(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 409 807**

(21) Número de solicitud: 201132110

(51) Int. CI.:

G06F 12/14 (2006.01)

(12) SOLICITUD DE PATENTE A2

(22) Fecha de presentación:

27.12.2011

(43) Fecha de publicación de la solicitud:

27.06.2013

(71) Solicitantes:

TELEFONICA, S.A. (100.0%)
GRAN VIA, N.28
28013 MADRID ES

(72) Inventor/es:

RODRIGUEZ SELA, Fernando;
ROSA, Jesús y
BARANDALLA TORREGROSA, Ignacio

(74) Agente/Representante:

ARIZTI ACHA, Monica

(54) Título: **MÉTODO PARA GESTIONAR COMUNICACIÓN SIN CONTACTO EN UN DISPOSITIVO DE USUARIO**

(57) Resumen:

Método para gestionar comunicación sin contacto en un dispositivo de usuario.

En el método de la invención, dicha comunicación sin contacto se realiza por un applet de un proveedor de servicios, estando instalado dicho applet en un dispositivo de usuario y accediendo dicho applet a una memoria de dicho dispositivo de usuario con el fin de realizar dicha comunicación sin contacto.

Está caracterizado porque comprende:

- dar acceso a una pluralidad de applets de dicho dispositivo de usuario a dicha memoria por medio de una entidad de gestor de servicios, estando instalada dicha pluralidad de applets en dicho dispositivo de usuario; y
- almacenar información de proveedores de servicios de dicha pluralidad de applets de manera aislada usando la seguridad de la tarjeta universal de circuito integrado de dicho dispositivo de usuario, teniendo cada uno de dicha pluralidad de applets un dominio de seguridad diferente en el que se almacena dicha información.

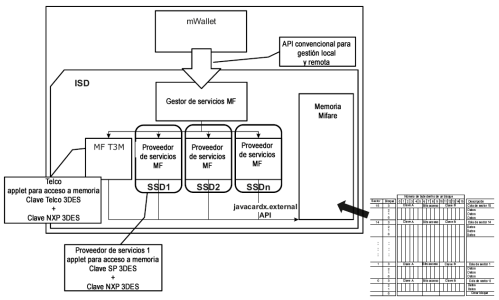

FIG. 2

ES 2 409 807 A2

DESCRIPCIÓN

Método para gestionar comunicación sin contacto en un dispositivo de usuario.

Campo de la técnica

La presente invención se refiere, en general, a un método para gestionar comunicación sin contacto en un dispositivo de usuario, realizándose dicha comunicación sin contacto por un *applet* de un proveedor de servicios, estando instalado dicho *applet* en un dispositivo de usuario y accediendo dicho *applet* a una memoria de dicho dispositivo de usuario con el fin de realizar dicha comunicación sin contacto, y más particularmente a un método que comprende dar acceso a una pluralidad de *applets* de dicho dispositivo de usuario a dicha memoria por medio de una entidad de gestor de servicios, estando instalada dicha pluralidad de *applets* en dicho dispositivo de usuario y almacenar información de proveedores de servicios de dicha pluralidad de *applets* de manera aislada usando la seguridad de la tarjeta universal de circuito integrado de dicho dispositivo de usuario, teniendo cada uno de dicha pluralidad de *applets* un dominio de seguridad diferente en el que se almacena dicha información.

Estado de la técnica anterior

Calypso es una tecnología propietaria sin contacto de tarjeta inteligente ampliamente usada en transportes en algunos países.

MiFare es una tecnología propietaria ampliamente implementada y la más extendida (propiedad de NXP, anteriormente Philips) para comunicaciones sin contacto. Sus capas inferiores se basan en la norma ISO 14443 [2] pero tiene comandos y organización de memoria propietarios, así como cifrado y autenticación de seguridad.

Dado que la mayoría de los sistemas sin contacto que existen en la actualidad se basan en esta tecnología, su emulación dentro del entorno de aparatos de teléfono SIM+móvil se ha convertido en un requisito importante para una adopción generalizada.

El protocolo MiFare es un protocolo propietario que usa (en algunos modos) cifrado en flujo basado en hardware para un procesamiento muy rápido. Debido al cifrado global de la comunicación completa y los fuertes requisitos de sincronismo, los mensajes de MiFare tienen que tunelizarse a través del extremo frontal sin contacto (CLF) usando el modo de transmisión sin contacto (modo de CLT) en la capa de enlace especificada en [3].

Con el fin de poder funcionar en estos sistemas, es necesario que el CLF funcione en el modo de CLT y la tarjeta universal de circuito integrado (UICC) tendrá que emular una memoria MiFare. Hay diferentes casos de MiFare:

- MiFare clásico: es una tarjeta de memoria con cierta seguridad de acceso. Esta memoria debe implementarse en la tarjeta. Entonces un lector externo podrá acceder a la misma a través del CLF y CLT como si fuera cualquier otra tarjeta MiFare.

- La UICC también debe implementar una manera para que las aplicaciones almacenadas en la tarjeta usen y gestionen esta memoria (interfaz de usuario de servidor web de tarjeta inteligente, o cualquier otra aplicación que haga uso de los datos en la memoria MiFare). Una interfaz de programación de aplicaciones (API) conocida debe usarse para este objeto. La UICC presentará una entidad central para gestionar la memoria MiFare especialmente para evitar colisiones en el uso de sectores de MiFare y sus claves de acceso.

- MiFare DESFire: es similar a las estructuras de tarjetas inteligentes existentes. También soporta la mayoría de comandos según la norma ISO 7816 [4]. La tarjeta debe implementar una emulación de este sistema.

MiFare4Mobile (o M4M) es una tecnología que proporciona una solución de extremo a extremo para entregar aplicaciones MiFare al teléfono móvil de comunicación de campo cercano (NFC) incluyendo la gestión de ciclo de vida de aplicación, modificación de contenido y visualización de información en la interfaz de usuario de servicio (UI).

En la actualidad, la única implementación M4M la ha realizado Gemalto.

NXP describió una API para la comunicación con la UI así como con la plataforma sobre el aire (OTA), tal como se mostrará en la figura 1.

Los componentes principales de la estructura presentada son:

- Gestor de servicios de MiFare4Mobile, que gestiona aplicaciones MiFare y datos relacionados. Actúa como un tipo de gestor de tarjetas JavaCard/GP (GlobalPlatform) para aplicaciones MiFare.

- Interfaz de usuario o “monedero”, como la interfaz de usuario gráfica para el gestor de servicios.

- Aplicación proxy, que establece la conexión entre sistema de extremo posterior y el gestor de servicios de

M4M. Retransmitirá los comandos de M4M relevantes desde la plataforma de extremo posterior hasta el gestor de servicios.

Debe observarse que la especificación no cubre la implementación de estos elementos sino sólo la interfaz entre los mismos (las API mencionadas anteriormente: API de MiFare de UI y API de MiFare de gestión remota). Esto deja la implementación abierta y sin definir incluyendo la manera en la que el SP debe funcionar para proporcionar aplicaciones, etc.

Esta especificación de API se divide en dos:

- API de gestor de servicios fiables (TSM), que incluye todos los comandos para instalar aplicaciones. El formato de los comandos es similar a los de la especificación de GP pero en realidad está fuera de GP y no incluye dominios de seguridad (SD). Los comandos son: INSTALL, ACTIVATE, DELETE, GETDATA y STORE DATA (para gestión de aplicaciones), INITIALIZE UPDATE y EXTERNAL AUTHENTICATE (para canales de seguridad SCP02) y PUTKEY y PUTMIFAREKEYS (para gestión de seguridad de MiFare).

- API de monedero, que incluye comandos para obtener información de MiFare a partir del monedero: ACTIVATE, RETRIEVE DATA y VERIFY PIN y CHANGE PIN.

Sin embargo, tanto la especificación de M4M como la implementación de Gemalto del gestor de servicios presenta la amenaza de dar demasiado poder a un TSM, puesto que Gemalto propone una estructura cerrada en la que sólo a través de la plataforma de Gemalto puede accederse a aplicaciones MiFare. Controlarán lo que se instala, y controlarán las claves de seguridad. Además M4M no soporta MiFare DESFire.

Las soluciones existentes no cubren algunos puntos importantes:

Privacidad y seguridad

- Toda la información del proveedor de servicios (SP) pasa a través del gestor de servicios

- Toda la información de configuración del SP está en el mismo dominio de seguridad

No se soporta MultiTSM

- Sólo se soporta 1 TSM

- Todos los SP usarán el mismo TSM

Descripción de la invención

Es necesario ofrecer una alternativa al estado de la técnica que cubra las lagunas encontradas en la misma, particularmente en relación con la falta de propuestas que realmente proporcionen un entorno de dominio de seguridad múltiple en el que sea posible proporcionar independencia a los proveedores de servicios y al operador de red móvil en sí mismos.

Para ello, la presente invención proporciona un método para gestionar comunicación sin contacto en un dispositivo de usuario, realizándose dicha comunicación sin contacto por un *applet* de un proveedor de servicios, estando instalado dicho *applet* en un dispositivo de usuario y accediendo dicho *applet* a una memoria de dicho dispositivo de usuario con el fin de realizar dicha comunicación sin contacto.

A diferencia de las propuestas conocidas, el método de la invención, de manera característica, comprende:

- dar acceso a una pluralidad de *applets* de dicho dispositivo de usuario a dicha memoria por medio de una entidad de gestor de servicios, estando instalada dicha pluralidad de *applets* en dicho dispositivo de usuario; y

- almacenar información de proveedores de servicios de dicha pluralidad de *applets* de manera aislada usando la seguridad de la tarjeta universal de circuito integrado de dicho dispositivo de usuario, teniendo cada uno de dicha pluralidad de *applets* un dominio de seguridad diferente en el que se almacena dicha información.

Otras realizaciones del método del primer aspecto de la invención se describen según las reivindicaciones adjuntas 2 a 14 y en una sección posterior relativa a la descripción detallada de varias realizaciones.

Breve descripción de los dibujos

Las anteriores y otras ventajas y características se entenderán más completamente a partir de la siguiente descripción detallada de realizaciones, con referencia a los dibujos adjuntos (algunos de los cuales ya se han descrito en la sección del estado de la técnica anterior), que deben considerarse de una manera ilustrativa y no limitativa, en los que:

La figura 1 muestra una estructura de NXP de MiFare5Mobile actual.

La figura 2 muestra la estructura de gestión móvil de MiFare Telco propuesta, según una realización de la presente invención.

La figura 3 muestra versiones avanzadas de la estructura de gestión móvil de MiFare Telco propuesta, según una realización de la presente invención.

La figura 4 muestra en detalle los componentes y las interfaces de la estructura de "gestión de Telco MiFare Mobile" (T3M), según una realización de la presente invención.

La figura 5 muestra el diagrama de flujo para un intercambio de proveedores de servicios, según una realización de la presente invención.

La figura 6 muestra el diagrama de flujo para la lectura de etiqueta de proveedor de servicios, según una realización de la presente invención.

Descripción detallada de varias realizaciones

La "gestión de Telco MiFare Mobile" (T3M) se ha diseñado como una solución para gestionar todos los tipos de MiFare y Calypso en un aparato de teléfono móvil.

Esta solución se ha diseñado para que sea más sencilla para el operador de red móvil (MNO) y también para que sea una solución muy segura de modo que se defina un entorno fiable para el proveedor de servicios.

La T3M usa la seguridad de la UICC (SIM) para almacenar la información de aplicaciones sin contacto (como MiFare o Calypso) de un SP de manera aislada.

T3M tiene soporte de TSM múltiple y toda la información de SP se gestiona por el *applet* de SP de modo que ninguna clave de seguridad fluye a través del gestor de servicios.

Se muestra un esquema genérico en la figura 2.

Se ha creado una estructura de T3M específica con el fin de implementar servicios MiFare cuando no hay disponible ninguna norma actual. Esta estructura podría usarse como la que se seleccione para normalización y adopción generalizada.

Los objetivos principales cuando se define la estructura de T3M son:

- Soporte de aplicaciones múltiples: NXP definió una solución de aplicaciones múltiples para las tarjetas MiFare usando el directorio de directorio de aplicaciones de MiFare (MAD). Estas soluciones se limitan a una memoria MiFare y todos los proveedores de servicios deben compartir la misma área de memoria. La solución propuesta permite que cada SP defina su propia estructura de memoria así como que también se soporten aplicaciones de MAD.

- Privacidad de proveedor de servicios: en las implementaciones actuales todos los SP deben almacenar sus códecs en el mismo dominio de seguridad y sólo un *applet* accede a los datos privados de SP en la memoria MiFare. La solución propuesta aísla cada SP en su propio SD y también el gestor de servicios actúa sólo como director dando acceso al recurso compartido (memoria MiFare) a cada *applet* de proveedor de servicios.

- Gestión OTA: Para las solución de T3M, la gestión OTA era obligatoria y cumple con las especificaciones de GP y M4M.

Una versión más avanzada puede añadir soporte para DESFire, Calypso y UID de 7 y 4 bytes clásico, tal como se muestra en la figura 3.

Los principales componentes en la solución de T3M son:

1. Gestor de servicios de T3M

- Organiza la gestión de aplicaciones MiFare y proporciona seguridad.

- Sirve de interfaz para el monedero a través de una API propietaria y a través de una API de monedero de MiFare4Mobile (es decir: la activación/desactivación de la aplicación pone los datos de SP de MiFare en la emulación de MiFare, lee/escribe datos de una aplicación dada, consigue una lista de aplicaciones MiFare disponibles).

- Puede servir de interfaz para la plataforma de gestión OTA para comandos de actualización de datos específicos de MiFare según la API de TSM de M4M. Sin embargo la mayor parte de la gestión de los *applets* puede realizarse directamente a través de la plataforma GP.

- NO mantiene o transporta en ningún momento ningún dato o clave de SP.

2. Proveedor de servicios de T3M

- Mantiene las claves MiFare de SP. Pueden producirse de manera dinámica dentro del *applet* o personalizarse usando procedimientos de GP remota o comandos de APDU sin contacto.

- Se instancia en un SD de GP específico de SP para gestión y personalización seguras.

- Su carga e instalación pueden realizarse a través del canal de GP usual.

- Mantiene un mapa de memoria de la aplicación MiFare que se cargará dentro de la emulación de memoria MiFare cuando se active la aplicación y el gestor de servicios ordene al *applet* de SP que lo haga. La carga se realiza directamente por el *applet* de SP, por tanto ninguna información pasa por el gestor de servicios. Usa javacardx.external para ello.

- Define un conjunto de etiquetas con datos a los que puede acceder la UI de servicio.

- Habrá un *applet* de SP de MiFare de T3M genérico con el fin de proporcionar seguridad cuando no haya ninguna aplicación seleccionada.

Esta implementación de un entorno de aplicaciones MiFare cumple con los requisitos establecidos anteriormente proporcionando independencia de seguridad de SP (según GP), acceso remoto de aplicaciones múltiples y una API de interfaz de usuario.

Ventajas de la invención

- Más privacidad: la información de cada proveedor de servicios reside en un dominio de seguridad aislado dentro de la UICC.

- Más seguro: Cada proveedor de servicios gestiona su información sin necesidad de un *cardlet* primario para gestionar la memoria.

- La memoria MiFare está protegida cuando ninguna otra aplicación MiFare está activa.

- Soporte de múltiples TSM.

Un experto en la técnica puede introducir cambios y modificaciones en las realizaciones descritas sin apartarse del alcance de la invención tal como se define en las reivindicaciones adjuntas.

SIGLAS

API	*Application Programming Language*; Lenguaje de programación de aplicaciones
CLF	*Contactless Fronted*; Extremo frontal sin contacto
CLT	*Contactless Transmission*; Transmisión sin contacto
GP	*Global Platform*; Plataforma global
M4M	*MiFare for Mobile*; MiFare para móvil
MNO	*Mobile Network Operator*; Operador de red móvil
NFC	*Near Field Communication*; Comunicación de campo cercano
OTA	*Over The Air*; Sobre el aire
SP	*Service Provider*; Proveedor de servicios
T3M	*Telco Mifare Mobile Management*; Gestión de Telco MiFare Mobile
TSM	Trusted Service Management; Gestión de servicios fiables
UI	*User Interface*; Interfaz de usuario
UICC	*Universal Integrated Circuit Card*; Tarjeta universal de circuito integrado
UID	User Identifier; Identificador de usuario

BIBLIOGRAFÍA

[1] NXP MIFARE4Mobile Interface Specification V1.01

[2] ISO/IEC 14443-1, ISO/IEC 14443-2, ISO/IEC 14443-3, ISO/IEC 14443-4

[3] ETSI TS 102 613 Smart Cards; UICC – Contactless Front-end (CLF) Interface; Part 1: Physical and data link layer characteristics v 7.8.0 (2010-04)

[4] ISO/IEC 7816

REIVINDICACIONES

1. Método para gestionar comunicación sin contacto en un dispositivo de usuario, realizándose dicha comunicación sin contacto por un *applet* de un proveedor de servicios, estando instalado dicho *applet* en un dispositivo de usuario y accediendo dicho *applet* a una memoria de dicho dispositivo de usuario con el fin de realizar dicha comunicación sin contacto, caracterizado porque comprende:

- dar acceso a una pluralidad de *applets* de dicho dispositivo de usuario a dicha memoria por medio de una entidad de gestor de servicios, estando instalada dicha pluralidad de *applets* en dicho dispositivo de usuario; y

- almacenar información de proveedores de servicios de dicha pluralidad de *applets* de manera aislada usando la seguridad de la tarjeta universal de circuito integrado de dicho dispositivo de usuario, teniendo cada uno de dicha pluralidad de *applets* un dominio de seguridad diferente en el que se almacena dicha información.

2. Método según la reivindicación 1, en el que dicha comunicación sin contacto se lleva a cabo por medio de tecnología MiFare y dicha memoria es una memoria MiFare.

3. Método según la reivindicación 2, que comprende permitir que cada proveedor de servicios de dicho uno o más *applets* defina su propia estructura de memoria o use estructuras de directorio de aplicaciones MiFare.

4. Método según la reivindicación 2 ó 3, que comprende proporcionar, dicha entidad de gestor de servicios, una interfaz con un monedero de dicho dispositivo de usuario a través de una interfaz de programación de aplicaciones, o API, propietaria y a través de una API de monedero de MiFare4Mobile, siendo dicho monedero una interfaz de usuario gráfica de dicho dispositivo de usuario.

5. Método según la reivindicación 4, que comprende proporcionar, dicha entidad de gestor de servicios, una interfaz con una plataforma en el aire, u OTA, para comandos de actualización de datos específicos de MiFare usando una API de gestión de servicios fiables de MiFare4Mobile.

6. Método según cualquiera de las reivindicaciones anteriores 2 a 5, que comprende instanciar un repositorio de proveedores de servicios en un dominio de seguridad de plataforma global, o GP, específica del proveedor de servicios, para gestión y personalización seguras.

7. Método según la reivindicación 6, que comprende cargar e instalar dicho repositorio de proveedores de servicios a través de un canal de GP.

8. Método según la reivindicación 7, que comprende, dicho repositorio de proveedores de servicios:

- mantener claves MiFare de proveedores de servicios; y/o

- mantener un mapa de memoria de una aplicación MiFare que se cargará en dicha memoria MiFare cuando se active dicha aplicación MiFare y dicha entidad de gestor de servicios se lo ordene al *applet* asociado a dicha aplicación MiFare, en el que dicha carga se realiza directamente por dicho *applet* sin que pase ninguna información por dicha entidad de gestor de servicios.

9. Método según la reivindicación 8, que comprende realizar dicha carga por medio de un javacardx.external.

10. Método según la reivindicación 8 ó 9, que comprende generar dichas claves MiFare de manera dinámica dentro de un *applet*, usando procedimientos de GP remota o usando comandos de unidad de datos de protocolo de aplicación sin contacto.

11. Método según cualquiera de las reivindicaciones anteriores 6 a 10, que comprende definir un repositorio de proveedores de servicios genérico con el fin de proporcionar seguridad cuando no se selecciona ninguna aplicación MiFare.

12. Método según cualquiera de las reivindicaciones anteriores 6 a 11, que comprende realizar un intercambio de proveedores de servicios realizando las siguientes etapas:

- enviar, dicha entidad de gestor de servicios, un mensaje de descargar al proveedor de servicios de repositorio del proveedor de servicios activo actual;

- enviar, dicho proveedor de servicios de repositorio del proveedor de servicios activo actual, un mensaje de leer datos de sector a dicha memoria MiFare;

- almacenar, dicho proveedor de servicios de repositorio del proveedor de servicios activo actual, datos de sector recibidos desde dicha memoria MiFare en una base de datos;

- enviar, dicho proveedor de servicios de repositorio del proveedor de servicios activo actual, un mensaje de

eliminar datos de sector a dicha memoria MiFare y un mensaje de establecer claves de transporte;

- enviar, dicha entidad de gestor de servicios, un mensaje de cargar a un proveedor de servicios de repositorio de un proveedor de servicios diferente;

- leer, dicho proveedor de servicios de repositorio de un proveedor de servicios diferente, un sector de datos desde una base de datos;

- enviar, dicho proveedor de servicios de repositorio de un proveedor de servicios diferente, un mensaje de escribir datos de sector a dicha memoria MiFare; y

- realizar una instrucción de establecer proveedor de servicios activo en dicha entidad de gestor de servicios.

13. Método según cualquiera de las reivindicaciones anteriores 6 a 11, que comprende realizar una lectura de etiqueta de proveedor de servicios realizando las siguientes etapas:

- enviar, dicha entidad de gestor de servicios, un mensaje de conseguir etiqueta a un proveedor de servicios de repositorio;

- realizar una instrucción de conseguir sector y bloque en dicho proveedor de servicios de repositorio;

- enviar, dicho proveedor de servicios de repositorio, datos leídos a dicha memoria MiFare; y

- enviar, dicho proveedor de servicios de repositorio, datos recibidos desde dicha memoria MiFare a dicha entidad de gestor de servicios.

14. Método según cualquiera de las reivindicaciones anteriores 6 a 14, que comprende añadir soporte a dicho repositorio de proveedores de servicios para DESFire, Calypso e identificador de usuario de 7 y 4 bytes clásico.

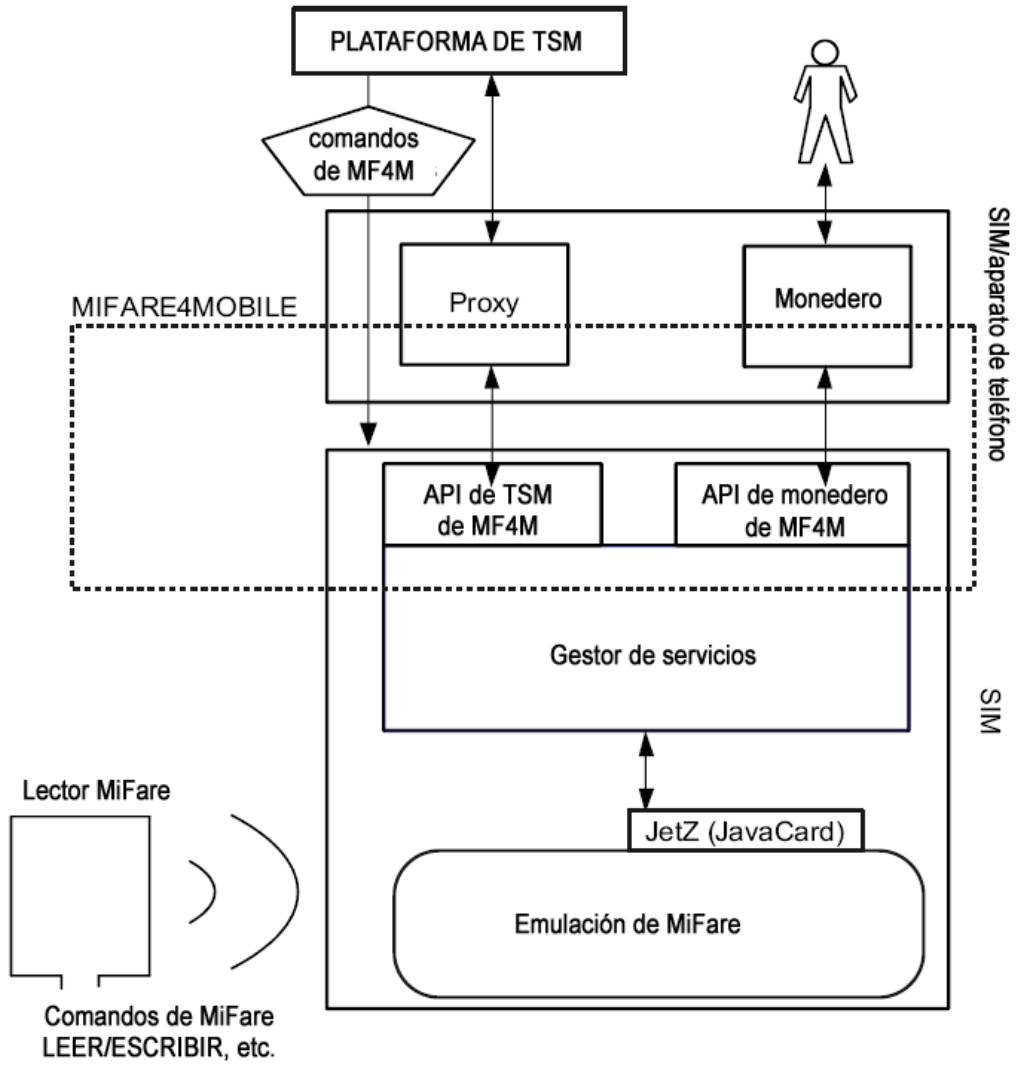
PLATAFORMA DE TSM
comandos de MF4M
MIFARE4MOBILE
Proxy
Monedero
SIM/aparato de teléfono
API de TSM de MF4M
API de monedero de MF4M
Gestor de servicios
SIM
Lector MiFare
JetZ (JavaCard)
Emulación de MiFare
Comandos de MiFare LEER/ESCRIBIR, etc.

FIG. 1

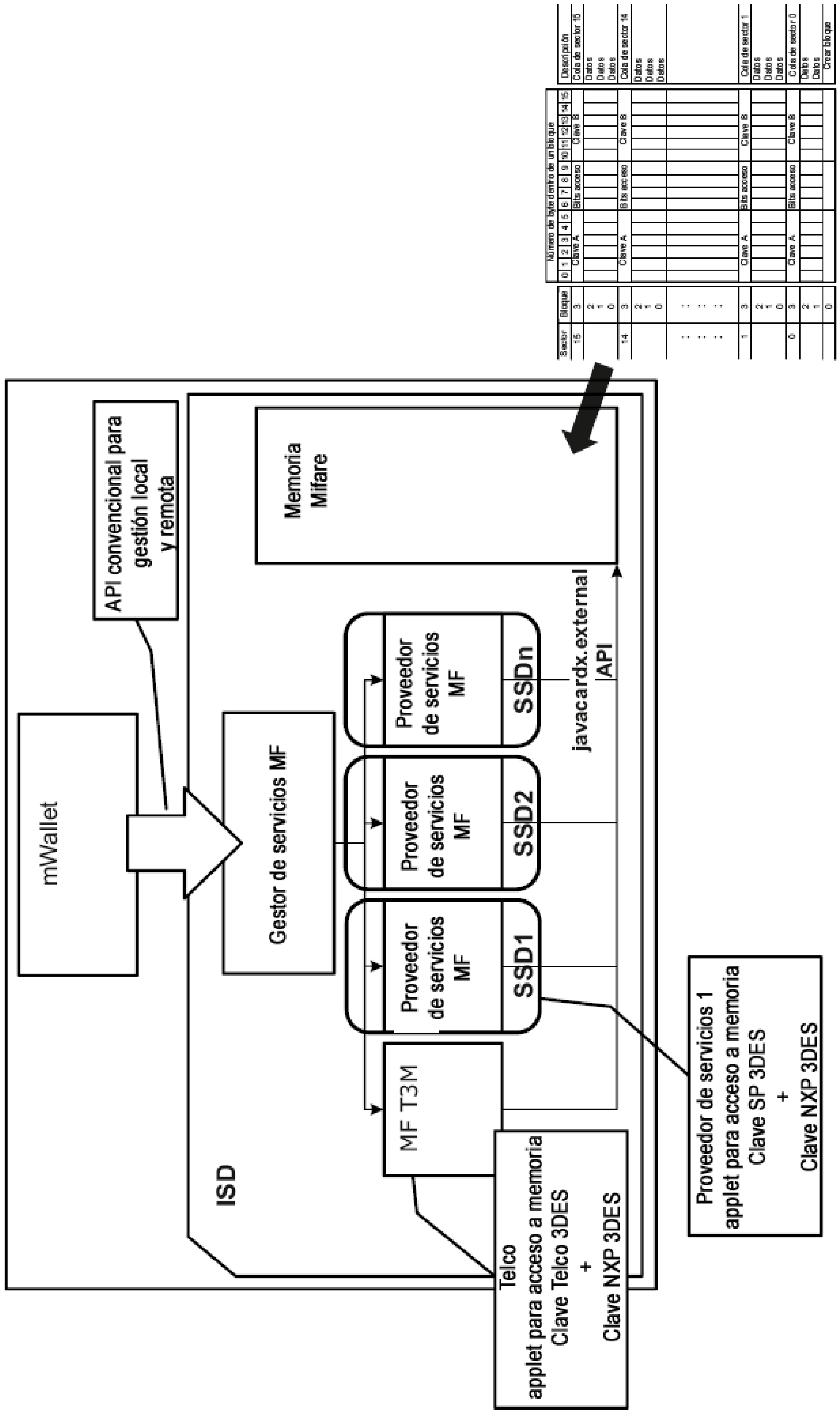
mWallet
API convencional para gestión local y remota
Gestor de servicios MF
Memoria Mifare
ISD
MF T3M
Proveedor de servicios MF
SSD1
Proveedor de servicios MF
SSD2
Proveedor de servicios MF
SSDn
javacardx.external API
Telco applet para acceso a memoria Clave Telco 3DES + Clave NXP 3DES
Proveedor de servicios 1 applet para acceso a memoria Clave SP 3DES + Clave NXP 3DES

FIG. 2

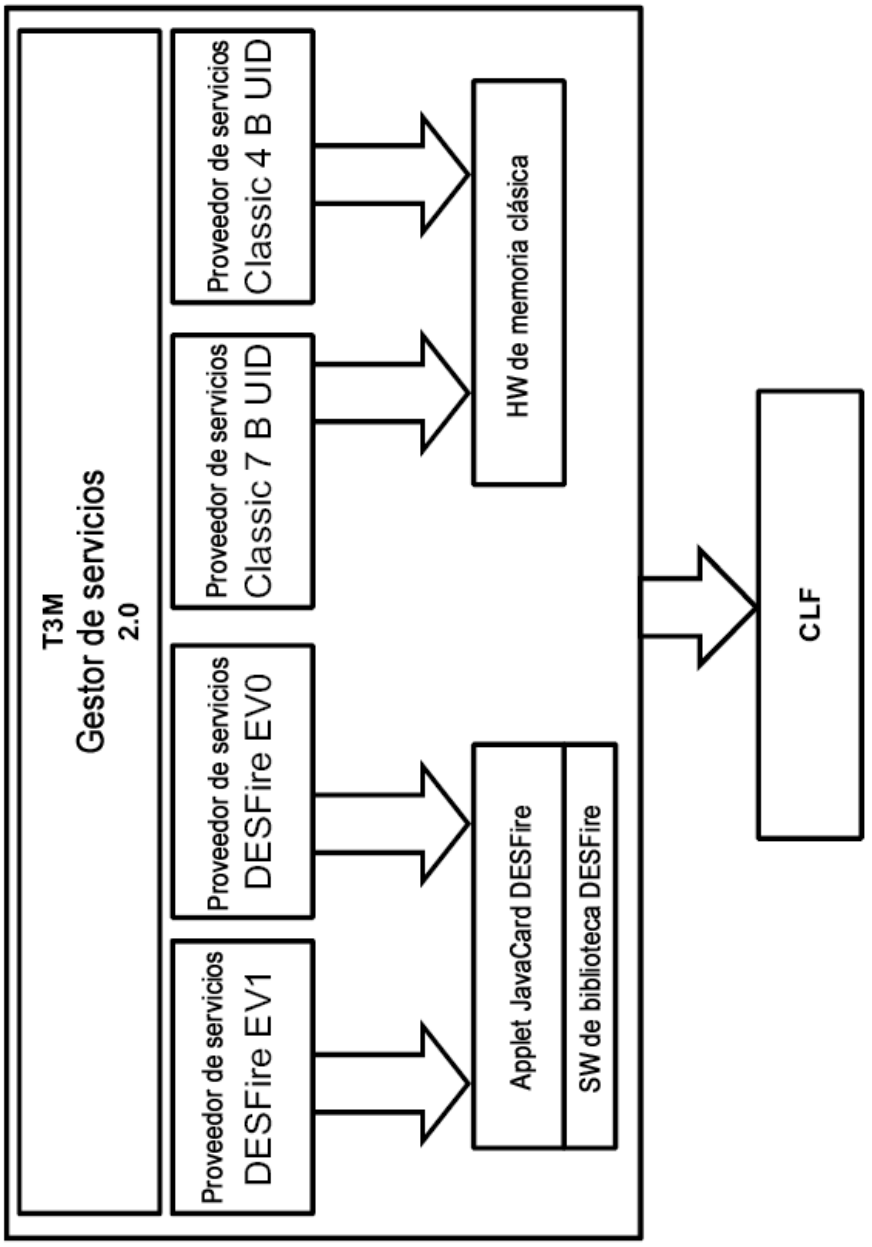
T3M
Gestor de servicios
2.0
Proveedor de servicios
DESFire EV1
Proveedor de servicios
DESFire EV0
Proveedor de servicios
Classic 7 B UID
Proveedor de servicios
Classic 4 B UID
Applet JavaCard DESFire
SW de biblioteca DESFire
HW de memoria clásica
CLF

FIG. 3

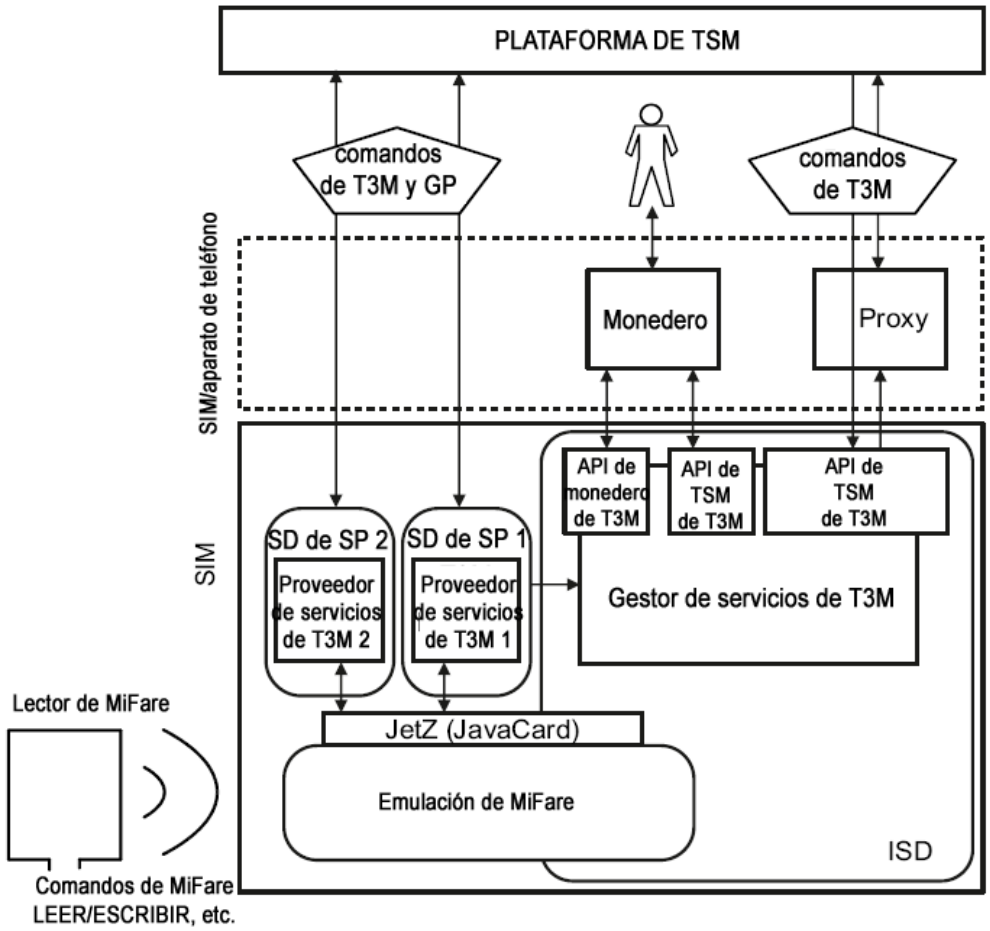
PLATAFORMA DE TSM
comandos de T3M y GP
comandos de T3M
SIM/aparato de teléfono
Monedero
Proxy
SIM
API de monedero de T3M
API de TSM de T3M
API de TSM de T3M
SD de SP 2
SD de SP 1
Proveedor de servicios de T3M 2
Proveedor de servicios de T3M 1
Gestor de servicios de T3M
JetZ (JavaCard)
Emulación de MiFare
ISD
Lector de MiFare
Comandos de MiFare
LEER/ESCRIBIR, etc.

FIG. 4

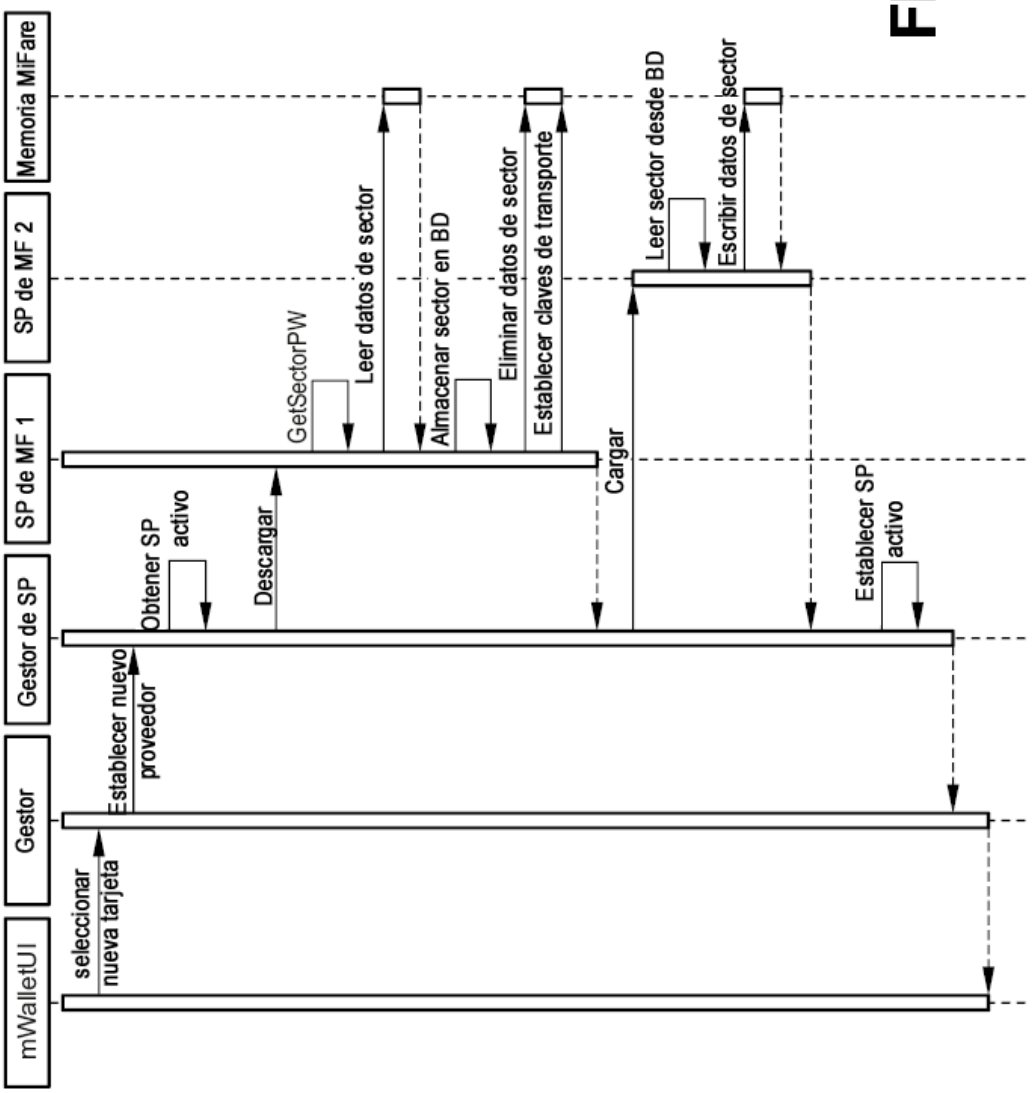
mWalletUI
Gestor
Gestor de SP
SP de MF 1
SP de MF 2
Memoria MiFare
seleccionar nueva tarjeta
Establecer nuevo proveedor
Obtener SP activo
Descargar
GetSectorPW
Leer datos de sector
Almacenar sector en BD
Eliminar datos de sector
Establecer claves de transporte
Cargar
Leer sector desde BD
Escribir datos de sector
Establecer SP activo

FIG. 5

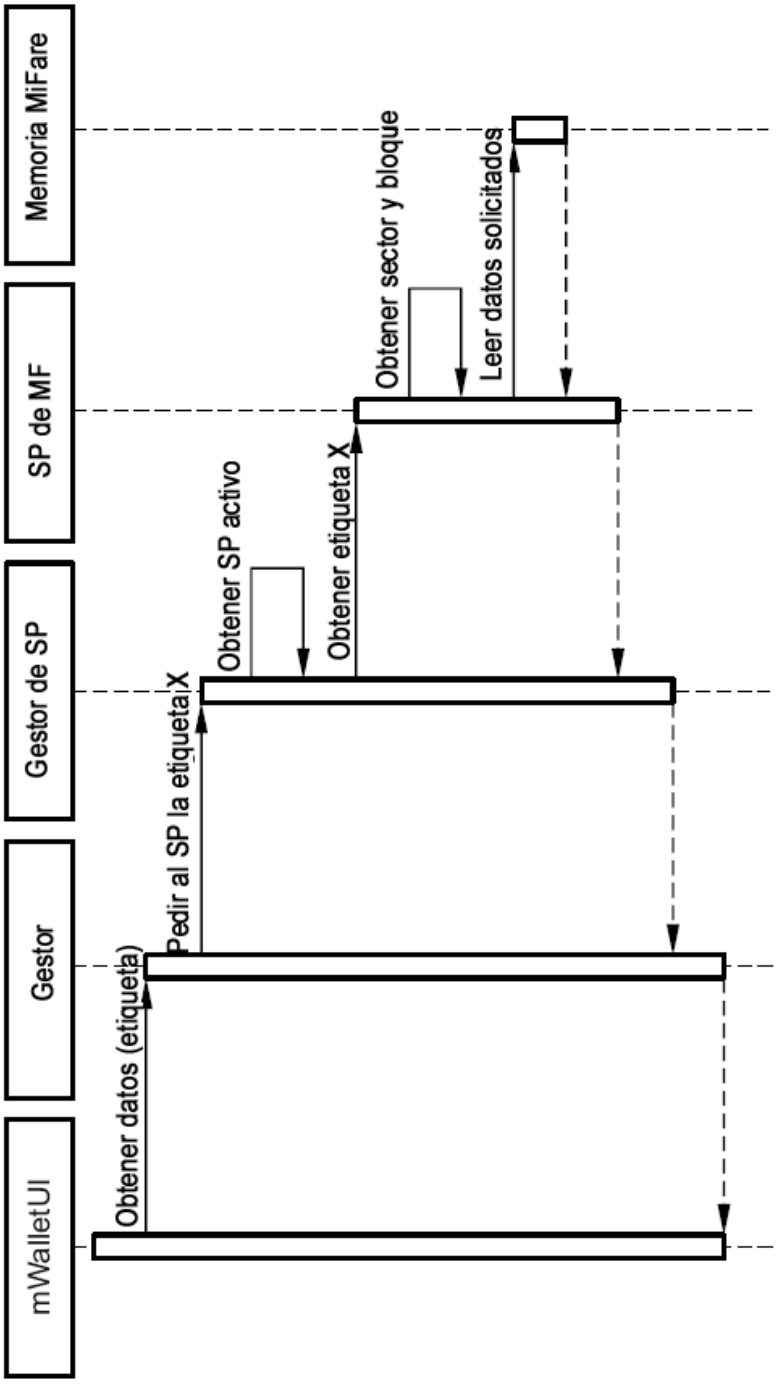
Memoria MiFare
SP de MF
Gestor de SP
Gestor
mWalletUI
Obtener sector y bloque
Leer datos solicitados
Obtener etiqueta X
Obtener SP activo
Pedir al SP la etiqueta X
Obtener datos (etiqueta)

FIG. 6