(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 881 123**

(51) Int. CI.:

***H04L 9/08*** (2006.01)
***H04L 9/12*** (2006.01)
***H04L 9/32*** (2006.01)
***H04L 29/06*** (2006.01)
***H04W 12/06*** (2011.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **19.04.2016 PCT/ES2016/070275**

(87) Fecha y número de publicación internacional: **26.10.2017 WO17182679**

(96) Fecha de presentación y número de la solicitud europea: **19.04.2016 E 16899312 (9)**

(97) Fecha y número de publicación de la concesión europea: **26.05.2021 EP 3432508**

(54) Título: **Método implementado por ordenador para generación de contraseñas y productos de programa informático del mismo**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
**26.11.2021**

(73) Titular/es:

**TELEFONICA CIBERSECURITY & CLOUD TECH S.L.U. (100.0%)**
**Ronda de la Comunicación s/n**
**28050 Madrid, ES**

(72) Inventor/es:

**MARTÍN RODRÍGUEZ, RICARDO;**
**DE LOS SANTOS VILCHEZ, SERGIO;**
**TORRANO GIMÉNEZ, CARMEN;**
**GUZMÁN SACRISTÁN, ANTONIO y**
**ALONSO CEBRIÁN, JOSÉ MARÍA**

(74) Agente/Representante:

**ARIZTI ACHA, Monica**

## DESCRIPCIÓN

Método implementado por ordenador para generación de contraseñas y productos de programa informático del mismo

Campo de aplicación

La presente invención concierne a un método implementado por ordenador y a productos de programa informático para generación de contraseñas que son robustas y desconocidas para el usuario y que solo funcionan en un lugar web que sea adecuado y validado como seguro.

Antecedentes de la invención

El uso seguro de contraseñas siempre ha supuesto un reto en seguridad. Las contraseñas protegen los “bienes online” de los usuarios frente a otros usuarios no legítimos. Custodiarlas adecuadamente es fundamental para mantener el servicio seguro o secreto. Los principales problemas de uso de una contraseña, o los retos a los que se enfrentan los usuarios para poder gestionarlas adecuadamente, son:

- Evitar la reutilización de contraseñas diferentes para cada servicio. Esto es fundamental para que, ante una posible revelación de la información, no se vean comprometidos todos los servicios de un usuario.
- Utilización de contraseñas seguras. Utilizar contraseñas predecibles, sencillas o deducibles no aporta seguridad real.
- Mantener las contraseñas secretas. El propio usuario puede divulgarlas consciente o inconscientemente.
- Resistir ataques de ingeniería social, en particular, el *phishing*. Una buena parte de contraseñas son reveladas por los propios usuarios porque no son conscientes de que las están introduciendo en un lugar no seguro.

Para solucionar el problema de la gestión de contraseñas de usuarios, existen gestores de contraseñas en el mercado que permiten almacenar tantas contraseñas como se desee de forma segura (cifrada), acceder a ellas a través de una sola contraseña maestra y generar contraseñas seguras.

Existen tres formas fundamentales de gestionar la contraseña y, con ellas, buena parte de la identidad digital.

- Manual: El usuario suele gestionar una serie de contraseñas que utiliza en un número limitado de servicios. Suelen usarse contraseñas fáciles de recordar y de escribir y suelen repetirse entre los diferentes servicios. Esto implica el uso de contraseñas débiles (para que puedan ser recordadas) y normalmente implica incluso que sean predecibles. Por el contrario, no suelen ser almacenadas. Guardar las claves en este paso puede resultar en un almacenaje inseguro (texto en claro).
- Gestión local: El usuario gestiona sus contraseñas a través de un gestor en local. Los programas de gestión de contraseñas suelen ocuparse de los aspectos descuidados en el proceso de gestión manual. A través de una contraseña maestra, abren un fichero (cifrado) donde se almacenan el resto de contraseñas. Además de esta funcionalidad básica, suelen permitir la generación de contraseñas seguras y diferentes para cada servicio.

Esto libera al usuario de los aspectos más incómodos de la gestión de las contraseñas. En este sentido, existen varios programas que cumplen con este objetivo: *PasswordSafe, KeePass* y *1Password* se consideran los más conocidos y utilizados. Estas soluciones suelen ser de código abierto, multiplataforma y gestionan todos los aspectos relacionados con las contraseñas, desde la generación hasta el almacenamiento seguro. La contraseña maestra no es almacenada, sino que se utiliza para el descifrado (habitualmente síncrono) del fichero con el resto de contraseñas.

Sin embargo, estas son soluciones pensadas fundamentalmente para el uso en local y carecen de sincronización. En otras palabras, el usuario debe llevar consigo el fichero cifrado con las contraseñas, abrirlo con la contraseña maestra y poder así utilizar las contraseñas de la manera tradicional.

En algunos casos como *1Password*, el programa incluye una funcionalidad para realizar la sincronización a través de terceros; sin embargo, esto no supone una sincronización real del sistema de contraseñas, sino una comodidad adicional para que el usuario disponga de su cartera de claves en diferentes lugares.

- Gestión en la nube: Para paliar este problema de sincronización, han surgido otros productos como *LastPass*, *Dashlane* o *Encryptr*. Suelen utilizarse en forma de multiplataforma o integrados en forma de plugin para el navegador (no es el caso de *Encryptr*). Su principal ventaja es la sincronización, puesto que la información relevante se almacena en los servidores de cada producto. Al tratarse de una información tan sensible, el

almacenamiento se basa en una política de cero conocimientos sobre lo que se almacena. Esto se resuelve habitualmente almacenando el contenido completamente cifrado en el servidor y descifrándolo en el cliente. Aun así, este sistema puede llegar a plantear dudas ante un único repositorio de contraseñas (cifradas) que lo convierte en un único punto de fallo. Además, los servicios de sincronización, ya sea integrada o usando servicios de terceros, requieren que el servicio esté unido de alguna forma al usuario, ya sea como usuarios del servicio en sí o dándose de alta (un registro) como un usuario en la propia plataforma. Un atacante con acceso al correo electrónico de la víctima podría ser capaz de recuperar la contraseña del servicio a través del correo electrónico. En cualquier caso, los datos son adjudicables al usuario y están almacenados en el servidor de un tercero.

Una debilidad común a todos los sistemas es que:

- En el caso de los sistemas con sincronización, estos obligan a utilizar sistemas de almacenamiento de terceros, donde (aunque cifrada) se almacena información sensible o que puede relacionarse con el usuario.
- No integran ninguna solución para que la contraseña sea introducida en el dominio correcto y no sufra *phishing*. Aunque la contraseña se genera y almacena de forma segura, si se introduce en el formulario incorrecto controlado por el atacante, queda totalmente expuesta.
- Todos los sistemas se basan por completo en el conocimiento de una contraseña maestra. Con acceso a esta contraseña, se podría tener acceso a cualquier otra contraseña almacenada en el mismo sistema.

Descripción de la invención

Realizaciones de ejemplo de la presente invención contribuyen, de acuerdo con un primer aspecto, a un método implementado por ordenador para generación de contraseñas, en el que un usuario accede mediante un primer dispositivo de computación por primera vez a una página o sitio web identificado por un dominio web que requiere la identificación, mediante la utilización de credenciales incluyendo un identificador de usuario y una contraseña, de dicho usuario en dicha página o sitio web.

El método propuesto comprende generar dicha contraseña requerida por la página o sitio web mediante la realización de las siguientes etapas:

- interceptar, por dicho primer dispositivo de computación, cuando el usuario accede a dicha página o sitio web;
- recuperar, por el primer dispositivo de computación, información referente a dicho dominio web y a dicho identificador de usuario;
- generar, por el primer dispositivo de computación, una firma en local, es decir, dentro de su propio ámbito, que identifique el dominio que posee el recurso solicitado, o también denominado *Domain Signature* local;
- comprobar, por el primer dispositivo de computación, si ya existe en una base de datos suya algún registro para dicho dominio web y para dicho identificador de usuario y comprobar si un valor criptográfico, o Id_hash (208), es distinto de cero (en otras palabras, este valor criptográfico se comprueba para poder determinar cuándo debe generarse una nueva contraseña);
- enviar, por el primer dispositivo de computación, si el resultado de dicha comprobación ha sido falso, información referente al dominio web y al identificador de usuario a un servidor remoto, para que este último pueda generar una *Domain Signature* para ese dominio web;
- recibir, por el primer dispositivo de computación, de dicho servidor remoto, la *Domain Signature* generada y generar un valor aleatorio para dicho valor criptográfico, o Id_hash; y
- comprobar, por el primer dispositivo de computación, que la *Domain Signature* recibida del servidor remoto coincide con la *Domain Signature* local previamente generada y, en caso de que coincidan, generar un input necesario para que un módulo generador de contraseñas genere la contraseña requerida por la página o servicio web mediante la realización de las siguientes acciones:

  i. solicitar, por el primer dispositivo de computación, al usuario la introducción en el primer dispositivo de computación de una contraseña maestra, o Master-Pass;

  ii. ejecutar, por el primer dispositivo de computación, una función matemática criptográfica sobre el identificador de usuario, el identificador del dominio web, el Id_Hash, la *Domain Signature* local y la Master-Pass solicitada, dando como resultado un valor encriptado que se concatena con unas políticas de contraseña relacionadas con el citado dominio;

iii. enviar, por el primer dispositivo de computación, la información resultante de dicha concatenación a dicho módulo generador de contraseñas para que este último pueda generar la contraseña requerida por la página o sitio web implementando un algoritmo de computación que utiliza la Master-Pass y el Id_Hash; y

iv. recibir, por el primer dispositivo de computación, la contraseña generada por el módulo generador de contraseñas y directamente, sin almacenarla, insertarla en la página o sitio web.

Por tanto, la generación de la contraseña para la citada página o sitio web implica incorporar etapas destinadas a la detección de cualquier intento de suplantación del dominio para el que se genera. Además, la generación de la contraseña se realiza de tal manera que previene que el usuario utilice contraseñas ya definidas para otros servicios o que genere contraseñas que no sean robustas. Para ello, el usuario solo necesita fijar la Master-Pass y a partir de ella y la información necesaria para proteger al usuario de los intentos de suplantación de dominio, el método propuesto genera diferentes contraseñas para todos los servicios a los que vaya a acceder el usuario.

En una realización de ejemplo, el mencionado módulo generador de contraseñas está incluido en el servidor remoto. Alternativamente, en otra realización de ejemplo, el citado módulo generador de contraseñas está incluido en un programa de computación dedicado que se ejecuta en el primer dispositivo de computación.

Según la presente invención, las políticas de contraseña relacionadas con el citado dominio incluyen una longitud mínima que la contraseña generada debe tener en relación con el número de letras mayúsculas y minúsculas y/o número de dígitos y/o número de caracteres no alfanuméricos.

Asimismo, según la presente invención, las políticas de contraseña relacionadas con el citado dominio se pueden recuperar directamente del propio dominio o alternativamente pueden ser definidas por el propio usuario o incluso pueden estar configuradas por el primer dispositivo de computación y almacenadas en su base de datos.

En una realización de ejemplo preferida, el citado algoritmo de computación genera una función criptográfica binaria, o Meta-Pass, a partir de la combinación de la Master-Pass y del Id_Hash y la realización de un mapeo de bits que forman la Master-Pass con unos caracteres asociados con las políticas de contraseña relativas al citado dominio.

Además, según dicha realización de ejemplo, el citado algoritmo de computación puede incluir caracteres aleatorios sobre un resultado obtenido por dicho mapeo para obtener una contraseña que cumpla con una longitud predeterminada de acuerdo con dichas políticas de contraseña relativas al citado dominio y puede mezclar los caracteres que forman parte de dicha contraseña de una longitud predeterminada.

En una realización de ejemplo, la citada etapa iv) comprende además almacenar, por el primer dispositivo de computación, en la base de datos y asociado al identificador de usuario y al identificador del dominio web, el Id_Hash asociado a las credenciales del usuario para la página o sitio web.

Asimismo, el método propuesto incluye un protocolo de sincronización para que el usuario pueda gestionar sus contraseñas desde diferentes dispositivos de computación. A este fin, en una realización de ejemplo, el método propuesto comprende además:

- acceder, por el usuario, a la página o sitio web con un segundo dispositivo de computación, en el que dicho segundo dispositivo de computación está sincronizado con el primer dispositivo de computación a través de la introducción de una contraseña de un solo uso, u OTP, en el segundo dispositivo de computación, habiendo sido dicha OTP previamente generada por el servidor remoto;

- enviar, por el segundo dispositivo de computación, la OTP al servidor remoto para que este último pueda realizar su validación;

- recibir, por el segundo dispositivo de computación, si el resultado de dicha validación ha sido correcto, un mensaje encriptado del servidor remoto, en el que dicho mensaje encriptado incluya dicha OTP y unos datos asociados a la página o sitio web encriptados mediante la utilización de la Master-Pass, y en el que dichos datos incluyen el identificador de usuario, el identificador del dominio web y el Id_Hash; y

- solicitar, por el segundo dispositivo de computación, que el usuario introduzca la Master-Pass para descifrar el contenido de dicho mensaje encriptado.

En aún otra realización de ejemplo, cuando el usuario, mediante el primer dispositivo de computación, accede por segunda vez a la página o sitio web, el método propuesto mediante la utilización del primer dispositivo de computación comprende:

- interceptar dicho segundo acceso a la página o sitio web y recuperar la información referente al dominio web y al identificador de usuario;

- generar una nueva *Domain Signature* local para el dominio web y comparar dicha *Domain Signature* local generada con la *Domain Signature* local previamente generada;

- comprobar si ya existe en la base de datos un registro para dicho dominio web y para dicho identificador de usuario y comprobar también si el Id_Hash es distinto de cero;

- si el resultado de dicha comprobación ha sido verdadero, solicitar que el usuario introduzca en el primer dispositivo de computación la Master-Pass;

- ejecutar una función matemática criptográfica sobre el identificador de usuario, el identificador del dominio web, el Id_Hash, la *Domain Signature* local y la Master-Pass solicitada, dando como resultado un valor encriptado que se concatena con las políticas de contraseña relativas al citado dominio; y

- enviar la información resultado de dicha concatenación al módulo generador de contraseñas para que este último pueda generar la contraseña requerida por la página o sitio web.

Otras realizaciones de la invención que se dan a conocer en el presente documento incluyen también productos de programas informáticos para realizar las etapas y operaciones del método propuesto en el primer aspecto de la invención. Más específicamente, un producto de programa informático es una realización que tiene un medio legible por ordenador que incluye instrucciones de programa informático codificadas en el mismo que, cuando se ejecutan en al menos un procesador de un sistema informático, hacen que el procesador realice las operaciones indicadas en el presente documento como realizaciones de la invención.

Por tanto, la presente invención proporciona un método de generación de contraseñas que son robustas y desconocidas para el usuario (lo cual evita que estas se puedan revelar) y que solo funcionan en un lugar web que es adecuado y validado como seguro, de modo que se evita el almacenamiento permanente de las contraseñas en un servicio Cloud o en local.

La definición de la arquitectura propuesta permite que el generador de contraseñas se ubique en un servidor propio o en un servidor de terceros.

Asimismo, la presente invención proporciona una solución totalmente portátil, que puede integrarse con cualquier navegador, desde cualquier lugar y que define una API que garantiza que el servicio se puede usar desde fuera del navegador.

Breve descripción de los dibujos

Las anteriores y otras características y ventajas se entenderán mejor a partir de la siguiente descripción detallada de realizaciones de ejemplo que deben tomarse de forma meramente ilustrativa y no limitativa, con referencia a los dibujos que la acompañan, en los que:

La figura 1 ilustra esquemáticamente los componentes que pueden formar parte del sistema propuesto para la generación de contraseñas.

La figura 2 ilustra esquemáticamente la secuencia de etapas implementada por la invención para generar una contraseña robusta a partir de un input recibido de un usuario.

La figura 3 es un diagrama de flujo que ilustra esquemáticamente el flujo de datos del método propuesto para la generación de contraseñas.

La figura 4 es un diagrama de flujo que ilustra el protocolo de sincronización propuesto por la presente invención para la generación de contraseñas entre diferentes *User equipments* (UE).

Descripción detallada de una realización de ejemplo

La figura 1 muestra una realización de ejemplo preferida de los módulos/elementos de un sistema propuesto por la presente invención para generar contraseñas. La figura muestra al usuario 100 que interacciona con una máquina del usuario o primer dispositivo de computación 200, por ejemplo un *user equipment* (UE). Asimismo, en la figura se ilustra un servidor remoto 300 accesible desde Internet.

El sistema se propone para garantizar la portabilidad del método propuesto y la integración de múltiples UE desde los

que el usuario 100 podrá gestionar sus contraseñas. Para ello los diferentes módulos del sistema facilitan la sincronización de la información local. En esta arquitectura se asume que el usuario 100 es el único que conoce la clave maestra (Master-Pass 101).

Según este ejemplo de realización, el UE 200 incluye:

- Un módulo interceptor de credenciales 201: Este módulo está configurado para interceptar la inserción de credenciales en páginas web. Asimismo, este módulo puede detectar cuándo el usuario 100 se encuentra visitando una página susceptible de necesitar una contraseña gestionada por el sistema.

- Una base de datos local 205: Almacena la información relativa a cada uno de los servicios web para los cuales se quiera proteger la contraseña generada. La información que se almacenará en la citada base de datos local 205 será:

  1. Nombre de usuario 206 (o identificador de usuario): Identifica al usuario 100 para el servicio alojado en el dominio. Permite el uso y gestión de diferentes nombres de usuario en el mismo dominio, aunque pertenezcan a la misma persona.

  2. Dominio 207 (o identificador del dominio web) para el cual se genera la contraseña.

  3. Id_Hash 208: Valor criptográfico que, combinado con la Master-Pass 101, permite generar la contraseña adecuada para cada dominio. Si se desea cambiar la contraseña para un dominio y usuario en concreto, solo será necesario cambiar este dato.

  4. Políticas de contraseña 209: Política relacionada con las características de la contraseña generada. Esto garantiza que la metodología propuesta genera contraseñas robustas. Éstas políticas incluyen el tamaño de la contraseña, además del mínimo número de elementos que debe contener la contraseña en relación con el número de letras en mayúsculas y minúsculas, número de dígitos y número de caracteres no alfanuméricos. Cuando sea posible, las políticas se recuperan del dominio. Si el dominio no tiene una política de contraseñas definida o no es posible su recuperación automática, el usuario puede introducir estos valores manualmente. En todo caso, si no hay una definición explicita de estas políticas, la presente invención proporciona una política por defecto para garantizar la robustez de las contraseñas generadas.

- Un módulo generador de firma de dominio local 202: Este módulo está configurado para permitir la obtención de la firma del dominio, en adelante denominada *Domain Signature* (*hash* del certificado) de un dominio dado, y que la compara con la almacenada en la base de datos 205. La *Domain Signature* mitiga los ataques de tipo hombre en el medio (*Man in the Middle*, MitM) en los que los certificados TLS han sido comprometidos o modificados. Este valor almacenará la información necesaria para identificar el lugar visitado. Si el usuario pretende hacer uso del generador de contraseñas en un dominio sospechoso de no estar protegido con la información criptográfica "habitual", el sistema le alertará.

- Un módulo de comunicación 203: Este módulo está configurado para recuperar y permitir el envío de la información necesaria para que el servidor remoto 300 pueda generar de forma segura la *Domain Signature* asociada a un dominio dado, durante la generación de las credenciales. Este módulo posibilitará la solicitud de forma segura de una contraseña al usuario 100 que actuará como una clave maestra y permitirá generar el input del sistema sin comprometer la información que identifica al usuario 100. Se pueden, por ejemplo, utilizar las funciones criptográficas de *hashin*g para producir una cadena cuyo conocimiento imposibilita deducir la información que se usó para generarla, pero que será diferente a otras cadenas generadas a partir de información diferente. Se genera entonces un *hash* a partir de la concatenación del Identificador de usuario 206, Identificador del dominio web 207, Id_Hash 208 y la Master-Pass 101. A esto se le añade la estructura de datos que ilustra qué política de mínimos está asociada a la contraseña establecida por el usuario 100. El *hash* y la política se denominan input. Este módulo envía el input al servidor remoto 300 y espera recibir la contraseña generada.

- Un módulo de actualización 204: Este módulo asegura que, en el caso de que el usuario 100 utilice varios UE 200, la información entre todos ellos sea coherente.

Además, el servidor remoto 300 incluye:

- Un módulo de comunicación 304: Este módulo está configurado para facilitar la comunicación con los clientes recibiendo solicitudes de generación de *Domain Signature* o de contraseñas y enviando el resultado de ambos procesos.

- Un módulo generador de firmas 301: Este módulo está configurado para generar, si es posible, la *Domain Signature* de un dominio dado.

- Un módulo generador de contraseñas 302: Este módulo está configurado para generar la contraseña solicitada por el usuario 100. Este módulo recibe como una entrada el input enviado por el cliente 100.

- Un módulo gestor de sincronización 303: Este módulo permite la actualización de la información entre varios UE 200 gestionados por dicho usuario 100. Esto permitirá que, por ejemplo, el usuario 100 pueda gestionar sus contraseñas desde diferentes dispositivos de computación, por ejemplo un PC, un teléfono móvil, una tablet, etc.

La distribución de los módulos mostrada en la figura 1 puede alterarse. Específicamente, el módulo generador de contraseñas 302, en otras realizaciones de ejemplo alternativas y en este caso no ilustradas, en lugar de estar incluido en el servidor remoto está incluido en un programa de computación dedicado que se ejecuta en el UE 200.

Con referencia ahora a la figura 2, se muestra una realización de ejemplo para la generación de la contraseña por el módulo generador de contraseñas 302. En primer lugar se genera una Meta-Pass que consiste en un *hash* representado en formato binario generado a partir de una combinación de la Master-Pass 101 y el Id_Hash 208. Para la generación de la contraseña del servicio, se mapean los bits de la Meta-Pass (tantos como sean necesarios para cumplir los requisitos de longitud de la contraseña) a diferentes parámetros *charsets* predefinidos. Se definen *charsets* para caracteres de diferentes tipos: caracteres alfabéticos en minúsculas, caracteres alfabéticos en mayúsculas, dígitos y caracteres no alfanuméricos. Para el mapeo, en primer lugar se garantiza que la contraseña cumpla la política de mínimos establecida por el usuario 100 respecto al tipo de caracteres que contiene. Para ello, se mapean directamente los primeros bits de la Meta-Pass a los *charsets* correspondientes en función de los criterios establecidos en la política. Por ejemplo, si la política establece que la contraseña debe contener al menos dos dígitos y un carácter no-alfanumérico, se mapearían los primeros bits de la Meta-Pass a estos *charsets* para generar los caracteres indicados.

Una vez garantizados los requisitos mínimos de la contraseña, se completa el resto de la contraseña con caracteres elegidos aleatoriamente hasta completar la longitud indicada (en el caso de que fuera necesario). Para ello, se usa un generador de números pseudoaleatorios (en inglés *Pseudo-Random Number Generator*, PRNG), que determina a qué *charset* se deben mapear el resto de los caracteres generados. Como semilla del generador de números pseudoaleatorios se puede utilizar una selección de los primeros bits del input recibido.

Finalmente, se barajan todos los caracteres para garantizar la robustez de la contraseña generada. El resultado final es la contraseña generada para el dominio, la cual enviará al cliente 100 el módulo de comunicación 304.

Con referencia ahora a la figura 3, en la misma se muestra una realización de ejemplo del método propuesto.

A partir del diagrama de flujo de la figura 3 es posible describir cuatro eventos distintos definidos para la gestión de contraseñas: la generación de una contraseña nueva, el uso de una contraseña ya generada, un cambio de contraseña y la revocación o caducidad del certificado del dominio.

Generación de una contraseña nueva:

Cuando un usuario 100 interactúa con un servicio web que le solita identificarse con un nombre de usuario y una contraseña, podrá utilizar la invención propuesta para generar una contraseña evitando la reutilización de contraseñas anteriores. Además esta contraseña se generará a partir de una información de confianza que identifica el dominio, de forma que sea posible detectar si el usuario es víctima de ataques de tipo *phishing*. Por último, el uso de esta invención permitirá introducir la contraseña siempre que se le solicite de forma automática sin que ello suponga tener la contraseña almacenada en alguna parte.

Inicialmente Id_Hash 208 toma el valor cero (NULL). Según el diagrama de la figura 3, una vez que se intercepta la inserción de la contraseña, el módulo encargado de obtener los parámetros que perfila cada dominio en local, es decir el generador de firmas local 202, necesitará recuperar la información que sea necesaria para identificar al dominio. Con esta información se genera la *Domain Signature* local 210.

Tras ello, se comprueba si ya existe algún registro correspondiente para este dominio y este nombre de usuario y, además, si Id_Hash 208 es distinto de NULL. En caso de que no se satisfagan estas condiciones, se envía la información necesaria al servidor remoto 300 para generar la *Domain Signature*. En el servidor remoto 300, el módulo generador de firmas 301 identifica el dominio y contrasta la información recuperada por el usuario 100 con la información recuperada desde fuentes de confianza. Este proceso se explicará más adelante en este documento. El resultado del proceso es una firma que determina qué información identifica legítimamente el dominio (*Domain Signature*). Esta *Domain Signature* se envía al UE 200, que genera un valor aleatorio para IdHash 208. A continuación se comprueba si la *Domain Signature* recibida del servidor remoto 300 (que es de confianza) se corresponde con la firma local 210. Si este no es el caso, se da una alerta de fallo. Si ambas coinciden, se notifica al usuario 100 que se va a crear una nueva contraseña para que compruebe que está en la página web adecuada y se continúa con el

proceso. El UE 200 ejecuta una función de *hashing* sobre los siguientes datos: Identificador de usuario 206, Identificador del dominio web 207, Id_Hash 208, *Domain Signature* local 210 y la Master-Pass 101 solicitada explícitamente al usuario 100. La información relativa a la política de contraseñas 209 definida para este dominio se concatena al *hash* resultante y se envía al servidor remoto 300. El servidor remoto 300, como se explicó anteriormente, genera una clave para este domino y la envía al UE 200. Sin embargo, esta clave no se almacena, si no que directamente se introduce en la página web desde la que se solicitan las credenciales al usuario 100 para registrar una nueva contraseña. Sí se almacenan, sin embargo, los datos que identifican al usuario 100 para este dominio (Identificador de usuario 206), el propio dominio (Identificador del dominio web 207) y el valor criptográfico que diferenciará este registro de otros (Id_Hash 208). El almacenamiento de estos datos no se considera peligroso y evita que se almacene la contraseña. En caso de que la contraseña falle, se fuerza un cambio de contraseña para el cual se introduce Id_Hash=NULL y se vuelve al punto del proceso en el que se recupera la información de dominio.

Uso de una contraseña ya generada:

Cada vez que el usuario 100 necesite introducir la contraseña en un proceso de autenticación deberá solicitar la generación de la contraseña por el sistema. El objetivo de este documento no es entrar en aspectos concretos de la implementación, ya que para cada plataforma o tecnología subyacente estos aspectos serán ligeramente diferentes. Cuando se solicita la contraseña, se recupera la información relativa al usuario 206 y al Identificador del dominio web 207 y se genera la *Domain Signature* local 210. A continuación se comprueba si esta contraseña se ha generado anteriormente. Esto significa que existe un registro que identifica el Identificador del dominio web 207, al usuario para este dominio 206 y un valor que distingue este registro de otros (Id_Hash 208). Como la contraseña que se había generado anteriormente, la *Domain Signature* utilizada en este caso proviene del UE 200 y no del módulo generador de firmas 301. Tras ello, se procede a solicitar la generación de la contraseña. Para ello, se envía el resultado de ejecutar una función de *hashing* sobre los siguientes datos: Identificador de usuario 206, Identificador del dominio web 207, Id_Hash 208, *Domain Signature* local 210 y la Master-Pass 101 solicitada explícitamente al usuario 100. La información relativa a la política de contraseñas 209 definida para este dominio se concatena al *hash* resultante y se envía al servidor remoto 300. El servidor remoto 300 generará una clave para este domino y se la enviará al UE 200. Cuando el UE 200 recibe la contraseña, este completará el proceso de autenticación. Si este proceso falla podría ser un síntoma de que el dominio ha sido comprometido o de que se ha revocado/caducado el certificado. Para ello, se fuerza el cambio de contraseña, poniendo Id_Hash=NULL antes de continuar el proceso.

Cambio de contraseña:

Como se ha comentado anteriormente, la regeneración de una contraseña para un sitio web específico solo implicará el cambio de Id_Hash 208. Cuando se requiere la generación de una nueva contraseña, el UE 200 sustituirá el valor almacenado en el campo Id_Hash 208 por NULL. Si el usuario 100 utiliza más de un UE 200 se forzará el cumplimiento del protocolo de sincronización diseñado para garantizar la coherencia de la información entre todos los nodos.

Revocación o caducidad del certificado del dominio:

Cuando la inserción de la contraseña provoca un fallo puede deberse a tres factores:

- Un error en la inserción de la contraseña maestra.
- Que el certificado del dominio se haya revocado o caducado.
- Que el dominio esté intentando ser suplantado.

En este caso, la presente invención proporciona un uso iterativo del flujo definido en la figura 3 y la definición de una estructura de datos que permita hacer sucesivas comprobaciones. La secuencia a implementar sería, en primer lugar, solicitar al usuario 100 que vuelva a introducir su contraseña maestra (Master-Pass 101). Los módulos que ilustrarían esta segunda introducción de la contraseña maestra no se han ilustran en la figura 3 por simplicidad. La implementación requeriría un contador asociado a cada dominio, que se pondría a cero en cada una de las generaciones de contraseña. Si esto falla, en segundo lugar se fuerza un cambio de contraseña. Esto implica poner a NULL el campo Id_Hash 208 y reiniciar el proceso desde el principio. Esto requeriría que se recuperara de forma legítima la *Domain Signature* asociada al dominio y que se comparara con la obtenida de forma local 210. Si esta comparación destaca alguna diferencia se puede concluir que el dominio era víctima de un ataque de suplantación.

Según la presente invención, para la generación de la *Domain Signature*, el servidor remoto 300 asume la responsabilidad de la generación de las contraseñas asociadas a dominios, además de proteger la confidencialidad de los usuarios, permitiendo protegerlos del robo de estas credenciales basado en la suplantación de sitios web (ataques de *phishing*). Para ello el módulo generador de firmas 301 facilita una instantánea de todos aquellos parámetros que perfilan un dominio y que, si se compromete, deberían alterarse. Cuando se usa la información de los certificados digitales y de la cadena de certificación para perfilar cada dominio, esta técnica se denomina *certificate*

*pinning*. El hecho de que esta característica se proponga para ser ejecutada desde el servidor remoto 300 ofrece una solución de confianza que elude posibles ataques sobre las máquinas locales desde las que opera el usuario 100. La presente invención propone ir más allá de lo que el estándar asociado al *certificate pinning* ofrece. El objetivo es ofrecer un dictamen sobre si el sitio visitado es realmente el que dice ser. Para ello, el sistema tiene varios puntos de datos del visitante y la respuesta será binaria.

Los datos que se pueden obtener del cliente 100 para realizar la comprobación serán: URL visitada completa (no solo el dominio), idioma del navegador (país), rango de IP, cadena de certificados del sitio visitado y nivel de comprobación. Puede haber varios tipos de comprobación o niveles deseados, que se pueden enviar como parámetro.

El servidor remoto 300 se utiliza además como un segundo visitante que, además de detectar los ataques, compare si las respuestas desde puntos diferentes son coherentes y, por tanto, que compruebe si el usuario 100 se encuentra en una red fiable o si sus respuestas están siendo manipuladas. Los casos posibles de ataque son ataques MiTM y *phishing* tradicional.

MiTM con *certificate pinning*:

La presente invención puede ser aprovechada para cubrir el “vacío” provocado por el uso de HSTS y HPKP. Estos protocolos confían en el primer uso para establecer los pines y la redirección TLS en el navegador. A partir de ahí, es el propio navegador el que se encarga de gestionar la seguridad de los sitios que tienen HPKP y HSTS. Se propone que el servidor remoto 300 simule su propio HPKP y almacene los pines de los sitios que visita, de modo que el usuario 100 pueda confiar en el *pinning* del servidor remoto 300 en lugar de hacerlo él mismo, reduciendo el riesgo de que sea comprometido. De esta manera se reduce el riesgo de que el certificado sea inválido la primera vez que se visita un dominio, solventando el inconveniente de los protocolos HSTS y HPKP. El servidor remoto 300 es el encargado de ofrecer esta confianza al usuario 100.

Además, la presente invención propone que el servidor remoto 300 pueda gestionar la seguridad de los sitios que no hayan implementado esta seguridad, reemplazándolos. En un ejemplo de realización, la secuencia sería la siguiente. El UE 200 envía el dominio, el idioma y la cadena de certificados del dominio visitado. Seguidamente, el módulo generador de firmas 301 que ha visitado previamente estos dominios y conserva los pines realiza una comparación de pines, por países si es posible. Si no, lo hará según el nivel de comprobación requerido. El nivel 0, 1, 2, 3 o 4 indicará qué certificado se quiere comprobar, si es el raíz, hoja, intermedio, los tres o ninguno. En el caso de español e inglés, el servidor remoto 300 debe almacenar una estructura como la siguiente, no asociada a ningún usuario en concreto:

Dominio1: es_ES(spkp1, spkp2, spkp3), en_US(spkp1, spkp2, spkp3)

Dominio2: es_ES(spkp1, spkp2, spkp3), en_US(spkp1, spkp2, spkp3),

Siendo spkp1, spkp2 y spsk3 el pin del certificado raíz, hoja e intermedio, respectivamente.

Esta estructura podría proporcionarse por cualquiera o de cualquier forma: desde listas de terceros hasta visitas del propio servidor remoto 300. Asimismo, el propio usuario 100 podría actualizarla dinámicamente.

MiTM sin certificados:

La presente invención en otras realizaciones de ejemplo puede intentar comprobar si la dirección IP del sitio visitado es la habitual o, al menos, si se encuentra en su mismo rango. Para ello, el US 200 envía el dominio, el idioma y la IP del dominio visitado. Luego, el servidor remoto 300, que ya ha visitado previamente estos dominios y que conserva la IP y su rango, realiza una comparación de IP, por países si es posible. Si no, lo hará según el nivel de comprobación requerido. El nivel 0, 1, 2, 3 o 4 indicará qué certificado se quiere comprobar, si es la IP exacta, su rango (primer, segundo octeto…) o nada. En el caso de español e inglés, el servidor remoto 300 debe almacenar una estructura como esta:

Dominio1: es_ES(IP), en_US(IP)

Dominio2: es_ES(IP), en_US(IP).

*Phishing* tradicional:

Aunque los casos anteriores pueden proteger del *phishing*, el *phishing* tradicional debe considerarse como el caso en el que el dominio no es el real que se quiere visitar. En estos casos, la contraseña introducida fallará, puesto que la *Domain Signature* local 210 no será la del dominio auténtico. En ese caso, se introduce Id_Hash=NULL y se itera de nuevo en el proceso. Esto requeriría que se recuperara de forma legítima la *Domain Signature* asociada al dominio y

que se comparara con la obtenida de forma local 210. Si esta comparación destaca alguna diferencia se podrá concluir que el dominio era la víctima de un ataque de suplantación.

Una de las principales ventajas de la presente invención es la alta portabilidad de su diseño. Un usuario 100 que haya confiado en la gestión de sus contraseñas a la solución propuesta por la invención podrá tener desplegados varios UE 200 a través de los cuales puede introducir las contraseñas allí donde se soliciten. Como se ha comentado anteriormente, la idea es que el usuario 100 solo tenga que usar una única contraseña maestra (Mater-Pass 101) y que el sistema sea capaz de generar la contraseña adecuada para cada entorno, siempre la misma, a partir de esta contraseña maestra (Master-Pass 101), de información que identifica el dominio desde el que se están solicitando las credenciales (Identificador del dominio web 207) y del usuario para ese dominio (Identificador de usuario 206). Para que se pueda utilizar la solución propuesta, la presente invención opta por almacenar de forma local esta información al considerarse que su posible exposición no compromete en ningún caso la identidad del usuario 100.

En un escenario en el que el usuario 100 interactuará con múltiples UE 200, sin embargo, esto sí plantea un problema ya que será necesario asegurar la coherencia de la información que se almacena de forma local en cada uno de estos UE 200. Solo garantizando esta coherencia la información usada para solicitar la generación de las contraseñas será la misma en cada uno de los UE 200 usados. Aunque existen alternativas para realizar la actualización de forma automática, sin que tenga que participar activamente el usuario 100, la presente invención opta por una alternativa en la que el usuario 100 debe activar explícita y temporalmente el envío de la versión más actualizada de los datos usados para generar el input. En el diseño del protocolo que resuelve esta actualización se han perseguido los siguientes objetivos:

- Prescindir de cualquier intercambio de credenciales que identifiquen al usuario 100 ante el sistema. Aunque es necesario identificar al usuario 100 para construir un canal seguro por el que enviar la información, esta identificación se hace en base a la generación de contraseñas de un solo uso (*One Time Password*, OTP) y se opta por ello porque el usuario 100 debe ser capaz de capturar esta OTP e introducirlo en cada uno de los UE 200. Esto es coherente con la forma en la que normalmente se resuelve el envío de la OTP con un canal extra y no aporta novedades desde el punto de vista de la etapa innovadora de la invención. Por esta razón no se explicarán las alternativas tecnológicas existentes para resolverlo.

- Para preservar la confidencialidad se opta por un cifrado simétrico basado en la clave maestra del usuario (Master-Pass 101). En este ámbito se supone que el usuario 100 es el mismo en todos los UE 200, así que no es necesario proporcionar soluciones que resuelvan el hecho de compartir la clave.

- Reducir el tiempo en el que la información está fuera de los UE 200. La información siempre se cifra con la clave maestra, pero optar por una estructura de actualización en la que el usuario 100 desencadena la actualización permite que la solución propuesta, independientemente de si se ha recibido la información, tenga un tiempo de vida asociado con la vida de la OTP generada para definir el intercambio.

En la figura 4 se ilustra una realización de ejemplo del intercambio de información que permite la actualización del sistema/método. Es importante destacar que esta estructura da soporte tanto a la primera vez que se utiliza un UE 200 diferente del habitual como en los sucesivos usos que se haga de él. Según el ejemplo de la figura 4, el usuario 100 que utiliza un UE_0 provoca la propagación de las modificaciones en la información que usa el sistema para generar el input del generador de contraseñas (Generador de contraseñas 302). Con este estímulo, el servidor remoto 300, a través del gestor de sincronización 303, genera una contraseña de un solo uso a la que se asigna un tiempo de vida limitado y que se usará para construir un canal lógico que identifique solo temporalmente al usuario 100 a través de los UE 200 que se vayan a usar. Esta OTP se envía al UE_0 en el que se realizan dos procesos: el primero supone cifrar la información a enviar y para ello se solicita al usuario 100 la clave maestra (Master-Pass 101). El resultado del proceso de cifrado, junto con la OTP que identificará temporalmente al usuario 100 se envía al servidor remoto 300. El servidor remoto 300 almacenará durante el tiempo de vida de la OTP el mensaje recibido. El otro proceso que debe facilitarse en el UE_0 es la captura de la OTP para su traslado al UE que se desea actualizar. Esta captura puede depender de la plataforma en la que se va a ejecutar el UE y podrá basarse en un código QR, una cadena de caracteres, etc.

A continuación el usuario 100, en esta ocasión desde el UE_1, solicitará la sincronización de los contenidos almacenados en local. El usuario 100 entonces insertará la OTP en este UE_1. El UE_1 entonces enviará este OTP al servidor remoto 300 que, tras validar si coincide con la OTP que identifica un mensaje que se va a enviar, realizará el envío de dicho mensaje. Este envío no supone la eliminación del mensaje del servidor remoto 300, que permanecerá en el servidor remoto 300 el tiempo de vida asignado a la OTP. De esta forma se facilita la sincronización de varios UE. Al recibir el mensaje, el UE_1 solicitará al usuario 100 que introduzca la clave maestra para descifrar el contenido. Si se trata de un usuario legítimo tendrá esta clave y se habrá actualizado la información.

Por tanto, la presente invención proporciona una solución para la gestión de contraseñas que es portátil, evita el almacenamiento de las contraseñas y alerta de los ataques de *phishing*.

La invención propuesta puede implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementa en software, las funciones pueden almacenarse en o codificarse como una o más instrucciones o código en un medio legible por ordenador.

El medio legible por ordenador incluye medio de almacenamiento informático. El medio de almacenamiento puede ser cualquier medio disponible al que pueda accederse mediante un ordenador. A modo de ejemplo, y no de limitación, tal medio legible por ordenador puede comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que pueda usarse para llevar o almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Disco (*disk*) y disco (*disc*), tal como se usan en el presente documento, incluyen discos compactos (CD), láserdiscs, discos óptico, discos versátil digitales (DVD), discos flexible y discos Blu-ray en los que los discos (*disks*) reproducen normalmente datos de forma magnética, mientras que los discos (*discs*) reproducen datos de forma óptica con láseres. Deberían incluirse también combinaciones de los anteriormente mencionados dentro del alcance de un medio legible por ordenador. Cualquier procesador y el medio de almacenamiento pueden residir en un ASIC. El ASIC puede residir en un terminal de usuario. Como alternativa, el procesador y el medio de almacenamiento pueden residir como componentes discretos en un terminal de usuario.

Como se usa en el presente documento, los productos de programa informático que comprenden medios legibles por ordenador incluyen todas las formas de medio legible por ordenador excepto en la medida en que dicho medio se considere que no son señales de propagación transitorias no establecidas.

El alcance de la presente invención se define en las reivindicaciones adjuntas.

**REIVINDICACIONES**

1. Método implementado por ordenador para generación de contraseñas, en el que un usuario (100) accede mediante un primer dispositivo de computación (200) por primera vez a una página o sitio web identificado por un dominio web que requiere la identificación, mediante la utilización de credenciales incluyendo un identificador de usuario y una contraseña, de dicho usuario (100) en dicha página o sitio web, comprendiendo el método la generación de dicha contraseña requerida por la página o sitio web mediante la realización de las siguientes etapas:

- interceptar, por dicho primer dispositivo de computación (200), cuando el usuario (100) accede a dicha página o sitio web;

- recuperar, por el primer dispositivo de computación (200), información referente a dicho dominio web (207) y a dicho identificador de usuario (206);

- generar, por el primer dispositivo de computación (200), una firma en local que identifica el dominio que posee el recurso solicitado, o *Domain Signature* local (210);

- comprobar, por el primer dispositivo de computación (200), si ya existe en su base de datos (205) un registro para dicho dominio web (207) y para dicho identificador de usuario (206) y comprobar si un valor criptográfico, o Id_Hash, (208) es distinto de cero;

- enviar, por el primer dispositivo de computación (200), si el resultado de dicha comprobación ha sido falso, información referente al dominio web y al identificador de usuario a un servidor remoto (300), para que este último genere una *Domain Signature* para ese dominio web;

- recibir, por el primer dispositivo de computación (200), de dicho servidor remoto (300), la *Domain Signature* generada y generar un valor aleatorio para dicho valor criptográfico, o Id_Hash (208); y

- comprobar, por el primer dispositivo de computación (200), que la *Domain Signature* recibida del servidor remoto (300) coincide con la *Domain Signature* local (210) previamente generada y, en caso de coincidir, generar un input necesario para que un módulo generador de contraseñas (302), que se incluye en el servidor remoto (300), genere la contraseña requerida por la página o servicio web mediante la realización de las siguientes acciones:

    i. solicitar, por el primer dispositivo de computación (200), al usuario (100) la introducción en el primer dispositivo de computación (200) de una contraseña maestra, o Master-Pass (101);

    ii. ejecutar, por el primer dispositivo de computación (200), una función matemática criptográfica sobre el identificador de usuario (206), el identificador del dominio web (207), el Id_Hash (208), la *Domain Signature* local (210) y la Master-Pass (101) solicitada, dando como resultado un valor encriptado que se concatena con unas políticas de contraseña relacionadas con el citado dominio (209);

    iii. enviar, por el primer dispositivo de computación (200), la información resultado de dicha concatenación a dicho módulo generador de contraseñas para que este último pueda generar la contraseña requerida por la página o sitio web implementando un algoritmo de computación que genera un valor Meta-Pass de la combinación de la Master-Pass (101) y el Id_Hash (208), y que mapea bits del valor Meta-Pass con los correspondientes caracteres asociados con las políticas de contraseña relacionadas con el citado dominio (209); y

    iv. recibir, por el primer dispositivo de computación (200), la contraseña generada por el módulo generador de contraseñas (302) y directamente, sin almacenarla, insertarla en la página o sitio web;

- acceder, por el usuario (100), a la página o sitio web con un segundo dispositivo de computación (200),

en el que dicho segundo dispositivo de computación (200) está sincronizado con el primer dispositivo de computación (200) a través de la introducción de una contraseña de un solo uso, u OTP, en el segundo dispositivo de computación (200), habiendo sido dicha OTP previamente generada por el servidor remoto (300);

- enviar, por el segundo dispositivo de computación (200), la OTP al servidor remoto (300) para que este último pueda realizar su validación;

- recibir, por el segundo dispositivo de computación (200), si el resultado de dicha validación ha sido correcto,

un mensaje encriptado del servidor remoto (300),

en el que dicho mensaje encriptado incluya dicha OTP y unos datos asociados a la página o sitio web encriptados mediante la utilización de la Master-Pass (101), y

en el que dichos datos incluyen el identificador de usuario (206), el identificador del dominio web (207) y el Id_Hash (208); y

- solicitar, por el segundo dispositivo de computación (200), al usuario (100) la introducción de la Master-Pass (101) para descifrar el contenido de dicho mensaje encriptado.

2. Método según la reivindicación 1, en el que las políticas de contraseña relacionadas con el citado dominio (209) incluyen una longitud mínima que la contraseña generada debe tener en relación con el número de letras mayúsculas y minúsculas y/o al número de dígitos y/o al número de caracteres no alfanuméricos.

3. Método según la reivindicación 1 o 2, que comprende recuperar las políticas de contraseña relacionadas con el citado dominio (209) directamente del propio dominio.

4. Método según la reivindicación 1 o 2, en el que las políticas de contraseña relacionadas con el citado dominio (209) son definidas por el propio usuario (100).

5. Método según la reivindicación 1 o 2, en el que las políticas de contraseña relacionadas con el citado dominio (209) están configuradas por el primer dispositivo de computación (200) y almacenadas en la base de datos (205).

6. Método según la reivindicación 1, que comprende además:

- incluir caracteres aleatorios sobre un resultado obtenido por dicho mapeo para obtener una contraseña que cumpla con una longitud predeterminada de acuerdo con dichas políticas de contraseña relativas al citado dominio (209); y

- mezclar los caracteres que forman parte de dicha contraseña de una longitud predeterminada.

7. Método según la reivindicación 1, en el que dicha etapa iv) comprende además almacenar, por el primer dispositivo de computación (200), en la base de datos (205) y asociado al identificador de usuario (206) y al identificador del dominio web (207), el Id_Hash (208) asociado a las credenciales del usuario (100) para la página o sitio web.

8. Método según la reivindicación 7, caracterizado porque cuando el usuario (100), mediante el primer dispositivo de computación (200), accede por segunda vez a la página o sitio web, el primer dispositivo de computación (200) comprende:

- interceptar dicho segundo acceso a la página o sitio web y recuperar la información referente al dominio web (207) y al identificador de usuario (206);

- generar una nueva *Domain Signature* local (210) para el dominio web y comparar dicha *Domain Signature* local (210) generada con la *Domain Signature* local (210) previamente generada;

- comprobar si ya existe en la base de datos (205) un registro para dicho dominio web (207) y para dicho identificador de usuario (206) y comprobar también si el Id_Hash (208) es distinto de cero;

- si el resultado de dicha comprobación ha sido verdadero, solicitar al usuario (100) la introducción en el primer dispositivo de computación (200) de la Master-Pass (101);

- ejecutar una función matemática criptográfica sobre el identificador de usuario (206), el identificador del dominio web (207), el Id_Hash (208), la *Domain Signature* local (210) y la Master-Pass (101) solicitada, dando como resultado un valor encriptado que se concatena con las políticas de contraseña relativas al citado dominio (209); y

- enviar la información resultado de dicha concatenación al módulo generador de contraseñas (302) para que este último genere la contraseña requerida por la página o sitio web.

9. Producto de programa informático que incluye instrucciones de código que, cuando se implementan en un sistema de computación, implementan un método según una cualquiera de las reivindicaciones 1 a 8.

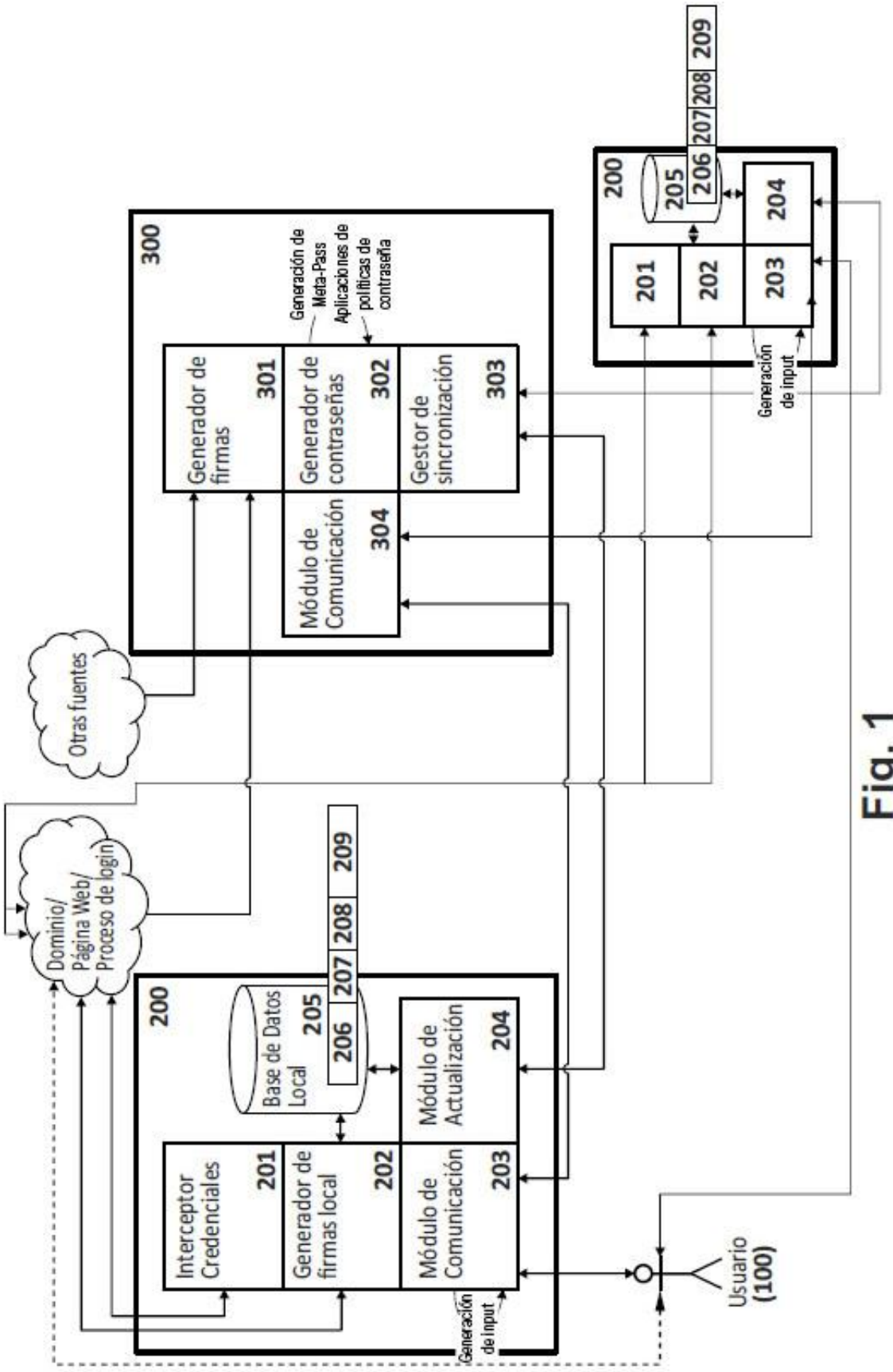
Usuario
(100)
Generación de input
Interceptor Credenciales
201
Generador de firmas local
202
Módulo de Comunicación
203
Módulo de Actualización
204
Base de Datos Local
205
206
207
208
209
200
Dominio/ Página Web/ Proceso de login
Otras fuentes
300
Generador de firmas
301
Generador de contraseñas
302
Gestor de sincronización
303
Módulo de Comunicación
304
Generación de Meta-Pass Aplicaciones de políticas de contraseña
Generación de input

Fig. 1


Generación de Meta-Pass
Mapeo para cumplir con unos mínimos de unas políticas de contraseña
Mapeo aleatorio para completar una longitud mínima de la contraseña
Barajeo

**Fig. 2**

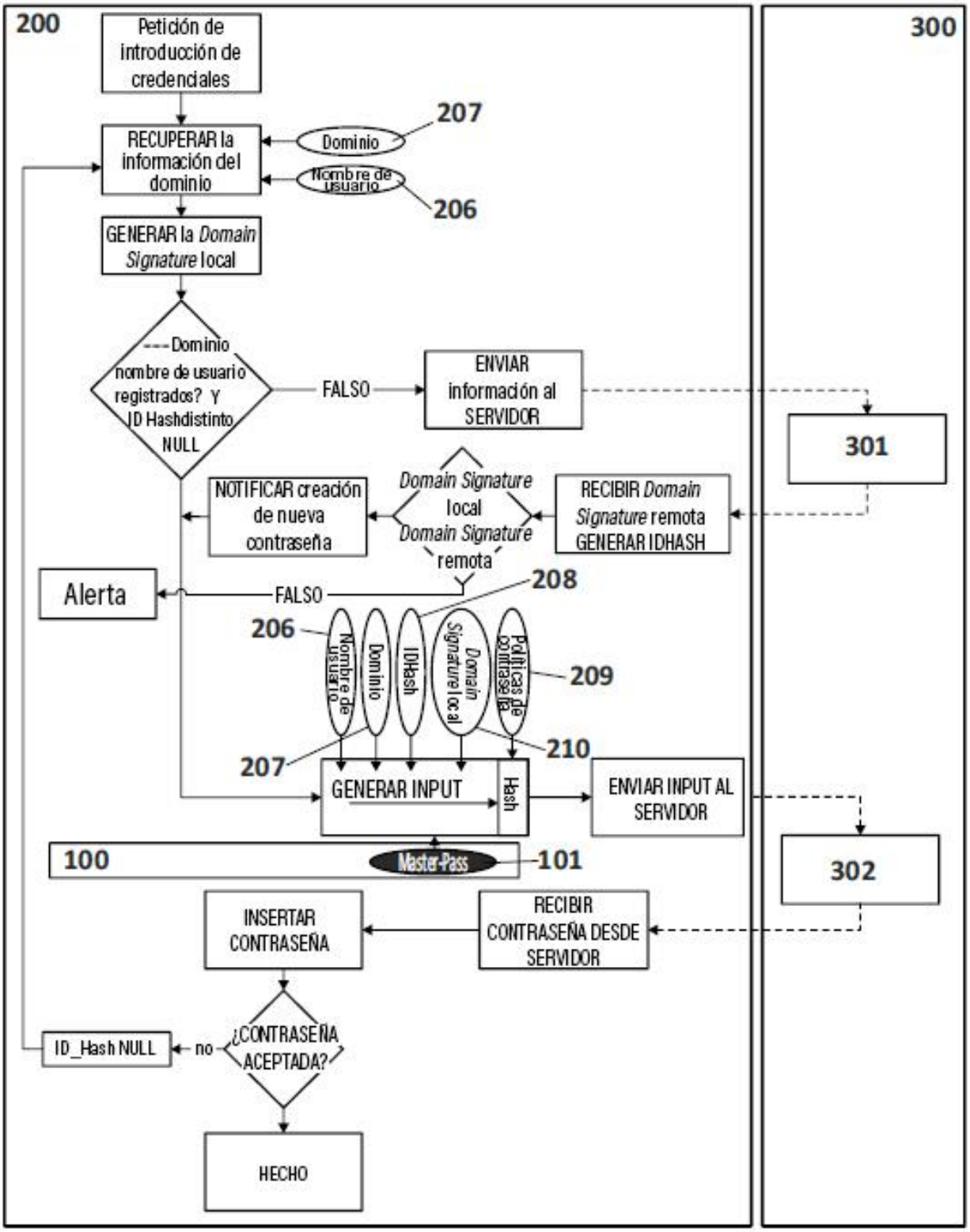
200
300
Petición de introducción de credenciales
RECUPERAR la información del dominio
Dominio
Nombre de usuario
207
206
GENERAR la Domain Signature local
---Dominio nombre de usuario registrados? Y ID Hashdistinto NULL
FALSO
ENVIAR información al SERVIDOR
301
Domain Signature local
Domain Signature remota
NOTIFICAR creación de nueva contraseña
RECIBIR Domain Signature remota GENERAR IDHASH
Alerta
FALSO
208
206
Nombre de usuario
Dominio
IDHash
Domain Signature local
Políticas de contraseña
209
210
207
GENERAR INPUT
Hash
ENVIAR INPUT AL SERVIDOR
100
Master-Pass
101
302
INSERTAR CONTRASEÑA
RECIBIR CONTRASEÑA DESDE SERVIDOR
ID_Hash NULL
no
¿CONTRASEÑA ACEPTADA?
HECHO

Fig. 3

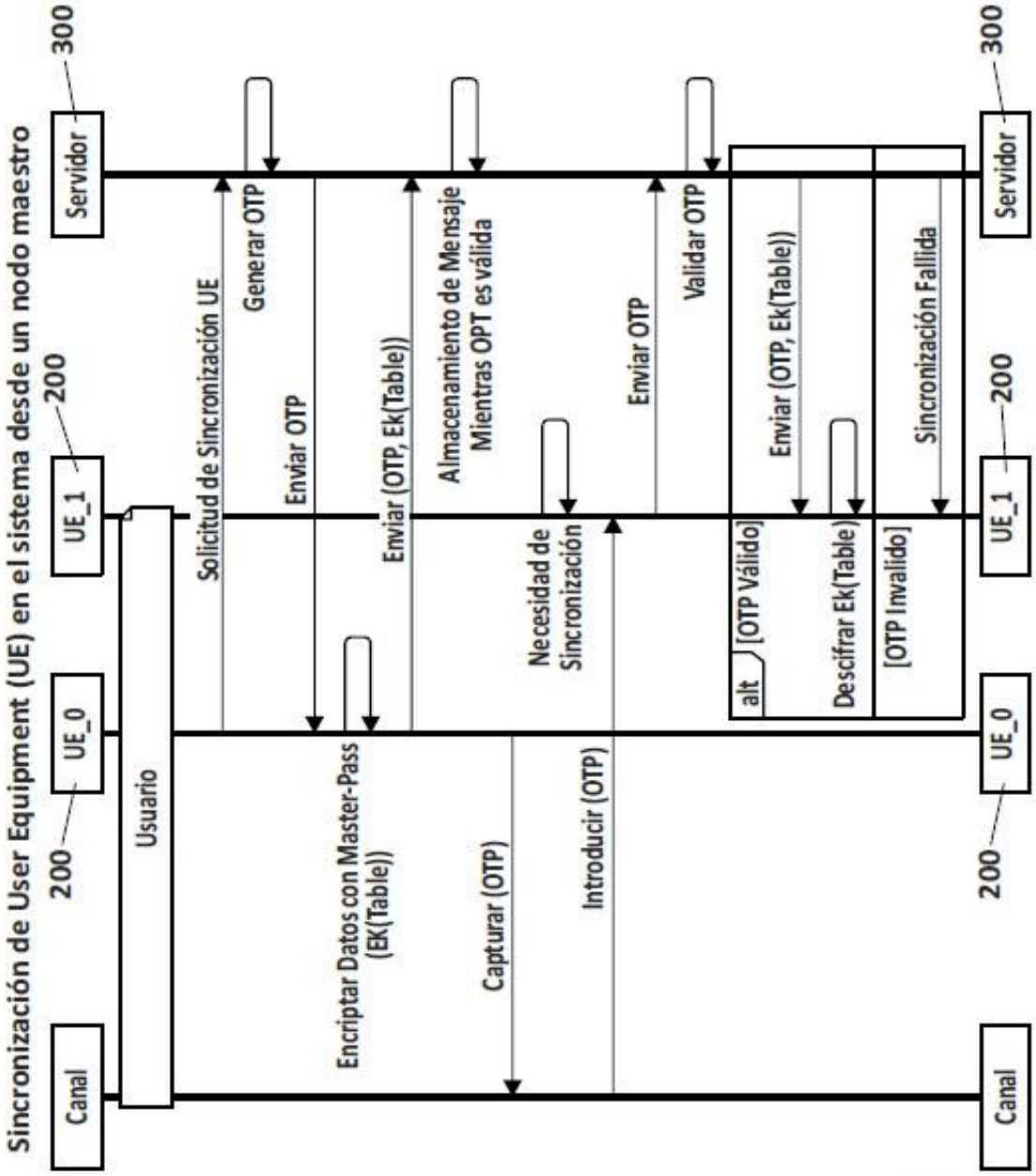
Sincronización de User Equipment (UE) en el sistema desde un nodo maestro
Servidor
300
UE_1
200
UE_0
200
Usuario
Canal
Solicitud de Sincronización UE
Generar OTP
Enviar OTP
Enviar (OTP, Ek(Table))
Almacenamiento de Mensaje Mientras OPT es válida
Enviar OTP
Validar OTP
Enviar (OTP, Ek(Table))
Sincronización Fallida
alt
[OTP Válido]
Descifrar Ek(Table)
[OTP Invalido]
Necesidad de Sincronización
Encriptar Datos con Master-Pass (EK(Table))
Capturar (OTP)
Introducir (OTP)

Fig. 4