(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 819 200**

(51) Int. CI.:

H04W 12/00 (2009.01)
G06F 21/35 (2013.01)
H04L 29/06 (2006.01)
H04L 9/32 (2006.01)
H04W 12/08 (2009.01)
H04W 12/06 (2009.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **27.11.2013 PCT/GB2013/053138**

(87) Fecha y número de publicación internacional: **05.06.2014 WO14083335**

(96) Fecha de presentación y número de la solicitud europea: **27.11.2013 E 13801699 (3)**

(97) Fecha y número de publicación de la concesión europea: **17.06.2020 EP 2926290**

(54) Título: **Un método y sistema para proporcionar autenticación del acceso del usuario a un recurso informático a través de un dispositivo móvil utilizando múltiples factores de seguridad separados**

(30) Prioridad:

28.11.2012 GB 201221433
05.12.2012 US 201213706307
01.03.2013 GB 201303677

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
15.04.2021

(73) Titular/es:

HOVERKEY LTD (100.0%)
Upper House, Spring Hill, Nailsworth
Gloucestershire, GL6 0LX, GB

(72) Inventor/es:

YAU, ARNOLD

(74) Agente/Representante:

ISERN JARA, Jorge

DESCRIPCIÓN

Un método y sistema para proporcionar autenticación del acceso del usuario a un recurso informático a través de un dispositivo móvil utilizando múltiples factores de seguridad separados

1. Introducción

La presente solicitud se refiere a un método y sistema para autenticar a un usuario en un recurso informático al que se accede a través de un dispositivo móvil utilizando un token de seguridad portátil (por ejemplo, una tarjeta inteligente sin contacto o una pulsera), junto con un código que el usuario puede fácilmente recordar (por ejemplo, un código PIN). Este código proporciona un segundo factor de seguridad separado, preferiblemente independiente, que puede salvaguardar el recurso del ordenador incluso si el token de seguridad portátil y el dispositivo móvil se pierden o son robados juntos. Una realización preferida se refiere a proporcionar protección de datos y acceso seguro a aplicaciones y datos almacenados a los que se accede a través de un dispositivo móvil (como un teléfono o tableta) utilizando un token de hardware de comunicación de campo cercano (NFC) o un token de Bluetooth de corto alcance.

La autenticación segura de un usuario a través de un dispositivo móvil se está volviendo importante en dos situaciones diferentes: en primer lugar, para la autenticación del acceso del usuario a un recurso informático en el dispositivo móvil y, en segundo lugar, en un servidor remoto.

La mayoría de los sistemas existentes emplean el uso de una contraseña simple o PIN para autenticar al usuario. A pesar de la ubicuidad de los sistemas basados en contraseñas, tiene muchos problemas. la contraseña ideal es la que el usuario recuerda con facilidad. Sin embargo, para que las contraseñas sean seguras, deben ser largas y difíciles de predecir, lo que contradice el requisito anterior. Esto se ve exacerbado por la proliferación de contraseñas para la multitud de aplicaciones que un usuario usa normalmente, para las cuales las mejores prácticas de seguridad recomiendan que se utilicen diferentes contraseñas.

Además del acceso a las aplicaciones, algunos usuarios móviles desean garantizar un alto nivel de seguridad para los datos (incluidos archivos completos y datos contenidos en un archivo o una estructura de datos) en su dispositivo, frente a una serie de escenarios de amenazas externas. Por ejemplo, un usuario puede usar una aplicación en una tableta u otro dispositivo portátil que sincronice archivos con su P.C. de escritorio a través de un servicio de almacenamiento en línea (por ejemplo, Dropbox, Box.com [*marcas comerciales*]). Algunos de los archivos descargados pueden contener información confidencial, como documentos comerciales. El usuario desea protegerse contra la posibilidad de una violación de datos en caso de robo del dispositivo.

Una forma práctica de lograr esto hoy en día es habilitar el cifrado del dispositivo en el sistema operativo móvil, que utiliza una clave de cifrado derivada de la contraseña de la pantalla de bloqueo del dispositivo. Para máxima seguridad, esta contraseña debe ser larga y compleja. Sin embargo, usar una contraseña larga y compleja como contraseña para desbloquear la pantalla de bloqueo es extremadamente inconveniente para el usuario.

Debido a esto, la mayoría de los usuarios son reacios a usar una contraseña más complicada que un código PIN de 4 dígitos para desbloquear la pantalla de bloqueo. Un atacante experto podrá descifrar cualquier archivo almacenado en un dispositivo robado con métodos de ataque de fuerza bruta. Además, los datos confidenciales se descifran cada vez que se desbloquea el dispositivo, incluso cuando el usuario no está utilizando los datos, lo que aumenta el riesgo de una violación de datos de forma innecesaria.

Otro enfoque posible para el cifrado de datos es que la aplicación genere su propia clave de cifrado. El problema con este enfoque es que la clave tendría que estar protegida por una contraseña o derivada de ella por motivos de seguridad, o tendría que almacenarse dentro de la aplicación en forma de texto sin formato para su usabilidad. El primer enfoque hereda el mismo problema de complejidad de contraseña que el método de cifrado del dispositivo anterior, mientras que el segundo ofrece poca seguridad, ya que el atacante que podría comprometer los datos de texto sin formato podría leer fácilmente la clave de texto sin formato y descifrar los datos. Una forma de proporcionar un nivel adicional de seguridad a los usuarios de dispositivos móviles es exigir que el usuario también lleve un token físico portátil que se comunique con el dispositivo utilizando un sistema de comunicación inalámbrico, por ejemplo, Bluetooth o Bluetooth de baja energía (BLE). El dispositivo móvil comprueba de manera constante la presencia del token. Este token, cuando está presente dentro de un rango de varios metros del dispositivo móvil, verifica constantemente que el usuario está en verdad presente. Cuando el usuario abandona el token y el dispositivo pierde contacto, el dispositivo se protege contra cualquier acceso hasta que se recupere la comunicación con el token.

Nicholson, Corner and Noble describen un ejemplo de tal sistema en IEEE Transactions on Mobile Computing, Vol. 5 No 11 de noviembre de 2006, “Mobile Device Security Using Transient Authentication”. Hay una serie de desventajas de tal sistema. El canal de comunicaciones basado en difusión entre el token y el dispositivo móvil está sujeto a la escucha de un atacante que está dentro del rango cercano del token y el dispositivo. A pesar de estar cifrado, debido a los numerosos eventos de autenticación transitorios que tienen lugar entre el token y el dispositivo, el atacante tiene

muchas oportunidades para criptoanalizar los mensajes de autenticación, así como para realizar análisis de tráfico sin siquiera tener que intentar un ataque criptoanalítico.

Un ladrón que roba el dispositivo móvil pero aún permanece dentro del alcance del token de seguridad que lleva el propietario del dispositivo podrá acceder a los recursos del dispositivo. El robo del dispositivo móvil y el token juntos inutiliza el sistema de seguridad.

En algunos otros sistemas existentes se ha proporcionado un nivel adicional de seguridad al exigir que un teléfono móvil con capacidad NFC o Bluetooth se autentique primero en la red móvil antes de ejecutar una aplicación. Luego, un token NFC/Bluetooth proporciona una clave asimétrica al teléfono que, a su vez, se autentica en un servicio de terceros al realizar una firma digital dentro del propio teléfono.

En el documento US-A-2011/0212707, se muestra un ejemplo genérico de tal sistema. Sin embargo, esto presenta una serie de desventajas. En particular, el cambio de la credencial de la aplicación requiere la reprogramación o el reemplazo del token; el número de credenciales de usuario aseguradas por el sistema está limitado por la (pequeña) capacidad de almacenamiento del token; y la pérdida del token plantea un riesgo directo de exposición de las credenciales del usuario. Además, las aplicaciones que se ejecutan en el dispositivo móvil y el servidor son capaces de hacer uso del sistema de seguridad descrito solo si se han programado específicamente para hacerlo. El sistema descrito no se puede utilizar con aplicaciones preexistentes.

Otro enfoque para la identificación de múltiples factores se describe en el documento US-A-2008/0289030. Aquí, un token sin contacto, tras la validación, se usa para permitir el acceso a las credenciales de autenticación aseguradas en el dispositivo móvil.

Esto tiene una serie de desventajas graves, incluida la necesidad de utilizar un almacenamiento seguro en el dispositivo. Normalmente, esto no está disponible para los desarrolladores de aplicaciones, ya que es mantenido y controlado por el fabricante del dispositivo (por ejemplo, teléfono móvil) o el proveedor del sistema operativo subyacente o un operador de red móvil. Además, es probable que no sea seguro hacer uso únicamente de un identificador de token como un medio para validar el token. Los token RFID normalmente se pueden leer con cualquier lector compatible y se pueden clonar con facilidad.

En el documento WO-A-2011/089423, se describe otro enfoque más. Describe un sistema en el que se utiliza la presencia de un token sin contacto para autorizar la ejecución de una función o aplicación segura, y está dirigido principalmente a usos de billetera móvil.

De nuevo, el sistema descrito tiene una serie de desventajas, principalmente que utiliza una forma de control lógico que es relativamente fácil de eludir.

De manera más general, en el entorno empresarial, existe un riesgo de seguridad significativo al permitir a los usuarios conectar dispositivos móviles a la red debido a una mayor probabilidad de acceso no autorizado a los datos (lo que lleva a la pérdida de la confidencialidad y/o integridad de los datos) como resultado de:

- códigos de acceso revelados inadvertidamente tales como PIN o códigos alfanuméricos, por ejemplo, de la navegación lateral,
- códigos de acceso fáciles de adivinar,
- dispositivos perdidos o robados que no están adecuadamente protegidos,
- uso no supervisado de dispositivos por un tercero,
- el sistema Hoverkey tiene como objetivo proporcionar soluciones para aplicaciones para contrarrestar estas amenazas.

Con las realizaciones de la presente divulgación, el usuario puede almacenar una clave maestra de alta capacidad criptográfica (128 bits o más en la actualidad) en el token de seguridad portátil, y esta clave puede usarse para proteger directamente la clave de cifrado de datos de una aplicación o una contraseña larga y compleja, de la que se puede derivar una clave de cifrado suficientemente larga y segura. Esto permite al usuario proteger cualquier dato almacenado en el dispositivo con una clave de cifrado muy fuerte. Si el dispositivo es robado, no es factible que cualquier atacante potencial descifre los datos cifrados en él sin el token asociado.

Las credenciales pueden almacenarse en el dispositivo móvil o, en forma remota, en la nube. El almacenamiento en la nube preferiblemente tiene las siguientes características:

- Las credenciales protegidas siempre se almacenan en la nube y se recuperan de la nube antes de su uso.

- El almacenamiento en caché local transparente es posible, pero no pretende ser un almacenamiento permanente (debe borrarse después de un período de tiempo de espera especificado).

• Si se pierde el dispositivo o el token, las credenciales pueden eliminarse con solo eliminar los archivos relevantes del servicio de almacenamiento en la nube para evitar un posible uso indebido.

• La sincronización de credenciales es posible entre dispositivos para el mismo usuario, lo que evita la necesidad de ingresar manualmente las mismas credenciales varias veces.

Menezes et al. divulgan en el “Handbook of Applied Cryptography”, 1997, CRC Press LLC, EE. UU., una reseña general de la criptografía como un estudio de técnicas matemáticas relacionadas con aspectos de la seguridad de la información tales como la confidencialidad, la integridad de los datos, la autenticación de entidades, y autenticación del origen de los datos.

El documento WO 2012/103584 A1 divulga la provisión de un token para autenticar dinámicamente a un usuario. El token incluye una memoria para almacenar datos seguros; un procesador para calcular las credenciales de autenticación del usuario basándose en los datos seguros y para construir una dirección de servidor basada en las credenciales de autenticación.

El documento GB 2423396 A revela un identificador que se lee de un token (por ejemplo, tarjeta inteligente, tarjeta SIM) y se utiliza para ubicar un registro en un almacén de datos (por ejemplo, un servidor remoto) que contiene datos de autenticación del usuario (por ejemplo, nombre de usuario, contraseña o máscara de bits). Los datos de autenticación del usuario se utilizan para autenticar al usuario. El acceso al token puede estar limitado por el uso de un PIN.

2. Antecedentes

2.1 Realizaciones y características preferibles de las mismas

La invención se define en las reivindicaciones independientes. En las reivindicaciones dependientes, se establecen realizaciones particulares. Algunas de las realizaciones descritas a continuación no son realizaciones de la invención según las reivindicaciones.

De acuerdo con la presente divulgación, se proporciona un método y sistema para autenticar el acceso a los recursos informáticos en un dispositivo móvil.

Según un primer aspecto, un método de autenticación de un recurso informático en un dispositivo móvil comprende:

almacenar una autorización de recurso cifrada;
transmitir la autorización cifrada a un token de seguridad portátil separado; en el token, descifrar la autorización cifrada y generar, al menos parcialmente, una respuesta de desbloqueo;
transmitir en forma segura la respuesta de desbloqueo al dispositivo móvil;
requerir que un usuario se autentique por separado en el dispositivo móvil; y
desbloquear el recurso si la respuesta de desbloqueo requerida y la autenticación separada son válidas.

La respuesta de desbloqueo puede comprender una autorización simple, obtenida descifrando la autorización cifrada.

La respuesta de desbloqueo puede comprender alternativamente una función (como un hash) de una autorización simple, obtenida descifrando la autorización cifrada, e información adicional. Por lo tanto, en un modo de uso, el token puede verificar y descifrar la autorización cifrada. Luego, en lugar de devolver una autorización simple al dispositivo, protegido por una sesión u otra clave de cifrado, el token puede realizar algunos cálculos sobre la autorización simple y posiblemente alguna otra información (por ejemplo, información basada en token), y devolver el resultado al dispositivo. Los ejemplos incluyen lo siguiente:

• Ejemplo 1: Firma digital: cálculo = función de firma digital, autorización simple = clave de firma privada; parámetro = hash del mensaje; salida = firma digital en el hash del mensaje
• Ejemplo 2: Derivación de clave: cálculo = función de derivación de clave; autorización simple = código maestro de derivación de claves; parámetros = información de contexto, longitud de salida; salida = clave derivada del código maestro
• Ejemplo 3: Reencriptación: cálculo = función de cifrado; autorización simple = clave de cifrado; parámetro = (otra) clave de cifrado; salida = la autorización simple cifrada con una clave diferente

La autorización puede comprender una contraseña, un PIN o una clave criptográfica.

La respuesta de desbloqueo puede transmitirse al dispositivo móvil bajo la protección de una clave de cifrado, como una clave de sesión.

El token puede almacenar credenciales de propiedad de usuario/token, y el descifrado del token se basa en las credenciales del usuario.

El método proporciona autenticación de dos factores (o de múltiples factores) al requerir que un usuario se autentique por separado en el dispositivo móvil, por ejemplo, mediante la validación de la autenticación en el dispositivo móvil en el token antes de enviar el código de desbloqueo. Preferiblemente, el método requiere una prueba de conocimiento del dispositivo (y en última instancia del usuario) antes de descifrar la autorización. Tal prueba de conocimiento puede incluir un PIN; una contraseña de muchos caracteres; un gesto de movimiento detectado por una cámara en el dispositivo; un gesto detectado por un panel táctil en el dispositivo, por ejemplo, un gesto que comprende patrones de deslizamiento que se ven comúnmente para desbloquear un teléfono con pantalla bloqueada; una secuencia de movimientos detectada por un giroscopio y/o acelerómetro del dispositivo; una contraseña hablada u otro sonido o secuencia de sonidos emitidos por el usuario; un patrón de toques realizados por el usuario en una carcasa del dispositivo, por ejemplo, que puede ser detectado por un micrófono del dispositivo; u otra prueba adecuada de conocimiento como será evidente para los expertos en la técnica.

En algunas realizaciones, un usuario puede autenticarse mediante información biométrica, por ejemplo, una huella digital o un escaneo del iris del usuario. Ventajosamente, el uso de una contraseña biométrica, como una huella dactilar, protege contra las escuchas de personas cercanas. También tiene la ventaja de que un usuario no tiene que recordar una contraseña o código.

La prueba podrá presentarse tras la autenticación mutua. Alternativamente, la autenticación del dispositivo puede ser completamente independiente de la autenticación del token.

Se puede ejecutar un servicio en el dispositivo móvil que controla las funciones criptográficas del dispositivo y el acceso al recurso. Se puede ejecutar un Applet en el token que proporciona funciones criptográficas del token.

Las credenciales de usuario pueden ser generadas por el token y nunca dejar el token (o la aplicación ejecutándose en el token).

En algunas realizaciones, las credenciales de usuario se almacenan en una ubicación segura dentro del token, por ejemplo, donde la ubicación es proporcionada por hardware y/o software.

Preferiblemente, la autorización cifrada almacenada en el dispositivo móvil se puede descifrar solo con las correspondientes credenciales de usuario almacenadas en el token.

El método puede incluir verificar la integridad en el token mediante un código de autenticación de mensaje (MAC) recibido del dispositivo.

El método puede incluir la verificación de la integridad de la autorización cifrada en el token antes del descifrado.

El dispositivo y el token pueden realizar una autenticación mutua criptográfica antes de la transmisión de la autorización cifrada.

El cifrado, el descifrado y/o la autenticación mutua pueden proporcionarse mediante criptografía de clave simétrica. En algunas realizaciones, la autenticación mutua puede proporcionarse mediante autenticación asimétrica. En tales realizaciones, la autenticación se puede realizar en ambas direcciones a través de dos pares de claves, un par de claves para cada dirección, de modo que cada uno de los dispositivos y token pueda identificar y autenticar al otro. En tales realizaciones, cada par de claves comprende una clave privada y una clave pública. Alternativamente, la autenticación puede realizarse cuando solo uno de los dispositivos está autenticado, por ejemplo, cuando algún otro aspecto del canal de comunicación proporciona la seguridad/autenticación adicional.

Un código de usuario puede pasarse del dispositivo al token y puede ser validado por el token antes de que tenga lugar la operación de descifrado.

El recurso puede comprender datos o una aplicación que se ejecuta o se almacena en el dispositivo móvil.

De acuerdo con otro aspecto se proporciona:

un dispositivo móvil;
un token que incluye almacenamiento de token para almacenar credenciales de usuario privadas, un sistema de comunicaciones de token y un procesador de token que proporciona funciones criptográficas;
y en el que, en uso, el sistema de comunicaciones del dispositivo transmite una autorización cifrada al token; se descifra en el token utilizando las credenciales de usuario; en donde el token genera, al menos parcialmente, una respuesta de desbloqueo, siendo transmitida en forma segura la respuesta de desbloqueo por el sistema de

comunicaciones del token al dispositivo móvil; requerir que un usuario se autentique por separado en el dispositivo móvil; y desbloquear el recurso si la respuesta de desbloqueo requerida y la autenticación separada son válidas.

El sistema de comunicaciones del dispositivo y el sistema de comunicaciones del token pueden comunicarse por aire, por ejemplo, mediante comunicación de campo cercano (NFC), Bluetooth o BLE. Alternativamente, el sistema de comunicaciones del dispositivo y el sistema de comunicaciones del token pueden comunicarse solo cuando el token está en contacto con el dispositivo a través de una interfaz física.

El sistema de comunicaciones del dispositivo puede enviar un código de usuario al token que es validado por el token antes de que tenga lugar la operación de descifrado.

El sistema de comunicaciones del dispositivo puede enviar un código de autenticación de mensaje (MAC) al token, que es validado por el token antes de que tenga lugar la operación de descifrado.

De acuerdo con otro aspecto, se proporciona:

un token de hardware para autenticar el acceso a un recurso informático a través de un dispositivo móvil, en donde el token comprende:

almacenamiento de token para el almacenamiento de una pluralidad de credenciales de usuario; un sistema de comunicaciones de token para comunicarse con un dispositivo móvil; un procesador de token que proporciona funciones criptográficas; y donde, en uso:

al recibir el sistema de comunicaciones de token una autorización cifrada, el procesador de token verifica la integridad y descifra la autorización cifrada y genera, al menos parcialmente, una respuesta de desbloqueo, y en el que el sistema de comunicaciones de token transmite en forma segura la respuesta de desbloqueo para su uso por un dispositivo móvil.

El dispositivo móvil puede comprender cualquier dispositivo de hardware móvil o portátil que sea capaz de ejecutar aplicaciones de usuario y manejar funciones de comunicación y criptográficas. Los dispositivos típicos incluyen teléfonos móviles, tabletas, ordenadores portátiles, teléfonos inteligentes, relojes inteligentes, gafas inteligentes, y similares. El token puede ser cualquier dispositivo portátil o token de hardware móvil que sea capaz de comunicarse (preferiblemente comunicación sin contacto) con un dispositivo móvil y que incluya almacenamiento y un sistema ejecutable que sea capaz de manejar comunicaciones y funciones criptográficas. Por ejemplo, el token puede ser un dispositivo móvil o portátil, como un teléfono inteligente, reloj inteligente, gafas inteligentes, teléfono móvil, tableta, ordenador portátil, o similar. En algunas realizaciones, el token y el dispositivo móvil son ambos teléfonos inteligentes móviles. En algunas realizaciones, el token puede comprender un llavero, una tarjeta inteligente NFC o un auricular inalámbrico.

El recurso informático protegido puede guardarse en la memoria o almacenamiento de un dispositivo o (donde una aplicación) puede mantenerse lista para su ejecución o puede estar ejecutándose en un entorno de ejecución. Con ese fin, el dispositivo puede incluir un almacenamiento, una memoria y un procesador.

En algunas realizaciones, el token será una tarjeta inteligente sin contacto, aunque serían igualmente posibles otros token que la persona sostenga, porte o lleve puesta. Los token adecuados pueden incluir un dispositivo portátil, como un anillo para llevar en el dedo del usuario, otra joya que lleve el usuario, un dispositivo incorporado en un reloj, cinturón, gafas, ropa, insignia, pulsera de fitness o cualquier otra cosa que lleve normalmente el usuario, o incluso un dispositivo incrustado debajo de la piel del usuario. En algunas realizaciones, el token puede coserse o fijarse a una prenda de vestir u otro accesorio que lleve el usuario. Las formas de realización en las que el usuario lleva el token son ventajosas, ya que es más probable que el usuario tenga el token portátil en su persona cuando sea necesario, y es menos probable que el usuario pierda el token que cuando el token es un artículo separado que debe llevarse, por ejemplo, en un bolso o billetera.

El token puede tener botón(es), área(s) sensible(s) al tacto u otros medios para permitir la retroalimentación/entrada manual o de otro usuario a través del token.

En algunas realizaciones, la clave maestra se almacena en el token en hardware y/o software dedicado a la función de proporcionar almacenamiento seguro. En algunas realizaciones, la clave maestra se almacena en el token en zonas seguras dedicadas proporcionadas por la arquitectura del hardware y/o software del dispositivo, por ejemplo, los dispositivos pueden utilizar tecnología ARMs TrustZone, una UICC (comúnmente conocida como tarjeta SIM) o un elemento seguro integrado. En algunas realizaciones, la propia clave podría almacenarse en un llavero seguro implementado por el sistema operativo y/u otro software presente en el dispositivo.

En algunas realizaciones, el token puede comprender un entorno de hardware y/o software seguro y confiable para el almacenamiento de la clave maestra y la ejecución de operaciones de cifrado y descifrado. Por ejemplo, el entorno

seguro puede comprender un elemento seguro, un módulo de plataforma fiable, un entorno de cifrado fiable o cualquier otro entorno adecuado como resultará evidente para los expertos en la técnica. Ventajosamente, muchos dispositivos inteligentes comprenden tales entornos, por ejemplo, Samsung Galaxy S4.

El token portátil puede ser un dispositivo dedicado con el propósito de cifrar/descifrar credenciales de autenticación en forma segura, o puede ser un dispositivo que, además de otras funciones, posiblemente primarias, también proporciona cifrado/descifrado seguro de credenciales de autenticación.

En el caso de que el token comprenda un dispositivo que tenga otras funciones, este puede ser un dispositivo existente en posesión del usuario (por ejemplo, un teléfono inteligente) y, por lo tanto, tiene la ventaja de que el usuario no necesita recordar o modificar su comportamiento para llevar un token o dispositivo adicional.

La autenticación de la aplicación almacenada en el dispositivo puede comprender una contraseña o PIN de la aplicación. Las credenciales de usuario almacenadas en el token pueden comprender una clave criptográfica privada.

Se prefiere que la comunicación entre el token y el dispositivo móvil utilice NFC, aunque también se podrían utilizar otros canales, incluidos Bluetooth, Bluetooth de baja energía (BLE) u otros tipos de comunicación por radiofrecuencia. También se prevén token que requieran contacto con el dispositivo móvil, incluidas tarjetas magnéticas y tarjetas de contacto eléctrico.

En algunas realizaciones, la comunicación entre el dispositivo móvil y el token portátil puede implementarse usando cualquier forma de comunicación que pueda ser suficientemente segura para la transmisión de credenciales de autenticación, como será evidente para los expertos en la técnica. Dicha comunicación podría incluir una red WiFi local, una conexión WiFi-Direct, un cable USB directo, una cámara frontal del dispositivo móvil y una pantalla del token (o viceversa) cuando se colocan uno frente al otro, una conexión infrarroja, frecuencias de audio usando un altavoz del dispositivo móvil y un micrófono del token (o viceversa), o cualquier otro medio de comunicación como será evidente para los expertos en la técnica.

Un sistema preferido comprende preferiblemente:

1. Uno o más dispositivos móviles
2. Un token NFC, Bluetooth o BLE programado para:

a) poder autenticarse mutuamente con cualquiera de los dispositivos del usuario
b) responder solo a los comandos emitidos por cualquiera de los dispositivos del usuario
c) realizar el cifrado y la protección de la integridad de los datos proporcionados por el dispositivo.
d) devolver los datos protegidos criptográficamente
e) realizar el descifrado y la verificación de la integridad de los datos previamente protegidos.
f) requerir la validación de un PIN de usuario antes de realizar operaciones de descifrado
3. Una aplicación de administración de contraseñas instalada en cada dispositivo móvil
4. Cualquier cantidad de aplicaciones de terceros protegidas por el sistema

De acuerdo con otro aspecto, se proporciona un método para autenticar a un usuario para que acceda a un recurso informático a través de un dispositivo móvil, que comprende:

almacenar una autorización de recursos cifrada;
transmitir la autorización cifrada a un token de seguridad portátil separado;
en el token, descifrar la autorización cifrada y generar, al menos parcialmente, una respuesta de desbloqueo; transmitir en forma segura la respuesta de desbloqueo al dispositivo móvil; y proporcionar acceso al recurso si la respuesta de desbloqueo requerida es válida;
en el que se requiere que un usuario se autentique en el dispositivo móvil y la autenticación en el dispositivo móvil se valida en el token antes de que se envíe la respuesta de desbloqueo.

De acuerdo con otro aspecto, se proporciona un sistema de autenticación de un usuario para acceder a un recurso informático a través de un dispositivo móvil con un token de seguridad portátil, que comprende:

un dispositivo móvil;
un token que incluye almacenamiento de token para almacenar credenciales de usuario privadas, un sistema de comunicaciones de token y un procesador de token que proporciona funciones criptográficas;
y en el que, en uso, el sistema de comunicaciones del dispositivo transmite una autorización cifrada al token; se descifra en el token utilizando las credenciales de usuario; en donde el token genera, al menos parcialmente, una respuesta de desbloqueo, siendo transmitida en forma segura la respuesta de desbloqueo por el sistema de comunicaciones del token al dispositivo móvil; y proporciona acceso al recurso si la respuesta de desbloqueo requerida es válida;

en el que se requiere que un usuario se autentique en el dispositivo móvil y la autenticación en el dispositivo móvil se valida en el token antes de que se envíe la respuesta de desbloqueo.

De acuerdo con otro aspecto, se proporciona un token de hardware para autenticar a un usuario para acceder a un recurso informático a través de un dispositivo móvil, que comprende:

almacenamiento de token para el almacenamiento de una pluralidad de credenciales de usuario; un sistema de comunicaciones de token para comunicarse con un dispositivo móvil; un procesador de token que proporciona funciones criptográficas; y en el que, en uso:

al recibir el sistema de comunicaciones de token una autorización cifrada, el procesador de token verifica la integridad y descifra la autorización cifrada y genera, al menos parcialmente, una respuesta de desbloqueo, en la que el sistema de comunicaciones de token transmite en forma segura la respuesta de desbloqueo para su uso por un dispositivo móvil, y en el que se requiere que un usuario se autentique en el dispositivo móvil y la autenticación en el dispositivo móvil se valida en el token antes de que se envíe la respuesta de desbloqueo.

De acuerdo con otro aspecto, se proporciona un método para autenticar a un usuario para que acceda a un recurso informático a través de un dispositivo móvil que comprende:

almacenar una autorización de recursos cifrada;
transmitir la autorización cifrada a al menos un dispositivo de token de seguridad portátil separado;
en el dispositivo token, descifrar la autorización cifrada y generar al menos parcialmente una respuesta de desbloqueo;
transmitir en forma segura la respuesta de desbloqueo al dispositivo móvil; y proporcionar acceso al recurso si la respuesta de desbloqueo requerida es válida;
en el que se requiere que un usuario se autentique en el dispositivo móvil o en el dispositivo token, y la autenticación se valida en el dispositivo token antes de que se envíe la respuesta de desbloqueo.

De acuerdo con otro aspecto, se proporciona un sistema de autenticación de un usuario para acceder a un recurso informático a través de un dispositivo móvil con al menos un dispositivo de token de seguridad portátil, que comprende:

un dispositivo móvil;
al menos un dispositivo de token que incluye almacenamiento de token para almacenar credenciales de usuario privadas, un sistema de comunicaciones de token y un procesador de token que proporciona funciones criptográficas; y en el que, en uso, el sistema de comunicaciones del dispositivo móvil transmite una autorización cifrada al dispositivo de token; se descifra en el dispositivo de token utilizando las credenciales del usuario; el dispositivo de token genera, al menos parcialmente, una respuesta de desbloqueo, siendo transmitida en forma segura la respuesta de desbloqueo por el sistema de comunicaciones de token al dispositivo móvil; y proporciona acceso al recurso si la respuesta de desbloqueo requerida es válida; en el que se requiere que un usuario se autentique en el dispositivo móvil o en el dispositivo token, y la autenticación se valida en el dispositivo token antes de que se envíe la respuesta de desbloqueo.

De acuerdo con otro aspecto, se proporciona un dispositivo de token de hardware para autenticar a un usuario para acceder a un recurso informático a través de un dispositivo móvil, en el que el dispositivo de token comprende:

almacenamiento de token para el almacenamiento de una pluralidad de credenciales de usuario; un sistema de comunicaciones de token para comunicarse con un dispositivo móvil; un procesador de token que proporciona funciones criptográficas; y donde, en uso:

al recibir el sistema de comunicaciones de token una autorización cifrada, el procesador de token verifica la integridad y descifra la autorización cifrada y genera, al menos parcialmente, una respuesta de desbloqueo, en la que el sistema de comunicaciones de token transmite en forma segura la respuesta de desbloqueo para su uso por un dispositivo móvil, y en el que se requiere que un usuario se autentique en el dispositivo móvil o en el dispositivo de token, y la autenticación se valida en el dispositivo de token antes de que se envíe la respuesta de desbloqueo.

En el caso de que un usuario se autentique en el dispositivo de token, se puede utilizar un dispositivo de token que comprenda una pantalla y/o un teclado integrado. Puede usarse una tarjeta inteligente, por ejemplo, las disponibles comercialmente como las producidas por NagraID. Alternativamente, el token puede ser un teléfono inteligente, reloj inteligente, tableta, o similar.

2.2 Nivel 1 de Hoverkey

Una realización preferida se incorpora preferiblemente dentro de un producto denominado Hoverkey. El diseño de Hoverkey está optimizado para facilitar la integración con aplicaciones móviles y aplicaciones web existentes, así como para facilitar su uso. Implementa un sistema seguro de almacenamiento y recuperación de credenciales de usuario (por ejemplo, contraseña), protegido mediante token NFC.

La presente solicitud se refiere particularmente a una realización que utiliza un diseño de seguridad específico, denominado en esta descripción como “nivel 1”. Las referencias al nivel 1 de Hoverkey (o Hoverkey L1) deben entenderse en consecuencia.

2.2.1 Concepto de seguridad

El concepto detrás de Hoverkey L1 está diseñado para funcionar con todas las aplicaciones existentes que autentican al usuario mediante una combinación de nombre de usuario y contraseña, aunque se pueden utilizar métodos de autenticación distintos de las contraseñas. Normalmente, sin ningún cambio en la aplicación a la que se accede, la tecnología simplemente reemplaza la entrada manual de la contraseña del usuario con un toque de un token NFC. Esta realización ofrece las siguientes ventajas:

• No se requieren cambios para el servidor de aplicaciones, lo que permite una fácil integración
• Los cambios en cualquier aplicación cliente existente se pueden implementar fácilmente mediante el uso de un componente Hoverkey.
• Mayor seguridad al permitir que la tecnología “recuerde” las contraseñas del usuario, lo que significa
  · El usuario puede elegir contraseñas más seguras (más largas y más “aleatorias”)
  · El usuario puede elegir una contraseña diferente para distintas cuentas sin el temor o la incomodidad de las contraseñas olvidadas
• Elimina la necesidad de ingresar contraseñas alfanuméricas en un teclado en pantalla, especialmente cuando se incluyen símbolos, que es lento y propenso a errores y sujeto a ataques de navegación por encima del hombro.

3. Descripción general

Las realizaciones de la presente divulgación se pueden llevar a la práctica de varias formas y ahora se describirá una realización específica, a modo de ejemplo, con referencia a los dibujos adjuntos, en los que:

Fig. 1 muestra la arquitectura de alto nivel Hoverkey L1;
Fig. 2 muestra la organización de la tarjeta Java y los applets
Fig. 3 muestra el protocolo de activación;
Fig. 4 muestra el método de agregar un nuevo dispositivo a una tarjeta activada;
Fig. 5a muestra el protocolo de registro para una aplicación web privada;
Fig. 5b muestra el protocolo de registro para una aplicación pública;
Fig. 6 muestra el protocolo de acceso por contraseña;
Fig. 7 muestra el proceso de cifrado de contraseña;
Fig. 8 muestra cifrado de recuperación de contraseña;
Fig. 9 muestra la jerarquía de claves; y
Fig. 10 muestra los estados del Applet y su secuencia.

3.1 Modelo de implementación

En un nivel alto, el modelo de implementación preferido de Hoverkey se resume a continuación:

• Cada usuario tiene uno o más dispositivos móviles habilitados para NFC, que pueden ser proporcionados por la empresa o propiedad del usuario.

• Cada usuario recibe un token de seguridad NFC único.

• Cada token NFC se puede emparejar con todos los dispositivos que pertenecen al mismo usuario.

Se toman los siguientes pasos para implementar una Hoverkey:

• Hoverkey compra token NFC en blanco a los revendedores
• Una vez recibida la prueba o la orden de compra, Hoverkey formatea los token NFC para el cliente o un emisor asociado.
• Al recibir el token NFC, el usuario invoca la función de activación
• El usuario luego configura sus aplicaciones habilitadas para Hoverkey con sus credenciales

3.2 Arquitectura

La arquitectura de alto nivel de Hoverkey L1 se ilustra en la Figura 1. Cada aplicación de desarrollador (aplicación 1, aplicación 2 y aplicación 3 en el diagrama) está integrada con el componente Hoverkey L1, lo que le permite comunicarse con el servicio Hoverkey a través de un protocolo de comunicación entre procesos (IPC).

En cada dispositivo móvil, hay una única instancia del servicio Hoverkey que acepta solicitudes de una aplicación y cuando se requiere una contraseña. El servicio Hoverkey recupera la contraseña en nombre de la aplicación a través de una serie de intercambios con el Applet Java Card a través de la interfaz NFC.

Las ventajas de utilizar un servicio incluyen:

• Elimina las claves de autenticación necesarias para compartir (para el acceso al Applet) entre aplicaciones
• No es necesario que las aplicaciones requieran permisos NFC
• Acceso centralizado y mediado al Applet que permite evitar el acceso simultáneo.

En la plataforma Android, los posibles mecanismos de IPC incluyen el método Intent para una integración simple y gruesa, o el método de Servicio Remoto usando el Lenguaje de Definición de Interfaz de Android (AIDL) para una integración fina y de bajo nivel.

Las contraseñas protegidas por Hoverkey son encriptadas por el Applet de la tarjeta al registrarse y almacenadas en el dispositivo móvil dentro de la aplicación Hoverkey. Cuando se requiere acceso, la aplicación registrada solicita la contraseña a través de la aplicación Hoverkey que, a su vez, solicita que el Applet descifre la contraseña.

3.3 Principales características de diseño de seguridad

• Activación y emparejamiento: un token de Hoverkey solo se puede utilizar con un dispositivo con el que se haya emparejado (en el momento de la activación). Cada dispositivo móvil solo puede emparejarse con un token. Cada token se puede emparejar con hasta cuatro dispositivos.

• Registro: para defenderse de aplicaciones maliciosas, las aplicaciones de terceros solo pueden usar los servicios de Hoverkey después de un proceso seguro de registro en el dispositivo. El acceso posterior con contraseña requiere prueba de registro previo.

• Dos factores: cada contraseña puede protegerse adicionalmente con un PIN elegido por el usuario para proporcionar una forma de autenticación de dos factores. Opcionalmente, se pueden proporcionar tres o más niveles de autenticación.

• Seguridad criptográfica: Hoverkey utiliza algoritmos y modos criptográficos estándar de la industria para proteger las contraseñas de los usuarios, respaldados por las mejores prácticas en la administración segura de claves.

• Seguridad de token: los token de Hoverkey se gestionan con seguridad durante todo su ciclo de vida para garantizar que los riesgos se minimicen en todas las etapas.

3.4 Uso de Hoverkey L1

Para utilizar Hoverkey L1, se siguen los siguientes pasos:

1. Una nueva organización de clientes solicita tarjetas Hoverkey L1 para sus usuarios móviles.

2. Hoverkey (o socio) genera un OrgID para el cliente.

a) Opcionalmente, se genera una clave de registro para el cliente si tiene la intención de desarrollar sus propias aplicaciones privadas, que se entrega al cliente o al desarrollador para que la incruste en sus aplicaciones.

3. Hoverkey formatea la cantidad requerida de tarjetas con OrgID, MasterAPIKey, Admin Key, User Authentication Key y PUK, y las envía al Cliente o al Desarrollador.

4. El equipo de desarrollo del cliente integra el componente Hoverkey en sus propias aplicaciones y las configura con su OrgID y RegKey durante el desarrollo.

5. El usuario instala la o las aplicaciones del cliente o desarrollador y la aplicación Hoverkey (de Google Play Store).

6. El usuario recibe un token (formateado) de Sys Admin y un correo electrónico de activación (que contiene una URL de activación).

7. El usuario activa el token desde la aplicación Hoverkey y establece un PIN:

a) La aplicación Hoverkey descarga un archivo de perfil de configuración
b) Se recuerda al usuario que elimine el correo electrónico de activación cuando se complete la activación.

8. Las aplicaciones de terceros se registran con la aplicación Hoverkey (normalmente con un nombre de usuario y contraseña, una vez para cada aplicación de cliente o desarrollador).

9. El usuario comienza a utilizar aplicaciones móviles habilitadas para Hoverkey.

10. El usuario puede emparejar dispositivos adicionales con el token hasta cuatro dispositivos.

a) Si se utiliza un servidor Hoverkey, los datos de la aplicación se pueden sincronizar desde el servidor
b) Todas las aplicaciones habilitadas para Hoverkey deben volver a registrarse en el nuevo dispositivo (según el paso 8).

4. Componentes del sistema

[4.1 Dispositivo móvil]

Hoverkey L1 es preferiblemente compatible con teléfonos inteligentes Android habilitados para NFC, aunque, por supuesto, otras plataformas son igualmente posibles.

4.2 Aplicación Hoverkey L1

Las siguientes subsecciones resumen las funciones proporcionadas por la aplicación Hoverkey L1.

1. Activación de token
a) Emparejamiento de token NFC con dispositivo móvil
b) Configuración de PIN en gestión de token
c) Cambio de PIN
d) Desbloqueo de PIN
e) Revocación de un token
2. Registro de la aplicación (configuración del nombre de usuario y la contraseña)
3. Gestión de aplicaciones
a) Cambio de contraseña
b) Dar de baja una aplicación

4.3 Aplicaciones móviles de terceros

• Incorporar el componente Hoverkey L1 de acuerdo con las pautas de implementación.

4.4 Token NFC

La Figura 2 muestra la organización del cable Java y los Applets.

El token NFC es un token sin contacto que admite especificaciones Java Card y GlobalPlatform. El token preferiblemente tiene un alto nivel de aprobación de seguridad bajo los esquemas de Criterios Comunes y/o FIPS. El producto inicial se implementa en el factor de forma de la norma ISO 7810 (tarjeta de crédito).

El token está diseñado para admitir múltiples Applets de Java Card. El sistema Hoverkey requiere la instalación de un Applet, dejando espacio en la tarjeta para Applets de terceros.

4.5 Servicio de almacenamiento de datos basado en la nube

Hoverkey admite la recuperación y sincronización de credenciales bajo demanda mediante un servicio de almacenamiento en la nube. Hay muchas implementaciones posibles de un servicio en la nube utilizando una variedad de protocolos y, de hecho, ya existen muchos.

Como mínimo, un servicio adecuado admite preferiblemente las siguientes funciones:

1. Identificación de un usuario con un identificador único
2. Almacenamiento de datos arbitrarios en el servidor en un archivo y directorio con nombres arbitrarios
3. Recuperación de datos almacenados previamente

Una implementación más preferible de un servicio de almacenamiento de credenciales de Hoverkey también proporciona:

1. Autenticación sólida del usuario
2. Comunicación con el dispositivo del usuario a través de un canal seguro

3. Medidas de alta disponibilidad
4. Gestión segura de las instalaciones

En la práctica, Hoverkey puede admitir servicios populares en la nube como DropBox o puede proporcionar su propio servicio personalizado para los usuarios de Hoverkey.

4.5.1 Applet de Hoverkey L1

El Applet implementa:

• El proceso de activación (también conocido como “personalización” en la terminología común de tarjetas inteligentes) que incluye:

• Emparejamiento de dispositivo/token
• Generación de clave de cifrado de contraseña (PEK)
• Configuración inicial del PIN de usuario
• Funciones de cifrado/descifrado de contraseña
• El protocolo de autenticación mutua criptográfica

El Applet de Hoverkey almacena y administra los siguientes objetos de datos:

Nombre/etiqueta	Descripción
TokenID [ID del token]	Un identificador único para cada instalación de Applet
DeviceIDs [ID del dispositivo]	Una lista de (hasta 4) ID de dispositivo asociadas con esta tarjeta; la ID debe admitir texto ASCII, por ejemplo, “GalaxyS3-894579”, “DavesTablet-9792234” (de modo que cuando se enumeran las ID, el usuario puede saber qué ID corresponde a qué dispositivo).
Password Encryption Key (PEK) [Clave de cifrado de contraseña]	Derivado de valores aleatorios, las claves para cifrar y descifrar las contraseñas de la aplicación del usuario, así como su protección y verificación de integridad
PIN de usuario	El PIN del usuario utilizado para acceder a las contraseñas. Siempre se configura durante la activación, pero cada aplicación puede decidir si se requiere un PIN. El PIN tiene asociado un contador de intentos restantes de PIN.
PUK de usuario	Las claves de desbloqueo del PIN del usuario. También hay un solo contador de intentos de desbloqueo restantes.
Logs	Registros de actividad para eventos auditables recientes
OrgID	Un identificador único para la organización Cliente o Desarrollador
MasterAPIKey	Una clave única asociada con OrgID para la autenticación de aplicaciones privadas de terceros

4.5.2 Ciclo de vida del token

A continuación, se describe el ciclo de vida de un token NFC:

1. El distribuidor proporciona tarjetas a Hoverkey
2. Formateo de la tarjeta
a) Implementaciones de bajo volumen: Hoverkey formatea tarjetas y suministros para el Cliente o Desarrollador.
b) Implementaciones de gran volumen: Hoverkey proporciona a una impresora de tarjetas de terceros de confianza:
• Gráficos superpuestos de tarjetas
• OrgID, MasterAPIKey y AdminKey
• Conjunto de claves de autenticación y PUK
3. El usuario activa la tarjeta
4. El token activado es:

a) revocado y reemplazado en caso de pérdida o robo
b) devuelto y reemplazado si se vuelve defectuoso
c) devuelto cuando el usuario abandona la organización del cliente
d) actualizado o reemplazado cuando un nuevo Applet o una nueva versión del Applet existente están disponibles para el usuario

5. Diseño de seguridad de alto nivel

5.1 Resumen

El usuario puede descargar la aplicación Hoverkey L1 desde Google Play Store y, por lo tanto, no tiene ninguna información específica del cliente durante la instalación.

Los token NFC son formateados por Hoverkey, que incluye la carga de datos del Cliente. Tras la activación, estos datos se transfieren a la aplicación Hoverkey L1 para permitir que las aplicaciones para desarrolladores se registren.

Las aplicaciones para desarrolladores deben registrarse con el servicio Hoverkey (parte de la aplicación Hoverkey L1) antes de que se habiliten para NFC. El registro implica asegurar la contraseña del usuario con su token NFC (activado).

5.2 Cifrado de contraseña

La función principal de Hoverkey L1 es proporcionar almacenamiento y recuperación seguros de contraseñas. La contraseña está encriptada y su integridad protegida junto con sus metadatos. Cuando se requiere la contraseña, el PEK almacenado en el token NFC se utiliza para verificar y descifrar las contraseñas protegidas.

5.3 Mensajería segura a través de NFC

La especificación de la Plataforma Global (GP) admite el intercambio seguro de mensajes APDU entre la tarjeta y el terminal. GP admite tres niveles de seguridad de mensajería:

1. Solo autenticación de entidad
2. (1) anterior más protección de integridad
3. (2) anterior más protección de confidencialidad.

Hoverkey L1 admite mensajería segura de nivel 3 mediante el protocolo GP Secure Channel Protocol versión 2 (SCP02).

5.4 PIN

Con el fin de admitir un nivel de seguridad mejorado, Hoverkey L1 admite el uso adicional de un PIN que comparten todas las aplicaciones de terceros (ya que es un PIN validado dentro del Applet de token). El usuario debe configurar un PIN en la activación, pero cada aplicación de terceros puede tener su propia política sobre dónde se requiere un PIN para acceder.

El administrador del sistema puede hacer cumplir el requisito de un código PIN de usuario (para todas las aplicaciones) en la activación mediante el proceso de configuración.

6. Protocolos y procedimientos de seguridad

6.1 Activación

La Figura 3 muestra el protocolo de activación.

Condiciones previas

- AuthKey (simple u ofuscada) obtenida de la URL de activación
- Datos de configuración descargados al servicio Hoverkey a través de la URL de activación, que incluyen: políticas de requisitos de PIN
  - Datos de marca compartida
  - Configuración de informes
- El Applet está formateado con OrgID y MasterAPIKey y no se ha activado

Objetivos

- Establecer una clave de autenticación compartida (emparejamiento) entre el Applet y el servicio Hoverkey
- Generar y almacenar la clave de cifrado de contraseña (PEK) en el token
- Inicializar PIN de usuario
- Transferir OrgID y MasterAPIKey al servicio Hoverkey (para la validación de aplicaciones para desarrolladores)

Pasos (refiriéndose a los números correspondientes establecidos en la Figura 3).

1. El servicio Hoverkey consulta el token para TokenID
2. AuthKey puede proporcionarse en texto plano o, para mayor seguridad, ofuscado con el TokenID.
a) Si está ofuscado, el servicio Hoverkey desofusca (desencripta) AuthKey con TokenID (como se muestra en la Figura 3)
b) Si está en texto sin formato, se omite el Paso 1 y el Paso 2 solo necesitará almacenar AuthKey (texto sin formato)
3. El servicio y el Applet realizan una autenticación mutua
4. El servicio envía una solicitud de activación, proporcionando un número aleatorio, PIN y DeviceID
5. Applet almacena PIN y DeviceID, y deriva PEK de Random
6. Applet devuelve TokenID, OrgID y MasterAPIKey. Estos son almacenados por el servicio Hoverkey, junto con RegKey después de derivar de MasterAPIKey.
7. El servicio regresa OK
8. Applet actualiza su estado a activado
9. Tras la activación exitosa, si el usuario no tiene más dispositivos para emparejar con su token, debe eliminar el correo electrónico de activación (y cualquier copia) de su cuenta de correo.

6.2 Agregar un nuevo dispositivo

La Figura 4 muestra el método de agregar un nuevo dispositivo a un token activado.

Condiciones previas

• El Applet ya ha sido activado (por otro dispositivo)

Objetivo

• Transferir OrgID y APIKey al servicio Hoverkey

Pasos (refiriéndose a los números correspondientes establecidos en la Figura 4)

1. El servicio Hoverkey recupera AuthKey del enlace proporcionado por el correo electrónico de activación
2. El servicio se autentica mutuamente con el Applet (ya activado)
3. El servicio proporciona un PIN para autenticarse en el Applet, junto con su propio DeviceID que se agregará
4. Applet valida el PIN, almacena DeviceID
5. Applet devuelve OrgID, MasterAPIKey y TokenID
6. El servicio almacena OrgID y APIKey, junto con RegKey después de derivar de MasterAPIKey.
7. Tras la activación exitosa, si el usuario no tiene más dispositivos para agregar (emparejar) su token, debe eliminar el correo electrónico de activación (y cualquier copia) de su cuenta de correo.

6.3 Registro de la aplicación

El propósito del registro es que la aplicación de terceros se autentique en la aplicación Hoverkey y, al mismo tiempo, proporcione a la aplicación Hoverkey las credenciales de usuario para su almacenamiento seguro.

Tras el registro exitoso, Hoverkey emite la aplicación de terceros con su APIKey aleatoria única para su posterior acceso a la API de Hoverkey (es decir, una APIKey incluso si está comprometida no será válida en un dispositivo diferente).

Hay dos métodos para el registro de aplicaciones:

1. Método de clave asimétrica, principalmente para aplicaciones públicas, es decir, aquellas disponibles en las tiendas de aplicaciones.

2. Método de clave simétrica, principalmente para aplicaciones privadas, es decir, aquellas desarrolladas internamente y distribuidas por medios no públicos.

Método de clave asimétrica

Un desarrollador de aplicaciones públicas que desee integrar Hoverkey en su aplicación debe obtener una clave de registro (RegKey) en forma de certificado, que está incrustado en la aplicación antes de su lanzamiento público. El certificado es emitido por Hoverkey y firmado con la clave privada de Hoverkey. La clave pública correspondiente está incrustada en la aplicación Hoverkey para verificar el certificado de la aplicación. La idea es que el certificado dé fe de varios atributos de la aplicación (que deben poder obtenerse en forma independiente del sistema operativo), lo que dificulta que una aplicación maliciosa se haga pasar por auténtica.

Atributos que deben certificarse incluyen (para la aplicación de Android):

• Su AppID única (nombre del paquete en Android cuya singularidad está garantizada si se descarga desde Play Store).

Método de clave simétrica

Una aplicación privada, es decir, una que no se implemente a través de la tienda de aplicaciones pública, utilizará un esquema de registro diferente. Dado que el desarrollador de la aplicación puede querer implementar sus aplicaciones en forma privada sin la participación de Hoverkey, empleamos un método alternativo que permite al desarrollador generar su propia RegKey (basada en claves simétricas).

La Figura 5 muestra el protocolo de registro. La Figura 5a ilustra el registro para una aplicación web privada y la Figura 5b ilustra el registro para una aplicación pública. Se aplica el mismo número de referencia a cada uno.

Condición previa

• El token NFC se ha activado correctamente (si no, la activación se invocará en el paso 2)

Objetivos

• Configurar el servicio Hoverkey para usar con esta aplicación
• Crear una contraseña protegida con token NFC para usar con los pasos del servicio Hoverkey (refiriéndose a los números establecidos en las Figuras 5a y 5b)

1. La aplicación se registra con OrgID (solo aplicación privada), APIKey, AppID, Política y la contraseña del usuario. En el caso de una aplicación pública, RegKey será un certificado firmado digitalmente. Para una aplicación privada, RegKey será una cadena de bytes pseudoaleatoria. Las políticas admitidas actualmente incluyen:

a) Si se requiere PIN para esta aplicación.

2. Hoverkey Service comprueba si se ha activado. Si está activado, valida la RegKey proporcionada por la aplicación. Para una aplicación pública, la clave de registro se valida mediante la clave pública de registro de la aplicación Hovkery. Para una aplicación privada, el OrgID proporcionado se verifica y RegKey se valida con el derivado de MasterAPIKey.

3. El servicio realiza la autenticación mutua con Applet. Además, Applet valida el DeviceID proporcionado por el Servicio.

4. El servicio envía la solicitud de cifrado de la contraseña, junto con la política y el PIN para su validación.

5. El Applet valida el PIN y cifra la contraseña y la política con PEK.

6. Para validar el cifrado exitoso, el Servicio envía una solicitud de descifrado con la contraseña cifrada, proporcionando Session PEKs (Session PEK ENC y Session PEK_MAC) y opcionalmente un PIN (según la política).

7. Applet descifra y devuelve la contraseña en texto plano, cifrada en SessionPEK.

8. El servicio descifra y verifica la contraseña de texto sin formato devuelta y devuelve el éxito a la aplicación.

9. El servicio guarda el ID de usuario, la política y la contraseña cifrada en el servidor de almacenamiento en la nube como AppID/DeviceID/credentials.dat.

6.4 Recuperación de contraseña

La Figura 6 muestra el protocolo de acceso por contraseña.

Condición previa

• La aplicación se ha registrado con el servicio Hoverkey y ha configurado una contraseña cifrada
• El Applet está en estado activado

Objetivo

• Recupera la contraseña especificada que ha sido protegida por el token NFC
• Opcionalmente, recupera la ID de usuario (si está almacenado)

Pasos (refiriéndose al número establecido en la Figura 6)

1. La aplicación envía el comando de solicitud de contraseña, proporcionando APIKey, AppID.

2. El servicio Hoverkey valida la solicitud.

3. El servicio obtiene la ID de usuario, la política y la contraseña cifrada de la aplicación al recuperar el archivo AppID/DeviceID/credencials.dat del almacenamiento en la nube y luego solicita un PIN al usuario.

4. El servicio se autentica mutuamente con Applet. Además, Applet valida la DeviceID proporcionado por el Servicio.

5. El servicio envía la contraseña cifrada al Applet para que la descifre, proporcionando claves de sesión (Session PEK_ENC y Session PEK_MAC) y un PIN.

6. Applet autentica y descifra la entrada y valida el PIN.

7. Applet devuelve la contraseña de texto sin formato cifrada bajo Session PEK y la integridad protegida con Session PEK_MAC.

8. La contraseña se descifra y se devuelve a la aplicación.

6.5 Cambio de contraseña para la aplicación

Para cambiar la contraseña de una aplicación, los servicios Hoverkey simplemente reemplazan la contraseña cifrada existente por una nueva, con los siguientes pasos:

1. Autenticación mutua, Applet valida DeviceID
2. Requiere PIN
3. El servicio envía una nueva contraseña y política
4. Applet devuelve una contraseña cifrada

6.6 Cambio de PIN

Para cambiar el PIN del token, se siguen los siguientes pasos:

1. Autenticación mutua, Applet valida DeviceID
2. Requiere PIN
3. El usuario ingresa nuevo PIN (dos veces).
4. Applet almacena nuevo PIN.

6.7 Dar de baja la aplicación

Eliminar la siguiente información de la aplicación:

(No se requiere el token de Hoverkey)

1. AppID
2. Cualquier contraseña encriptada
3. Cualquier nombre de usuario guardado
4. Política

6.8 Revocación del token NFC

Si se pierde el token, realice una vez con cada dispositivo asociado:

(No se requiere el token Hoverkey)

- Limpiar la clave de autenticación de la aplicación Hoverkey
- Borrar todas las contraseñas cifradas
- Restablecer la aplicación Hoverkey al estado preactivado

La aplicación Hoverkey también descarga una lista de ID de token revocados periódicamente, lo que le permite revocar el token si una entrada coincide con la que está emparejada.

6.9 Listar dispositivos

- Puede ser llevado a cabo
  - · por cualquier dispositivo emparejado
  - · autenticación mutua, Applet valida DeviceID o autenticación mutua con la clave de administración
  - · o después de la autenticación mutua con la clave de administración
- No se requiere PIN
- Applet devuelve la lista de ID de dispositivo asociados

6.10 Revocación de un dispositivo

Por lo general, se lleva a cabo después de la lista de dispositivos, ya que no se espera que la aplicación Hoverkey recuerde la lista de ID de dispositivo.

- Puede realizarse desde cualquier dispositivo emparejado
- Autenticación mutua, Applet valida DeviceID
- Requiere PIN
- Elimina DeviceID del applet

6.11 Bloqueo de PIN

- Dentro del Applet, el PIN de usuario tiene un valor de PIN intentos restantes (PTR) asociado, inicializado a un número específico.

- El Applet también tiene un número fijo (5) claves de desbloqueo personal (PUK) de 8 dígitos, etiquetadas PUK1, PUK2, etc., que se generan y cargan aleatoriamente al formatear. Se proporciona una copia de los PUK de cada token al administrador del sistema. El Applet mantiene un único valor de intentos de desbloqueo restantes (UTR), inicializado a un número específico.

- Cada vez que el PIN se valida con éxito, PTR se restablece a su valor inicial.

- Cada vez que se detecta un PIN incorrecto, el PTR se reduce en uno.

- Si PTR llega a cero, el PIN de usuario se bloquea. El Applet también vuelve al servicio que PUK el usuario debe usar para desbloquear el PIN e intenta permanecer para ese PUK.

- Para desbloquear y restablecer el PIN, el usuario debe solicitar su código PUK a SysAdmin como se indica en la IU bloqueada con PIN o recuperando el estado del Applet (ver la Sección 0). Si es la primera vez que el usuario desbloquea el PIN, solicitará el código PUK1; la segunda vez requerirá PUK2, etc., es decir, cada código PUK solo se puede usar una vez.

- Si los códigos PUK del usuario están agotados, tan pronto como PTR llegue a cero nuevamente, el Applet se bloqueará. Se debe reemplazar el token NFC.

- Cada vez que se ingresa incorrectamente un PUK, el UTR disminuye. Si UTR llega a cero, el Applet se bloquea. Se debe reemplazar el token NFC.

6.12 Obtener el estado de Applet

- Puede realizarse desde cualquier dispositivo
- Si no está autenticado
  - · Applet devuelve TokenID, estado de Applet
- Si está autenticado (con clave de autenticación o clave de administrador)
  - · Si está en estado formateado: devuelve TokenID, estado de Applet, contador restante de intentos de PIN = Máx, índice PUK actual, contador restante de intentos PUK actual (esto puede no ser el máximo, ya que el Applet puede haberse restablecido a formateado, lo que no restablece el estado del PUK, es decir, los PUK usados se siguen usando). El índice PUK actual es el índice del código PUK que el usuario debe solicitar si el PIN actual se bloquea.
  - · Si está en estado activado: devuelve TokenID, estado de Applet, contador restante de intentos de PIN, índice PUK actual, contador restante de intentos de PUK = Máx.
  - · Si está en estado de PIN bloqueado: devuelve TokenID, estado de Applet, contador restante de intentos de PIN = 0, índice PUK actual, contador restante de intentos de PUK
  - · Si está en estado bloqueado: devuelve TokenID = 0, estado del Applet.

6.13 Funciones de administración

Todas las funciones de esta sección requieren autenticación mutua con la clave de administrador.

6.13.1 Reformatear token

Para volver a formatear el token (por ejemplo, para emitirlo a un nuevo usuario)
• Autenticación mutua con clave de administrador
• Enviar comando de reformateo a:
  · Eliminar el PIN de usuario existente (y restablecer el contador de reintentos)
  · Eliminar las claves de protección de contraseña existentes PEK ENC, PEK_MAC
  · Restablecer el Applet al estado FORMATTED
  · (No restablecer los PUK; los PUK usados permanecen en uso)

6.13.2 Restablecimiento de PIN

Para que el administrador del sistema restablezca el PIN,
• Autenticación mutua con clave de administrador
• Enviar comando de restablecimiento de PIN con el nuevo PIN del usuario
  · (no requiere PUK)

6.14 Acceso de emergencia

6.14.1 Token NFC perdido/defectuoso

Para el acceso en línea de emergencia, el usuario puede simplemente iniciar sesión manualmente con su contraseña. Si el usuario no conoce/recuerda su contraseña (debido al uso de una contraseña compleja, por ejemplo), la función de restablecimiento de contraseña de la aplicación se puede utilizar para establecer una nueva contraseña (y también cambiar la contraseña protegida de Hoverkey).

6.14.2 PIN olvidado/bloqueado

Si la política de una aplicación requiere un PIN que el usuario no recuerda, podría:

• probar diferentes PIN hasta que se bloquee el PIN (si aún no lo ha hecho) y solicitar un PUK al administrador del sistema para desbloquear y restablecer el PIN.

• iniciar sesión manualmente si recuerda la ID de usuario y la contraseña (aunque tendrá que recuperar o restablecer el PIN eventualmente para continuar usando Hoverkey LI).

6.15 Sincronización de credenciales entre dispositivos

Condiciones previas:

• El usuario tiene dispositivos con IDs DeviceA y DeviceB respectivamente
• El token del usuario se ha activado y está listo para usar en ambos dispositivos
• El usuario ha registrado una aplicación con una ID AppX en DeviceA
• AppX no se ha registrado en DeviceB

Objetivo:

• Las credenciales de AppX para el usuario están disponibles para su uso en DeviceB

Pasos

1. En DeviceB, AppX se registra con Hoverkey Service utilizando el método de clave simétrica o asimétrica, pero sin proporcionar las credenciales del usuario.

2. El servicio recupera el archivo AppX/DeviceA/credentials.dat del almacenamiento en la nube.

3. El servicio carga el mismo archivo, sin alteraciones, que AppX/DeviceB/credentials.dat.

4. Las credenciales ya están listas para usarse en DeviceB.

7. Especificación criptográfica

7.1 Gestión de claves

Por motivos de seguridad, las claves utilizadas para cifrar y proteger la integridad de las contraseñas de usuario para el almacenamiento (generadas por el Applet en la activación) nunca abandonan el Applet (ni el token físico). Las claves de sesión también se utilizan (generadas por la aplicación Hoverkey) para cifrar y proteger la integridad de las contraseñas a través de NFC después del descifrado. Estos se limpian inmediatamente después de su uso.

7.2 Proceso de cifrado de almacenamiento de contraseña

La Figura 7 muestra el proceso de cifrado de contraseña.

Cifrado de contraseña para almacenamiento, para realizar por el Applet.

a) Combinar la política, la longitud de la contraseña y la contraseña recibida del dispositivo, aplicar relleno para alinear con la longitud del bloque de cifrado
2. Generar un vector de inicialización (IV) aleatorio de la longitud del bloque de cifrado de cifrado
3. Cifrar el bloque generado en el Paso 1 en modo CBC usando IV del Paso 2, usando la Clave PEK_ENC
4. Cifrar el IV con PEK_ENC en modo ECB
5. Calcular una MAC activada (salida del Paso 4 + salida del Paso 3) utilizando una MAC basada en hash (HMAC) con la clave PEK_MAC
6. (Salida del paso 5 + salida del paso 3 + MAC del paso 4) se devuelve al dispositivo para su almacenamiento

7.3 Proceso de cifrado de recuperación de contraseña (sesión)

La Figura 8 muestra el cifrado de recuperación de contraseña.

Para ser realizado por Applet, después de la verificación del MAC, el descifrado del objeto cifrado proporcionado por el dispositivo y la validación del campo de política.

1. La contraseña de texto plano se rellena a la izquierda con un campo de longitud de dos bytes y a la derecha se rellena con bytes aleatorios para hacer el bloque más grande permitido (se ajusta a una R-APDU) cuyo tamaño es un múltiplo de la longitud del bloque de cifrado.

2. Pasos 2 a 5 según el Proceso de cifrado de almacenamiento de contraseñas, excepto que Session_PEK_ENC y Session_PEK_MAC se utilizan para el cifrado y la protección de la integridad.

7.4 Jerarquía de derivación de claves de registro de aplicaciones (clave simétrica)

La Figura 9 muestra la jerarquía de claves. Las claves se derivan utilizando el KDF basado en HMAC como se describe en la Publicación especial 800-108 del NIST, [: L. Chen, Recommendation for Key Derivation Using Pseudorandom Functions (Revised), NIST SP 800-108, October 2009, disponible en http://csrc.nist.gov/publications/nistpubs/800-108/sp800-108.pdf. Este documento se incorpora como referencia.

Claves de emisor

IssuerMasterKey = bytes aleatorios generados por claves de organización seguras de RNG

OrgID = OrgID único asignado
AppID = (globalmente) identificador de aplicación único

8. Estado del Applet de Hoverkey

La Figura 10 ilustra los estados del Applet y su secuencia.

Estado	Descripción
Instalado	El applet está instalado pero aún no se puede seleccionar.
Seleccionable	El applet ahora se puede seleccionar y está listo para personalizarse.
Formateado	Paso de personalización 1: Hoverkey (o un tercero de confianza) ha generado y cargado OrgID, APIKey, Auth Key, Admin Key y PUK. El administrador puede restablecer las tarjetas activadas a este estado. Todos los objetos de datos se restablecen, excepto los PUK que se hayan utilizado.
Activado	Paso de personalización 2: Token entregado al Usuario que también ha recibido su correo electrónico de activación personalizado. Ha seguido las instrucciones para activar el token y establecer el PIN. El Applet ahora está listo para usarse operativamente. En este momento, se pueden agregar dispositivos adicionales.

(continuación)

Estado	Descripción
PIN bloqueado	Si el PIN del usuario intenta que el contador restante llegue a cero (con al menos un PUK no utilizado restante), el Applet entra en este estado y no realizará las funciones principales hasta que se desbloquee con un PUK.
Bloqueado	Si el contador de intentos de PUK llega a cero o el contador de intentos de PIN llega a cero sin más PUK restante, el Applet se bloquea. El token debe ser revocado, luego puede ser destruido o enviado de regreso a Hoverkey.

9. Glosario

Término	Definición
Applet	Programa de software que se ejecuta en una tarjeta inteligente compatible con la plataforma global y las especificaciones de la tarjeta (por ejemplo, Java Card).
Application Protocol Data Unit (APDU)	Mensajes de comunicación básicos entre una tarjeta inteligente y el terminal (lector).
Reg. Ap.	Validación de una aplicación de terceros por Hoverkey en el primer uso y emisión de la clave API para el acceso posterior.
Bluetooth/BLE	Un conjunto de estándares de comunicación inalámbrica diseñados para el intercambio de datos de corto alcance entre dispositivos. Normalmente lo utilizan pequeños dispositivos personales para crear una red de área personal. Bluetooth de baja energía (BLE) es un estándar de Bluetooth que permite que los dispositivos de bajo consumo que solo se comunican en forma intermitente consuman una fracción de la energía requerida por el Bluetooth normal.
Clientes	La persona u organización responsable de la gestión diaria del token Hoverkey. En particular, son responsables de enviar correos electrónicos de activación y, cuando un usuario requiere el desbloqueo del PIN, autenticar al usuario y emitir códigos PUK. Al vender directamente a los usuarios finales, Hoverkey desempeñará de hecho el papel del cliente.
Desarrolladores	Desarrolladores de aplicaciones móviles, especialmente aquellos que integran funciones de Hoverkey en sus aplicaciones.
DeviceID	Un identificador único para un dispositivo móvil (o uno que probablemente sea único)
Apps del desarrollador	Los desarrolladores pueden mejorar la seguridad de sus aplicaciones móviles existentes mediante la creación de una aplicación de desarrollador, utilizando Hoverkey iOS y Android u otros tipos de bibliotecas de código.
Usuario final (o Usuario)	Un miembro de una organización del Cliente que usa aplicaciones habilitadas para Hoverkey.
Acceso de emergencia	Un servicio opcional que permite el acceso a servicios protegidos por Hoverkey sin un token NFC en funcionamiento mediante un método de autenticación de respaldo preespecificado.
Plataforma Global	Una organización responsable de especificar estándares para la gestión de aplicaciones de tarjetas inteligentes (es decir, Applets).
Hoverkey L1 App	Una aplicación instalada y ejecutada en el dispositivo móvil del usuario que proporciona el servicio Hoverkey y funciones de gestión.
Componente de Hoverkey	Componente de software proporcionado por Hoverkey para la integración en aplicaciones de terceros.
Socio emisor	Una organización con una relación establecida con Hoverkey para emitir el token de Hoverkey a su Cliente.
Personal Identification Number (PIN)	Una secuencia de dígitos que el usuario mantiene en secreto para la autenticación del token NFC.

(continuación)

Término	Definición
Administrador del sistema (Sys Admin)	Por lo general, la persona en la organización del Cliente que es responsable de implementar las políticas de seguridad de TI y tendrá influencia sobre cualquier producto de seguridad que pueda seleccionar la organización. Tienen un conjunto de habilidades técnicas. También pueden asumir el rol de administrador de usuarios (ver más abajo) en implementaciones pequeñas.
Activación de token	El proceso por el cual un usuario final configura el primer uso de su token NFC.
Formateo de token	El proceso mediante el cual se preparan las tarjetas inteligentes en blanco para el cliente.
Admin de usuario	Esta es la persona de la organización del Cliente que es responsable del funcionamiento de los sistemas de seguridad de TI.

REIVINDICACIONES

1. Un método de autenticación de un usuario para acceder a un recurso informático a través de un dispositivo móvil que comprende:

almacenar una autorización de recursos cifrada;
transmitir la autorización de recursos cifrados a al menos un dispositivo de token de seguridad portátil separado;
en el al menos un dispositivo de token de seguridad portátil separado, descifrar la autorización de recursos cifrada y generar al menos parcialmente a partir de ella una respuesta de desbloqueo;
transmitir en forma segura la respuesta de desbloqueo generada al dispositivo móvil; y
proporcionar acceso a través del dispositivo móvil al recurso informático si la respuesta de desbloqueo requerida es válida; en el que se requiere que un usuario se autentique en el dispositivo móvil o en al menos un dispositivo de token de seguridad portátil separado, y la autenticación de usuario se valida en el al menos un dispositivo de token de seguridad portátil separado antes de que se envíe la respuesta de desbloqueo.

2. Un método de acuerdo con la reivindicación 1, en el que la respuesta de desbloqueo comprende una autorización simple, obtenida descifrando la autorización de recursos cifrada.

3. Un método de acuerdo con la reivindicación 1, en el que la respuesta de desbloqueo comprende una función de una autorización simple, obtenida al descifrar la autorización de recursos cifrada, e información adicional.

4. Un método de acuerdo con cualquiera de las reivindicaciones anteriores, en el que la respuesta de desbloqueo se transmite al dispositivo móvil bajo la protección de una clave de cifrado, como una clave de sesión.

5. Un método de acuerdo con cualquiera de las reivindicaciones anteriores, en el que el al menos un token de seguridad portátil separado almacena las credenciales de usuario, en donde el descifrado en el al menos un dispositivo de token de seguridad portátil separado se basa en las credenciales del usuario, en el cual, opcionalmente, las credenciales de usuario son generadas por el al menos un dispositivo de token de seguridad portátil separado y nunca dejan el al menos un dispositivo de token de seguridad portátil separado.

6. Un método de acuerdo con la reivindicación 5, en el que la autorización de recursos cifrados se puede descifrar únicamente con las correspondientes credenciales de usuario almacenadas en al menos un dispositivo de token de seguridad portátil separado.

7. Un método de acuerdo con cualquiera de las reivindicaciones anteriores, en el que el dispositivo móvil y el al menos un dispositivo de token de seguridad portátil separado realizan una autenticación mutua criptográfica antes de la transmisión de la autorización de recursos cifrados.

8. Un método de acuerdo con cualquiera de las reivindicaciones anteriores, en el que la autenticación del usuario en el dispositivo móvil es a través de información biométrica, por ejemplo, una huella dactilar y/o un patrón de iris.

9. Un sistema para autenticar a un usuario para que acceda a un recurso informático a través de un dispositivo móvil, que comprende un dispositivo móvil y al menos un dispositivo de token de seguridad portátil separado, que comprende medios para llevar a cabo el método de acuerdo con la reivindicación 1.

10. Un sistema de acuerdo con la reivindicación 9, en el que el al menos un dispositivo de token de seguridad portátil separado está configurado para transmitir la respuesta de desbloqueo al dispositivo móvil bajo la protección de una clave de cifrado, como una clave de sesión.

11. Un sistema de acuerdo con las reivindicaciones 9 o 10, en el que la autenticación del usuario se realiza mediante información biométrica, por ejemplo, una huella dactilar y/o un patrón de iris.

12. Un sistema de acuerdo con cualquiera de las reivindicaciones 9 a 11, en el que al menos un dispositivo de token de seguridad portátil separado está configurado para verificar la integridad de la autorización de recursos cifrados antes del descifrado.

13. Un sistema de acuerdo con cualquiera de las reivindicaciones 9 a 12, en el que el dispositivo móvil y el al menos un dispositivo de token de seguridad portátil separado están configurados para realizar una autenticación mutua criptográfica antes de la transmisión de la autorización de recursos cifrados.

14. Un sistema de acuerdo con cualquiera de las reivindicaciones 9 a 13, en el que el al menos un dispositivo de token de seguridad portátil separado está configurado para enviar la respuesta de desbloqueo solo con la confirmación positiva del usuario, por ejemplo, presionando un botón en el token.

15. Un sistema de acuerdo con una cualquiera de las reivindicaciones 9 a 14, en el que el al menos un dispositivo de token de seguridad portátil separado es proporcionado por cualquiera de un llavero, una insignia, una tarjeta inteligente NFC, un reloj, un anillo portátil, una pulsera de fitness, un auricular inalámbrico, una joya o un dispositivo informático móvil.

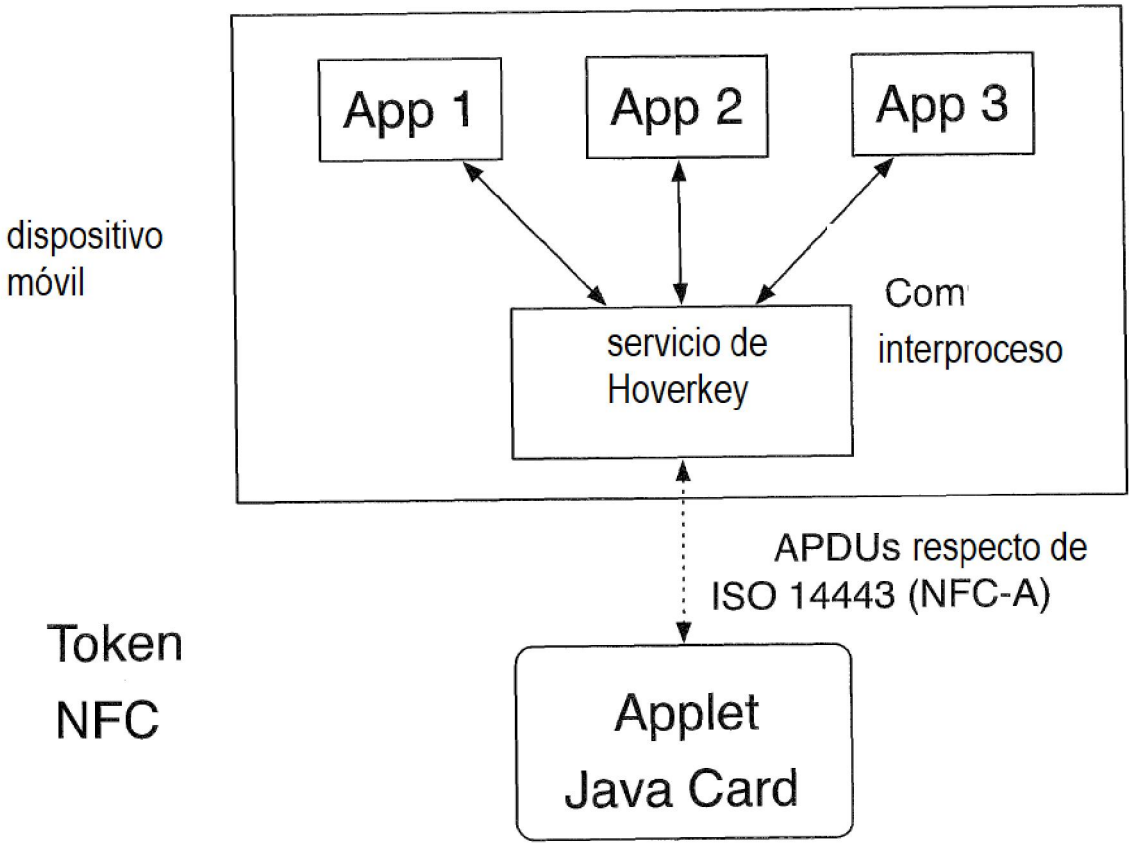

App 1
App 2
App 3
dispositivo
móvil
servicio de
Hoverkey
Com
interproceso
APDUs respecto de
ISO 14443 (NFC-A)
Token
NFC
Applet
Java Card

Figura 1

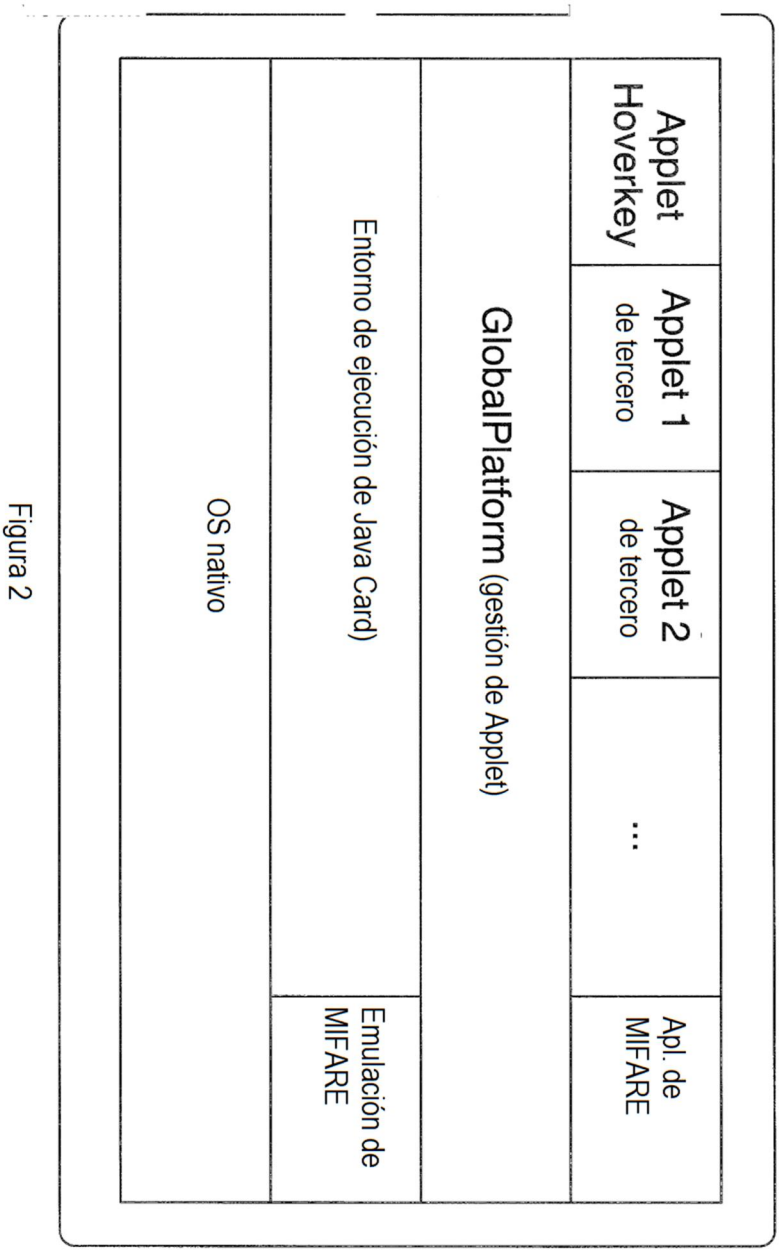

Applet Hoverkey
Applet 1 de tercero
Applet 2 de tercero
...
Apl. de MIFARE
GlobalPlatform (gestión de Applet)
Entorno de ejecución de Java Card)
Emulación de MIFARE
OS nativo

Figura 2

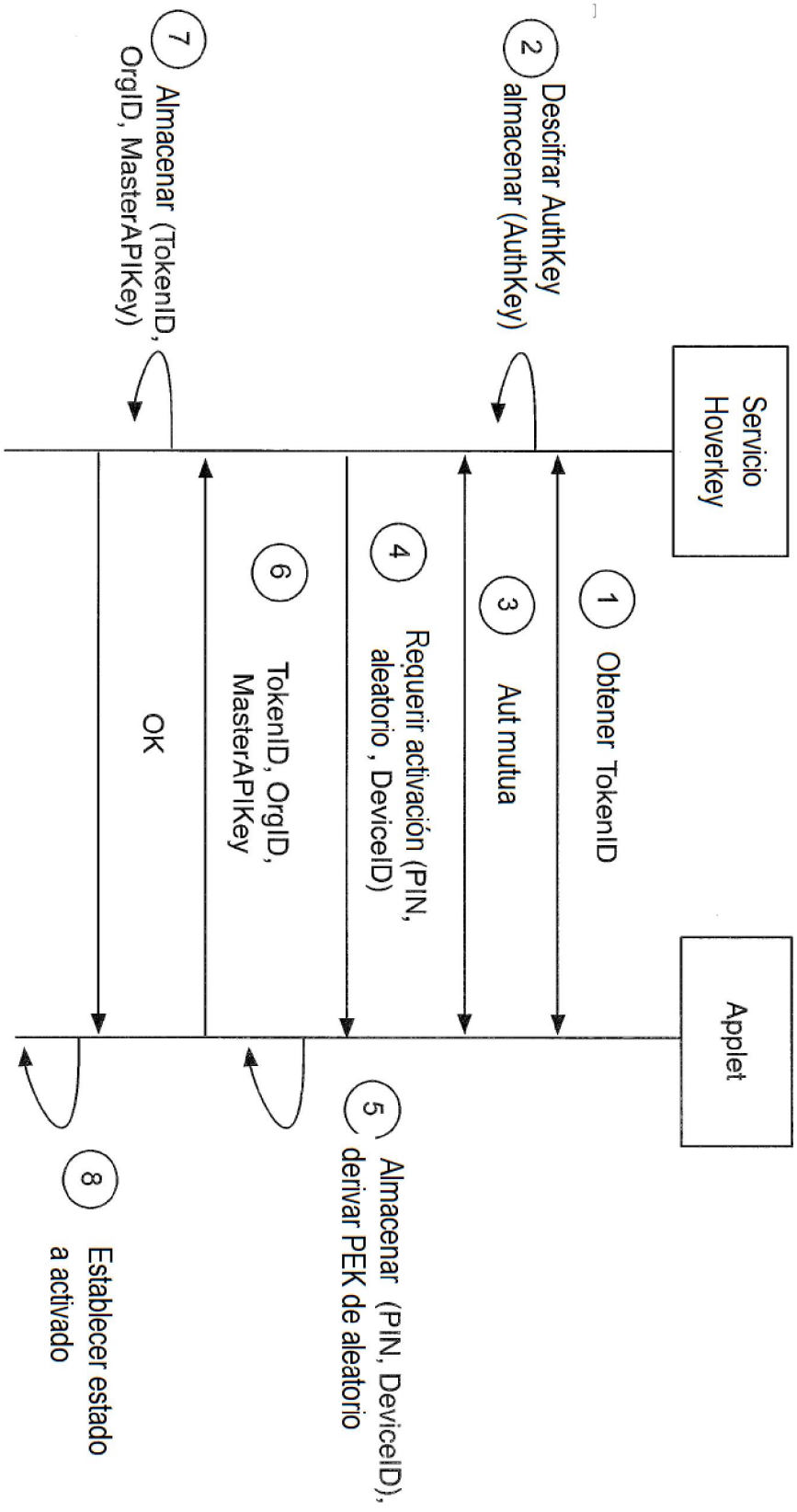

Servicio Hoverkey
Applet
1 Obtener TokenID
2 Descifrar AuthKey almacenar (AuthKey)
3 Aut mutua
4 Requerir activación (PIN, aleatorio , DeviceID)
5 Almacenar (PIN, DeviceID), derivar PEK de aleatorio
6 TokenID, OrgID, MasterAPIKey
7 Almacenar (TokenID, OrgID, MasterAPIKey)
OK
8 Establecer estado a activado

Figura 3

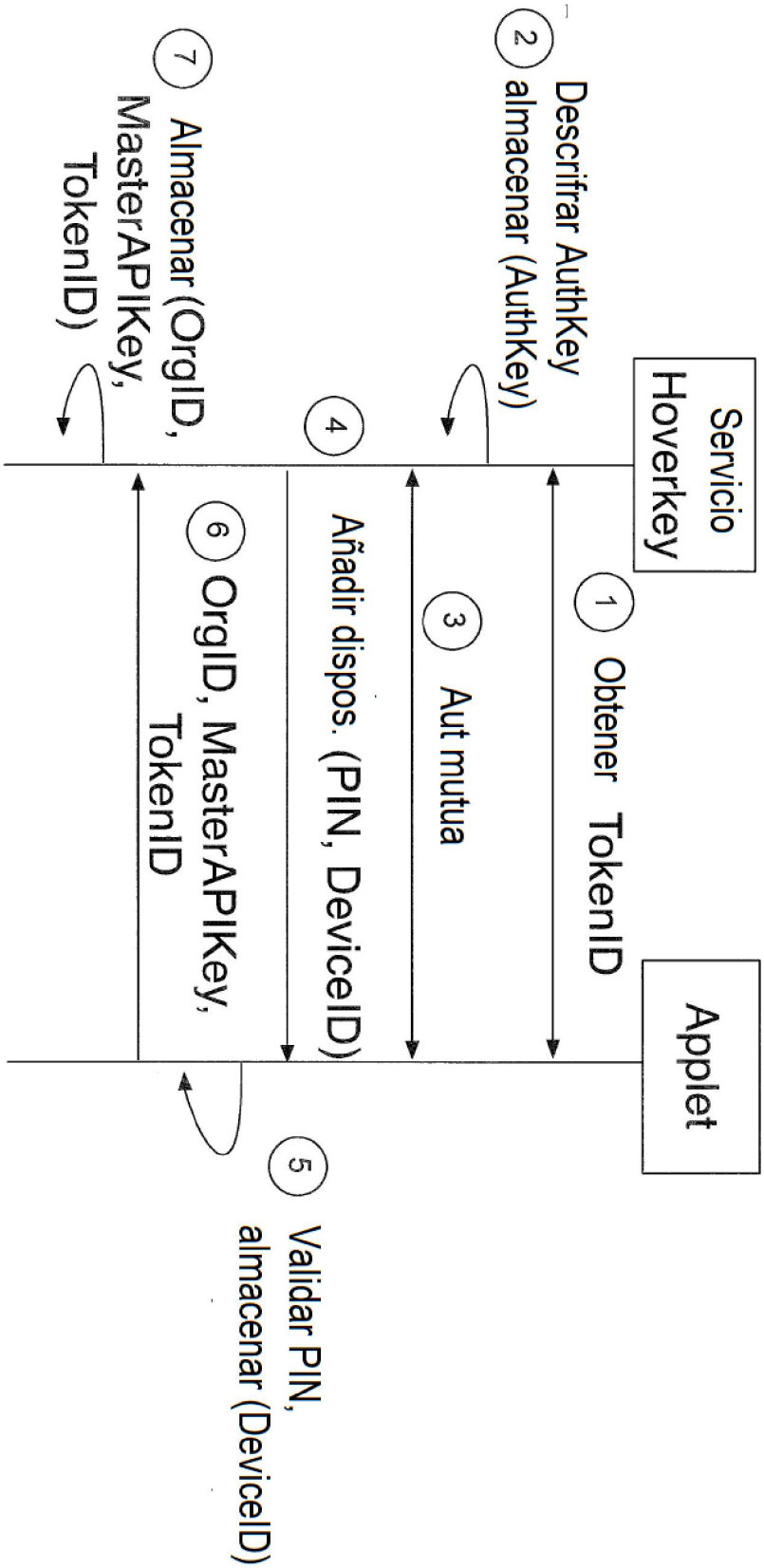

Servicio Hoverkey
Applet
1 Obtener TokenID
2 Descrifrar AuthKey almacenar (AuthKey)
3 Aut mutua
4 Añadir dispos. (PIN, DeviceID)
5 Validar PIN, almacenar (DeviceID)
6 OrgID, MasterAPIKey, TokenID
7 Almacenar (OrgID, MasterAPIKey, TokenID)

Figura 4

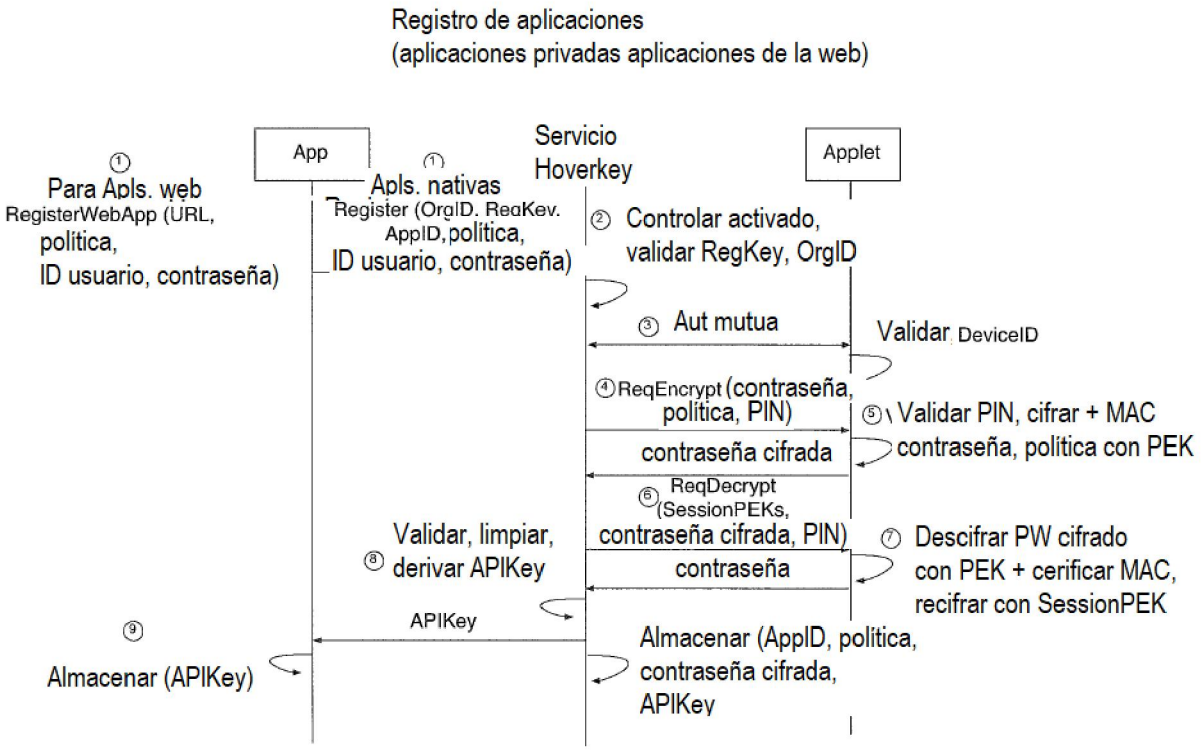

Registro de aplicaciones
(aplicaciones privadas aplicaciones de la web)
App
Servicio Hoverkey
Applet
① Para Apls. web
RegisterWebApp (URL, política, ID usuario, contraseña)
① Apls. nativas
Register (OrgID. RegKey. AppID, política, ID usuario, contraseña)
② Controlar activado, validar RegKey, OrgID
③ Aut mutua
Validar DeviceID
④ ReqEncrypt (contraseña, política, PIN)
⑤ Validar PIN, cifrar + MAC contraseña, política con PEK
contraseña cifrada
⑥ ReqDecrypt (SessionPEKs, contraseña cifrada, PIN)
⑦ Descifrar PW cifrado con PEK + cerificar MAC, recifrar con SessionPEK
⑧ Validar, limpiar, derivar APIKey
contraseña
APIKey
⑨ Almacenar (APIKey)
Almacenar (AppID, política, contraseña cifrada, APIKey

Figura 5a

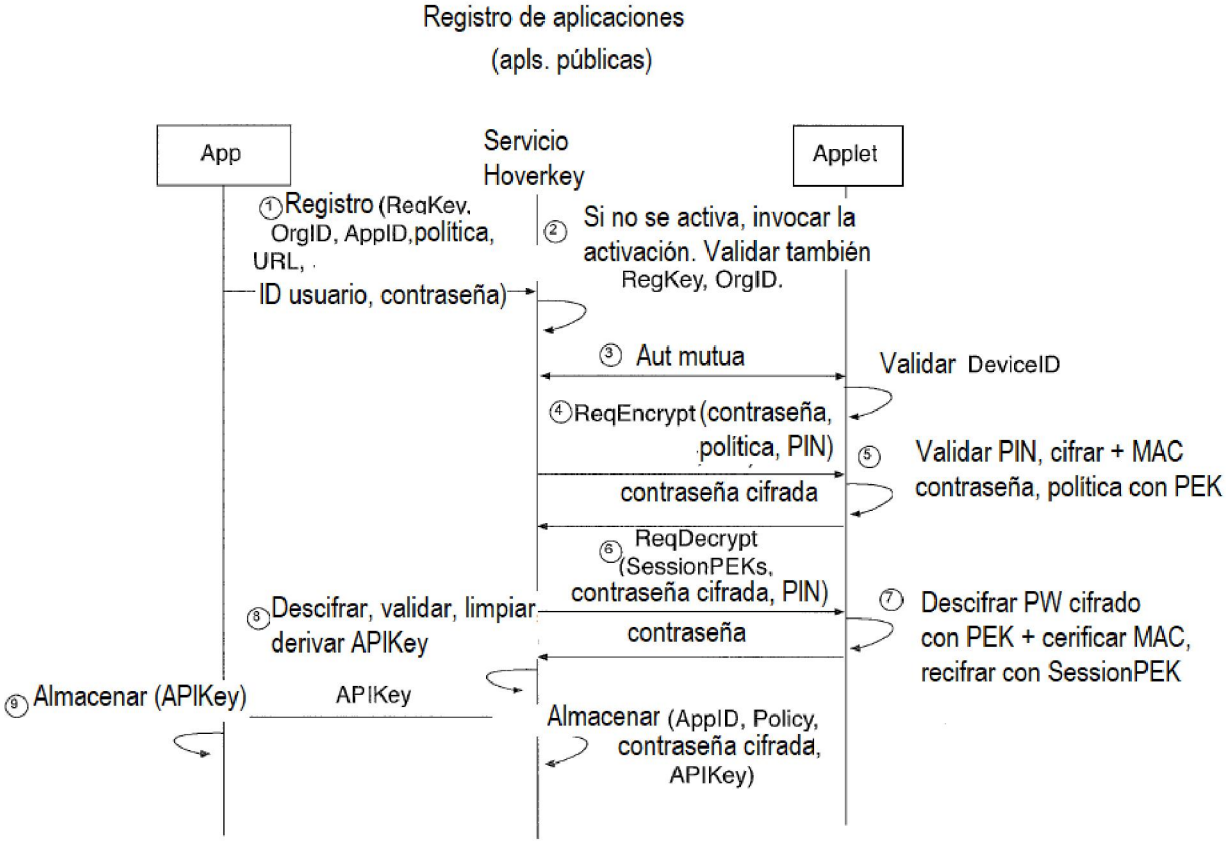

Registro de aplicaciones
(apls. públicas)
App
Servicio
Hoverkey
Applet
1 Registro (RegKey, OrgID, AppID, política, URL, .
ID usuario, contraseña)
2 Si no se activa, invocar la activación. Validar también RegKey, OrgID.
3 Aut mutua
Validar DeviceID
4 ReqEncrypt (contraseña, política, PIN)
5 Validar PIN, cifrar + MAC contraseña, política con PEK
contraseña cifrada
6 ReqDecrypt (SessionPEKs, contraseña cifrada, PIN)
7 Descifrar PW cifrado con PEK + cerificar MAC, recifrar con SessionPEK
8 Descifrar, validar, limpiar derivar APIKey
contraseña
9 Almacenar (APIKey)
APIKey
Almacenar (AppID, Policy, contraseña cifrada, APIKey)

Figura 5b

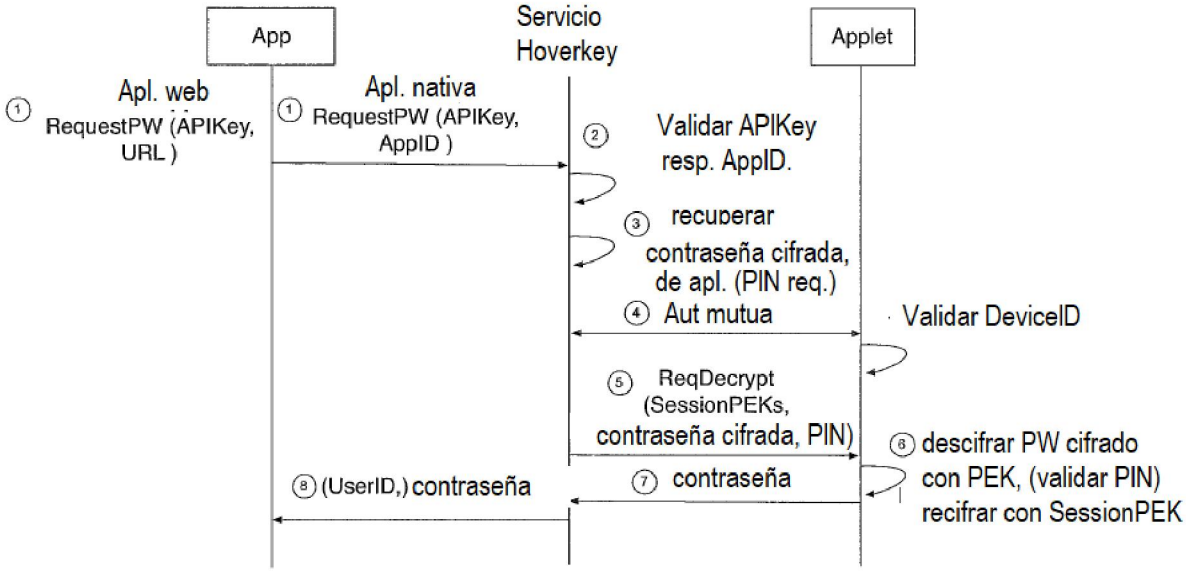

App
Servicio Hoverkey
Applet
Apl. web
1 RequestPW (APIKey, URL)
Apl. nativa
1 RequestPW (APIKey, AppID)
2 Validar APIKey resp. AppID.
3 recuperar contraseña cifrada, de apl. (PIN req.)
4 Aut mutua
Validar DeviceID
5 ReqDecrypt (SessionPEKs, contraseña cifrada, PIN)
6 descifrar PW cifrado con PEK, (validar PIN) recifrar con SessionPEK
7 contraseña
8 (UserID,) contraseña

Figura 6

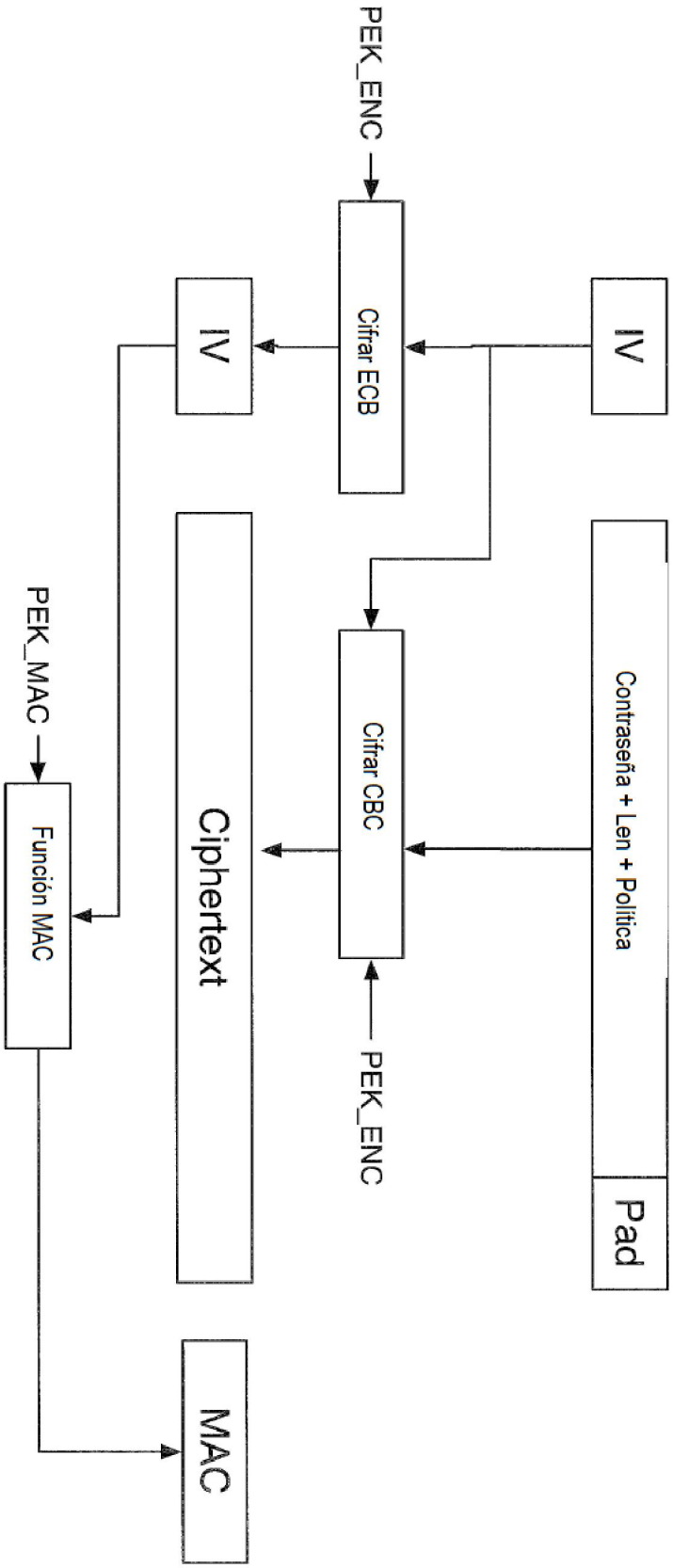
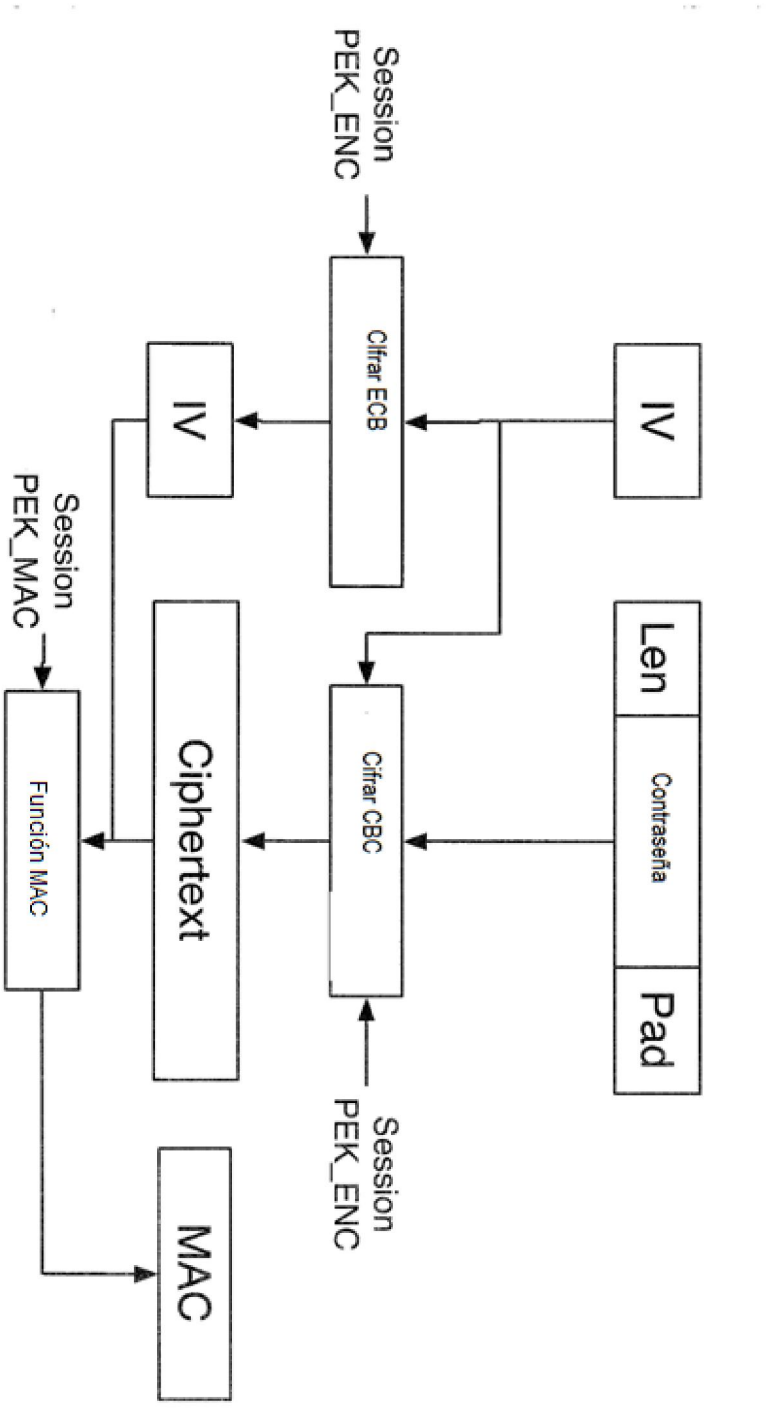

Esquema de cifrado de contraseña (con PEK)
PEK_ENC
Cifrar ECB
IV
IV
Contraseña + Len + Política
Pad
Cifrar CBC
PEK_ENC
Ciphertext
PEK_MAC
Función MAC
MAC

Figura 7



Esquema de cifrado de contraseña (con SessionPEK)

Figura 8

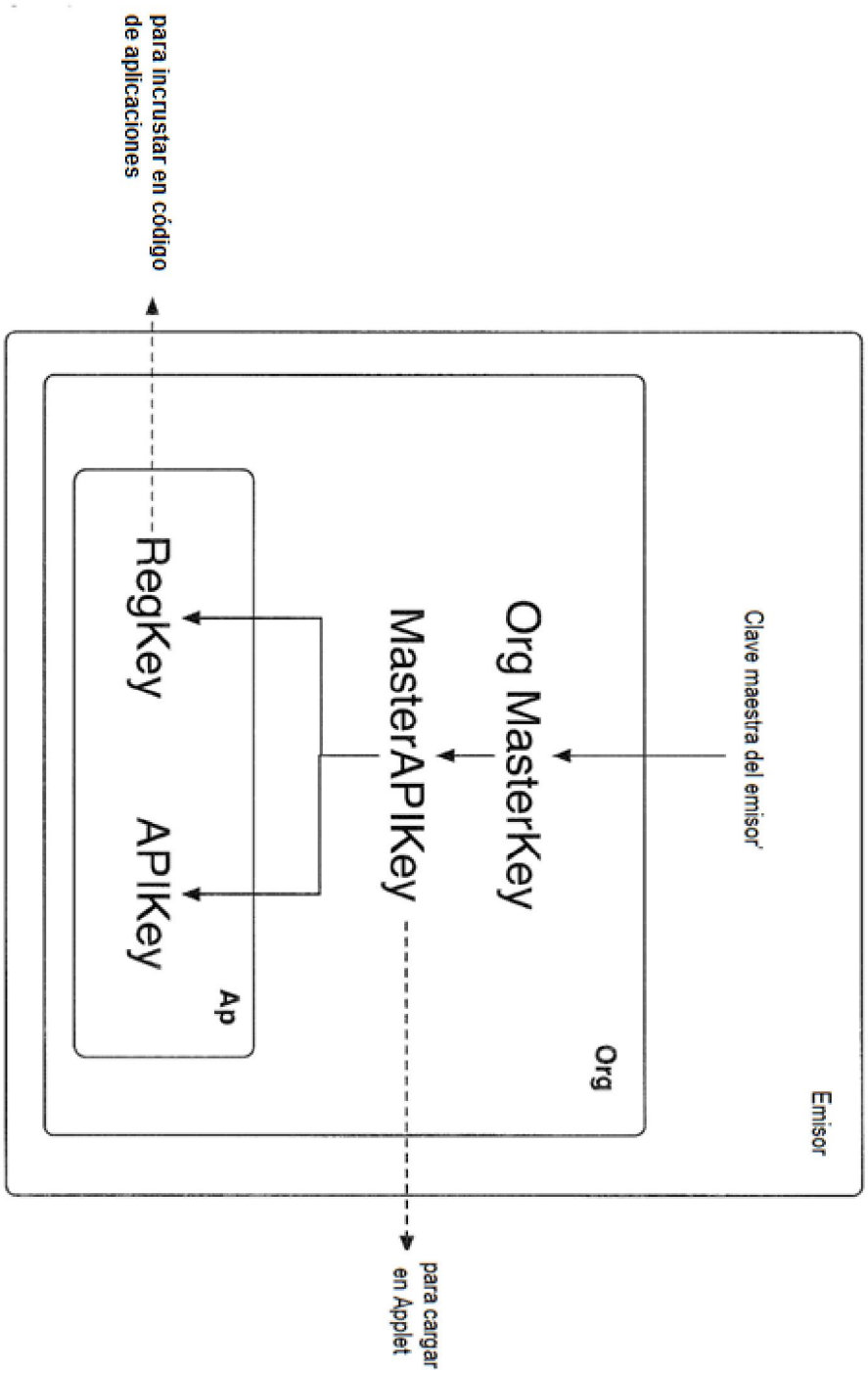

para incrustar en código de aplicaciones
Emisor
Clave maestra del emisor
Org MasterKey
Org
MasterAPIKey
RegKey
APIKey
Ap
para cargar en Applet

Figura 9

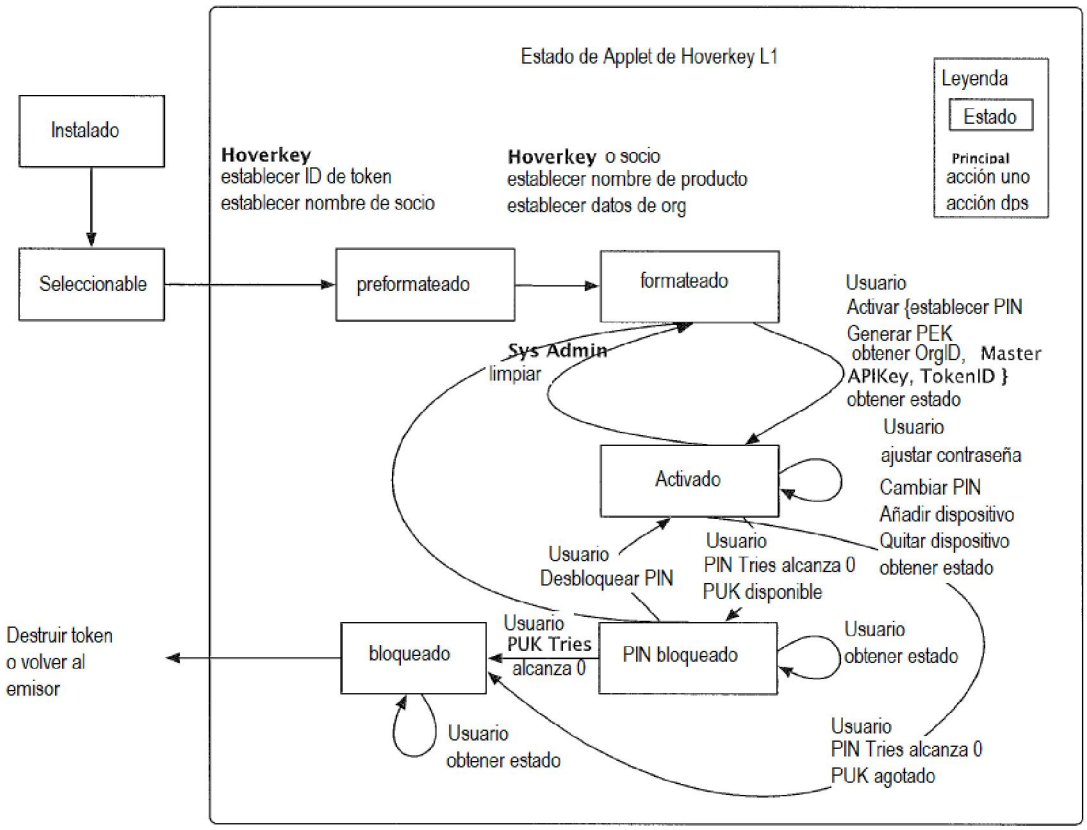

Estado de Applet de Hoverkey L1
Leyenda
Estado
Principal
acción uno
acción dos
Instalado
Seleccionable
preformateado
formateado
Hoverkey
establecer ID de token
establecer nombre de socio
Hoverkey o socio
establecer nombre de producto
establecer datos de org
Usuario
Activar {establecer PIN
Generar PEK
obtener OrgID, Master
APIKey, TokenID }
obtener estado
Sys Admin
limpiar
Usuario
ajustar contraseña
Cambiar PIN
Añadir dispositivo
Quitar dispositivo
obtener estado
Activado
Usuario
Desbloquear PIN
Usuario
PIN Tries alcanza 0
PUK disponible
Destruir token
o volver al
emisor
bloqueado
Usuario
PUK Tries
alcanza 0
PIN bloqueado
Usuario
obtener estado
Usuario
obtener estado
Usuario
PIN Tries alcanza 0
PUK agotado

Figura 10