador. En una etapa posterior, el software y la clave de autenticación se pasan a la unidad de autenticación, en la que el software y la clave de autenticación se introducen en el esquema de autenticación. El esquema de autenticación selecciona una función de troceo de una familia de funciones de troceo usando la clave de autenticación y calcula un primer valor de troceo del software. La unidad de procesamiento solicita a la unidad de almacenamiento los datos de uso de la clave de autenticación y marca la clave de autenticación como inutilizable en los datos de uso. El primer valor de troceo y el identificador se pasan al sistema de distribución de software en línea. Finalmente, el software, el identificador y el primer valor de troceo se almacenan en el paquete de instalación, que se transmite a través de la unidad emisora inalámbricamente al módulo de instalación de software del vehículo.

La unidad receptora del módulo de instalación de software del vehículo recibe el paquete de instalación y desempaqueta el software, el identificador y el primer valor de troceo del mismo. A continuación, el software y el identificador se pasan a la unidad de autenticación. La unidad de autenticación solicita a la unidad de almacenamiento la clave de autenticación que corresponde al identificador, así como los datos de uso de la clave de autenticación. Si los datos de uso indican que la clave de autenticación no se ha usado, la unidad de autenticación procede e introduce el software y la clave de autenticación en el esquema de autenticación para calcular un segundo valor de troceo del software. En una siguiente etapa, la unidad de procesamiento solicita a la unidad de almacenamiento los datos de uso de la clave de autenticación y marca la misma como inutilizable en los datos de uso.

Si el primer valor de troceo recibido en el paquete de instalación coincide con el segundo valor de troceo calculado en el módulo de autenticación de la entidad receptora, el software es válido y el proceso de instalación procede a su conclusión. Sin embargo, si los valores de troceo no coinciden, el software no es válido y se rechaza.

Si los valores de troceo no coinciden, son posibles una pluralidad de diferentes reacciones, por ejemplo, incluyendo proporcionar un mensaje al usuario del vehículo, retardar una posible actualización del software en el vehículo durante un tiempo determinado y/o proporcionar una alerta al fabricante del vehículo.

En particular, el número de intentos no exitosos de actualización del software, es decir, si los valores de troceo no coinciden, puede limitarse a un cierto umbral. El efecto de limitar estos intentos no exitosos es evitar un ataque de fuerza bruta, por ejemplo.

A continuación, se describen realizaciones ilustrativas de la presente invención con referencia a las figuras adjuntas.

Breve descripción de los dibujos

La Figura 1 muestra una primera realización de una secuencia de etapas de método para la ejecución segura de software en una entidad receptora;

La Figura 2 muestra una segunda realización de una secuencia de etapas de método para la ejecución segura de software en una entidad receptora;

La Figura 3 muestra una realización de una entidad receptora para la ejecución segura de software;

La Figura 4 muestra una realización de una secuencia de etapas de método para la transmisión y ejecución seguras de software;

La Figura 5 muestra una red de Petri de una realización de la entidad emisora;

La Figura 6 muestra una red de Petri de una realización de la entidad receptora;

La Figura 7 muestra una red de Petri que ilustra una realización de la modificación de las claves de autenticación almacenadas en la entidad receptora mediante los datos de modificación; y

La Figura 8 muestra una red de Petri que ilustra una realización adicional de la modificación de las claves de autenticación almacenadas en la entidad receptora mediante los datos de modificación.

Si no se indican de otra manera en las figuras se han asignado los mismos signos de referencia a elementos similares o de funcionalidad similar.

Descripción detallada de las realizaciones

En la Figura 1, se representa una realización de una secuencia de etapas de método para la ejecución segura de software 1 en una entidad receptora 3. Por ejemplo, la entidad receptora 3 es un coche, el software 1 es el sistema operativo de un controlador del coche y la entidad emisora 2 es un servidor operado por un fabricante de coches.

En este contexto, la Figura 2 muestra un sistema con una entidad emisora 2 y la entidad receptora 3, en donde el software 1 se transfiere desde la entidad emisora 2 a la entidad receptora 3 y el software 1 transferido se ejecuta condicionalmente en la entidad receptora 3.

El método de la Figura 1 incluye las siguientes etapas 301-307 que se ejecutan en la entidad receptora 3: en la etapa 301 representada por las subetapas 301a y 301b, se ejecuta un esquema de autenticación 4 (subetapa 301a) con la entidad emisora 2 usando una clave de autenticación 5 para seleccionar una función de troceo 6 de una familia de funciones de troceo 13 (subetapa 301b). En particular, tanto la entidad emisora 2 como la entidad receptora 3 usan la clave de autenticación común 5 para seleccionar la misma función de troceo 6 de la familia de funciones de troceo 13. La familia de funciones de troceo 13 puede almacenarse en la entidad receptora 3 o puede proporcionarse a la entidad receptora 3 como alternativa. Por ejemplo, el esquema de autenticación 4 puede ser teóricamente seguro para la información. Como un ejemplo, en este punto puede utilizarse un esquema de autenticación de Wegman y Carter (véanse las referencias [1] y [2]).

En la etapa 302, el software 1 y un primer valor de troceo 7 transferido desde la entidad emisora 2 se reciben en la entidad receptora 3. El primer valor de troceo 7 se calcula por la entidad emisora 2 introduciendo el software 1 en la función de troceo 6 seleccionada. Por ejemplo, la entidad emisora 2 puede comunicarse con una pluralidad de entidades receptoras 3. En un caso de este tipo, la entidad emisora 2 puede transmitir el software 1 y un respectivo primer valor de troceo 7 individual a una respectiva entidad receptora 3. En otras palabras, una respectiva entidad receptora 3 recibe el software 1 y un primer valor de troceo 7 que se calcula individualmente para la respectiva entidad receptora 3.

En la etapa 303, la clave de autenticación 5 y los datos de uso 8 se almacenan en la entidad receptora 3. Los datos de uso 8 incluyen información acerca del uso de la clave de autenticación 5 almacenada. Por ejemplo, los datos de uso 8 se incorporan como una bandera que indica el uso de la clave de autenticación 5. Por ejemplo, la bandera puede ser 0, si la clave de autenticación 5 no se ha usado aún, y puede ser 1, si la clave de autenticación 5 se ha usado. Como una alternativa, los datos de uso 8 pueden incorporarse como un contador o como una lista blanca. Si los datos de uso 8 son un contador, se permite un número predefinido de usos (por ejemplo, tres) de la clave de autenticación 5.

Si los datos de uso 8 se incorporan como una lista blanca, dicha lista blanca puede indicar que se permiten las claves de autenticación 5. En otras palabras, una clave de autenticación recibida 5 que no está en la lista blanca indica que la clave de autenticación recibida 5 es inutilizable o tiene que borrarse.

Además, la entidad receptora 3 puede almacenar un conjunto de N claves de autenticación 5, siendo $N \geq 1$. En particular, las N claves de autenticación 5 se almacenan en la entidad receptora 3 en un proceso de producción de la entidad receptora 3. Por ejemplo, las N claves de autenticación 5 pueden almacenarse en un hardware seguro en la entidad receptora 3. Además, la clave de autenticación 5 puede protegerse, en particular usando criptografía de caja blanca. En una realización adicional, la clave de autenticación 5 almacenada en la entidad receptora 3 puede

modificarse por medio de datos de modificación 10. Dichos datos de modificación 10 pueden recibirse en la entidad receptora 3, por ejemplo, a través de una red inalámbrica. Los datos de modificación 10 recibidos pueden contener una clave de modificación. La clave de autenticación 5 almacenada puede procesarse mediante una función XOR con la clave de modificación recibida para formar una nueva clave de autenticación 5. Usando claves de autenticación 5 nuevas o actualizadas, puede mejorarse la seguridad.

En la etapa 304, los datos de uso 8 se actualizan en la entidad receptora 3 con cada uso de la clave de autenticación 5 almacenada. Si, para el ejemplo, los datos de uso 8 son un contador, el contador se incrementa con cada uso de la clave de autenticación 5.

En la etapa 305, la clave de autenticación 5 almacenada se marca como inutilizable, cuando se alcanza un límite de uso definido de acuerdo con los datos de uso 8 actualizados. Como una alternativa en la etapa 305, la clave de autenticación 5 almacenada se borra cuando se alcanza el límite de uso definido de acuerdo con los datos de uso 8 actualizados.

En particular, actualizar 304 los datos de uso 8 con cada uso de la clave de autenticación 5 almacenada puede incluir marcar 305 la clave de autenticación 5 almacenada como inutilizable. Si, por ejemplo, los datos de uso 8 son una bandera, entonces el valor de bandera 0 significa que la clave de autenticación 5 no se ha usado, y puede usarse la clave de autenticación 5. El valor de bandera 1 significa que la clave de autenticación 5 se ha usado y la clave de autenticación 5 ya no puede usarse más, por tanto, es inutilizable. Adicionalmente, los datos de marcado pueden ser parte de los datos de uso 8. Decidir si una clave de autenticación 5 se marca como inutilizable puede hacerse introduciendo los datos de uso en una función. El resultado del cálculo de función determinará si la clave de autenticación 5 se marca como inutilizable. La función puede calcularse antes del uso de una clave de autenticación 5 para seleccionar una función de troceo 6 de una familia de funciones de troceo 13. Si la clave de autenticación 5 es inutilizable, el software 1 puede rechazarse.

En la etapa 306, el software 1 recibido se introduce en la función de troceo 6 seleccionada en la entidad receptora 3 para calcular un segundo valor de troceo 9.

En la etapa 307, se comparan el primer valor de troceo 7 y el segundo valor de troceo 9. El software 1 se ejecuta en la entidad receptora 3 únicamente si el primer valor de troceo 7 y el segundo valor de troceo 9 son iguales.

En particular, las etapas del método de la Figura 1 no se ejecutan consecutivamente. Por ejemplo, la etapa 303 puede ser la primera etapa.

La Figura 2 muestra una segunda realización de una secuencia de etapas de método para la ejecución segura de software en la entidad receptora 3. En la Figura 2, las etapas de método 201a, 201b, 202a, 202b se ejecutan por la entidad emisora 2, en donde las etapas 301a, 301b y 302-307 se ejecutan por la entidad receptora 3.

La entidad emisora 2 (etapa 201a) y la entidad receptora 3 (etapa 301a) ejecutan un esquema de autenticación 4 usando una clave de autenticación 5.

Adicionalmente, la entidad emisora 2 (etapa 201b) y la entidad receptora 3 (etapa 301b) seleccionan una función de troceo 6 de una familia de funciones de troceo 13 usando dicha clave de autenticación 5, respectivamente.

En la etapa 202a, la entidad emisora 2 introduce el software 1 en la función de troceo seleccionada para calcular un primer valor de troceo 7.

En la etapa 202b, la entidad emisora 2 transmite el software 1 junto con el primer valor de troceo 7 calculado a la entidad receptora 3.

En la etapa 302, la entidad receptora 3 recibe el software 1 y el primer valor de troceo 7 transmitido por la entidad emisora 2.

En la etapa 303, la entidad receptora 3 almacena la clave de autenticación 5 y los datos de uso 8 acerca del uso de la clave de autenticación 5 almacenada.

En la etapa 304, la entidad receptora 3 actualiza los datos de uso 8. Esta actualización se hace con cada uso de la clave de autenticación 5 almacenada.

En la etapa 305, la entidad receptora 3 marca la clave de autenticación 5 almacenada como inutilizable o borra la clave de autenticación 5 almacenada cuando se alcanza un límite de uso definido de acuerdo con los datos de uso 8 actualizados.

En la etapa 306, la entidad receptora 3 introduce el software 1 recibido en la función de troceo 6 seleccionada para calcular un segundo valor de troceo 9.

En la etapa 307, la entidad receptora 3 compara el primer valor de troceo 7 con el segundo valor de troceo 9. La entidad receptora 3 ejecuta el software 1 recibido únicamente si el primer valor de troceo 7 y el segundo valor de troceo 9 son iguales entre sí. Si el primer valor de troceo 7 y el segundo valor de troceo 9 no son iguales entre sí, la entidad receptora 3 no ejecuta el software 1, sino que lo rechaza.

La Figura 3 muestra una realización de una entidad receptora 3 para una ejecución segura del software 1. Se recibe el software 1 desde una entidad emisora 2 (véase la Figura 2) y se ejecuta condicionalmente en la entidad receptora 3. La entidad receptora 3 comprende una unidad de autenticación 31, una unidad de recepción 32, una unidad de almacenamiento 33 y una unidad de procesamiento 34.

La unidad de autenticación 31 está configurada para ejecutar un esquema de autenticación 4 con la entidad emisora 2 usando una clave de autenticación 5 para seleccionar una función de troceo 6 de una familia de funciones de troceo 13.

La unidad de recepción 32 está configurada para recibir el software 1 y el primer valor de troceo 7 desde la entidad emisora 2. En la entidad emisora 2, el primer valor de troceo 7 se calcula introduciendo el software 1 en la función de troceo 6 seleccionada.

La unidad de almacenamiento 33 está configurada para almacenar la clave de autenticación 5 y los datos de uso 8 acerca del uso de la clave de autenticación 5 almacenada.

Además, la unidad de procesamiento 34 está configurada para actualizar los datos de uso 8 con cada uso de la clave de autenticación 5 almacenada. Además, la unidad de procesamiento 34 marca la clave de autenticación 5 almacenada como inutilizable cuando se alcanza un límite de uso definido de acuerdo con los datos de uso 8 actualizados. Como alternativa, o adicionalmente, la unidad de procesamiento 34 puede borrar la clave de autenticación 5 almacenada cuando se alcanza el límite de uso definido de acuerdo con los datos de uso 8 actualizados. Adicionalmente, la unidad de procesamiento 34 introduce el software 1 recibido en la función de troceo 6 seleccionada para calcular un segundo valor de troceo 9. El software 1 se ejecuta en la entidad receptora 3 únicamente si el primer valor de troceo 7 es igual al segundo valor de troceo 9.

La Figura 4 muestra una realización de una secuencia de etapas de método para la transmisión y ejecución seguras de software 1, en donde el software 1 se ejecuta condicionalmente en una entidad receptora 3.

El método de la Figura 4 incluye las siguientes etapas de método 401-407:

En la etapa 401, se ejecuta un esquema de autenticación 4 entre la entidad emisora 2 y la entidad receptora 3 usando una clave de autenticación 5 para seleccionar una función de troceo 6 de una familia de funciones de troceo 13. En particular, tanto la entidad emisora 2 como la entidad receptora 3 usan la clave de autenticación común 5 para seleccionar la misma función de troceo 6 de la familia de funciones de troceo 13.

En la etapa 402, el software 1 y un primer valor de troceo 7 se transmiten desde la entidad emisora 2 a la entidad receptora 3, en donde la entidad emisora 2 calcula el primer valor de troceo 7 introduciendo el software 1 en la función de troceo 6 seleccionada.

En la etapa 403, la clave de autenticación 5 y los datos de uso 8 acerca del uso de la clave de autenticación 5 almacenada se almacenan en la entidad receptora 3.

En la etapa 404, los datos de uso 8 se actualizan en la entidad receptora 3 con cada uso de la clave de autenticación 5 almacenada.

En la etapa 405, la clave de autenticación 5 se marca como inutilizable o se borra por la entidad receptora 3, cuando se alcanza un límite de uso definido de acuerdo con los datos de uso 8 actualizados.

En la etapa 406, el software 1 recibido se introduce en la función de troceo 6 seleccionada por la entidad receptora 3 para calcular un segundo valor de troceo 9.

En la etapa 407, el software 1 se ejecuta por la entidad receptora 3 únicamente si el primer valor de troceo 7 y el segundo valor de troceo 9 son iguales.

Además, las Figuras 5 a 8 muestran diferentes redes de Petri. Cada una de las redes de Petri de las Figuras 5 a 8 usan testigos k, posiciones p y transiciones t.

La Figura 5 muestra una red de Petri de una realización de la entidad emisora 2. La Figura 6 muestra una red de Petri de una realización de la entidad receptora 3. Aunando las Figuras 5 y 6, la entidad emisora 2 de la Figura 5 y la entidad receptora 3 de la Figura 6 se vinculan mediante dos transiciones de transmisión $t_3$ y $t_4$. Por medio de la transición de

transmisión $t_3$, el primer valor de troceo 7 calculado se transfiere desde la entidad emisora 2 de la Figura 5 a la entidad receptora 3 de la Figura 6. Por medio de la transición de transmisión $t_4$, el software 1 se transfiere desde la entidad emisora 2 de la Figura 5 a la entidad receptora 3 de la Figura 6.

A continuación, se analizan los detalles de las Figuras 5 y 6:
La posición $p_1$ de la Figura 5 contiene un testigo $k_3$ que incluye cuatro claves de autenticación 5 ($clave_1$ de autenticación, $clave_2$ de autenticación, $clave_3$ de autenticación y $clave_4$ de autenticación). Además, el testigo $k_3$ incluye datos de uso 8 que indican el uso de la respectiva clave de autenticación 5. En el ejemplo de la Figura 5, la respectiva x en los datos de uso 8 indica que se han usado las correspondientes claves de autenticación 5. En detalle, de acuerdo con la Figura 5, se han usado la $clave_1$ de autenticación y la $clave_2$ de autenticación, mientras que la $clave_3$ de autenticación y la $clave_4$ de autenticación no se han usado.

La posición $p_2$ contiene un testigo $k_2$ para una familia de funciones de troceo 13. En la transición $t_1$, se selecciona una cierta función de troceo 6 de la familia de funciones de troceo 13 usando una cierta clave de las claves de autenticación 5. La posición $p_3$ entonces contiene la función de troceo 6 seleccionada.

La posición $p_4$ contiene un testigo $k_1$ para el software 1. En la transición $t_2$, el software 1 se introduce en la función de troceo 6 para calcular el primer valor de troceo 7 (véase la posición $p_5$).

Como ya se ha indicado anteriormente, en la transición $t_3$, el primer valor de troceo 7 calculado se transfiere a la entidad receptora 3 de la Figura 6 y, en la transición $t_4$, el software 1 se transfiere a la entidad receptora 3.

Con referencia a la Figura 6, la posición $p_6$ tiene el primer valor de troceo 7 como recibido desde la entidad emisora 2. Además, la posición $p_{10}$ ha recibido el software 1 desde la entidad emisora 2.

La posición $p_7$ tiene un testigo $k_4$ que corresponde al testigo $k_3$ de la Figura 5. Además, la transición $t_5$ corresponde a la transición $t_1$ de la Figura 5, la posición $p_8$ corresponde a la posición $p_2$ de la Figura 5, y el testigo $k_5$ corresponde al testigo $k_2$ de la Figura 5.

La posición $p_9$ contiene la función de troceo 6 según se proporciona por la transición $t_5$ para seleccionar la función de troceo 6 la entrada de los testigos $k_4$ y $k_5$.

En transición $t_6$, la función de troceo 6 se calcula con la entrada del software 1 (véase $p_{10}$) y la función de troceo 6 (véase $p_9$). A continuación, la posición $p_{11}$ incluye el segundo valor de troceo calculado 9 como resultado de la transición 6. En transición $t_7$, se comparan los contenidos de $p_6$ y $p_{11}$, en concreto el primer valor de troceo 7 y el segundo valor de troceo 9. Si el primer valor de troceo 7 es igual al segundo valor de troceo 9, entonces la transición $t_7$ pondrá un testigo en la posición $p_{12}$. El software 1 de la posición $p_{10}$ puede ejecutarse con la transición $t_8$ únicamente si un testigo está en la posición $p_{12}$.

Adicionalmente, la Figura 7 muestra una red de Petri que ilustra que las claves de autenticación 5 almacenadas en la entidad receptora 3 (véase la posición $p_7$ que tiene el testigo $k_4$) pueden modificarse por los datos de modificación 10. Lo mismo se aplica para la posición $p_1$ que tiene el testigo $k_3$ para la entidad emisora 2 de la Figura 5.

En la Figura 7, la posición $p_{13}$ tiene un testigo $k_6$ con dichos datos de modificación 10. En la transición $t_9$, las claves de autenticación 5 del testigo $k_4$ pueden modificarse usando los datos de modificación 10 del testigo $k_6$. Por ejemplo, dichos datos de modificación 10 pueden contener una clave de modificación. En la transición $t_9$, la clave de autenticación 5 puede procesarse mediante una función XOR con la clave de modificación para formar una nueva clave de autenticación 5. La nueva clave de autenticación 5 puede incorporarse en el testigo $k_4$.

La Figura 8 muestra una red de Petri que ilustra una realización adicional para modificar las claves de autenticación 5.

En comparación con la red de Petri de la Figura 7, la red de Petri de la Figura 8 se amplía mediante una transición $t_{10}$ y una posición $p_{14}$. En la transición $t_{10}$, se amplían los datos de modificación 10 del testigo $k_6$. Por ejemplo, si los datos de modificación 10 son una semilla para una función pseudoaleatoria, la ampliación de los datos de modificación 10 se hace introduciendo dicha semilla en la función pseudoaleatoria y usando el resultado de la función pseudoaleatoria como datos de modificación ampliados 14. La posición $p_{14}$ contiene los datos de modificación ampliados 14. Dichos datos de modificación ampliados 14 se usan, a continuación, en la transición $t_{10}$ para modificar las claves de autenticación 5.

Mientras la presente invención se ha descrito con referencia a ciertas realizaciones, se entenderá por los expertos en la materia que pueden hacerse diversos cambios y pueden sustituirse equivalentes sin alejarse del alcance de la presente invención. Además, pueden hacerse muchas modificaciones para adaptar una situación particular a los contenidos de la presente invención sin alejarse de su alcance. Por lo tanto, se concibe que la presente invención no se limita a las realizaciones particulares divulgadas, sino que la presente invención incluirá que todas las realizaciones pertenecen al alcance de las reivindicaciones adjuntas.

NÚMEROS DE REFERENCIA

| | |
|---|---|
| 1 | software |
| 2 | entidad emisora |
| 3 | entidad receptora |
| 4 | esquema de autenticación |
| 5 | clave de autenticación |
| 6 | función de troceo |
| 7 | primer valor de troceo |
| 8 | datos de uso |
| 9 | segundo valor de troceo |
| 10 | datos de modificación |
| 13 | familia de funciones de troceo |
| 14 | datos de modificación ampliados |
| 31 | unidad de autenticación |
| 32 | unidad de recepción |
| 33 | unidad de almacenamiento |
| 34 | unidad de procesamiento |
| 201 - 202 | etapa de método |
| 301 - 307 | etapa de método |
| 401 - 407 | etapa de método |
| k; k1 - k6 | testigo |
| p; p1 - p14 | posición |
| t; t1 - t10 | transición |

REFERENCIAS

[1] J. Lawrence Carter y Mark N. Wegman, Universal Classes of Hash Functions, Journal of Computer and System Sciences 18, N.º 2, p. 143-154, abril de 1979

[2] Mark N. Wegman y J. Lawrence Carter, New Hash Functions and Their Use in Authentication and Set Equality, Journal of Computer and System Sciences 22, p. 265-279, 1981

[3] Aysajan Abidin y Jan-Åke Larsson, Direct Proof of Security of Wegman-Carter Authentication with Partially Known Key, Quantum Information Processing 13, p. 2155-2170, 2013

[4] Brecht Wyseur, White-Box Cryptography, Tesis Doctoral, Katholieke Universiteit Leuven, marzo de 2009

[5] Thomas Monz *et al.*, 14-Qubit Entanglement: Creation and Coherence, Physical Review Letters, 2011

[6] IBM, avance extraordinario de IBM: La informática cuántica finalmente 'al alcance', www.foxnews.eom/tech/2012/02/28/ibm-quantum-computing-as-little-as-10-years-off/, 2012

**REIVINDICACIONES**

1. Un método para la ejecución segura de software (1) en una entidad receptora (3), en donde el software (1) se ejecuta condicionalmente en la entidad receptora (3), comprendiendo el método las siguientes etapas en la entidad receptora (3):

ejecutar (301) un esquema de autenticación (4) con una entidad emisora (2) usando una clave de autenticación (5) para seleccionar una función de troceo (6) de una familia de funciones de troceo (13),
recibir (302) el software (1) y un primer valor de troceo (7) desde la entidad emisora (2), en donde el primer valor de troceo (7) se calcula introduciendo el software (1) en la función de troceo (6) seleccionada, almacenar (303) la clave de autenticación (5) y datos de uso (8) acerca del uso de la clave de autenticación (5) almacenada, en donde los datos de uso (8) se incorporan como un contador o como una lista blanca,
actualizar (304) los datos de uso (8) con cada uso de la clave de autenticación (5) almacenada, marcar (305) la clave de autenticación (5) almacenada como inutilizable o borrar la clave de autenticación (5) almacenada cuando se alcanza un cierto límite de uso definido por un umbral de acuerdo con los datos de uso actualizados (8),
introducir (306) el software (1) recibido en la función de troceo (6) seleccionada para calcular un segundo valor de troceo (9), y
ejecutar (307) el software (1) únicamente si el primer valor de troceo (7) es igual al segundo valor de troceo (9),
en donde el primer valor de troceo recibido (7) es un primer valor de troceo (7) individual respectivo calculado individualmente para cada una de una pluralidad de entidades receptoras (3).

2. El método de la reivindicación 1,
en donde el esquema de autenticación (4) es teóricamente seguro para la información.

3. El método de la reivindicación 2,
en donde el esquema de autenticación (4) es el esquema de autenticación de Wegman y Carter.

4. El método de una de las reivindicaciones 1 a 3,
en donde la entidad receptora (3) es un vehículo, en particular un coche, un camión, una aeronave, un vehículo espacial o una embarcación.

5. El método de una de las reivindicaciones 1 a 3,
en donde la entidad receptora (3) es un dispositivo, en particular un dispositivo móvil o un dispositivo crítico para la seguridad, en particular un dispositivo médico.

6. El método de una de las reivindicaciones 1 a 5,
en donde un conjunto de N claves de autenticación (5) se almacena en la entidad receptora (3) en un proceso de producción de la entidad receptora (3), siendo $N \geq 1$.

7. El método de la reivindicación 6,
en donde el conjunto de N claves de autenticación (5) se almacena en un hardware seguro, en donde el hardware seguro se instala en la entidad receptora (3).

8. El método de una de las reivindicaciones 1 a 7,
en donde la clave de autenticación (5) se protege usando criptografía de caja blanca.

9. El método de una de las reivindicaciones 1 a 8,
en donde la clave de autenticación (5) almacenada en la entidad receptora (3) se modifica por medio de datos de modificación (10).

10. El método de la reivindicación 9,
en donde los datos de modificación (10) contienen una clave de modificación, en donde la clave de autenticación (5) almacenada se procesa mediante una función XOR con la clave de modificación para formar una nueva clave de autenticación (5).

11. El método de la reivindicación 9,
en donde los datos de modificación (10) son más pequeños que la clave de autenticación (5), o más pequeños que N claves de autenticación (5), en donde los datos de modificación (10) se introducen en un algoritmo para generar datos de modificación ampliados (14).

12. Una entidad receptora (3) para la ejecución segura de software (1), en donde el software (1) se ejecuta condicionalmente en la entidad receptora (3), comprendiendo la entidad receptora (3):

una unidad de autenticación (31) para ejecutar un esquema de autenticación (4) con una entidad emisora (2) usando una clave de autenticación (5) para seleccionar una función de troceo (6) de una familia de funciones de troceo (13),

una unidad de recepción (32) para recibir el software (1) y un primer valor de troceo (7) desde la entidad emisora (2), en donde el primer valor de troceo (7) se calcula introduciendo el software (1) en la función de troceo (6) seleccionada,

una unidad de almacenamiento (33) para almacenar la clave de autenticación (5) y datos de uso (8) acerca del uso de la clave de autenticación (5) almacenada, en donde los datos de uso (8) se incorporan como un contador o como una lista blanca,

una unidad de procesamiento (34) que está configurada para actualizar los datos de uso (8) con cada uso de la clave de autenticación (5) almacenada, para marcar la clave de autenticación (5) almacenada como inutilizable o para borrar la clave de autenticación (5) almacenada cuando se alcanza un cierto límite de uso definido por un umbral de acuerdo con los datos de uso actualizados (8), para introducir el software (1) recibido en la función de troceo (6) seleccionada para calcular un segundo valor de troceo (9), y para ejecutar el software (1) únicamente si el primer valor de troceo (7) es igual al segundo valor de troceo (9),

en donde el primer valor de troceo recibido (7) es un primer valor de troceo (7) individual respectivo calculado individualmente para cada una de una pluralidad de entidades receptoras (3).

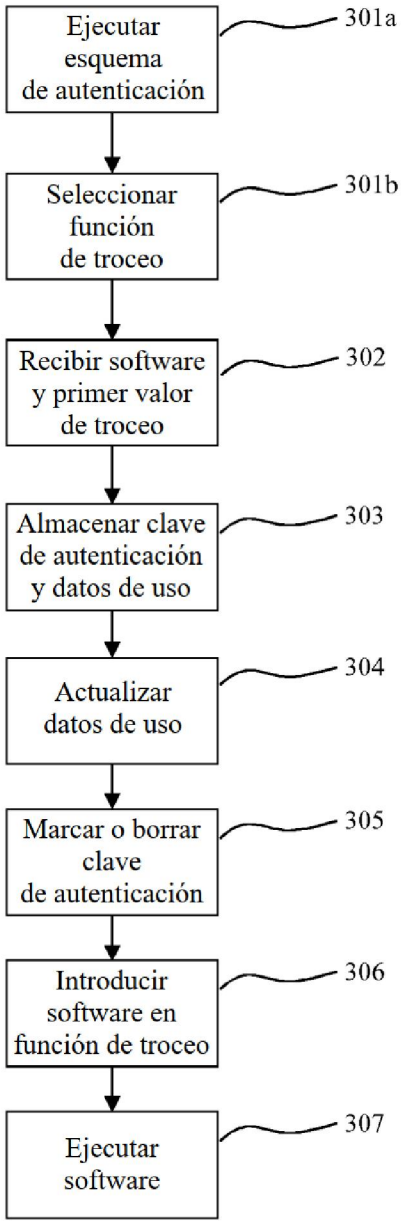
Ejecutar esquema de autenticación
301a
Seleccionar función de troceo
301b
Recibir software y primer valor de troceo
302
Almacenar clave de autenticación y datos de uso
303
Actualizar datos de uso
304
Marcar o borrar clave de autenticación
305
Introducir software en función de troceo
306
Ejecutar software
307

Fig. 1

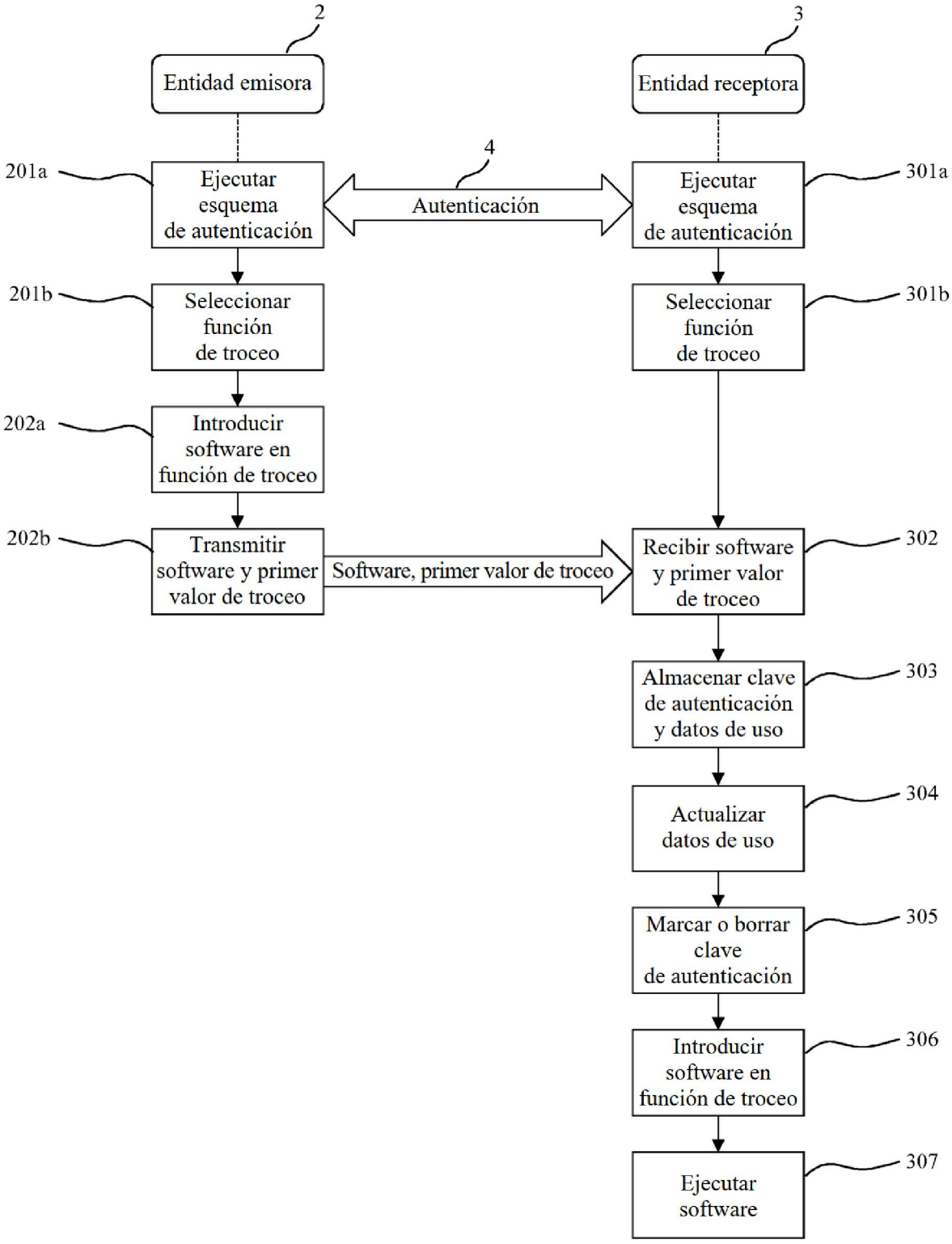
2
Entidad emisora
3
Entidad receptora
4
Autenticación
201a
Ejecutar esquema de autenticación
301a
Ejecutar esquema de autenticación
201b
Seleccionar función de troceo
301b
Seleccionar función de troceo
202a
Introducir software en función de troceo
202b
Transmitir software y primer valor de troceo
Software, primer valor de troceo
302
Recibir software y primer valor de troceo
303
Almacenar clave de autenticación y datos de uso
304
Actualizar datos de uso
305
Marcar o borrar clave de autenticación
306
Introducir software en función de troceo
307
Ejecutar software

Fig. 2

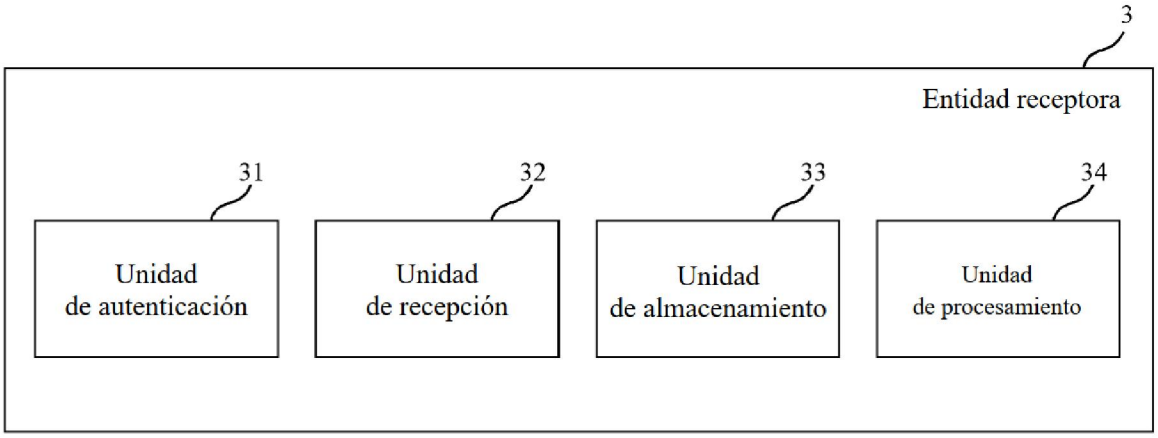
3
Entidad receptora
31
Unidad de autenticación
32
Unidad de recepción
33
Unidad de almacenamiento
34
Unidad de procesamiento

Fig. 3

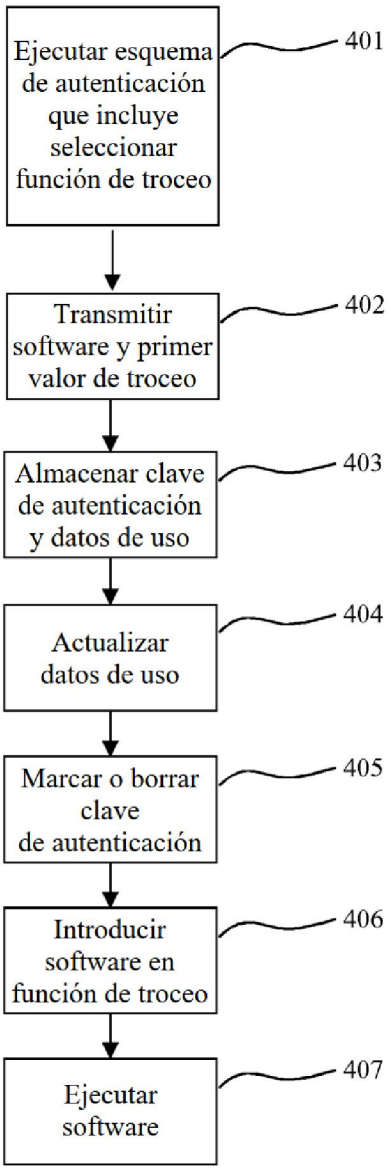
Ejecutar esquema de autenticación que incluye seleccionar función de troceo
401
Transmitir software y primer valor de troceo
402
Almacenar clave de autenticación y datos de uso
403
Actualizar datos de uso
404
Marcar o borrar clave de autenticación
405
Introducir software en función de troceo
406
Ejecutar software
407

Fig. 4

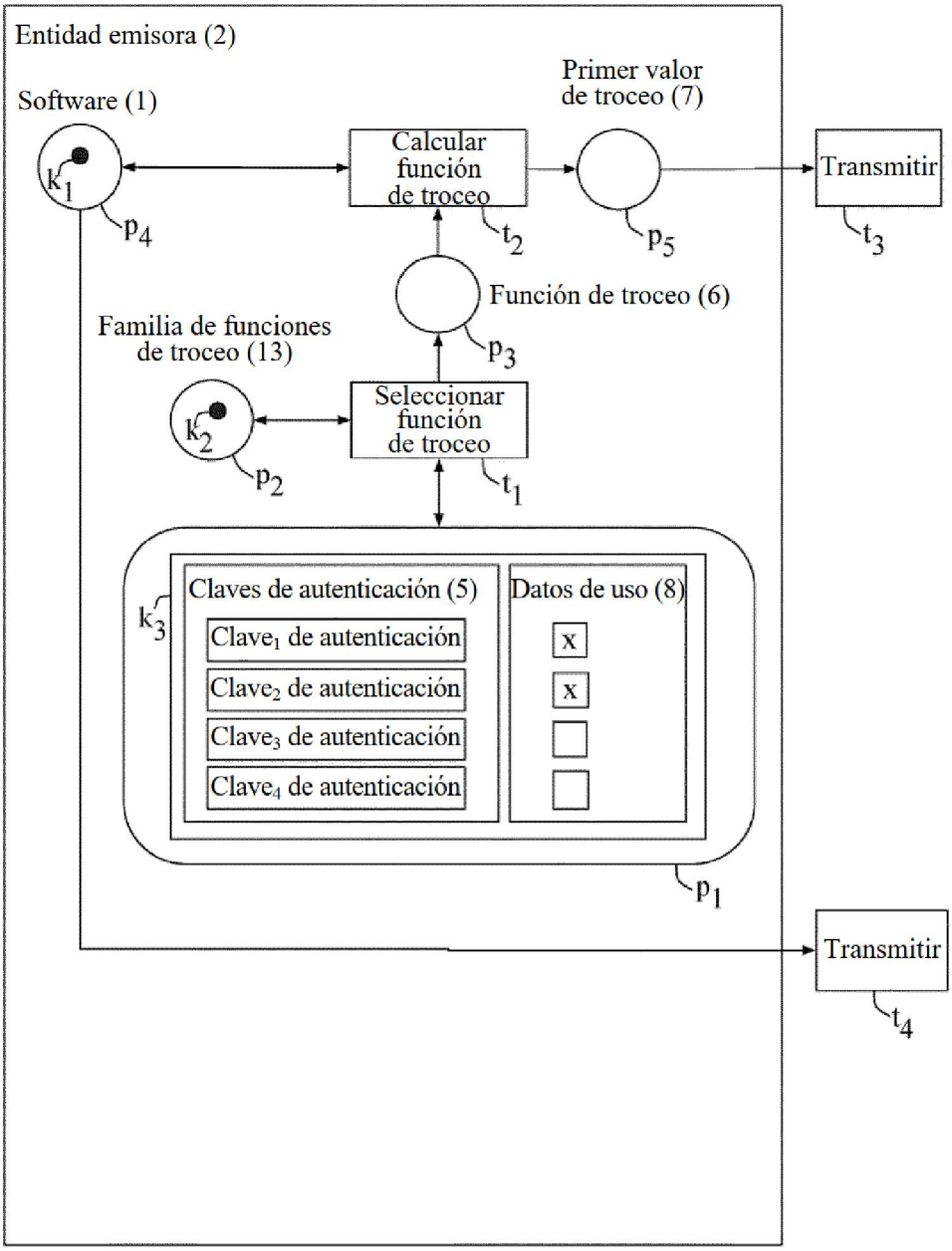
Entidad emisora (2)
Software (1)
Primer valor de troceo (7)
Calcular función de troceo
Transmitir
Función de troceo (6)
Familia de funciones de troceo (13)
Seleccionar función de troceo
Claves de autenticación (5)
Datos de uso (8)
Clave1 de autenticación
Clave2 de autenticación
Clave3 de autenticación
Clave4 de autenticación
x
x
Transmitir
k1
k2
k3
p1
p2
p3
p4
p5
t1
t2
t3
t4

Fig. 5

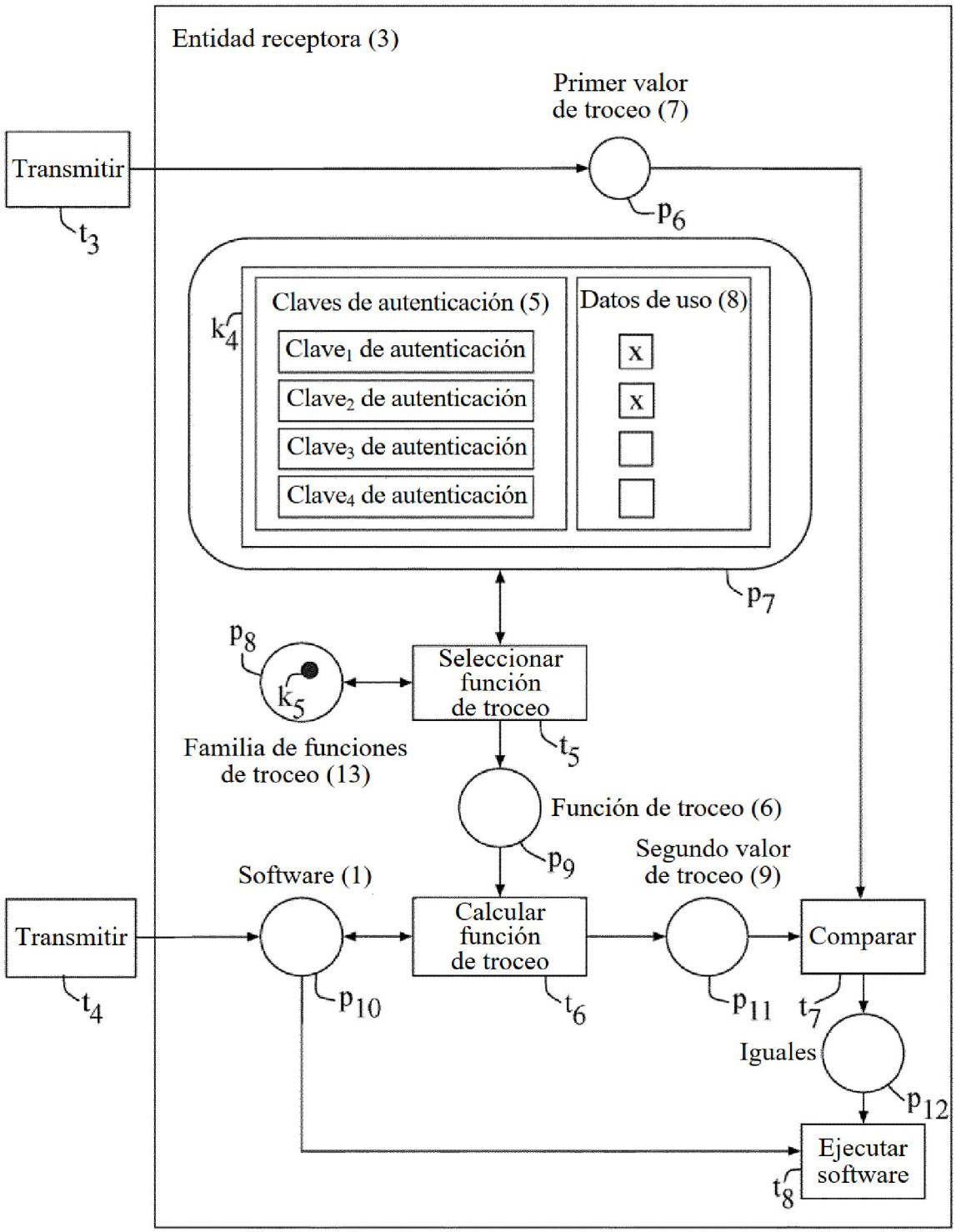
Entidad receptora (3)
Primer valor de troceo (7)
Transmitir
t3
p6
k4
Claves de autenticación (5)
Datos de uso (8)
Clave1 de autenticación
Clave2 de autenticación
Clave3 de autenticación
Clave4 de autenticación
x
x
p7
p8
k5
Seleccionar función de troceo
t5
Familia de funciones de troceo (13)
Función de troceo (6)
p9
Software (1)
Segundo valor de troceo (9)
Transmitir
t4
p10
Calcular función de troceo
t6
p11
Comparar
t7
Iguales
p12
Ejecutar software
t8

Fig. 6

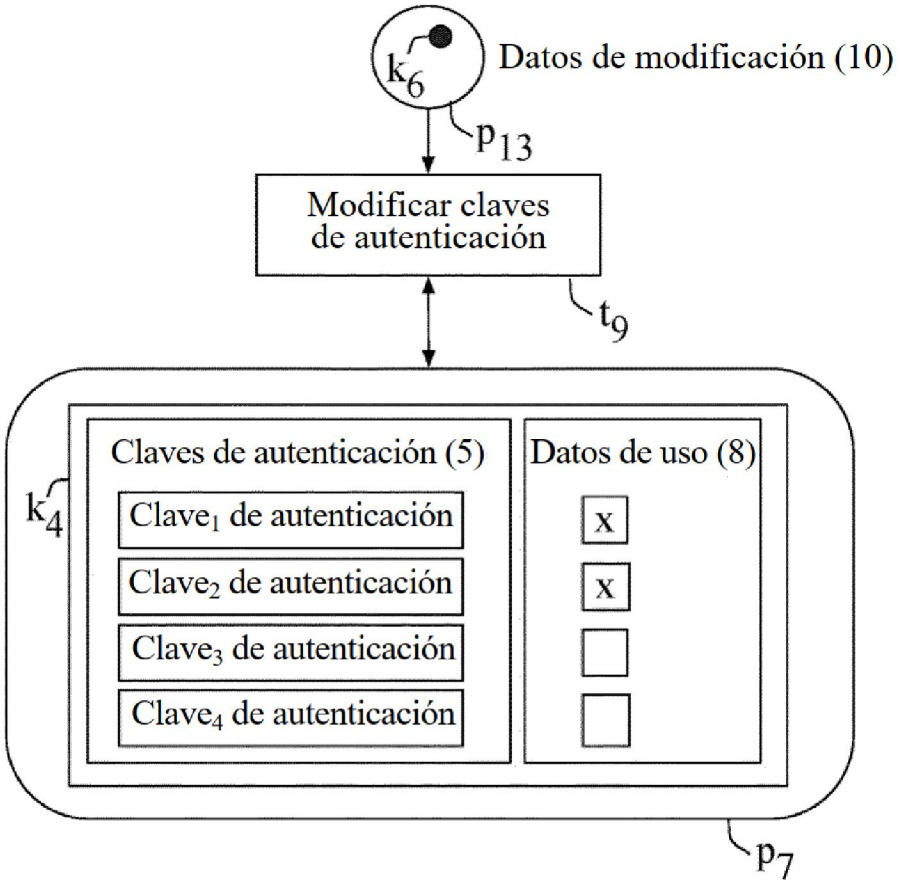
k6
Datos de modificación (10)
p13
Modificar claves de autenticación
t9
k4
Claves de autenticación (5)
Clave1 de autenticación
Clave2 de autenticación
Clave3 de autenticación
Clave4 de autenticación
Datos de uso (8)
x
x
p7

Fig. 7

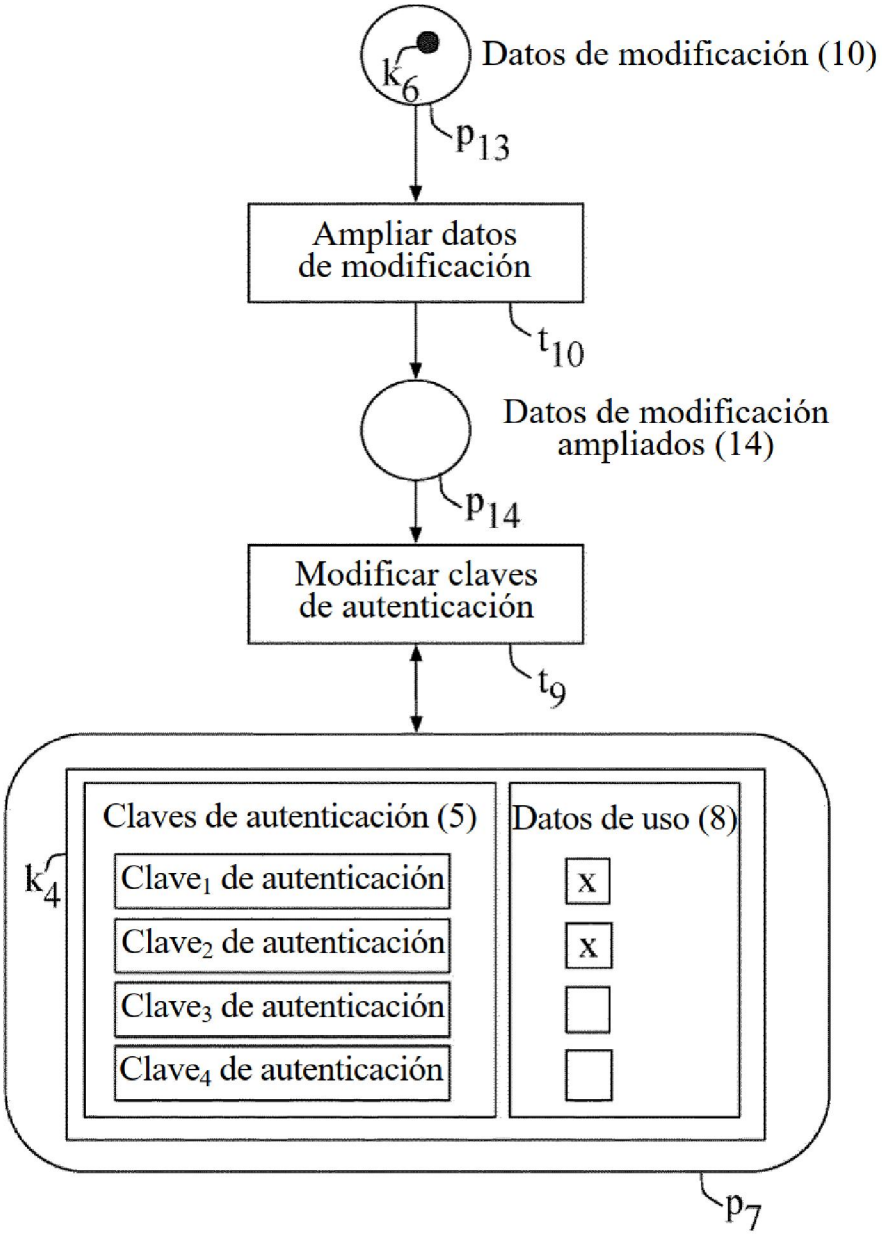
Datos de modificación (10)
k6
p13
Ampliar datos de modificación
t10
Datos de modificación ampliados (14)
p14
Modificar claves de autenticación
t9
k4
Claves de autenticación (5)
Clave1 de autenticación
Clave2 de autenticación
Clave3 de autenticación
Clave4 de autenticación
Datos de uso (8)
x
x
p7

Fig. 8