2. ESTADO DEL ARTE

Con los dispositivos móviles pertenecientes a la primera generación, era impensable poder realizar firmas electrónicas avanzadas, debido a la limitación criptográfica existente en estos dispositivos, así como a las limitaciones correspondientes a operaciones computacionales. Sin embargo, los dispositivos actuales están capacitados para poder generar firmas electrónicas basadas en algoritmos criptográficos.

El principal objetivo a la hora de realizar una firma digital es que sea equivalente a una firma manuscrita. Para poder satisfacer esto, son necesarios una serie de requisitos:

- La solución adoptada para generar la firma electrónica debe estar basada en algoritmos criptográficos asimétricos.
- Es necesario que la firma sea generada a partir de un certificado digital expedido por una autoridad de confianza.
- El formato de la firma electrónica debe ser un formato estándar reconocido: PKCS1 o PKCS7/CMS.
- Con el objetivo de que la solución adoptada pueda ser realizada en diferentes sistemas operativos, su generación debe ser independiente del sistema operativo del dispositivo.

Existen diferentes formas de poder clasificar las soluciones existentes de firma digital para dispositivos móviles. Se pueden clasificar de acuerdo a una serie de criterios tales como:

- Tecnología hardware
- Tecnología software
- Tecnología híbrida
- Soluciones independientes del dispositivo

En los siguientes apartados las diferentes soluciones existentes atendiendo a la clasificación realizada anteriormente.

Otro criterio para clasificar las soluciones de firma electrónica implementadas suele ser atendiendo a la tecnología empleada. En este caso, existen soluciones basadas en criptografía asimétrica, soluciones independientes del sistema operativo, soluciones basadas en tarjetas inteligentes (o más comúnmente conocidas *smart cards*) y soluciones basadas en tecnología Java.

Resulta también interesante clasificar las posibles soluciones existentes de acuerdo al estándar de firma soportado. Desde este punto de vista, podemos encontrar soluciones basadas en firma electrónicas avanzadas, SSCD, PKCS1, PKCS7/CMS..

2.1 Tecnologías hardware

Desde el punto de vista de las plataformas de firma usadas para implementar el proceso de firma electrónica en dispositivos móviles se pueden distinguir cuatro grupos o categorías. El primero de ellos involucra soluciones basadas en el uso de la tarjeta SIM, teniendo como objetivo realizar el proceso de firma electrónica en el interior de la tarjeta SIM del dispositivo móvil, a través de su propio proceso criptográfico. El segundo grupo incluye todas las soluciones basadas en el propio dispositivo móvil. El tercer grupo de clasificación se basa en una mezcla híbrida entre el primero y el segundo, es decir, consisten en implementar el proceso de firma electrónica mediante la tarjeta SIM y el dispositivo móvil en sí. Y el último grupo de clasificación está orientado a una serie de servicios de más alto nivel independientes de la tecnología de firma específica del dispositivo.

2.1.2 Tecnologías basadas en la tarjeta SIM

Las tarjetas SIM actuales son tarjetas multi-aplicaciones, en la cual múltiples servicios están trabajando al mismo tiempo, tal y como se muestra en la figura 2.1. Cada una de estas aplicaciones tiene su propia funcionalidad y algunas de ellas ofrecen soporte a la firma electrónica. Una tarjeta SIM normalmente posee dos clases de memoria: ROM y EEPROM.

En la memoria ROM se encuentra la capa física que contiene el sistema operativo de la tarjeta SIM, la parte correspondiente a gestión de memoria y la interfaz de entrada y salida. En la parte superior se encuentra la máquina virtual de Java: JavaCard Virtual Machine que interpreta la aplicación que se ejecute, el gestor de tarjetas que gestiona el ciclo de vida de cada aplicación, así como la SIM Toolkit Security, que añade cabecera de seguridad a los mensajes, entre otras cosas. En esta zona de memoria también reside la aplicación GSM que es colocada por el fabricante y no se puede eliminar de la tarjeta SIM.

En la memoria EEPROM se encuentran las diferentes aplicaciones. Esta zona de memoria puede ser modificada en el ciclo de vida de la tarjeta por el gestor de tarjeta. La aplicación GSM controla la comunicación entre redes GSM y almacena los archivos o ficheros GSM en el interior de la memoria EEPROM. Estos ficheros contienen información relativa a las claves GSM, libreta de direcciones, mensajes, etc. Las applets USAT son aplicaciones desarrolladas con la tecnología SIM Application Toolkit. La aplicación WIM permite desarrollar operaciones criptográficas en el interior de la tarjeta SIM.

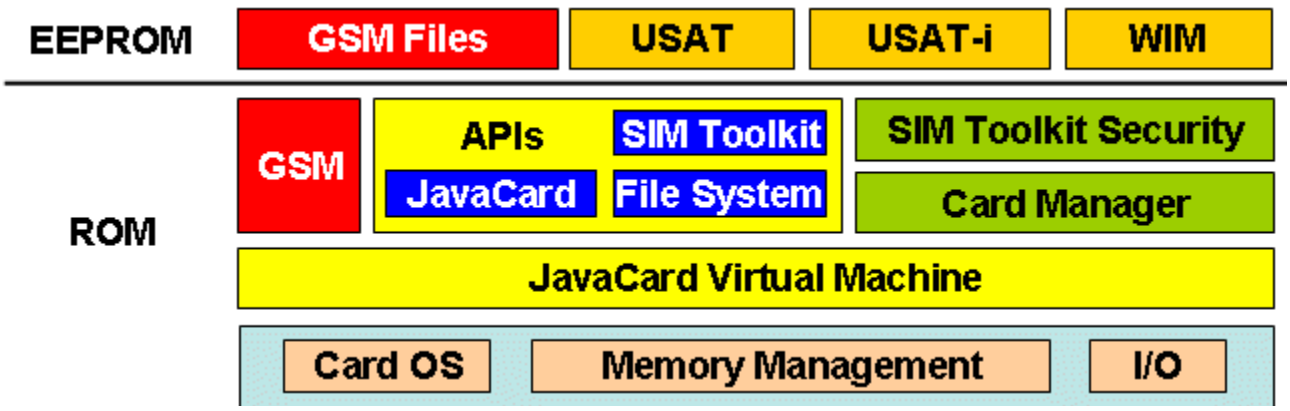

Figura 2.1. Arquitectura de la tarjeta SIM multi-aplicaciones

A continuación vamos a describir y analizar diferentes tecnologías que facilitan la implementación de soluciones de firma electrónica en aplicaciones de las tarjetas SIM.

2.1.2.1 SMS

El Servicio de Mensajes Cortos (Short Message Service) es un servicio disponible en la mayoría de los dispositivos móviles que permite el envío de mensajes de texto entre diferentes aplicaciones y usuarios. Este servicio define la posibilidad de autenticar y cifrar la comunicación mediante el uso de paquetes de seguridad. Estos paquetes son cabeceras especiales, de forma que la parte emisora prepara los datos y los reenvía a la entidad emisora de la tarjeta SIM, que añade la cabecera a los datos, los empaqueta y lo envía como un mensaje al dispositivo móvil receptor. La entidad receptora recibe el mensaje y lo desempaqueta atendiendo a los parámetros de existentes en la cabecera de seguridad.

Desde el punto de vista de la firma electrónica, la cabecera de seguridad contiene tres elementos importantes: el parámetro indicador de seguridad SPI (Security Parameter Indication), el identificador de clave KID (Key Identifier), y la firma digital DS (Digital Signature). El SPI codifica el tipo de encriptación en dos bytes, el KID firma la clave y algoritmos usados en un byte y el DS contiene la firma electrónica de los datos en un número variable de bytes. Los primeros cuatro bits del byte KID codifica el algoritmo de firma digital entre las opciones DES y triple-DES. Así, la firma digital en los dispositivos móviles está basada en el uso de códigos de mensajes MAC (Message Authentication Codes). Los cuatro últimos bits del byte KID indica la clave usada en el proceso de firma digital. El número de claves depende de la implementación.

Este mecanismo fue la primera aproximación en ofrecer firmas electrónicas mediante la tarjeta SIM, pero tiene importantes desventajas: sólo soporta firma basada en criptografía simétrica, depende del fabricante de la tarjeta a la hora de crear las claves simétricas, por lo que no genera una firma electrónica legal puesto que el control de claves no depende del usuario final. Además, SMS tiene un ancho de banda muy pequeño que sólo permite enviar mensajes con textos cortos. Esta última desventaja es contrarrestada con la aparición de MMS, que permite transportar firma electrónica además del certificado. Sin embargo, MMS no está soportado en la tecnología SIM, sino en el dispositivo en sí.

2.1.2.2 SAT – USAT

La tecnología SIM Toolkit Application SAT define un set completo de comandos y eventos entre un teléfono GSM y una tarjeta SIM 2G. Esta comunicación facilita el desarrollo de nuevos servicios basados en tarjetas SIM pero independientes del dispositivo móvil o del fabricante de la tarjeta. Como se muestra en la figura 2.2, estos servicios intercambian datos con la red GSM a través del servicio SMS. La figura muestra un escenario común de un evento de entrada a la tarjeta SIM y la respuesta que se produce al dispositivo. Actualmente, la tecnología SAT está extendida a una gran cantidad de dispositivos y ha evolucionado hasta lo que se conoce como Universal SIM ApplicationToolkit (USAT), en los teléfonos de tercera generación. La principal mejora que implementa USAT es la posibilidad de abrir conexiones HTTP con dispositivos IP.

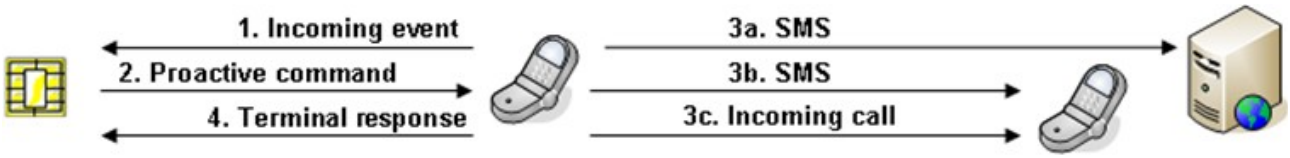

Figura 2.2 Modelo de comunicación SAT

Una aplicación SIM Toolkit es normalmente un applet JavaCard. La tecnología JavaCard ha significado un importante avance en el campo del desarrollo de aplicaciones basadas en tarjetas SIM. Entre sus ventajas más importantes se encuentran: el entorno de ejecución seguro en que se desarrollan, la posibilidad de desarrollar diferentes applets en la misma tarjeta, y el uso del lenguaje Java de programación para desarrollar las aplicaciones. Este lenguaje tiene dos APIs específicas: JavaCard API y SIM Toolkit API. El API JavaCard posee diferentes métodos que permiten mejorar la generación de claves simétricas y asimétricas. Estas applets pueden ser incluidas en el proceso de fabricación del dispositivo o dinámicamente descargadas e instaladas a través de un servidor OTA (Over The Air).

Esta solución presenta algunas desventajas. La primera de ellas está relacionada con la descarga de applets a través de servidores OTA, ya que es un proceso complejo e implica la fragmentación del applet en cortos mensajes. Además, el proceso de instalación del applet supone algunos riesgos de seguridad. Otro de los inconvenientes de esta tecnología es la lentitud en la generación de claves y, por tanto, en el proceso de firma.

2.1.2.3 WIM

Wireless Identity Module (WIM) es una especificación de seguridad que define cómo almacenar y generar credenciales criptográficos: claves simétricas y asimétricas, certificados de usuarios y terceras partes y objetos de identificación tales como números de identificación personal. También define cómo crear procesos de firma electrónica en una tarjeta SIM. El estándar WIM está basado en PKCS15, que permite un formato de información flexible en una ficha criptográfica. Está definido como una aplicación de tarjeta SIM independiente, como GSM o applets de SAT. De esta forma, puede ser usada para generar operaciones criptográficas con protocolos diferentes desde un dispositivo móvil, como TLS o S/MIME.

Por todas lo anterior, la tecnología WIM puede ser considerada como básica en el desarrollo de aplicaciones de firma electrónicas, desde el punto de vista de generación y gestión de claves. Actualmente, los fabricantes están incluyendo esta aplicación en tarjetas SIM como un estándar de la firma electrónica.

2.2 Tecnologías software

La capacidad criptográfica de los dispositivos móviles es cada vez mayor. Existen diferentes tecnologías móviles, como sistema operativo Symbian, sistema operativo Windows Mobile y Java

ME, que nos permiten desarrollar firmas electrónicas en los dispositivos. A continuación analizaremos cada una de ellas.

2.2.1 Sistema Operativo Windows Mobile

El sistema operativo Windows Mobile es el sistema operativo desarrollado por Microsoft para dispositivos móviles. Este sistema operativo constituye la base para el desarrollo de los dos principales tipos de plataformas: Pocket PC y Smartphone. Un Pocket PC, también llamado PDA, es un ordenador de pequeño tamaño y un SmartPhone está más orientado a teléfonos móviles en sí, aunque también posee funcionalidad de datos.

El sistema criptográfico de Microsoft, consta básicamente de varios componentes: aplicaciones, sistemas operativos, y algunos proveedores de servicios criptográficos (CSP). Las aplicaciones se comunican con el sistema operativo a través de la API criptográfica, denominada CryptoAPI, y el sistema operativo se comunica con los proveedores de servicio CSPs a través de la interfaz de servicios criptográficos (CryptoSPI), tal y como muestra la siguiente figura:

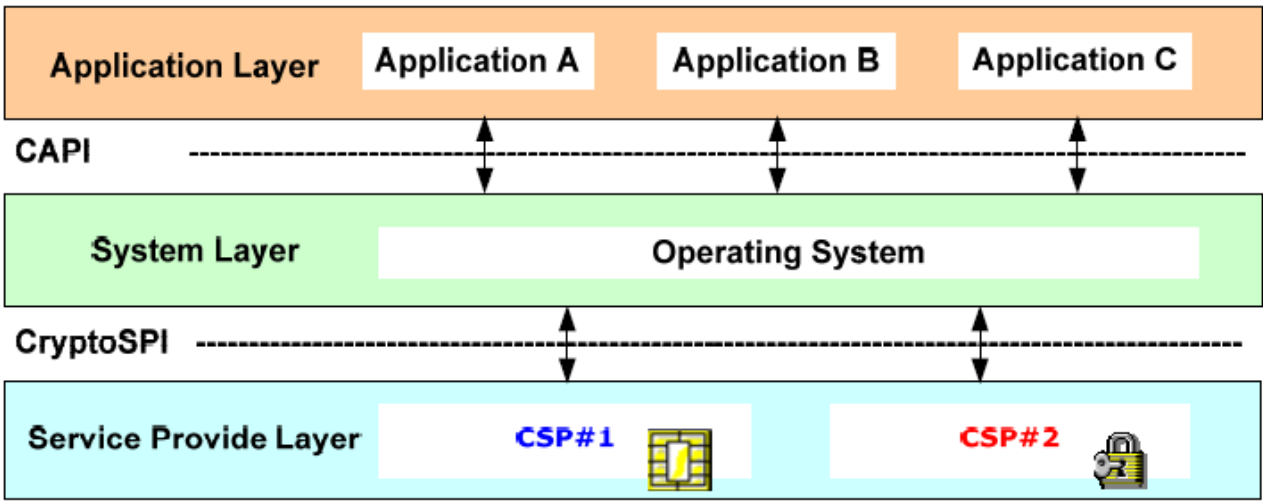

Figura 2.3. Arquitectura de seguridad de Windows Mobile

Un CSP es un módulo independiente que contiene implementaciones criptográficas de diversos algoritmos y estándares para autenticación, encriptación, codificación, almacenamiento de claves, cifrado y descifrado de datos, codificación y decodificación de certificados, gestión y almacenamiento de certificados, etc. Para proveer estas funcionalidades, se requiere el uso de librerías externas que permitan el desarrollo de las funciones CryptoAPI.

Se puede concluir que el desarrollo de CSPs permite la fácil incorporación de la creación de firmas digitales y el almacenamiento seguro de claves. Existen lectores de tarjetas que son compatibles con dispositivos Window Mobile, tales como SDIO o lectores de tarjeta Bluetooth. Así, es posible desarrollar una tarjeta CSP que use una tarjeta externa para almacenar las claves y computar las firmas digitales. La firma obtenida puede estar en formato PKCS1 o CMS/PKCS7.

2.2.1 Sistema Operativo Symbian

El sistema operativo Symbian es un sistema operativo que ha sido diseñado especialmente para dispositivos móviles. En cuanto a seguridad se refiere, entre sus principales características, podemos señalar importantes aspectos relacionados con la autenticidad, confidencialidad de datos e integridad. También merece la pena mencionar mecanismos de seguridad para la instalación de aplicaciones que permiten la autorización y autenticación del proceso de instalación de software mediante la verificación de firmas digitales. Además, proporciona un módulo de criptografía para los certificados, algoritmos de criptografía estándar (cifrado de claves simétricas y asimétricas), funciones hash y generación de claves. Estas funciones no pueden ser usadas directamente, pero se usan por otros módulos de seguridad como el módulo de gestión de certificados.

En la figura 4 podemos ver que la arquitectura de seguridad de un sistema operativo Symbian consiste básicamente en dos componentes de alto nivel. Uno de ellos es la gestión de certificados, cuyo principal propósito es proporcionar almacenamiento de certificados y garantizar su veracidad; el otro componente es un elemento criptográfico, cuyo propósito ha sido mencionado con anterioridad.

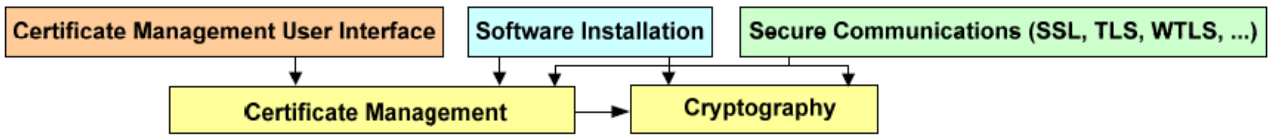

Figura 2.4. Arquitectura de seguridad de un sistema operativo Symbian

Estos módulos son la base de una serie de componentes de alto nivel, que incluyen una interfaz de usuario que proporciona gestión de certificados, instalación de software y comunicaciones seguras. De forma similar a Windows Mobiles, el sistema operativo Symbian define un esquema que permite integrar dispositivos criptográficos como WIMs.

A pesar del hecho de poseer mecanismos que permiten implementar la firma electrónica, resulta difícil utilizarlos porque están limitados a sistemas operativos Symbian y el acceso está muy limitado.

2.2.2 Java ME

La plataforma de Java edición para móviles (Java Mobile Edition), también conocida como J2ME, es una tecnología para el desarrollo de aplicaciones Java para dispositivos móviles. Se basa en una máquina virtual que permite crear un programa codificado en Java para el dispositivo móvil. Los servicios Java ME están basados en programas locales llamados *MIDlets* que el usuario puede importar directamente al dispositivo. Existen dos configuraciones para Java ME: CLDC (Connected Limited Device Configuration) y CDC (Connected Device Configuration). CLDC define un conjunto de interfaces y una máquina virtual Java, la máquina virtual K o KVM para dispositivos pequeños. CDC extiende las funcionalidades de CLDC y añade una nueva máquina virtual CVM, para trabajar con dispositivos de mayor capacidad.

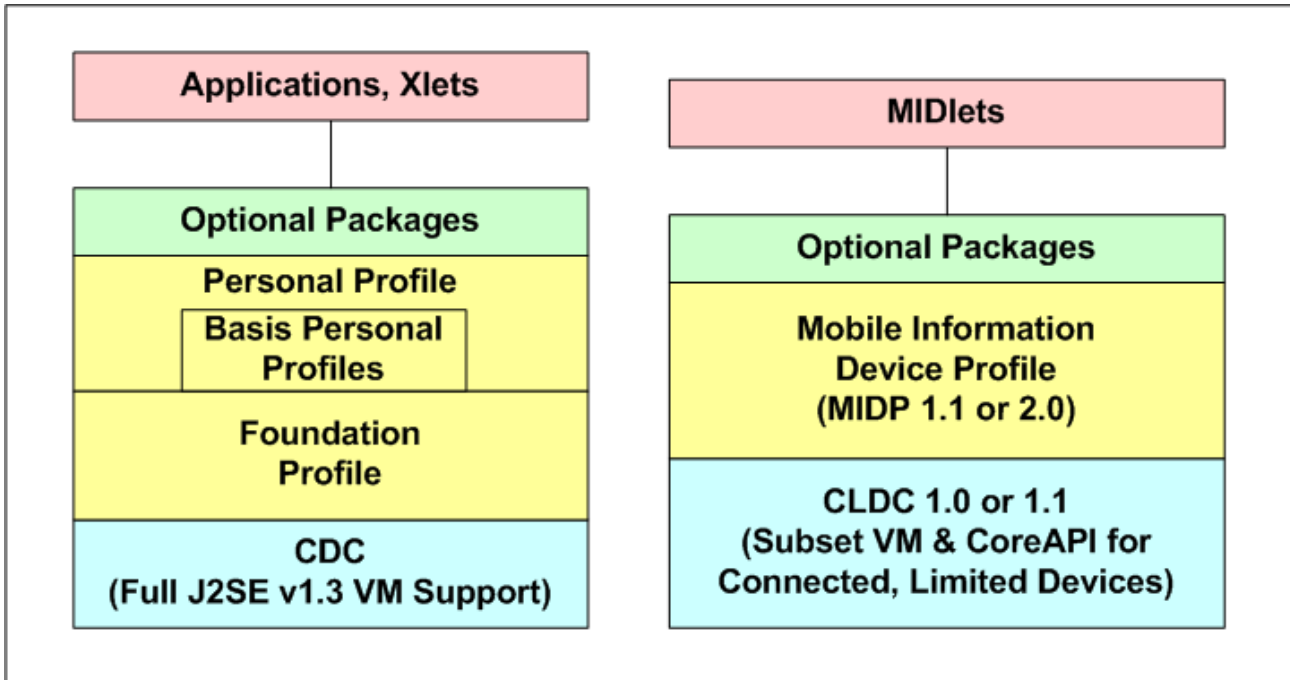

Figura 2.5. Plataforma Java 2 Micro Edition

A continuación, se analizarán los diferentes mecanismos usados para la generación de claves, la gestión y almacenamiento de certificados. También se hará mención de cómo se puede realizar el proceso de firmas electrónicas en dispositivos móviles usando esta tecnología.

Las operaciones criptográficas se consiguen mediante la incorporación de librerías, que proporcionan funciones criptográficas para dispositivos J2ME. Las librerías existentes son:

- Bouncy Castle. Se trata de una API criptográfica gratuita que proporciona métodos para las operaciones criptográficas más importantes: importación de clave privada y pública y certificados (PKCS12, PKCS8, PKCS7), generación de ambas claves y realización de firma digital y encriptación entre otras.

- IAIK Micro Edition: Es una API criptográfica cuyas principales características son encriptación simétrica/asimétrica, gestión de claves y certificados. Es compatible con todos los MIDP y soporta PKCS12, PKCS8, PKCS7, X.509, ASN.1 BER/DER y Base64. Presenta el inconveniente de no ser gratuita.

- Phaos Micro Foundation: API criptográfica que puede ser usada con las configuraciones CLDC y CDC y MIDP. Sus principales características criptográficas son; encriptación AES, 3DES, RC4, RC2, y RSA, firma DSA y RSA, algoritmos SGA1 y MD5, y certificados PKCS12, PKCS8, PKCS5 y x509v3. Esta librería tampoco es gratuita.

- NTRU Neo para Java: Esta librería contiene un conjunto de algoritmos especialmente diseñados para dispositivos con límites de recursos. Contiene encriptación simétrica y asimétrica y su propio algoritmo NTRU. Además, puede ser instalada en configuraciones CDLC y CDC. Sin embargo, no es gratuita.

Estas librerías proporcionan diferentes métodos para realizar operaciones criptográficas que requieren la ejecución de algoritmos. Sin embargo, aparte de esta funcionalidad, es necesario

proporcionar materiales necesarios para realizar estas operaciones, como son certificados, uso de claves, etc. La siguiente lista muestra una serie de posibles métodos para suministrar a las librerías criptográficas todo lo necesario:

- Generación de claves privadas en el dispositivo móvil. Las claves privadas se generarían en el interior del terminal. Posteriormente la petición de firma es generada y enviada a la autoridad registradora RA vía HTTP. Finalmente, el certificado es almacenado en el dispositivo. Esta funcionalidad puede ser proporcionada, por ejemplo, por la librería Bouncy Castle.

- Navegación por el sistema de archivos de dispositivo móvil. Las claves y certificados se deben almacenar en el dispositivo, por lo que es necesario acceder al sistema de archivos. Esto es posible mediante JSR-75, denominada Api File Connection.

- Uso de HTTPS. Es específico de MIDP 2.0. Es similar a HTTP, pero con el uso de protocolos SSL/TLS, por lo que se añade una capa de seguridad adicional que encripta las comunicaciones.

Además de todo lo anterior, necesitamos un mecanismo que permita almacenar las claves y los certificados en el entorno J2ME.

- Almacenamiento en el dispositivo usando Java Me Record Management System (RMS). Los MIDlets pueden almacenar datos usando el sistema de registro y éstos pueden ser recuperados posteriormente.

- Almacenamiento en el sistema de ficheros del dispositivo. El almacenamiento de información criptográfica depende de si la máquina virtual soporta la API JSR-75 o no. En caso de que sea soportado, se puede acceder a información criptográfica almacenada en el sistema de ficheros, por ejemplo, acceder a un fichero PKCS12 que contiene una clave privada y el certificado.

Después de este breve análisis, se puede concluir que el entorno Java Me presenta todos los componentes necesarios para el desarrollo de firmas electrónicas.

2.3 Tecnologías híbridas

Existen un gran número de servicios que no es posible implementar completamente en la memoria SIM ni tampoco en el dispositivo en sí. En esta sección se analizarán las principales tecnologías que se pueden aplicar en ambos sentidos (memoria SIM y en el dispositivo).

2.3.1 SATSA

SATSA (Security And Trust Services API) es una especificación para Java ME que proporciona la posibilidad de establecer una comunicación entre un MIDlet Java y un elemento de seguridad (por ejemplo, una tarjeta SIM). Mediante el uso de SATSA, también conocido como JSR-177, una tarjeta SIM puede implementar procesos de seguridad, como por ejemplo, firma electrónica, en

aplicaciones Java ME.

SATSA define tres formas de comunicación con la tarjeta SIM. El modo de comunicación depende del tipo de aplicación SIM. Estas alternativas están basadas en el uso de tres paquetes: el paquete APDU para el uso de comandos APDU a través del protocolo ISO 7816; el paquete JCRMI para peticiones del objeto JavaCard RMI, y el paquete PKI para el uso de la aplicación WIM de la tarjeta SIM en procesos criptográficos.

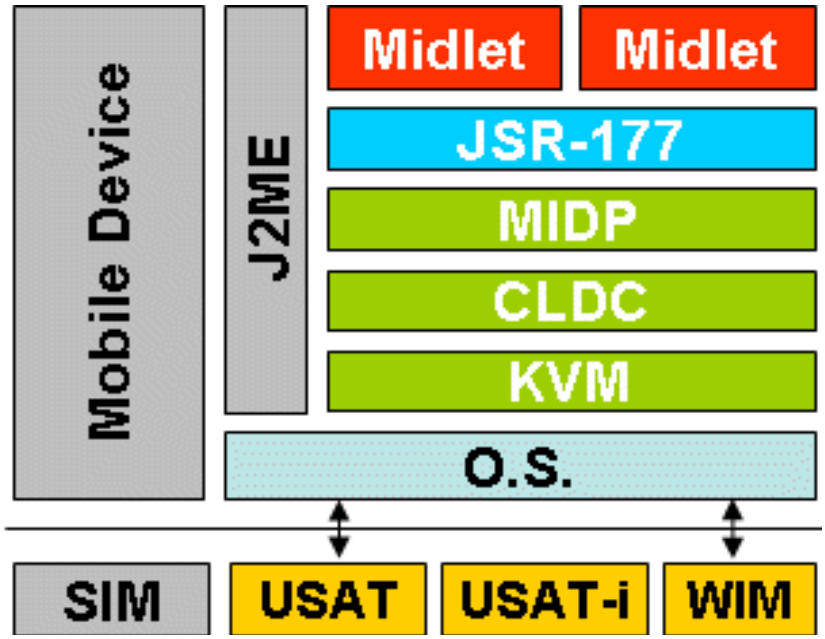

Figura 2.6. Arquitectura SATSA

Aunque esta especificación proporciona un nivel alto de abstracción a las aplicaciones Java ME, no existen muchos dispositivos móviles que soporten SATSA. Sin embargo, los fabricantes están incorporándola a los dispositivos con más frecuencia cada vez.

2.4 Soluciones independientes del dispositivo

Como ya se ha comentado anteriormente, la firma electrónica es un asunto de relevancia en el mundo del comercio electrónico para proporcionar servicios seguros. Esto significa que una aplicación tiene que desarrollar un software que debe implementar la firma electrónica en dispositivos móviles distintos. Debido a la cantidad de variedad de dispositivos, esto supone un problema de desarrollo e inversión. Para solucionar este número de problemas, existen aplicaciones que realizan peticiones de firma directamente a un servidor. A continuación se realizará un detalle más exhaustivo de este tipo de soluciones.

2.4.1 Firmas basadas en un servidor

Inicialmente, los dispositivos móviles han tenido grandes limitaciones de seguridad. Este hecho

implica que los dispositivos eran incapaces de firmar información usando criptografía asimétrica. Para solucionar esta situación, la solución propuesta fue introducir un servidor con la función de poseer la información necesaria para desarrollar el proceso de firma electrónica con el consentimiento del usuario. En este tipo de soluciones el dispositivo móvil se usa para validar la firma generada en el servidor. De esta forma, el servidor, antes de generar la firma, requiere autenticación por parte del usuario desde el dispositivo móvil. El proceso seguido es el de la figura 2.7.

Figura 2.7. Proceso de firma electrónica generada en un servidor

Este tipo de soluciones son usadas por dispositivos con pocos recursos. La principal ventaja es su simplicidad, además del fácil desarrollo de la solución. Esto es porque cualquier módulo SIM contenido en un dispositivo móvil es capaz de generar y usar claves simétricas puesto que estas capacidades son requeridas por el estándar GSM. Es importante señalar que este tipo de soluciones no está estandarizado y que depende de la implementación del servidor de firma. Otro de los inconvenientes importantes que presenta es el hecho de no estar bajo el control único del usuario, por lo que la firma generada no se puede considerar como una firma totalmente legal. Además, el servidor debe almacenar la clave privada del usuario y su certificado, lo que no deja de resultar seguro totalmente. El servidor puede ser proporcionado por el operador de red o por una tercera parte de confianza, puesto que almacena información de confianza. Otro problema importante es la no estandarización de la interfaz que describe la comunicación entre el servidor de firma y la aplicación.

2.4.2 Servicio de firma móvil

El servicio de firma móvil (MSS) nació para facilitar el desarrollo de soluciones basadas en firma para dispositivos móviles. A continuación, se describirá cómo funciona este tipo de servicios.

Se pueden diferenciar varios roles: usuario final, autoridad registradora, autoridad certificadora, el servicio proveedor del servicio de firma móvil MSSP (Mobile Signature Service Provider), el proveedor de la aplicación AP, entre otros. Estos elementos pueden verse en la figura 2.8.

El usuario final posee un dispositivo móvil que contiene una tarjeta inteligente proporcionada normalmente por el operador móvil. Esta tarjeta contiene una aplicación de firma que puede generar un proceso de firma para validar la identidad del usuario. Esta aplicación de firma está protegida por un PIN proporcionado por la tarjeta inteligente. En ningún momento el usuario puede cambiar este PIN.

La aplicación de firma es capaz de generar nuevas claves además de almacenar los certificados asociados a la generación de claves. Los certificados deben ser obtenidos por un autoridad

certificadora. La aplicación puede ser invocada por el MSSP para obtener la firma electrónica de los datos. Antes de que la firma sea generada, el usuario introduce el PIN de firma en su dispositivo móvil para validar que él es el usuario que va a firmar los datos. De esta forma, el proceso de firma está siempre bajo el control del usuario.

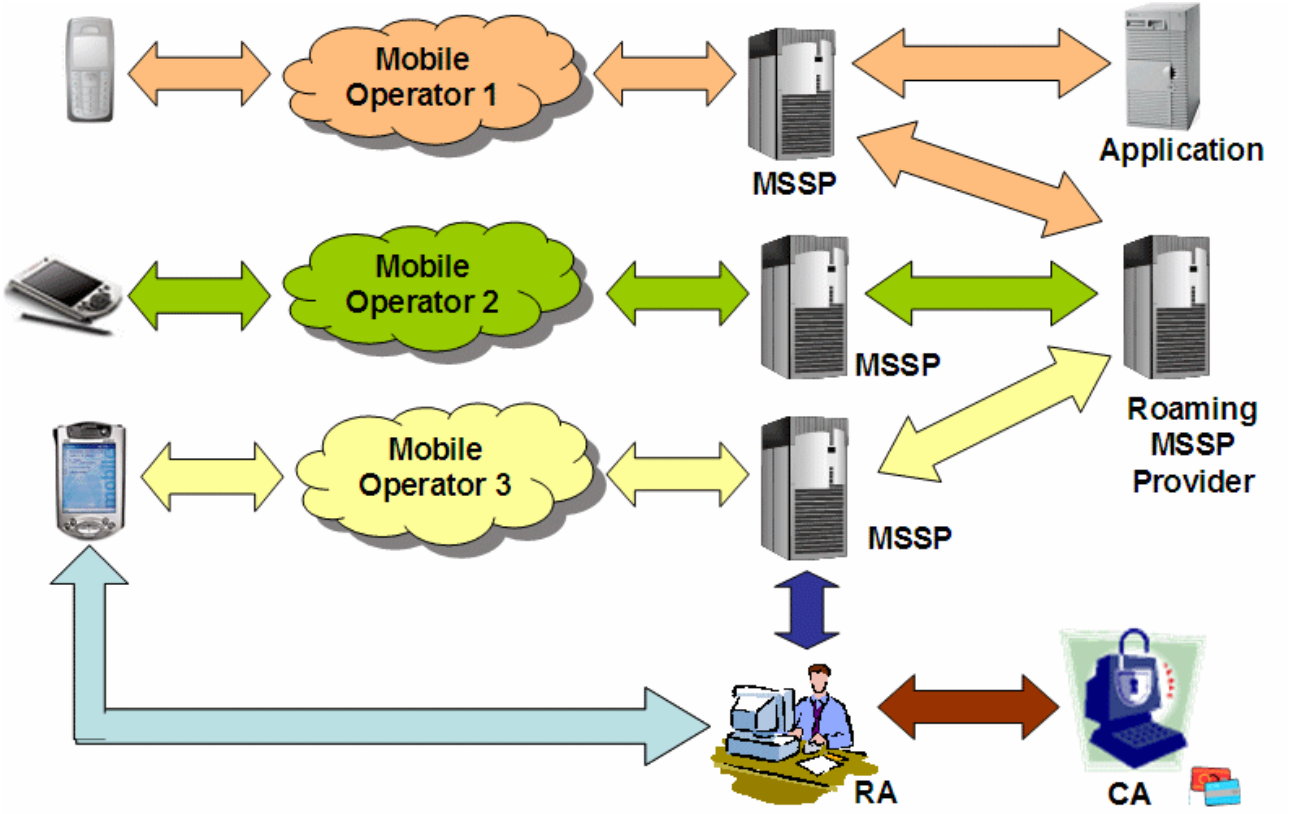

Figura 2.8. Servicio de firma móvil

Actualmente, existen diversas compañías que proporcionan servicios basados en este propósito. La principal ventaja es que la aplicación no tiene por qué saber nada del desarrollo del proceso de firma. La tarea es responsabilidad del MSSP. Otra característica importante es que, aunque la aplicación no tenga parte en el proceso de firma, sí que puede especificar aspectos relacionados con la firma como el formato, los datos que pueden ser mostrados, las claves a usar, etc. Sin embargo, presenta algunos inconvenientes, tales como que la aplicación tiene que ponerse de acuerdo con el MSSP que proporciona el servicio y que no todos los operadores móviles ofrecen esta posibilidad.