(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 807 547**

(51) Int. Cl.:

G06F 21/31 (2013.01)
G06Q 20/10 (2012.01)
G06Q 20/32 (2012.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **14.10.2013** **PCT/GB2013/052669**

(87) Fecha y número de publicación internacional: **23.04.2015** **WO15055972**

(96) Fecha de presentación y número de la solicitud europea: **14.10.2013** **E 13783625 (0)**

(97) Fecha y número de publicación de la concesión europea: **01.04.2020** **EP 3058498**

(54) Título: **Núcleo de seguridad CRM**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
23.02.2021

(73) Titular/es:

CRYPTOMATHIC LTD (100.0%)
329 Cambridge Science Park Milton Road
Cambridge, Cambridgeshire CB4 0WG, GB

(72) Inventor/es:

BOND, MIKE;
LANDROCK, MATTHEW y
LANDROCK, PETER

(74) Agente/Representante:

FLORES DREOSTI, Lucas

ES 2 807 547 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Núcleo de seguridad CRM

ANTECEDENTES DE LA INVENCIÓN

Campo de la invención

[0001] La presente invención se refiere en general a dispositivos y mecanismos de seguridad informáticos y, de forma más concreta, a núcleos de seguridad de aplicaciones con servicios de inscripción y autenticación para soportar aplicaciones en red de próxima generación.

Antecedentes de la invención

[0002] La banca móvil representa dinero en la moneda moderna. Al igual que el dinero en efectivo pasado de moda, atrae todo tipo de ataques criminales potenciados con cualquier herramienta, arma, dispositivo e información disponibles para ayudar. Por lo tanto, las aplicaciones de banca móvil y en la nube deben ser capaces de proteger el dinero, el usuario, el banco y cualquier otra persona con una inversión en el sistema financiero y sus transacciones. De lo contrario, las aplicaciones de banca móvil y en la nube perderán confianza y el dinero se trasladará a un lugar seguro.

[0003] La ganancia monetaria no es el único objetivo que un delincuente puede tener en mente. Otros objetivos incluyen venganza, anarquía, curiosidad y el bien público percibido. Los recursos que puede utilizar un atacante pueden variar de insignificantes a verdaderamente sustanciales. Las unidades militares gubernamentales muy sofisticadas ahora incluso están comenzando a participar en estas actividades maliciosas para sus propios objetivos estratégicos y a veces insondables.

[0004] La mejora constante de los niveles de cómputo y las capacidades de comunicación de los teléfonos inteligentes modernos ahora hace posible hacerlos adecuadamente seguros para las aplicaciones de banca móvil nativas. Pero las aplicaciones actuales de banca móvil en teléfonos inteligentes son en su mayoría poco ambiciosas, principalmente las basadas en navegador que ofrecen una funcionalidad limitada de banca por Internet, p. ej., localizadores de sucursales y cajeros automáticos. Del mismo modo, la seguridad de las aplicaciones del lado del servidor, a menos que se aplique utilizando HSM (módulos de seguridad de *hardware*), es igualmente débil. La seguridad mejorada es la clave para que las aplicaciones de banca móvil y en la nube admitan una funcionalidad más nativa, p. ej., saldo, transferencias, configuración del beneficiario, etc. Los núcleos de seguridad son una forma de lograrlo. Se debe conservar un modelo mixto proporcionando aún algunas partes de la experiencia de banca móvil del usuario a través de controles de vista de web para una buena flexibilidad.

[0005] Otro desafío para las aplicaciones de banca móvil y en la nube es que deben mantener un nivel adecuado de seguridad en un entorno en rápido cambio donde las plataformas y los sistemas operativos pueden transformarse completamente en cuestión de meses en lugar de años.

[0006] El Informe de amenazas móviles maliciosas de Juniper Networks de 2010-2011 dice: «Las amenazas a los dispositivos móviles son reales y van mucho más allá de los simples virus para incluir *malware*, pérdida y robo, interceptación de comunicación de datos, explotación y mala conducta, y ataques directos [...] A medida que aumenta el uso de los dispositivos móviles, la ausencia de productos de seguridad móvil instalados está desempeñando un papel facilitador en la vulnerabilidad de los dispositivos móviles y la explotación de datos sensibles e información de identificación personal (PII, por sus siglas en inglés)».

[0007] Los vectores de amenazas que desafían las aplicaciones de banca de clientes móviles o en red incluyen *malware* como *spyware*, virus, troyanos y gusanos; pérdida y robo, como pérdida de datos debido a dispositivos móviles extraviados o robados; interceptación de comunicación de datos, como escucha informática de comunicaciones, incluidos correos electrónicos, mensajes de texto, llamadas de voz, etc., que se originan o se envían a un dispositivo móvil o proceso de cliente en red; explotación y mala conducta, como el uso inadecuado de dicho dispositivo para diversión personal o ganancia monetaria; y ataques directos, como el servicio de mensajes cortos (SMS) y vulnerabilidades de seguridad (*exploits*) del navegador.

[0008] Las contraseñas estáticas han demostrado ser poco fiables y fáciles de comprometer. Por lo tanto, las aplicaciones de teléfonos inteligentes de contraseña de un solo uso (OTP, por sus siglas en inglés) que generan contraseñas dinámicas han comenzado a aparecer en Android de Google, iOS de Apple, Blackberry y otras plataformas, así como esquemas criptográficos simétricos o asimétricos en instancias cliente-servidor que no involucran al usuario. Como ejemplo, el conjunto de aplicaciones de seguridad Mobile AuthApp de Cryptomathic (Aarhus, Dinamarca) se basa en las normas industriales, como la contraseña de un solo uso basada en tiempo móvil (TOTP), la contraseña de un solo uso basada en el código de autenticación de mensajes con *hash* (HMAC) (HOTP), los algoritmos de pregunta respuesta OATH (OCRA) y la firma de transacciones del programa de autenticación de chip (CAP) Europay-MasterCard-Visa (EMV), p. ej., Mobile AuthApp ™ de Cryptomathic. La solución de Cryptomathic está totalmente integrada en sus servidores *backend*, por ejemplo, Authenticator ™ y Signer ™ de Cryptomathic.

[0009] Las soluciones de *token* de *software* también se han generalizado, son fáciles de implementar y altamente rentables, en comparación con los *tokens* heredados, y ofrecen una formidable seguridad de dos factores (2FA).

[0010] US2004039924 da a conocer un sistema y método para proteger un dispositivo informático utilizando una clave criptográfica maestra que está asociada al dispositivo.

[0011] WO2013056104 da a conocer una plataforma para llevar a cabo transacciones personalizadas seguras en un ecosistema multidominio que incluye un nivel de personalización que permite la personalización del proveedor de servicios para uno o más elementos del ecosistema almacenados en un dispositivo móvil.

[0012] US2013263212 da a conocer sistemas y métodos para una estructura móvil segura para conectar de manera segura aplicaciones que se ejecutan en dispositivos móviles a servicios dentro de una empresa.

[0013] "Mobile Proximity Payment Issues and Recommendation" Mobile Payment Forum, versión 1.0, octubre de 2006, está escrito para que lo consideren los que implementan los sistemas, o componentes de un sistema, de pago de proximidad móvil. También contiene información de uso para foros que definen tecnología que puede formar una base para la implementación de un sistema de pago de proximidad móvil.

SUMARIO DE LA INVENCIÓN

[0014] La invención se define por las reivindicaciones independientes adjuntas. En las reivindicaciones dependientes se definen modos de realización adicionales. Según un primer aspecto de la invención, se proporciona un núcleo de seguridad para soportar funcionalidades nativas en una aplicación de banca de un dispositivo informático creado para la comunicación por red con un servidor, que comprende:

interfaces de programación de aplicaciones (API) de inscripción y autenticación para una aplicación de banca;

un flujo de trabajo de registro incluido en las API y configurado para establecer claves e identificadores únicos;

una API para inscripción en todo el dispositivo para establecer una clave y un identificador de usuario (UID, por sus siglas en inglés) que sea visible en diferentes aplicaciones en el mismo dispositivo del cliente;

una API para inscripción de instalación específica diseñada para establecer una clave y un (UID) que son únicos para una instalación individual de dicha aplicación móvil;

una API para caracterizaciones de dispositivos que incluye identificación, detección de *malware* y estado de versión del *software*/parche: y

una API para utilizar datos de dispositivos inscritos para permitir el propósito del cliente o la autenticación del dispositivo, para el inicio de sesión en aplicaciones móviles, para la aprobación de transacciones y para la autenticación/cifrado a nivel de transacciones;

donde, el núcleo de seguridad en su totalidad es extensible en una serie continua de versiones de *software* y se puede implementar simultáneamente en una pluralidad de plataformas de teléfonos inteligentes; y

donde, la autenticación del usuario y la transacción son soportadas con un nivel predeterminado de seguridad en una aplicación de banca.

[0015] La API para inscripción en todo el dispositivo puede establecer una clave y un identificador de usuario (UID) creado para permanecer constante.

[0016] El núcleo de seguridad puede comprender además algunos o todos los siguientes: una API para la autenticación y cifrado de datos a nivel de secuencia para soportar la seguridad de las comunicaciones de aplicaciones de uso general; una API para la generación de vales/*cookies* para almacenamiento en caché de los resultados de autenticación entre sesiones; la transferencia de vales/*cookies* dentro de un ecosistema de aplicaciones y entre aplicaciones nativas, aplicaciones mixtas y aplicaciones web puras; gestión de compromisos del dispositivo, incluidas listas negras; verificación/aseguramiento de autointegridad de la aplicación; denegación de resistencia de servicio en el diseño del protocolo manteniendo un procesamiento mínimo que puede ser causado antes de la autenticación.

[0017] El núcleo de seguridad puede comprender además algunos o todos los siguientes: una interfaz de programación de aplicaciones (API) de almacenamiento de datos, una API de control de inscripción, una API de sesión, una API de transacción y una API de secuencia; un ofuscador de enrutamiento (RO, por sus siglas en inglés) con un controlador RO que incluye una tabla de saltos configurable y está conectado a cada una de las API de almacenamiento de datos, control de inscripción, sesión, transacción y secuencia; una API de almacenamiento de datos ficticios, una API de control de inscripción ficticia, una API de sesión ficticia, una API de transacción ficticia y una API de flujo ficticia, todas visibles externamente y todas enrutables internamente a través del RO a sus homólogos API correspondientes de acuerdo con una inicialización de tiempo de ejecución de dicha tabla de saltos; y una API de comunicaciones conectada a un protocolo de registro y módulos de protocolo de sesión a través de ofuscadores de protocolo.

[0018] El núcleo de seguridad puede comprender además una mitad pública a largo plazo de una clave K que se puede entregar a la aplicación móvil configurada para ser compartida con un proveedor de servicios de autenticación y que puede utilizarse para implementar protocolos de autenticación específicos; donde, un servidor está configurado para conservar una mitad privada de la clave en un coprocesador seguro para evitar robos de la clave. Por ejemplo, una clave de sesión KS puede generarse y cifrarse bajo una clave pública codificada de forma rígida de un servidor, a saber «Kpub». Una clave de sesión cifrada, {KS}Kpub, puede entregarse entonces al servidor utilizando HTTP o HTTPS, en función de una URL proporcionada para el servidor que está codificada de forma rígida en la aplicación de banca móvil. Puede generarse un número de serie único para el dispositivo móvil con el servidor cuando recibe la clave de sesión bajo Kpub. Puede enviarse una solicitud procedente del servidor a un módulo de seguridad de *hardware* (HSM) para generar una clave a largo plazo K para el dispositivo móvil, y para utilizar la clave de sesión con el fin de cifrar la clave a largo plazo, el número de serie y algunos parámetros de configuración seleccionados por el servidor [{K,serial,settings}]KS, para el almacenamiento en una base de datos a largo plazo como un registro. La [{K,serial,settings}]KS puede reenviarse en un mensaje desde el HSM hasta el dispositivo móvil y el mensaje [{K,serial,settings}]KS puede devolverse desde el servidor hasta la aplicación de banca móvil a través de un enlace HTTP-HTTPS. La clave de sesión KS puede descifrarse y verificar su integridad para recuperar la clave K, y recoger los datos de configuración. La clave a largo plazo K y los datos de configuración pueden almacenarse en un almacenamiento local que es accesible solo por esa aplicación de banca móvil específica, y que está ofuscada por cifrado utilizando una clave codificada de forma rígida KO.

[0019] Al menos una de las API del núcleo de seguridad puede tener dos implementaciones separadas producidas de manera independiente del mismo módulo funcional que pueden intercambiarse a voluntad para crear un delta de código artificialmente alto entre actualizaciones menores de una aplicación para aumentar el coste de adaptación del *malware* para atacar nuevos lanzamientos de dicha aplicación.

[0020] El núcleo de seguridad puede comprender además una pluralidad de mecanismos de seguridad técnicos configurados para resistir infecciones de *malware* e intentos de ingeniería inversa mediante el uso de ofuscación, *stubs* y/o camuflaje. El núcleo de seguridad puede comprender además una aplicación descargable instalable en el dispositivo de un cliente; y un sistema operativo alojado en una plataforma estandarizada; donde, las claves de registro y autenticación pueden protegerse para que un dispositivo en red participe en transacciones bancarias.

[0021] El núcleo de seguridad puede comprender además un ofuscador de enrutamiento que opera en una capa externa; al menos una versión anterior de un núcleo de seguridad incluido como camuflaje y que proporciona resistencia a los esfuerzos de ingeniería inversa; y una biblioteca de seguridad de la capa de transporte (TLS, por sus siglas en inglés) interna configurada para utilizarse en lugar de una capa de TLS del sistema operativo; donde, las *cookies* se administran internamente en el núcleo de seguridad en lugar de hacerlo mediante las bibliotecas del sistema operativo.

[0022] El núcleo de seguridad puede comprender además *stubs* de detectores antimanipulación internos configurados para notar las desviaciones del funcionamiento normal del programa que pueden surgir debido a la modificación del código, y para establecer indicadores para una acción posterior. El núcleo de seguridad puede comprender además *stubs* de detectores antimanipulación externos configurados para detectar la manipulación fuera de un núcleo principal y para unir más estrechamente el núcleo principal a una aplicación, y para camuflar su perfil de llamadas e interacciones API.

[0023] El núcleo de seguridad puede comprender además algoritmos criptográficos no estándar configurados para aumentar los costes de los intentos de aplicar ingeniería inversa al núcleo de seguridad.

[0024] El núcleo de seguridad puede comprender además al menos una tabla de saltos y/o enrutamiento API configurado para inicializarse solo en el tiempo de ejecución, o solo cuando se extrae de una red como parte del lanzamiento de la aplicación, como defensa contra el análisis estático y para frustrar los esfuerzos de ingeniería inversa manual.

[0025] Según otro aspecto de la invención, se proporciona un método de banca móvil, que comprende:

la publicación de una aplicación de banca móvil en una tienda de aplicaciones en internet;

la descarga de dicha aplicación de banca móvil en un dispositivo móvil desde internet;

el lanzamiento de dicha aplicación de banca móvil con el dispositivo móvil;

la detección de una primera instalación y la entrada en una fase de inicialización;

la generación de una clave de sesión KS y su cifrado bajo una clave pública codificada de forma rígida de un servidor, a saber «Kpub».

la entrega de una clave de sesión, {KS}Kpub, al servidor utilizando HTTP o HTTPS, en función de una URL proporcionada para el servidor que está codificada de forma rígida en la aplicación de banca móvil;

la generación de un número de serie único para el dispositivo móvil con el servidor cuando recibe la clave de sesión {KS}Kpub;

el envío de una solicitud procedente del servidor a un módulo de seguridad de *hardware* (HSM) para generar una clave a largo plazo K para el dispositivo móvil, y para utilizar la clave de sesión con el fin de cifrar la clave a largo plazo, el número de serie y algunos parámetros de configuración seleccionados por el servidor [{K,serial,settings}]KS, para el almacenamiento en una base de datos a largo plazo como un registro;

el reenvío de la [{K,serial,settings}]KS en un mensaje desde el HSM hasta el dispositivo móvil;

la devolución del mensaje de la [{K,serial,settings}]KS desde el servidor hasta la aplicación de banca móvil a través de un enlace HTTP-HTTPS;

el descifrado de la clave de sesión KS y la verificación de la integridad del mensaje para recuperar la clave K, y la recogida de los datos de configuración;

el almacenamiento de la clave a largo plazo K y de los datos de configuración en un almacenamiento local que es accesible solo por esa aplicación de banca móvil específica, y que está ofuscada por cifrado utilizando una clave codificada de forma rígida KO.

[0026] Las características del núcleo de seguridad se aplican igualmente al método.

[0027] En resumen, el modo de realización del núcleo de seguridad de la presente invención soporta una aplicación de banca móvil descargable para un teléfono inteligente, en forma de un conjunto de funcionalidades integrables para aplicaciones en la nube y móviles conectadas. Proporciona un entorno seguro para las aplicaciones en red en plataformas abiertas para realizar los flujos de trabajo de registro, inscripción y transacción con los servidores *backend* correspondientes en la red. Puede incluir defensas contra el análisis estático, intentos de ingeniería inversa y fraude de transacciones en tiempo real. Una defensa principal que puede emplearse es la ofuscación de los protocolos, API, algoritmos y código de programa. Puede detectar, frustrar, desviar e informar de forma activa los intentos de ingeniería inversa y la actividad de *malware* que detecta. Un ofuscador de enrutamiento puede configurarse para operar en la capa externa. Los diseños de núcleo anteriores pueden conservarse como camuflaje. Se puede utilizar una biblioteca TLS interna en lugar de la capa TLS del SO. Las *cookies* pueden administrarse internamente en el núcleo en lugar de en las bibliotecas del sistema operativo.

[0028] Estos y otros objetos y ventajas de la presente invención resultarán sin duda evidentes para los expertos en la materia después de haber leído la siguiente descripción detallada de los modos de realización preferidos que se ilustran en las diversas figuras de los dibujos.

EN LOS DIBUJOS

[0029]

La Fig. 1 es un diagrama de bloques funcional de un flujo de trabajo de registro de aplicaciones de banca en un modo de realización de la presente invención;

La fig. 2 es un diagrama de bloques funcional de un modo de realización de aplicación de banca de la presente invención y un servidor *backend*; y

La Fig. 3 es un diagrama de bloques funcional de un modo de realización de núcleo de seguridad CRM de la presente invención, como parte de esos dispositivos y sistemas de las Fig. 1-2.

DESCRIPCIÓN DETALLADA DE LOS MODOS DE REALIZACIÓN PREFERIDOS

[0030] Se proporciona un sistema de transacciones financieras y aplicación de banca con un flujo de trabajo de registro para establecer claves, identificadores únicos y soporte para la inscripción en todo el dispositivo. Las claves y la identificación de usuario (UID) son tales que permanecerán constantes y serán visibles en muchas aplicaciones diferentes en el mismo dispositivo, pero variarán con dispositivos diferentes. Una inscripción de instalación específica debe establecer una clave y una ID de usuario que sean únicas para una instalación individual de una aplicación. Esto proporciona además soporte para la caracterización, identificación, detección de *malware* y estado del parche-versión de *software* del dispositivo. Los datos de la plataforma inscrita se utilizan para la autenticación del cliente o de intentos automatizados, para el inicio de sesión en aplicaciones y para la aprobación de transacciones, incluida la autenticación-cifrado a nivel de transacciones. La autenticación y cifrado de datos a nivel de secuencia también puede incluirse para soportar la seguridad de las comunicaciones de aplicaciones de uso general.

[0031] Los vales-*cookies* o ID de sesión se generan para almacenar en caché los resultados de autenticación en las sesiones. Las transferencias de vales se proporcionan dentro de un ecosistema de aplicaciones y entre aplicaciones nativas, aplicaciones mixtas (nativas con componentes adicionales de vista web) y aplicaciones web puras. La gestión de compromisos del dispositivo incluye listas negras, y la aplicación de banca móvil realiza su propia verificación-aseguramiento de autointegridad. El diseño del protocolo es resistente a la denegación de servicio, manteniendo un procesamiento mínimo que puede ser causado antes de la autenticación.

[0032] En la Fig. 1, un flujo de trabajo de registro de aplicación de banca 100 forma la columna vertebral de una arquitectura de seguridad. El flujo de trabajo de registro 100 entrega una clave a largo plazo «Kpub» a una aplicación de banca 102 alojada en un dispositivo que aloja la aplicación 104. La Kpub se comparte con un proveedor de servicios de autenticación independiente 106, y puede utilizarse para implementar OATH, HOTP o TOTP, y otros protocolos de autenticación específicos como, p. ej., Kerberos o RADIUS, o utilizarse para iniciar la entrega de nuevas credenciales.

[0033] Una aplicación que contiene una clave pública programada «Kpub» correspondiente a un servidor 108 se descarga o se entrega de otra manera desde una tienda de aplicaciones. El servidor 108 oculta su mitad privada de la clave, «Kpriv», en un módulo de seguridad de *hardware* (HSM) 110. El HSM 110 es un coprocesador seguro de alto rendimiento destinado a evitar robos de la mitad privada de la clave Kpriv.

[0034] En un modo de realización del método de la presente invención, el proveedor de servicios de autenticación 106 publica la aplicación de banca 102 en una tienda de aplicaciones. Un usuario de la banca descarga la aplicación 102 en su dispositivo 104. El usuario lanza la aplicación 102. La aplicación 102 está configurada para detectar una primera instalación y entrar en una fase de inicialización. La aplicación 102 genera una clave de sesión KS* y la cifra bajo una clave pública codificada de forma rígida de un servidor 108, a saber «Kpub», indicado [KS]Kpub. La aplicación 102 entrega una clave de sesión, {KS}Kpub, al servidor 108 utilizando HTTP o HTTPS en una red de dispositivo o internet, en función de un localizador uniforme de recursos (URL) proporcionado para el servidor que también está codificado de forma rígida en la aplicación 102. Cuando el servidor 108 recibe la clave de sesión KS cifrada bajo Kpub, genera un número de serie único para el dispositivo 104, p. ej., un número largo aleatorio. El servidor 108 envía una solicitud al HSM 110 para generar una clave a largo plazo K para el dispositivo 104, y para utilizar la clave de sesión con el fin de cifrar la clave a largo plazo, el número de serie y algunos parámetros de configuración seleccionados por el servidor, p. ej., [{K,serial,settings}]KS, para almacenarse en una base de datos a largo plazo 112 como un registro 114. El HSM 110 reenvía la [{K,serial,settings}]KS en un mensaje al móvil 104. El servidor 108 devuelve el mensaje de la [{K,serial,settings}]KS a la aplicación de banca 102 a través del enlace HTTP-HTTPS. La aplicación de banca 102 utiliza la clave de sesión KS para descifrar y verificar la integridad del mensaje, para recuperar la clave K, y para recoger los datos de configuración. La aplicación de banca 102 almacena la clave a largo plazo K y los datos de configuración en un almacenamiento local 116 que es accesible solo por esa aplicación específica, y que está ofuscada por cifrado utilizando una clave codificada de forma rígida KO. La clave KO se deriva del cálculo de clave (*hashing*), p. ej., utilizando el algoritmo *hash* seguro de la Agencia de Seguridad Nacional (NSA, por sus siglas en inglés), SHA-1. El servidor 108 puede entregar posteriormente cualquier clave K a otro servicio bajo una clave de transporte, una clave maestra de zona (ZMK, por sus siglas en inglés), o puede proporcionar los servicios de autenticación en sí.

[0035] El flujo de trabajo de registro 100 entrega un secreto compartido entre dos partes conectadas con un identificador único. Depende del proveedor de servicios de autenticación permitir al usuario vincular su identificador de *token* de autenticación junto con su cuenta.

[0036] El flujo de trabajo 100 utiliza criptografía estándar, a saber, Rivest-Shamir-Adleman (RSA) o criptografía de curva elíptica (ECC, por sus siglas en inglés) y cifrado de Estándar de cifrado avanzado (AES, por sus siglas en inglés) con formatos de datos que son adecuados para su uso con un HSM en el lado del servidor. Dicha configuración permite que los servidores de registro escalen para soportar millones de solicitantes de registro sin hacer que el servidor sea vulnerable, puesto que el HSM se utiliza para proteger las claves. Los mismos algoritmos criptográficos se pueden implementar también de manera eficiente y efectiva en *software* puro en el dispositivo 104, sin depender de bibliotecas subyacentes que puedan cambiar. Por lo tanto, el núcleo de seguridad de la aplicación de banca es autónomo.

[0037] Si el proveedor de servicios de autenticación 106 tiene que cambiar el sitio de registro, pueden hacerlo con la asignación de DNS, o pueden publicar una actualización de la aplicación. El mismo proceso permite cambiar la clave pública codificada de forma rígida Kpub. El dispositivo 104 envía preferiblemente su información de modelo y número de serie durante el proceso de registro para permitir que el servidor 108 recopile estadísticas del uso de la plataforma.

[0038] La aplicación de banca 102 está configurada para no almacenar su clave de autenticación en ningún área del sistema de archivos que esté sujeta a copia de seguridad. Por lo tanto, cuando el dispositivo 104 es sustituido o recuperado de la copia de seguridad, será necesario volver a registrarlo para que se genere una nueva clave.

[0039] El enfoque de seguridad anterior depende totalmente de los controles de acceso a los archivos del sistema operativo para evitar el robo de cualquier clave. De esta manera, en cualquier dispositivo en el que el *malware* opera normalmente dentro de los permisos de la aplicación que se concedieron, las claves pueden permanecer seguras. Por ejemplo, puesto que no hay permiso para anular el acceso a todos los archivos del sistema de archivos.

[0040] Pero para el *malware* que tiene acceso a raíz, dichas claves serán triviales de recuperar. El almacenamiento de *keychain-keyring* (llaveros) de la clave no se utiliza deliberadamente ya que para muchos algoritmos de autenticación el acceso temporal a la clave es equivalente al robo de la clave, por lo que actualmente confiere pocas ventajas. Sin embargo, el uso de un llavero puede provocar la interacción con los servicios de copia de seguridad basados en la nube y es preferible evitar el almacenamiento de credenciales de usuario de manera centralizada con un proveedor de servicios externo.

[0041] En un modo de realización de arquitectura de seguridad mejorada de la presente invención, los usuarios reales del servicio de autenticación están sujetos a ataques, y esos ataques se pueden detectar. Si el ataque es una amenaza de *malware*, puede detectarse mediante: (1) la monitorización de las tiendas de aplicaciones en busca de aplicaciones de autenticación con nombres similares que pretenden ser la aplicación de banca móvil real 102; (2) la monitorización de registros de transacciones por fraude y servicios al cliente por conflictos; (3) la monitorización de informes antivirus de los proveedores de AV para móviles o establecimiento de una colaboración; y, (4) la recopilación de información.

[0042] Debido a la arquitectura de la aplicación de banca 102, es probable que, si se alcanza una segunda fase de amenaza, se vendan las vulnerabilidades de raíz de los dispositivos como, p. ej., los teléfonos inteligentes y sean ampliamente accesibles para los delincuentes en el mercado negro. En la segunda fase de amenaza, se deben incorporar contramedidas intrínsecas para limitar las capacidades del *malware* para robar credenciales de usuario. Y esto debe contener los ataques al menos hasta que los proveedores del sistema operativo hayan tenido tiempo suficiente para parchear sus vulnerabilidades, y todos los usuarios afectados puedan recuperar el control de sus dispositivos.

[0043] Si se entregó una vulnerabilidad de seguridad procedente de una tienda de aplicaciones descargando una aplicación maliciosa que utilizaba la escalada de privilegios, el proveedor de la tienda de aplicaciones estará en disposición de colaborar y proporcionar una lista de todos los usuarios que se han descargado la aplicación de autenticación y la aplicación maliciosa. Estos registros se pueden utilizar para enviar una advertencia de seguridad muy dirigida, y daría lugar a muy pocos falsos positivos.

[0044] El *malware* que infecta desde el *drive-by* de un sitio web no se registrará y se contabilizará tan fácilmente. Sin embargo, debe ser menos frecuente en la segunda fase de amenaza, puesto que requiere que coincidan dos vulnerabilidades, una que tome el control a través del navegador web, y una segunda que escale sus privilegios al nivel de la raíz. El reto del defensor es modificar el funcionamiento de la aplicación para disminuir el impacto del ataque actual, aunque solo sea la duración, y hasta que la vulnerabilidad se pueda parchear de manera permanente. Se pueden emplear varias técnicas para neutralizar temporalmente el *malware*.

[0045] La ofuscación de los datos almacenados en archivos en el disco se puede diseñar para disuadir a los usuarios de extraer claves trivialmente analizando los archivos almacenados al ejecutar la aplicación en un emulador, o en un teléfono inteligente liberado. Dicha ofuscación puede amplificarse para requerir esfuerzos de alto nivel por parte del atacante, p. ej., para volver a diseñar el algoritmo de recuperación de claves. Algunas técnicas básicas incluyen evitar que los atacantes recopilen los datos requeridos. Un atacante poco inteligente aplicaría ingeniería inversa en la aplicación de autenticación, implementaría la recuperación de claves en el *malware* y después cargaría solo la clave en el controlador de *malware*. Como contramedida, el simple cambio del algoritmo requeriría que el atacante rediseñara este tipo de *malware*, y les daría a los defensores un par de semanas de seguridad. Un atacante más inteligente podría hacer que el *malware* cargara todos los archivos almacenados creados por la aplicación del dispositivo, de manera que, si se cambiara un algoritmo de recuperación, el *malware* no necesitaría actualizarse y la recuperación podría realizarse sin conexión en una localización central.

[0046] Por lo tanto, los defensores deben aumentar la cantidad de almacenamiento utilizado para la clave, p. ej., al menos 10 MB de datos almacenados que podrían contener la clave. La complejidad de los datos de archivo almacenados puede aumentarse para crear archivos en una estructura compleja de directorio que aparece y desaparece en un proceso evolutivo durante varios días. Asegurarse de que los archivos que faltan o adicionales contribuyan al algoritmo de recuperación para que deje de funcionar. Los datos de configuración del dispositivo eventuales pueden integrarse en el formato de almacenamiento y los algoritmos de recuperación. En el caso de un teléfono inteligente, esto podría incluir el IMEI, el UDID, la fecha actual, los ajustes de volumen y localidad, etcétera. Cosas que un atacante puede haber olvidado recopilar cuando su *malware* carga los archivos de la aplicación en el operador de *malware*. Esto podría destrozar los datos que se recopilaron.

[0047] Las técnicas de confusión y ofuscación utilizadas en combinación pueden aumentar el tiempo que le lleva a un atacante hacer avanzar su *malware* hasta el punto en el que pueda recopilar los datos de registro a partir de nuevas infecciones.

[0048] La ofuscación de códigos se puede utilizar junto con la ofuscación de datos para retrasar a los atacantes que intentan verificar los algoritmos que utiliza la aplicación de banca móvil 102 para recuperar las claves. Forzar el *malware* para que trabaje, en lugar de simplemente interpretar los datos a largo plazo de la aplicación estática, observar la aplicación y después robar la clave en el momento de su uso.

[0049] Las aplicaciones bien diseñadas y resistentes utilizarán un mínimo de llamadas al sistema, especialmente llamadas al sistema criptográfico. Dichas llamadas son conexiones fáciles para el *malware* con control de raíz para acceder a las claves a pesar de los mecanismos de ofuscación que intervengan. La ofuscación de códigos es más fácil de realizar para el código «C» que para Java, lo que favorece la plataforma iPhone. Pero todavía tiene valor en ambas plataformas. Cualquier ofuscación que requiera que un atacante implemente un *rootkit* de monitorización para observar un proceso en ejecución en lugar de simplemente leer archivos del almacenamiento ya ha elevado el desafío y ganado una batalla significativa en la guerra global.

[0050] Los coprocesadores seguros pueden estar disponibles para almacenar claves criptográficas como tarjetas SIM, elementos seguros y módulos de plataforma segura (TPM). Pero si el coprocesador usa un algoritmo de autenticación con entradas de desafío predecibles, puede que no ayude. Por ejemplo, las contraseñas de un solo uso (OTP) totalmente basadas en tiempo y basadas en contador, y la firma de transacciones tienen desafíos predecibles. Un atacante puede simplemente utilizar acceso temporal al coprocesador de seguridad desde el sistema operativo (SO) de un procesador principal infectado con *malware* para recopilar un libro de claves o diccionario de entradas y salidas que represente todo lo necesario en el futuro para pretender tener posesión de la clave.

[0051] Solo cuando el servicio de autenticación proporciona un número de desafío verdaderamente aleatorio, como en un protocolo de respuesta de desafío, un atacante puede no tener ni idea antes de un intento de autenticación de qué datos es necesario procesar utilizando la clave almacenada.

[0052] Desafortunadamente, pasar a la autenticación de desafío-respuesta conlleva importantes sanciones de uso, ya que el usuario debe transferir el desafío al dispositivo móvil para firmarlo, ya sea introduciéndolo en el teclado o usando alguna transferencia automática, como códigos de barras 2D o patrones de parpadeo.

[0053] Una alternativa es implementar lógica inteligente en el coprocesador seguro que limite el tipo de mensaje que se puede procesar utilizando la clave. Por ejemplo, si la clave se utiliza para crear un HMAC de una entrada de autenticación, podría utilizar un contador monotónico interno para implementar un generador de contraseña HOTP. Aquí, un atacante puede utilizar su acceso comprometido al coprocesador para recopilar muchos cientos o miles de OTP, pero no podrán reestablecer el procesador seguro a su estado original, y, por lo tanto, es probable que se detecten automáticamente.

[0054] Por lo tanto, para un beneficio adecuado un coprocesador seguro necesita tener características de seguridad específicas disponibles, y los coprocesadores que simplemente almacenan una clave de manera segura, pero otorgan acceso para utilizarla para cualquier, fin tienen poco valor. En una estrategia a largo plazo, habrá límites en la efectividad de los coprocesadores seguros para la defensa y su valor probable puede estar principalmente en la oscuridad y crear una nueva barrera contra la cual el atacante aplique ingeniería inversa. Por lo tanto, pueden emplearse solo en la segunda fase de amenaza si son rentables.

[0055] A largo plazo, considerando especialmente un alejamiento de las OTP hacia la autenticación de transacciones, los coprocesadores seguros necesitarán tener trayectorias fiables para el usuario. Por ejemplo, para mostrar datos que están a punto de autenticarse y para buscar la aprobación o el rechazo en un lugar en el que el *malware* no puede interferir. Sin embargo, es poco probable que dicha interfaz de usuario (IU) fiable sea gráficamente muy atractiva, y es poco probable que las propuestas de IU fiables obtengan mucho apoyo de los fabricantes de teléfonos y es crucial contar con ellos.

[0056] Una sospecha creciente se dirige a la infraestructura de SSL-TLS global debido a una proliferación de autoridades de certificado raíz. Dichas sospechas representan una barrera importante para la utilización de cualquiera de las bibliotecas TLS proporcionadas por los fabricantes de dispositivos. Es posible que los fabricantes no tengan un conjunto configurable de certificados fiables, y eso hace que las aplicaciones de autenticación estén abiertas al ataque de adversarios poderosos que pueden suprimir las autoridades de certificación.

[0057] La capa externa de autenticación TLS puede considerarse poco fiable y eliminarse. En su lugar, se puede confiar en los algoritmos de cifrado patentados RSA-AES internos para proporcionar la protección necesaria que se perdería de otra manera. La eliminación de la TLS del protocolo podría mitigar los ataques de los *hackers* blancos y de reputación. En cuanto a la resistencia al *malware*, incluso si la TLS externa es ineficaz, no habría problema en dejarlo. No obstante, para protegerse contra un rango más amplio de amenazas a aplicaciones de autenticación, debe existir un plan para eliminar y reemplazar los mecanismos de TLS.

[0058] Los núcleos de seguridad de Cryptomathic tienen mecanismos para mitigar los riesgos de que un lego robe algunas credenciales de autenticación después de obtener acceso temporal al dispositivo de un individuo. Un PIN nunca es lo suficientemente bueno para evitar dicho acceso.

[0059] Un mecanismo para detectar la manipulación del reloj, donde en el caso de un *token* de OTP basado en tiempo, un atacante podría pedir prestado el dispositivo, adelantar el reloj varios días, luego ejecutar la aplicación de banca móvil 102 y recopilar futuros valores TOTP para usar en la autenticación, antes de devolver el tiempo a la normalidad. Se necesita un detector de manipulación del reloj para detectar este cambio en la secuencia de tiempo. Y para iniciar un mensaje de advertencia que pueda mostrarse al usuario, en un momento desconocido en el futuro, para que el atacante no pueda establecer el momento justo en el que va a aparecer este mensaje y luego descartarlo.

[0060] Del mismo modo, se incluye una verificación de IMEI-UDID al recuperar datos ofuscados del almacenamiento para detectar un ataque que intentó obtener una copia de seguridad completa de, p. ej., un teléfono inteligente y utilizarla para restaurar los datos en un teléfono diferente con la esperanza de obtener acceso a las credenciales de autenticación originales. También se debe incluir un mecanismo para evitar la copia de seguridad del material de claves en el PC del usuario o en la nube.

[0061] Las estrategias más avanzadas para proteger las aplicaciones de banca pueden centrarse mejor en romper el modelo económico criminal, en lugar de intentar defenderse de un ataque real. Una manera es reducir la rentabilidad de

los ataques exitosos, y en realidad es un trabajo mucho más sencillo. Al defenderse de las infracciones técnicas, debe protegerse un gran código base, y un atacante puede entrar en cualquier lugar, por lo que el desarrollador de la aplicación está en desventaja. Sin embargo, cuando la estrategia es atacar la base comercial de un delincuente, entonces sus operaciones deben ser resistentes en todas partes, y aquí radica la ventaja.

[0062] Los delincuentes han podido especializarse con la ayuda de un gran mercado y economía de mercado negro. Hoy en día, las vulnerabilidades pueden comprarse y venderse en el mercado, se pueden alquilar las redes de robots, etcétera. La economía del mercado negro no tiene acceso a la misma calidad de resolución de conflictos y marcos de trato justo que están disponibles para el mundo legítimo. Por lo tanto, un mercado distante es difícil de controlar para cualquiera de los malos.

[0063] Cualquier trampa de diseño que pueda introducir incertidumbre en el valor de «un compromiso en la aplicación X», o «1000 conjuntos de inicio de sesión recopilados para la aplicación X», u otro producto delictivo, reducirá considerablemente el valor percibido por los compradores. Cuanto más impredecibles sean las trampas, mejor para frustrar a los compradores. Por ejemplo, se podría utilizar una estrategia de actualización probabilística para una aplicación de banca móvil. Si el 75 % de los usuarios recibe una nueva versión A, y el 25 % recibe una versión B, entonces después de la actualización cuando un delincuente recopila datos, no conseguirá los datos de uno de los tipos de actualización. No se dan cuenta rápidamente de que necesitan soporte para ambos tipos de aplicaciones, por lo que una gran parte de los datos resulta inválida. Si se sorprenden por esto, una fricción en el mercado generaría problemas en la venta de los datos. Como los datos parcialmente malos pueden inhibir la cooperación. Los delincuentes tendrían que desarrollar formas de amortizar las recopilaciones de datos en conjuntos lo suficientemente grandes como para que no importe si algunas de las credenciales no son válidas.

[0064] Los excesos repentinos en la disponibilidad de los datos fácilmente recopilados, seguido de sequías puede hacer que los delincuentes se sobrepasen y, si no proyectan la cantidad de datos que podrán robar correctamente, entonces se comprometen en exceso a un orden y tienen que dejar a la otra parte.

[0065] Se puede utilizar la diversidad artificial para crear múltiples versiones de aplicaciones, rutinas de inicio de sesión, credenciales y contraseñas. Esto puede aumentar el esfuerzo necesario para un ataque en un ritmo mayor que el que se necesita para defender. Las versiones similares de aplicaciones con diferencias sutiles son mucho más difíciles de subyugar que las versiones totalmente diferentes.

[0066] Por lo tanto, un objetivo a largo plazo puede ser convencer a los delincuentes de que no ataquen la aplicación X, sino que ataquen las aplicaciones Y Z en su lugar. No con el argumento de que X es más seguro (es una carrera costosa entre X, Y y Z), sino porque hacer negocios continuos vendiendo credenciales de la aplicación X es demasiado arriesgado.

[0067] Este tipo de ataque de modelo de negocio era común en la industria de la TV de pago donde un grupo de atacantes bien financiado y organizado aplicaba ingeniería inversa y clonaba las tarjetas de suscripción de la TV de pago para vender suscripciones a un precio más barato. Después de las deficiencias iniciales de la industria, una estrategia principal fue implementar nuevas medidas de seguridad de forma iterativa con el siguiente lanzamiento justo antes de un gran evento deportivo, lo que significó que las tarjetas pirateadas dejaron de funcionar. Esto provocó que los clientes se quejaran a su proveedor del mercado negro, y dejó tiempo suficiente para que el cliente cambiara de opinión y comprara una suscripción genuina antes del comienzo del partido.

[0068] La naturaleza de una carrera armamentista en una plataforma cerrada como la X-Box o un teléfono móvil inteligente en lugar de en un PC es tal que estas defensas de modelos comerciales a menudo son muy efectivas.

[0069] A nivel técnico, si puede pagar a largo plazo para utilizar estos principios de alto nivel de aumentar el coste de un ataque a un nivel técnico profundo, así como a un nivel estratégico. Por ejemplo, se puede incluir criptografía no estándar para la que no haya implementaciones de referencia fáciles. Esta es una estrategia común para la seguridad militar, para mantener tanto la clave como el algoritmo en secreto. La criptografía no estándar no es lo mismo que la criptografía casera, siendo la diferencia que la primera consiste en modificaciones conservadoras realizadas a los algoritmos de criptografía existentes, como AES con rondas adicionales, o DES con diferentes cajas S, mientras que la segunda normalmente es un algoritmo nuevo pensado en la parte posterior de un sobre por un desarrollador sin experiencia criptográfica, y normalmente son variantes de los cifrados Viginère o Caesar, cuyos diseños tienen cientos de años y son sencillos de romper para un experto.

[0070] Muchos proveedores tienen diseñadores especializados de cifrado criptográfico simétrico y asimétrico en sus nóminas y están bien ubicados para desarrollar de forma segura variaciones de algoritmos no estándar. En resumen, la criptografía no estándar es fácil de escribir, pero es más difícil para aplicar ingeniería inversa y depurar, especialmente al mover implementaciones entre diferentes lenguajes.

[0071] En realidad, la comunidad delictiva carece de buenos conocimientos de criptografía (aunque está creciendo), como lo muestra la cantidad de virus nuevos que todavía utilizan algoritmos muy obsoletos como el cifrado de flujo RC4. Los autores de *malware* parecen reacios a utilizar código que no pueden copiar y pegar desde otro lugar, y aquí la criptografía no estándar es lo último en turbiedad. Los modos de realización avanzados de la presente invención amplían y subcontratan inteligencia y monitorización de las amenazas de seguridad móvil. Los servicios los ofrecen empresas de

antivirus y empresas más pequeñas especializadas en servicios de inteligencia. Las empresas de antivirus utilizan su amplia base de despliegue para monitorizar y recopilar estadísticas, y las empresas más pequeñas especializadas en servicios de inteligencia ponen la mayor parte de sus acciones en inteligencia humana (HUMINT).

[0072] HUMINT es una solución viable a largo plazo para infiltrarse y vigilar la economía y el mercado delictivo para obtener un aviso anticipado de un par de semanas a un par de meses sobre cómo se desarrolla la capacidad del mercado para proporcionar servicios para nuevos ataques.

[0073] En el caso de la seguridad de las tarjetas de pago, muchos observadores han participado en la supervisión y evaluación colaborativa de la información de inteligencia de los agentes de inteligencia humana para evaluar el ritmo al que las vulnerabilidades conocidas de seguridad de las tarjetas de pago, como la clonación de tarjetas SDA, se han extendido en la economía delictiva, en contraposición a la limitación de estas técnicas a experimentadores y aficionados. De esta manera, se puede detectar la diferencia entre un ataque futuro que no se ve desplegado a escala, y un ataque que está a punto de aumentar a escala.

[0074] Dicha inteligencia es inestimable para extender la duración de las medidas de seguridad en función de la oscuridad y el ataque económico. Las defensas pueden implementarse en el momento adecuado. Y lo que es más importante, puede vigilar la economía en busca de pruebas de un aumento inesperado de innovación que pueda amenazar con promover el fraude por encima de un nivel aceptable. Por lo tanto, la compra de tiempo adicional para implementar un conjunto de contramedidas aún mayor.

[0075] Por lo tanto, aunque los datos estadísticos de alto nivel sobre *malware* y amenazas pueden ser útiles para establecer presupuestos departamentales generales para determinados tipos de seguridad a un alto nivel en empresas, la inteligencia humana y la infiltración pueden proporcionar y proporcionan información realmente útil para el desarrollo iterativo de aplicaciones seguras.

[0076] La Fig. 2 representa un modo de realización de aplicaciones de banca de la presente invención y se hace referencia en el presente documento mediante el número de referencia general 200. La Fig. 2 caracteriza un dispositivo móvil 202 por sus componentes internos extraídos, incluidos sus componentes de banca móvil, una capa de Web-App de internet 204, y sus sistemas *backend* 206.

[0077] El dispositivo de aplicación de cliente 202 incluye una plataforma comercialmente disponible como, p. ej., una plataforma de teléfono inteligente que puede descargar y ejecutar «aplicaciones», como el iPhone de Apple o el Android de Google, y el WINDOWS Phone de Microsoft u ordenadores conectados en red como, p. ej., portátiles, PC, ordenadores en red con funcionalidad del lado cliente. Los modos de realización de la presente invención pueden cargarse y utilizarse en cualquiera de ellos o en todos por un cliente de banca móvil. Dicho cliente de banca móvil ya tendría normalmente un teléfono inteligente, y los modos de realización y funciones aquí se instalarían y se utilizarían bajo la dirección y el control de un banco particular u otra institución financiera.

[0078] El dispositivo de aplicación de cliente 202 incluye una capa de presentación de interfaz de usuario (IU) 210, una aplicación de banca móvil 212 que incluye un núcleo de seguridad de Cryptomathic (CRM) 214 con varios *stubs*, y un sistema operativo 216 correspondiente, como iOS, Blackberry y Android. La interfaz de usuario (IU) puede implementarse a través de una combinación de pantallas locales y datos remotos presentados por vista web proporcionados por un servidor.

[0079] La seguridad del dispositivo y del sistema es de suma importancia, y el hecho de que ningún *hardware* de seguridad especial o reservado esté disponible en el teléfono inteligente normal o en el dispositivo de aplicación de cliente 202 hace que una realización exitosa sea más desafiante. En lo que respecta al dispositivo de aplicación de cliente 202, toda la seguridad debe implementarse en el *software* y debe resistir los intentos de los estafadores de interponerse en el medio de las transacciones y también frustrar los intentos de ingeniería inversa. Por lo tanto, al menos la aplicación de banca 212 y los sistemas *backend* 206 están configurados para buscar y detectar de forma activa *malware*, intentos de ingeniería inversa, ataques de intermediarios y una serie de otras cosas malas.

[0080] La capa de Web-App de internet 204 incluye una capa de seguridad de perímetro 220, una capa de servidor HTTP 222, una capa de inscripción-registro-autenticación 224, y una aplicación Websphere Application Server 226 con un autenticador de sesión CRM 228. Hay dos clases de comunicación entre la aplicación de banca móvil 212 y su sistema de banca móvil correspondiente, (1) protocolos generales utilizados por inscripción, registro y autenticación, y todas las demás aplicaciones, y (2) protocolos específicos de aplicación como se utilizan en las transacciones.

[0081] El componente más desafiante de construir es el núcleo de seguridad de la aplicación en sí, soportado en varias plataformas y extendido con lanzamientos frecuentes según un plan de desarrollo y un plan de seguridad. Los horarios de lanzamiento internos frecuentes permitirían que los cambios en los componentes del lado del cliente se implementaran habitualmente al menos trimestralmente y en cuestión de días-semanas en caso de respuesta de emergencia a amenazas.

[0082] Los sistemas *backend* 206 incluyen un servidor de registro 230 y un administrador de autenticación de registro interno 232 que depende de un módulo de seguridad de hardware (HSM) 234. Los sistemas *backend* 206 incluyen además una base de datos específica de canal 236, una base de datos de cliente 237, un servidor de banca por Internet (IB, por

sus siglas en inglés) 238, y un sistema de banca 239 de núcleo de seguridad de CRM 214. La infraestructura del lado de servidor de banca móvil utilizaría normalmente la terminación TLS de la cuenta, IDS, monitorización de eventos de seguridad, detección de anomalías, y otros métodos de la mejor práctica.

[0083] El HSM 234 se puede implementar con HSM compatibles con PKCS#I1, p. ej., HSM de tarjeta PCI SafeNet ProtectServer o HSM conectado a la red nCipher Connect de Thales o incluso se puede ejecutar en *software* si se considera aceptable para el modelo de amenaza de seguridad. En criptografía, PKCS#11 forma parte de una familia de estándares llamados *Public-Key Cryptography Standards* (PKCS) (estándares de criptografía de clave pública), publicados por RSA Laboratories. Define una interfaz de programación de aplicaciones (API) independiente de la plataforma para los *tokens* criptográficos, como los módulos de seguridad de *hardware* y las tarjetas inteligentes. El estándar PKCS#11 se refiere a una API «Cryptoki» que es una concatenación de «*cryptographic token interface*» (interfaz de *token* criptográfico) pronunciada «*crypto-key*» (clave criptográfica). El «PKCS#11» también se puede utilizar para referirse a la API.

[0084] El servidor de registro 230 maneja cada establecimiento de claves de instancia de la aplicación y del dispositivo a largo plazo entre su módulo de seguridad de *hardware* 234 y el punto de conexión. Este vuelve a cifrar estas claves para un almacenamiento seguro como una instancia de dispositivo-aplicación en la base de datos 236. A dichas claves de registro almacenadas de forma segura también se puede acceder a través de la aplicación utilizando el núcleo de seguridad, el administrador de autenticación de registro interno 232 utiliza el mismo almacenamiento de claves para proporcionar un acceso seguro a estas claves desde la capa del Websphere Application Server (servidor de aplicaciones Websphere) 226 para la autenticación de sesiones-transacciones por parte del autenticador de CRM 228.

[0085] Los servidores de inscripción y aplicación pueden configurarse con módulos de Java que se encuentran en la capa de Websphere Application Server 204. Estos implementan diversos protocolos y mecanismos de verificación de autenticación que se combinan con los servicios proporcionados por el núcleo de seguridad de la aplicación de banca móvil 214.

[0086] En general, los modos de realización de la presente invención incorporan técnicas de resistencia de ingeniería inversa, identificación de dispositivo y detección de *malware*. Ya existen muchos dispositivos y métodos convencionales para implementar estas características. Sería importante mantener como secreto comercial las cosas exactas incorporadas en cualquier implementación y cómo están destinadas a responder.

[0087] El núcleo de seguridad de CRM 214 contiene módulos de Cryptomathic configurados para soportar la administración de múltiples claves almacenadas en un almacenamiento con sistema operativo o *hardware* seguros. Dichas claves son tanto por dispositivo como por instalación de aplicación. Los módulos de almacenamiento seguro en el núcleo de seguridad de CRM 214 aprovechan las claves almacenadas del llavero y los secretos entregados externamente. El núcleo de seguridad de CRM 214 soporta la administración de sesiones, un protocolo de autenticación para el inicio de sesión dentro de una sesión, exportación de *cookies*-vales de autenticación y seguridad de comunicaciones basadas en secuencias a través de un servidor proxy interno dentro del núcleo de seguridad de CRM 214. La identificación de dispositivos de terceros y la tecnología de detección (compromiso) de *malware* se pueden integrar en el núcleo de seguridad de CRM 214, p. ej., similar a los productos comercializados por Trusteer (Boston, MA) y ThreatMetrix, Inc. (San José, CA).

[0088] Una evolución estratégica del núcleo de seguridad de CRM 214 incluye la migración a la protección segura de elemento-SIM.

[0089] El mantenimiento puede incluir el mantenimiento y monitorización de señuelos, con alertas en una semana de actividad y resúmenes trimestrales.

[0090] La Fig. 3 representa una versión del núcleo de seguridad 214 de la aplicación de banca, en un modo de realización de la presente invención al que se hace referencia en el presente documento mediante el número de referencia general 300. El núcleo de seguridad 300 incluye en sus extremos exteriores varias interfaces de programación de aplicaciones (API) ficticias 302-306 y *stubs* de manipulación 307 conectados mediante un ofuscador de enrutamiento 308 que redirige a un conjunto correspondiente de API reales 312-316. Los *stubs* externos de detector de antimanipulación 307 se utilizan para detectar la manipulación desde fuera del núcleo principal, y para permitir que el núcleo esté unido de manera más estrecha a la aplicación. Esto puede servir también para camuflar los perfiles de llamada y las interacciones API.

[0091] Un controlador de ofuscador de enrutamiento (RO) 319 proporciona un mecanismo dinámico para cambiar la ofuscación según tablas que se inicializan solo en tiempo de ejecución. La inicialización del enrutamiento API o tablas de saltos solo en tiempo de ejecución proporciona una defensa efectiva contra el análisis estático y puede frustrar los intentos de ingeniería inversa manual. Las API externas incluyen API de almacenamiento de datos 302 & 312, API de control de inscripción 303 & 313, API de sesión 304 & 314, API de transacción 305 & 315 y API de secuencia 306 & 316.

[0092] La inspiración de un ofuscador de enrutamiento es que puede ocultar qué llamadas en las API externas 303-308 corresponderán a qué cuadros de la fila de arriba, p. ej., API de almacenamiento de datos...ctl. RO. Es como una permutación secreta en un cifrado de sustitución, salvo que cambia cada vez que se realiza una llamada, como el cifrado de rotor de la máquina Enigma de las fuerzas armadas alemanas que se hizo famoso en la Segunda Guerra Mundial. Por

esta razón, la fila de arriba es casi idéntica a la fila de abajo, a excepción de las llamadas de «control del ofuscador de enrutamiento» que nunca están expuestas al usuario final, y los «*stubs* de manipulación» que permiten llamadas externas ficticias que no invocan nada en el interior. El ofuscador de enrutamiento también se mezcla en las asignaciones potenciales, pero no reales, con las funciones en las versiones de núcleo anteriores al mismo tiempo, haciendo que parezca factible que también se pueda llamar a ese código.

[0093] Las API orientadas hacia el interior del sistema operativo incluyen una interfaz de administración de claves iOS 320, una interfaz de administración de claves 321, un módulo de protección 322, una interfaz de llamada de identificación (ID) del dispositivo SO 323 e interfaces diversas 324. Una interfaz de comunicaciones 325 y una API de almacenamiento del sistema operativo (SO) 326 están a la izquierda en la ilustración.

[0094] Dentro del núcleo de seguridad de CRM 300 hay un núcleo interno 330 construido con una base de código independiente de la plataforma que se puede compartir económicamente en todas las plataformas que soportan el código nativo. Las versiones de núcleo antiguas 332 y el código asociado se integran en versiones más nuevas de la aplicación como código de camuflaje. El conjunto aparece como un globo no diferenciado, lo que obliga a analizar las rutas de tiempo de ejecución para ordenar la masa. El código original, aunque obsoleto, es, con mucho, el mejor camuflaje posible, ya que es 100 % plausible en la primera inspección. Dicha funcionalidad del núcleo anterior se incluye en paralelo para la resistencia de ingeniería inversa y es hipotéticamente, pero no realmente, accesible a través del ofuscador de enrutamiento 302.

[0095] Las aplicaciones de banca acceden al núcleo de seguridad 300 para administrar sus procesos de inscripción, establecer sesiones seguras, autenticarse, comunicarse de forma segura con los servidores de banca móvil y almacenar de forma segura sus propios datos en el sistema de archivos. Los historiales de transacciones recientes pueden mantenerse localmente para verlos sin conexión a través de la API de almacenamiento de datos 302 & 312.

[0096] La API de control de inscripción 303 & 313 lanza los procesos de inscripción para crear identidades de dispositivo a través de la creación de perfiles, para autenticar las credenciales de usuario existentes en los servidores de banca móvil, registrar la identidad del dispositivo y luego compartir claves criptográficas. La API de inscripción 303 & 313 manipula una máquina de estado de inscripción 341 e incluye instrucciones del programa para iniciar la creación de perfil de un dispositivo para el estado de identidad, *malware* y liberación, para comenzar a autenticarse en servidores *backend* utilizando datos proporcionados por el cliente, y para intentar crear un cliente para las conexiones de instancia de la aplicación.

[0097] La API de sesión 304 & 314 establece un túnel externo TLS y lleva a cabo protocolos de autenticación de instancias de aplicaciones de banca para demostrar la identidad de la aplicación a los servidores. Los comandos de llamada «authUp» se utilizan para elevar el nivel de autenticación de una sesión al obtener credenciales de usuario y factores de seguridad adicionales. Estos pueden incluir respuestas a desafíos a través de comunicaciones fuera de banda para lo que tiene, lo que sabe, quién es, etc. Luego puede exportar los niveles de autorización de sesión actuales a la aplicación usando vales-*cookies*.

[0098] La API de transacción 305 & 315 protege los mensajes de transacción utilizando el código de autenticación de mensajes (MAC, por sus siglas en inglés) simétrico interno MACing y la seguridad de TLS externa para la confidencialidad, y proporciona integridad de transacción e integración en el registro de auditoría. Con el código de autenticación de mensajes basado en *hash* (HMAC), el cifrado es opcional. El valor predeterminado es dejar esto a la TLS para habilitar la inspección de seguridad perimetral en la banca móvil. Dichos servicios API se incluyen para satisfacer los requisitos legales y de cumplimiento.

[0099] La API de secuencia 306 & 316 permite que las solicitudes HTTP de WebView, el componente de control de la IU del navegador u otros componentes externos de la aplicación se enruten a través del núcleo de seguridad de CRM 300 a través de un proxy 342 que maneja *cookies* de autenticación adicionales, encapsulación de TLS y realiza algunos filtros de seguridad (URL, enlaces, etc.). La API de secuencia 306 & 316 permite que las aplicaciones se comuniquen de manera segura sin depender de los desarrolladores para administrar de manera segura el almacenamiento y eliminación de *cookies*, y protege las capacidades del navegador web.

[0100] Los *stubs* de seguridad 307 permiten que se utilicen llamadas API ficticias además de las llamadas principales, p. ej., para disfrazar las firmas de interacción entre el núcleo principal y una aplicación externa, y también para permitir que el núcleo de seguridad 300 enriquezca su interacción y se conecte a la aplicación principal.

[0101] Un administrador de claves de instancia de aplicación 350 proporciona una interfaz para claves criptográficas únicas para una instancia particular de una aplicación de banca concreta. Estos generalmente consisten en un identificador de la aplicación (aplicación y versión de la aplicación), una clave de almacenamiento simétrica, un identificador único de instalación y una clave asimétrica para la autenticación de la aplicación que se puede usar para firmar mensajes de protocolo. El administrador de claves de instancia de aplicación 350 almacena sus claves a través de una interfaz de uso y administración de claves unificada que abstrae los detalles específicos de la plataforma del almacenamiento de claves disponible y, si es necesario, emula servicios seguros.

[0102] Un administrador de claves de dispositivo 351 proporciona una interfaz para claves criptográficas de todo el dispositivo que se configuran durante un primer registro de una primera aplicación de banca móvil instalada, y que permanecen persistentes e invariables dentro de este grupo. Estas se almacenan a través de una interfaz de administración de claves 352 de la misma manera que las claves de instancia de aplicación, pero con pequeñas diferencias de configuración para permitir diferentes niveles de visibilidad. Las claves de todo el dispositivo administradas por este módulo se usan solo durante un flujo de trabajo de registro 100 (Fig. 1) donde tienen un papel que desempeñar para demostrar que una instancia de aplicación recién instalada puede desarrollar la confianza existente establecida.

[0103] Un administrador de identidad 353 proporciona servicios de identificación de dispositivos y huellas digitales. Los poderes asignados al administrador de identidad 353 dependen del nivel de acceso que necesita la aplicación de banca móvil para acceder a los datos personales protegidos del sistema operativo. Por ejemplo, la información sobre los contactos, la ubicación, el historial de ubicaciones, la zona horaria, las *cookies*, y similares, actuales del dispositivo pueden crear un perfil de un dispositivo, pero también se puede transformar con un algoritmo *hash* en un código de identificación aproximado. Dicho código se puede compartir con los servidores de banca para identificar el dispositivo más allá de las métricas explícitas del dispositivo que ofrecen algunos sistemas operativos, p. ej., el UDID del dispositivo.

[0104] Un módulo de fuente de tiempo protegido 354 proporciona marcas de tiempo a otros módulos en función de un reloj del sistema, pero con monitorización y métricas adicionales para detectar cualquier intento de manipulación del reloj y ataques de actualización.

[0105] Un módulo de almacenamiento seguro 355 proporciona almacenamiento ofuscado (cifrado con una clave codificada de forma rígida) y almacenamiento seguro basado en claves de almacenamiento a las que se accede desde la instancia de la aplicación y los administradores de claves del dispositivo 350 y 351. El módulo de almacenamiento seguro 355 permite que las aplicaciones de banca contengan archivos de datos seguros para mantener un estado sensible a largo plazo. El módulo de almacenamiento seguro 355 también puede mantener una capa secundaria de claves que permite un procesamiento criptográfico uniforme, independientemente de los niveles de soporte del algoritmo ofrecidos a través de la interfaz de administración de claves en las propias API criptográficas del dispositivo.

[0106] El módulo de almacenamiento seguro 355 incluye un mecanismo antirrobo 356 que dispersa y expande el almacenamiento utilizando algoritmos de expansión patentados y técnicas resistentes de ingeniería inversa. Esto significa, por ejemplo, que una clave podría calcularse de manera determinista a partir de un archivo de 50 MB en *flash*, o más bien un árbol de entrelazado y modificación de archivos, que es eficiente de realizar localmente, pero que crea desafíos de escala en caso de que un autor de *malware* desee escribir *malware* simple que simplemente recopile archivos de datos y los cargue en un servidor central. Esto obliga al *malware* a detectar y robar los datos de los archivos locales durante el tiempo de ejecución y este tipo de *malware* es mucho más costoso y difícil de escribir. El elemento de almacenamiento seguro incluye protección contra ataques de reversión de almacenamiento del dispositivo utilizando un contador de escritura de almacenamiento creciente que se mantiene en un llavero.

[0107] Un módulo de servicios criptográficos 358 implementa combinaciones de criptografía RSA, AES, funciones *hash* y una construcción HMAC que están incorporadas en el núcleo de seguridad y no dependen de API criptográficas externas con estándares que pueden estar bajo flujo. Las operaciones criptográficas se pueden realizar localmente y sin aceleración, ya que solo se manejan mensajes cortos de protocolo de unas pocas decenas de kilobytes como máximo y se utiliza el almacenamiento local para los datos de la transacción. Dado que el módulo de servicios criptográficos 358 está separado, se mejora la portabilidad del núcleo de seguridad entre plataformas. Eso, a su vez, economiza en las consiguientes pruebas y costes de mantenimiento.

[0108] Los algoritmos no estándar, como las versiones modificadas de AES, se pueden utilizar para aumentar el trabajo requerido para la ingeniería inversa. Dichos algoritmos exigen que se reconstruyan tediosamente a mano y que un atacante los compruebe antes de que puedan usar los resultados para descifrar datos de mensajes de protocolo o partes polimórficas del código. La mayoría de los que utilizan ingeniería inversa, en algún momento, implementarán rutinas de descifrado en un lenguaje de *scripting* adjunto al desensamblador, p. ej., IDA-python, y esto aumenta el coste de hacerlo. Si existen problemas de cumplimiento, los algoritmos estándar y no estándar se pueden utilizar de forma secuencial.

[0109] Un módulo de protocolo de registro 360 y un módulo de protocolo de sesión 362 implementan protocolos seguros de distribución de claves que establecen claves criptográficas a largo plazo y a nivel de sesión entre el núcleo de seguridad 214 y los sistemas *backend* 206 de la banca móvil. Estos protocolos se operan a través de la canalización de comunicaciones de TLS externa. Es posible la generación de claves a largo plazo en el dispositivo y la generación central de claves a largo plazo con entrega al dispositivo bajo una clave efímera.

[0110] Los ofuscadores de protocolo 364 y 366 dificultan el acceso al módulo de protocolo de registro 360 y al módulo de protocolo de sesión 362. Los protocolos de registro y administración de sesión 360 y 362 suelen estar en un estado lento de flujo y evolución a medida que se agregan nuevas características y a medida que madura el ecosistema de la aplicación. La tasa aparente de dicho flujo se puede acelerar artificialmente para una mejor resistencia a la ingeniería inversa. Esto puede ayudar a generar aplicaciones malvadas ficticias y modificadas que no funcionan según lo previsto. Por lo tanto, estos protocolos incluyen un ofuscador de protocolo y varias versiones arbitrariamente diferentes del protocolo destinadas a crear confusión y complejidad en un punto crítico en el desarrollo de un ataque.

[0111] Los *stubs* internos de detector antimanipulación 370 se distribuyen a través de monitores que intentan detectar las desviaciones del funcionamiento normal del programa que pueden surgir debido a la modificación del código, y marcar estas excepciones de una manera para que se pueda actuar sobre ellas más tarde después de una revisión adicional.

[0112] El tráfico de cobertura se puede utilizar de manera efectiva para ocultar el tráfico importante a simple vista. El núcleo de seguridad puede cogenerar mensajes de protocolo y tráfico superfluos para molestar a los atacantes y disfrazar mecanismos que dispersan y expanden el almacenamiento utilizando algoritmos de expansión patentados y técnicas resistentes de ingeniería inversa. El *malware* simple que simplemente recopila archivos de datos y los carga en un servidor central no hará el trabajo. El elemento de almacenamiento seguro 355 incluye protección contra ataques de reversión de almacenamiento del dispositivo utilizando un contador de escritura de almacenamiento creciente que se mantiene en el llavero.

[0113] En general, el núcleo de seguridad 214, 300 se centra principalmente en la seguridad a través de la ofuscación. Un ofuscador de enrutamiento opera en la capa externa. Los núcleos anteriores se incluyen como camuflaje, se utiliza una biblioteca TLS interna en lugar de la capa TLS del sistema operativo, y las *cookies* se administran internamente en el núcleo en lugar de en la capa del navegador webkit.

[0114] Los modos de realización del núcleo de seguridad de la presente invención implementan un protocolo de registro, almacenamiento cifrado y flujos de trabajo de registro e inscripción de banca móvil. Además, soportan flujos de trabajo criptográficos TOTP-HOTP y tiempo seguro, que ahora no se incluyen en los flujos de trabajo de banca móvil convencionales. El núcleo de seguridad de CRM 214, 300 se puede implementar tanto en Java como en "C" con un contenedor Objective-C compacto, y preferiblemente se aloja de forma simultánea, p. ej., en las plataformas de teléfonos inteligentes iOS de Apple, Android de Google y Blackberry.

[0115] Las aplicaciones OTP móviles convencionales de Cryptomathic ya tienen un marco de informe de errores y pautas de diseño de IU que asignan códigos categorizados a estados de error y manejan fácilmente las condiciones de error. Depende en gran medida de los proveedores de banca móvil recopilar los datos adecuados en los centros de atención telefónica de los clientes y configurar canales de retroalimentación para devolver información a los equipos de desarrollo para informes de errores.

[0116] Se proponen varias estrategias de resistencia de ingeniería inversa para resistir diferentes clases de atacantes, dependiendo de las habilidades, el nivel educativo y la ubicación de los atacantes. Cada una de las estrategias tiene diferentes niveles de efectividad e incurre en diferentes tipos de costes: algunas incurren en costes de licencia, algunas cuestan horas de mano de obra durante el desarrollo y algunas cuestan horas de mano de obra durante la depuración de pruebas de producción.

[0117] Las técnicas para la resistencia de ingeniería inversa se pueden aplicar en el tiempo de compilación para frustrar los descompiladores y los desensambladores comerciales, y en el tiempo de diseño para crear confusión conceptual y engañar sistemáticamente a los que aplican ingeniería inversa, lo que aumenta sus costes de adaptación a las posteriores actualizaciones de la aplicación.

[0118] Las técnicas se pueden preparar y probar con anticipación, pero la implementación se mantiene mejor durante el mayor tiempo posible. La implementación de contramedidas se basa en la recopilación de información y los programas de monitorización de fraude de banca móvil, y está optimizada para el máximo trastorno financiero para la empresa del fraude en comparación con el coste de la contramedida que se ha de desarrollar y el tamaño actual de las existencias de contramedidas. Durante un primer año de implementación, las contramedidas pueden implementarse más rápidamente al atribuir un mayor valor monetario a la contramedida como una forma de proteger la marca de banca móvil en la transición a la plena aceptación de la banca móvil en toda la base de clientes. Una vez que se alcanza la aceptación total, se ha de aceptar un cierto nivel de fraude.

[0119] Finalmente, se pueden usar varias implementaciones separadas producidas independientemente del mismo módulo funcional, y que pueden intercambiarse a voluntad, para crear un delta de código aparentemente alto entre actualizaciones menores de una aplicación. La promulgación de una implementación para «actualizar» otra se utiliza para aumentar sintéticamente el coste de adaptación del *malware* para atacar «nuevos e importantes» lanzamientos de aplicaciones.

[0120] Aunque la presente descripción se ha descrito con relación a los modos de realización preferidos actualmente, ha de entenderse que la exposición no ha de interpretarse como limitativa. Tras la lectura de la exposición anterior, resultarán evidentes varias alteraciones y modificaciones para los expertos en la materia.

REIVINDICACIONES

1. Método implementado por una aplicación de banca móvil (102) instalada en un dispositivo móvil (104) para proporcionar un entorno seguro para la aplicación de banca móvil (102), conteniendo la aplicación de banca móvil una clave pública codificada de forma rígida Kpub de un servidor (108) y suministrada al dispositivo móvil desde una tienda de aplicaciones, comprendiendo el método:

el lanzamiento en el dispositivo móvil;

la detección de una primera instalación y la entrada en una fase de inicialización;

la generación de una clave de sesión KS* y el cifrado de la clave de sesión KS* bajo la clave pública codificada de forma rígida Kpub para generar una clave de sesión cifrada {KS}Kpub;

la entrega de la clave de sesión cifrada, {KS}Kpub, al servidor utilizando un enlace que comprende un enlace HTTP o un enlace HTTPS en función de una URL proporcionada para el servidor que está codificada de forma rígida en la aplicación de banca móvil;

la recepción de información cifrada [{K,serial,settings}]KS* en un mensaje procedente del servidor por el enlace, comprendiendo la información cifrada [{K, serial, settings}]KS* un número de serie único para el dispositivo móvil, una clave a largo plazo K para el dispositivo móvil, y parámetros de configuración seleccionados por el servidor, cifrándose la información cifrada [{K, serial, settings}]KS* utilizando la clave de sesión KS*;

la utilización de la clave de sesión KS* para descifrar y verificar la integridad del mensaje para recuperar la clave a largo plazo K, y recoger los parámetros de configuración; y

el almacenamiento de la clave a largo plazo K y de los parámetros de configuración en un almacenamiento local (116) que es accesible solo por la aplicación de banca móvil, donde la clave a largo plazo K está ofuscada por cifrado utilizando una clave codificada de forma rígida KO.

2. Dispositivo móvil (104) que comprende una aplicación de banca móvil (102) instalada en el dispositivo móvil, y un almacenamiento local (116) que es accesible solo por la aplicación de banca móvil, donde la aplicación de banca móvil está configurada para:

el lanzamiento en el dispositivo móvil;

la detección de una primera instalación y la entrada en una fase de inicialización;

la generación de una clave de sesión KS* y el cifrado de la clave de sesión KS* bajo una clave pública codificada de forma rígida Kpub de un servidor (108) para generar una clave de sesión cifrada {KS}Kpub;

la entrega de la clave de sesión cifrada, {KS}Kpub, al servidor utilizando un enlace que comprende un enlace HTTP o un enlace HTTPS en función de una URL proporcionada para el servidor que está codificada de forma rígida en la aplicación de banca móvil;

la recepción de información cifrada [{K,serial,settings}]KS* en un mensaje procedente del servidor por el enlace, comprendiendo la información cifrada [{K, serial, settings}]KS* un número de serie único para el dispositivo móvil, una clave a largo plazo K para el dispositivo móvil, y parámetros de configuración seleccionados por el servidor, cifrándose la información cifrada [{K, serial, settings}]KS* utilizando la clave de sesión KS*;

la utilización de la clave de sesión KS* para descifrar y verificar la integridad del mensaje para recuperar la clave a largo plazo K, y recoger los parámetros de configuración; y

el almacenamiento de la clave a largo plazo K y los parámetros de configuración en dicho almacenamiento local, donde dicha clave a largo plazo K está ofuscada por cifrado utilizando una clave codificada de forma rígida KO.

3. Método llevado a cabo por un servidor (108) en comunicación por red con un dispositivo móvil (104), comprendiendo el método:

la recepción de una clave de sesión cifrada, {KS}Kpub, procedente de una aplicación de banca móvil (102) instalada en el dispositivo móvil mediante un enlace que comprende un enlace HTTP o un enlace HTTPS, la clave de sesión cifrada generada por la aplicación de banca móvil al cifrar una clave de sesión KS* bajo una clave pública codificada de forma rígida Kpub del servidor;

la generación de un número de serie único para el dispositivo móvil;

el envío de una solicitud procedente del servidor a un módulo de seguridad de *hardware* (110) para generar una clave a largo plazo K para el dispositivo móvil, y para utilizar la clave de sesión con el fin de cifrar la clave a largo plazo, el número de serie y algunos parámetros de configuración seleccionados por el servidor para generar

información cifrada [{K,serial,settings}]KS*, para el almacenamiento en una base de datos a largo plazo (112) como un registro;

la recepción de la información cifrada [{K,serial,settings}]KS* en un mensaje procedente del módulo de seguridad de *hardware*; y

la transmisión del mensaje desde el servidor hasta la aplicación de banca móvil en el dispositivo móvil por el enlace.

4. Servidor (108) en comunicación por red con un dispositivo móvil (104), estando configurado el servidor para:

la recepción de una clave de sesión cifrada, {KS}Kpub, procedente de una aplicación de banca móvil (102) instalada en el dispositivo móvil mediante un enlace que comprende un enlace HTTP o un enlace HTTPS, siendo generada la clave de sesión cifrada por la aplicación de banca móvil al cifrar una clave de sesión KS* bajo una clave pública codificada de forma rígida Kpub del servidor;

la generación de un número de serie único para el dispositivo móvil;

el envío de una solicitud procedente del servidor a un módulo de seguridad de *hardware* (110) para generar una clave a largo plazo K para el dispositivo móvil, y para utilizar la clave de sesión con el fin de cifrar la clave a largo plazo, el número de serie y algunos parámetros de configuración seleccionados por el servidor para generar información cifrada [{K,serial,settings}]KS*, para el almacenamiento en una base de datos a largo plazo (112) como un registro;

la recepción de la información cifrada [{K,serial,settings}]KS* en un mensaje procedente del módulo de seguridad de *hardware*; y

la transmisión del mensaje desde el servidor hasta la aplicación de banca móvil en el dispositivo móvil por el enlace.

5. Servidor de la reivindicación 4, donde la clave pública Kpub forma parte de un par de claves de servidor, y el servidor está configurado para conservar una mitad privada, Kpriv, del par de claves de servidor en el módulo de seguridad de *hardware*.

6. Servidor de la reivindicación 4 o 5, donde el servidor está configurado para entregar la clave a largo plazo K a un proveedor de servicios de autenticación bajo una clave de transporte.

7. Servidor de cualquiera de las reivindicaciones 4 a 6, donde el servidor está configurado para almacenar la información cifrada [{K,serial,settings}]KS* en la base de datos a largo plazo como un registro.

8. Sistema que comprende:

el dispositivo móvil de la reivindicación 2; y

el servidor de cualquiera de las reivindicaciones 4 a 7.

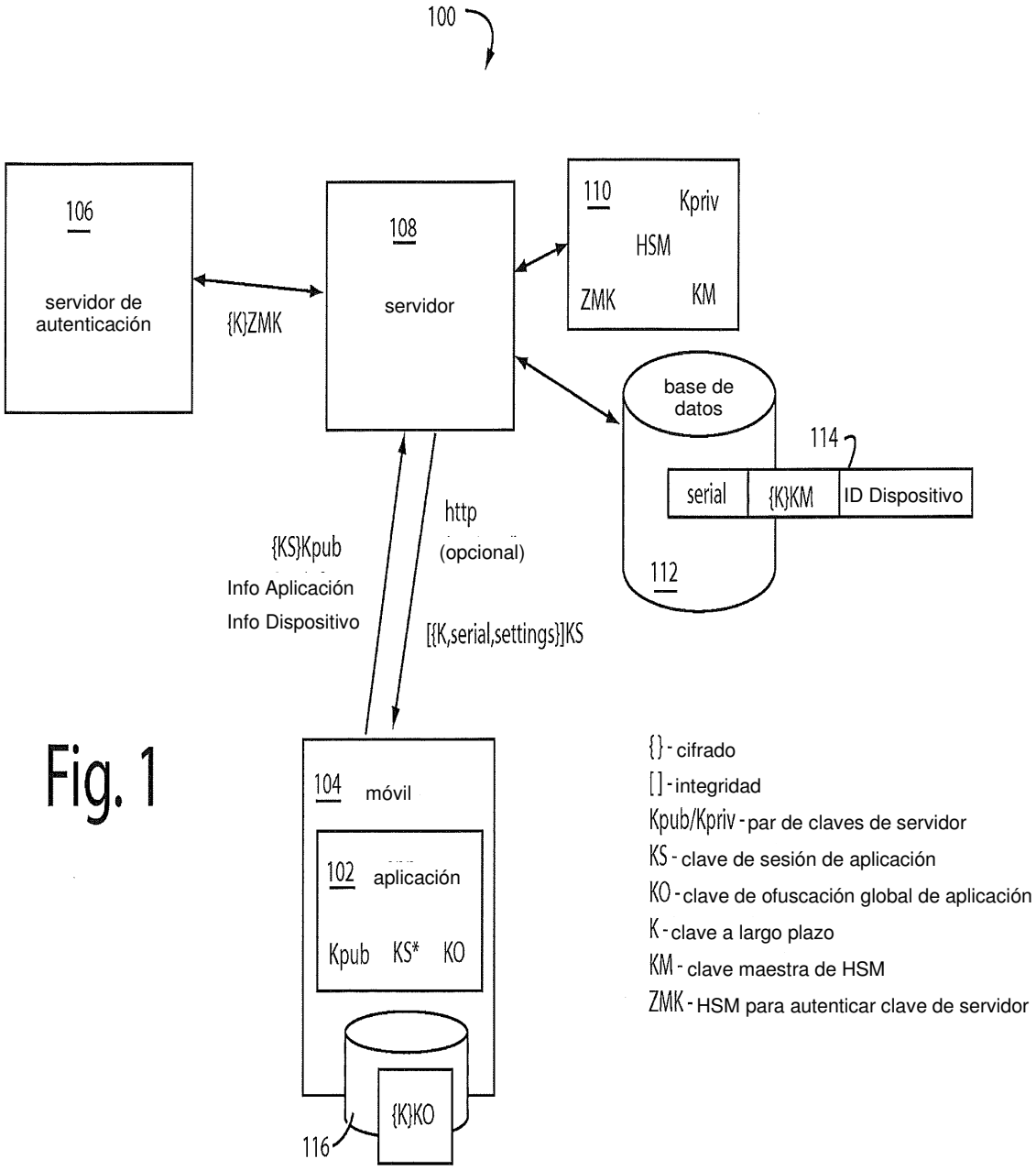

100
106
servidor de autenticación
{K}ZMK
108
servidor
110
Kpriv
HSM
ZMK
KM
base de datos
114
serial
{K}KM
ID Dispositivo
112
http
(opcional)
{KS}Kpub
Info Aplicación
Info Dispositivo
[{K,serial,settings}]KS
104
móvil
102
aplicación
Kpub
KS*
KO
{K}KO
116
{} - cifrado
[] - integridad
Kpub/Kpriv - par de claves de servidor
KS - clave de sesión de aplicación
KO - clave de ofuscación global de aplicación
K - clave a largo plazo
KM - clave maestra de HSM
ZMK - HSM para autenticar clave de servidor

Fig. 1

Fig. 2

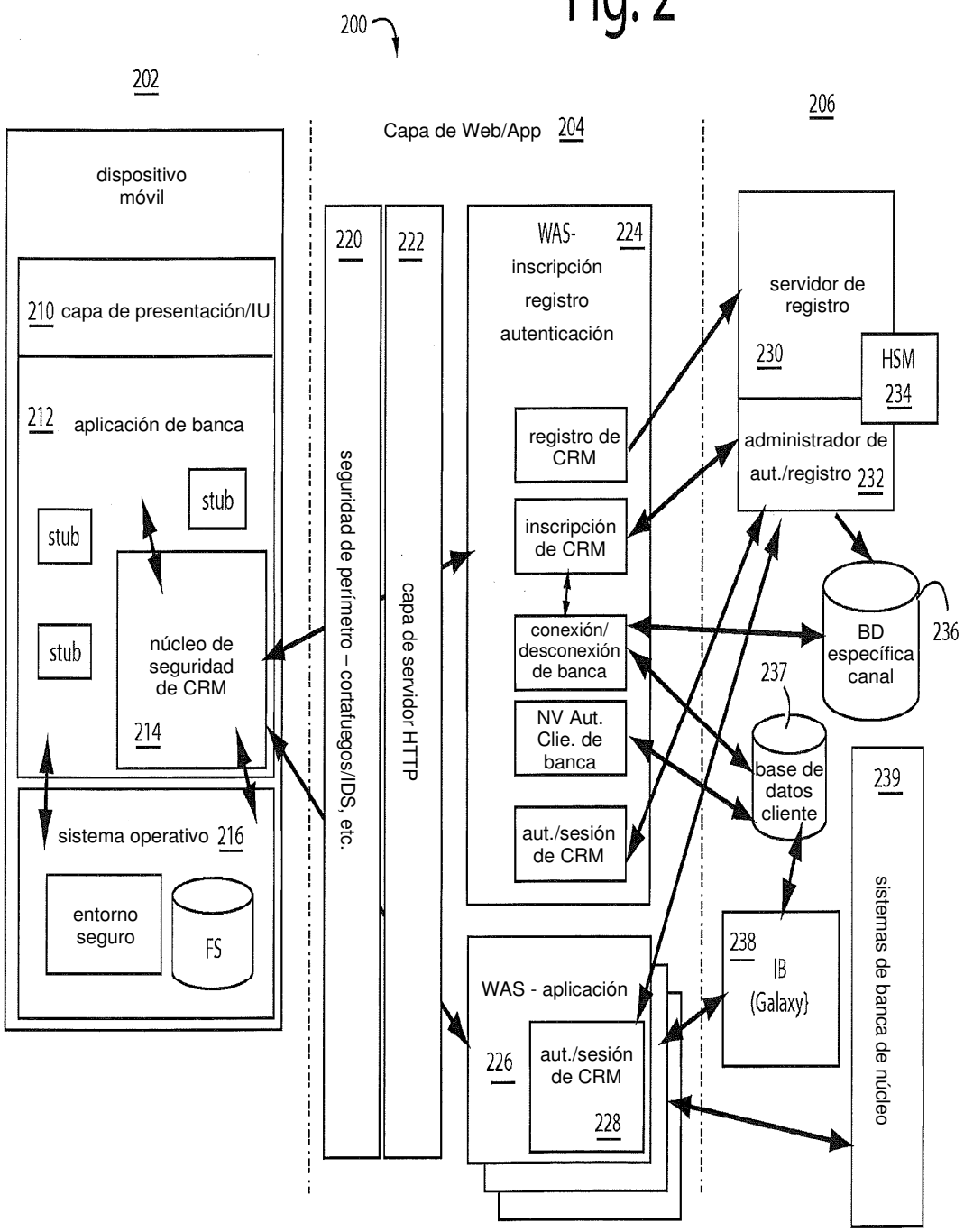
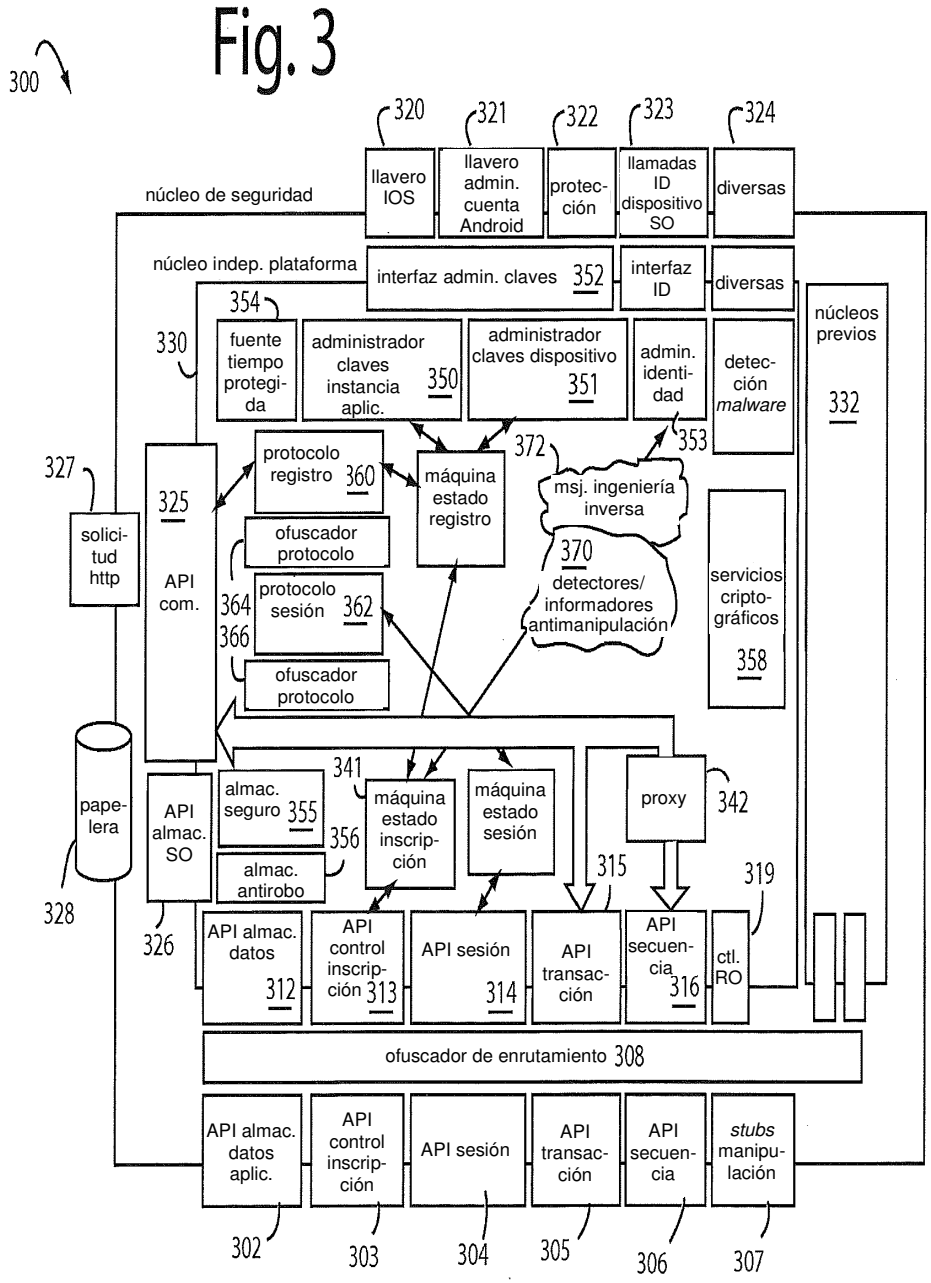

200
202
206
Capa de Web/App 204
dispositivo móvil
210 capa de presentación/IU
212 aplicación de banca
stub
stub
stub
núcleo de seguridad de CRM
214
sistema operativo 216
entorno seguro
FS
220
222
seguridad de perímetro – cortafuegos/IDS, etc.
capa de servidor HTTP
WAS- 224
inscripción
registro
autenticación
registro de CRM
inscripción de CRM
conexión/ desconexión de banca
NV Aut. Clie. de banca
aut./sesión de CRM
WAS - aplicación
226
aut./sesión de CRM
228
servidor de registro
230
HSM
234
administrador de aut./registro 232
BD específica canal
236
237
base de datos cliente
239
sistemas de banca de núcleo
238
IB
(Galaxy)

Fig. 3


300
320
321
322
323
324
llavero IOS
llavero admin. cuenta Android
protec-ción
llamadas ID dispositivo SO
diversas
núcleo de seguridad
núcleo indep. plataforma
interfaz admin. claves 352
interfaz ID
diversas
354
330
fuente tiempo protegi-da
administrador claves instancia aplic. 350
administrador claves dispositivo 351
admin. identi-dad
detec-ción malware
núcleos previos 332
327
325
API com.
protocolo registro 360
máquina estado registro
372
353
msj. ingeniería inversa
solici-tud http
ofuscador protocolo
370
detectores/ informadores antimanipulación
servicios cripto-gráficos 358
364
366
protocolo sesión 362
ofuscador protocolo
pape-lera
341
API almac. SO
almac. seguro 355
máquina estado inscrip-ción
máquina estado sesión
proxy
342
356
almac. antirobo
315
319
328
326
API almac. datos 312
API control inscrip-ción 313
API sesión 314
API transac-ción
API secuen-cia 316
ctl. RO
ofuscador de enrutamiento 308
API almac. datos aplic.
API control inscrip-ción
API sesión
API transac-ción
API secuen-cia
stubs manipu-lación
302
303
304
305
306
307