Comparación de los Mecanismos de Seguridad Brindados por .Net Compact Framework y MSA para el Desarrollo de Aplicaciones Móviles

Fecha de Recepción: Febrero de 2010
Fecha de Aprobación: Mayo de 2010

LUIS EDUARDO MELÉNDEZ CAMPIS
Estudiante de Doctorado
en Ingeniería Telemática,
Universidad de Vigo.
Especialista en Telecomunicaciones,
Universidad Autónoma de Bucaramanga
Ingeniero de Sistemas,
Politecnico Grancolombiano
melendezcampis@hotmail.com

RESUMEN

La idea central del texto no es crear un manual de cómo implementar mecanismo de seguridad en ambas plataformas, el texto busca dar una visión general de cómo cada una de la plataformas en cuestión atacan la temática de la seguridad de la información, que mecanismos proponen para garantizar la misma, y posteriormente realizar una comparación desde un punto de vista crítico de estos items.

2. INSEGURIDAD INFORMÁTICA EN APLICACIONES MÓVILES

En los últimos años el fenómeno de la inseguridad informática a tomado fuerza y dejo de ser un problema aislado para convertirse en una problemática real para todas las organizaciones del mundo. El entorno móvil no ha sido ajeno a este fenómeno, ha sido uno de los más golpeados debido a la característica de inseguridad que son inherente al medio de transmisión que este posee, el aire, otro apremiante ha sido la poca importancia que dieron los fabricantes de terminales y sistema operativos móviles a la seguridad de la información, lo que convirtió a estos dispositivos y sus aplicaciones en un blanco apetecido por los atacantes.

Debido a la masificación de las terminales móviles inteligentes, al gran numero de servicios y valores agregados que estas comenzaron a brindar y a la importancia que tomaron dentro de las actividades productivas de los usuarios finales, obligaron a los que antes ignoraron a tan importante ítem dentro de las comunicaciones y la informática a desarrollar mecanismos para garantizar esta en las terminales, aplicaciones y en el transporte de los datos. Ahora entraremos a dar un vistazo como Microsoft [1] y Sun Microsystem [2] atacan esta problemática desde el punto de vista de la inclusión de mecanismos y controles de seguridad dentro del desarrollo de aplicaciones móviles.

3. SEGURIDAD EN .NET COMPACT FRAMEWORK

El .Net Compact Framework divide la seguridad de la información en sus aplicaciones en tres capas o niveles [3]:

- Seguridad del dispositivo
- Seguridad de la aplicación
- Seguridad en la comunicación

La seguridad del dispositivo es la capa en la que los mecanismos y controles de seguridad son brindados directamente por la terminal movil o a través de la instalación de software fabricados por terceros, estos mecanismos se encargaran de garantizar el control en el acceso a la terminal movil, la seguridad en la instalación del sistema operativo y de software fabricados por terceros. Entre estos mecanismos tenemos los mecanismos de autenticación, la protección con antivirus y los bloqueos de seguridad.

Los mecanismos de autenticación son todos aquellos que nos permiten tener certeza de que el usuario que está tratando de hacer uso de la terminal movil es quien dice ser. Estos mecanismos se pueden clasificar según el método usado para autenticar a los usuarios en:

- Basados en algo que se tiene
- Basados en algo que se sabe
- Basados en algo que se es

Los sistemas de autenticación basados en algo que se tiene son aquellos que utilizan para este proceso tokens o tarjetas que contienen información personal del usuario al que esta pertenezca y es usada para el proceso de verificación de identidad. Los que se

basan en algo que se sabe, son aquellos que usan para el proceso de verificación combinaciones de caracteres alfanuméricos que se asocian con la identificación del usuario a autenticar. Y los que se basan en algo que se es, usan características biológicas únicas propias de cada usuario para autenticar la identidad del mismo.

En las terminales móviles el uso de sistemas de autenticación basados en algo que se sabe es común, a través del uso de contraseñas alfanuméricas de longitud variable para la autenticación de los usuarios en el momento del encendido de la terminal movil.

Figura 1. Sistema de autenticación por contraseña en terminales moviles

Aunque en actualidad ya existen terminales móviles que utilizan autenticación basada en algo que se es, ejemplo de ello es el proyecto Mofira de Sony Corporations, este sistema biométrico permite identificar al usuario mediante la lectura de las venas de un dedo, se vale de un sensor CMOS y de una luz infrarroja para hacerlo, el patrón de la vena es extraído mediante una imagen del dedo, es comprimido y almacenado en la memoria del dispositivo, esta tecnología ofrece una respuesta rápida (toma 0,25 segundos al teléfono movil realizar la autenticación) y de alta precisión [5].

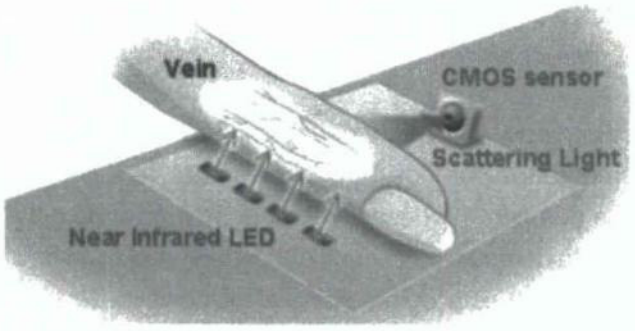


Figura 2. Sistema autenticación biométrico para terminales móviles MOFIRA

Los sistemas de antivirus para dispositivos móviles aparecen como una solución a la actual problemática provocada por la producción y distribución masiva de malware creado para afectar el normal funcionamiento de los sistemas operativos para terminales móviles como Windows Mobile, Symbian, Android, entre otros, esto debido a que las terminales móviles han dejado de ser simples aparatos telefónicos, para convertirse en oficinas móviles las cuales son usadas como medio de almacenamiento y procesamiento alterno de información altamente sensible para las actividades laborales de los usuarios móviles. Estas soluciones por lo general desarrolladas por terceros, permiten neutralizar la instalación y ejecución de código

malware en las terminales móviles, permitiendo así garantizar la integridad del sistema operativo y posibles daños a su estructura o la de los archivos en los ellos almacenados. Dentro del abanico de soluciones existentes se destaca la propuesta por Kaspersky Labs, la Kaspersky Mobile Security, es una suite de herramientas para terminales móviles inteligentes (Smartphones o PDA's) que permiten un nivel optimo de protección ante programas maliciosos (antivirus en tiempo real), spam SMS (filtros según remitente y contenido), ataques virtuales que afectan a las plataformas de las terminales móviles (firewall) y protege la información confidencial almacenada en el dispositivo en caso de robo o perdida del equipo (sms block, sms clean y sms watch) [6].

Los bloqueos de seguridad son un mecanismo implementado por los fabricantes de terminales móviles que permite desactivar algunas funciones del dispositivo de acuerdo al perfil del usuario o el nivel de seguridad que este quiera darle al dispositivo. Estos bloqueos permiten limitar la capacidad del usuario para escribir y almacenar información en la terminal móvil, proteger con password o bloquear las conexiones por bluethooth, restringir la carga de datos desde tarjetas smartcard o dispositivos USB, entre otros tipos de bloqueos.

La seguridad en la aplicación este nivel de aseguramiento es implantado directamente en la aplicación en desarrollo, por lo tanto este nivel de aseguramiento debe ser definido por el programador. El .Net Compact Framework ofrece una serie de mecanismos que permiten garantizar el aseguramiento de las aplicaciones desarrolladas bajo este marco de trabajo, algunos de ellos son:

- SqlClient
- SQLCE connectivity
- XML Web Services
- **HTTP Pluggable Protocol**

Estos mecanismos permiten realizar un control sobre el proceso de autenticación y autorización de los usuarios que hacen uso del sistema informático desde una plataforma móvil

- SQLCE Encryption
- File Encryption

Estos permiten crear un compendio criptográfico que se encarga de garantizar la confidencialidad de los datos usados y almacenados por la aplicación dentro del sistema informático móvil.

- **Validación de las entradas del usuario**

Los mecanismos de validación de entradas de usuario son desarrollados en su totalidad por el programador y para la implementación de estos se requiere un mínimo conocimiento de técnicas ofensivas para crear los patrones de los ataques a filtrar.

Estos controles permiten evitar inserción de caracteres especiales o cadenas de caracteres que provocan errores en el comportamiento de la aplicación, dichos errores facilitan el proceso obtención de información y evasión de otros mecanismos de seguridad implementados en la aplicación o en el dispositivo móvil. Las técnicas que comúnmente son filtradas por este tipo de controles en aplicaciones desarrolladas bajo el .Net Compact Framework son el SQL Inyection y el Cross Site Scripting o XSS.

```
private string SanitizeIntoSqlLiteral(string strQ)
{
    //Convert single quote into two single quotes
    return strQ.Replace("' ", "'' ");
}
```

Figura 3. Ejemplo de una función que valida entradas de usuarios contra ataques de tipo SQL Inyection

La seguridad de las comunicaciones es la capa en la que se garantizara la integridad y confidencialidad de la información que usan o generan las aplicaciones desarrolladas bajo el .Net Compact Framework mientras viajan a través de la red de transporte.

Anque estas redes de transporte ofrecen mecanismos de seguridad embebidos dentro de la tecnología que soporta a estas, como el encriptamiento WEP o WPA en redes inalámbricas IEEE 802.11a/b/g, o el fuerte encriptamiento sobre las redes de telefonía móvil GSM, deben existir mecanismos alternos que garantizen la seguridad de la información en caso de que algunos de los mecanismos brindados por las redes de transporte falle.

Por otro lado la seguridad de las redes de transporte llega hasta el core de la red que interconecta con la red pública de hay en adelante dependemos de los mecanismos que el desarrollador implemente para el uso y transporte de la información por parte de la aplicación.

Los mecanismos comúnmente usados por este marco de trabajo son:

- **VPN (Virtual Private Network)**

Una red privada virtual o también conocidas por su sigla VPN, son un túnel que se crea sobre una red pública usando para ellos algoritmos de encriptamiento sobre la información que se va a transmitir y autenticando los extremos de la comunicación.

En la siguiente grafica se puede ver cómo opera una VPN sobre un sistema móvil.

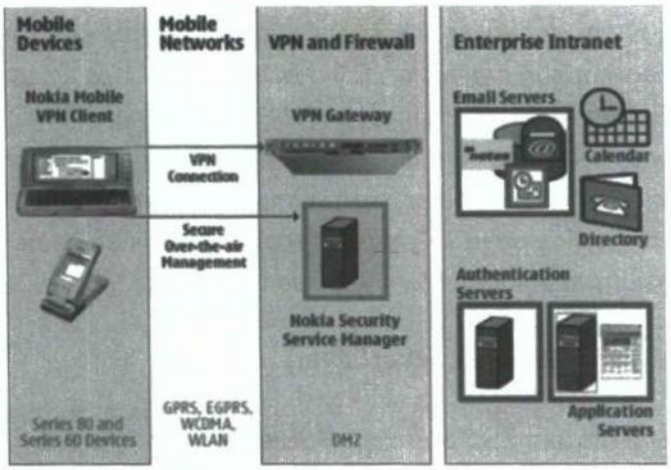


Figura 4. Funcionamiento de una VPN sobre un sistema móvil

El principal uso de las VPN es permitir a los usuarios conectarse a los recursos informáticos de las redes privadas de sus organizaciones a través de una red pública insegura. Para la implementación de este tipo de soluciones sobre .Net Compact Framework se utilizan protocolos como IPSEC [10] o PPTP [11].

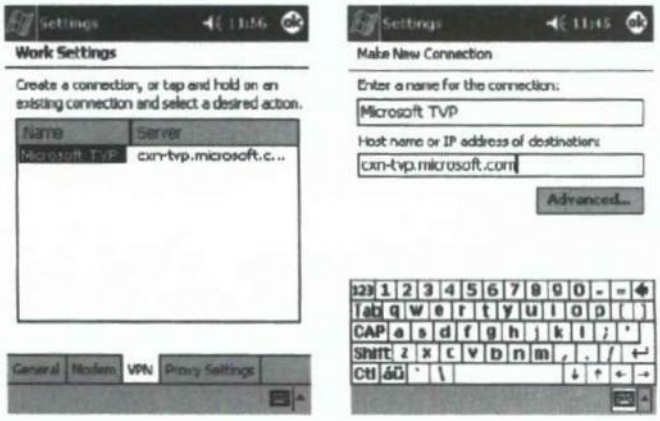


Figura 5. Pantalla de configuración de VPN en una terminal móvil con Windows Mobile

Una vez establecida una conexión VPN a la red privada, su utilización es transparente para. NET Compact Framework. Se Puede utilizar la clase

System.Net.Sockets.Socket para comunicarse con un servidor con protocolo TCP o UDP, el uso de la clase System.Net.WebRequest y WebResponse para iniciar una comunicación usando el protocolos HTTP, o simplemente trabajar con archivos en una carpeta compartida de red. Todas las transferencias de datos están protegidos por el cifrado de VPN [4].

- **SSL (Secure Socket Layer)**

El protocolo SSL o Secure Socket Layer proporciona cifrado de los datos que intercambia entre un servidor que cuenta con una extensión SSL (HTTPS, SMTPS, POP3S, etc ...) y un cliente que invoca el recurso de este, usando para esta labor un algoritmo de cifrado RC4 o IDEA y cifrando la clave privada de RC4 o IDEA mediante un algoritmo de cifrado de clave pública como RSA. A diferencia de las VPN, SSL no autentica los extremos de la comunicación, solo encripta la información lo que lo convierte en un protocolo que solo puede garantizar la confidencialidad de la información que transporta.

.Net Compact Framework soporta el uso de SSL con llaves de encriptamiento de 128 bits, este soporte incluye el uso de socket, de peticiones web y de llamados a XML Web service.

- **Personalización de las SOAP Extensions**

Las SOAP extensions ***permiten ampliar las funcionalidades de un web service mediante la modificación de los mensajes SOAP enviados y recibidos por un servicio o cliente de un web service*** [12], un ejemplo claro del uso de esta técnica enfocada a la seguridad en el transporte de la información seria la implementación de un algoritmo de cifrado para que se ejecute con un web service existente y sea consumido por los clientes al momento que estos lo invoquen. Para realizar esta labor se debe tener en cuenta el ciclo de vida de un web service, en la siguiente gráfica podemos observar el ítem en cuestión.

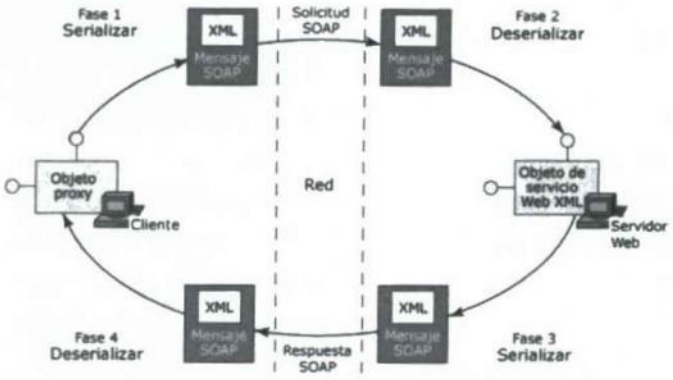


Figura 6. Ciclo de vida de un web service

Como puede ver, .NET Framework serializa y deserializa el código XML durante las fases tanto en el equipo donde reside el servicio Web como en el equipo del cliente de servicios Web. Una extensión SOAP se puede insertar en la infraestructura para inspeccionar o modificar los mensajes SOAP antes y después de cada una de estas fases de serialización y deserialización [12]. Teniendo en cuenta esto, si se desea encriptar los mensajes SOAP esta operación se realizara en ambos extremos y deberá ser realizada posterior al proceso de serializacion para el envio y antes de la deserializacion para la recepción de mensajes SOAP.

SEGURIDAD EN MOBILE SERVICE ARCHITECTURE (MSA)

A diferencia de .Net Compact Framework, la Mobile Service Architecture (MSA), cuenta con un conjunto de API's que se encargan en gran parte de mantener los mecanismos de seguridad usados por los programadores en el desarrollo de aplicaciones para ambientes móviles, la Security and Trust Services API for J2ME mas conocida como SATSA "provee

servicios de seguridad a las aplicaciones y dispositivos móviles, incluyendo el almacenamiento seguro para proteger la información sensible, las operaciones criptográficas para apoyar las operaciones de pago y la integridad de los datos, y un entorno de ejecución seguro que permite el despliegue de las características de seguridad personalizadas".

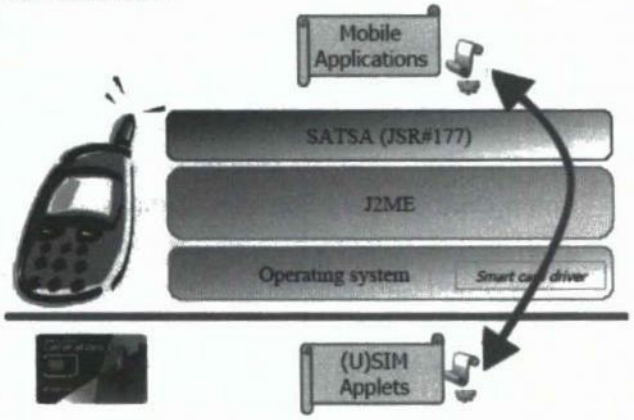
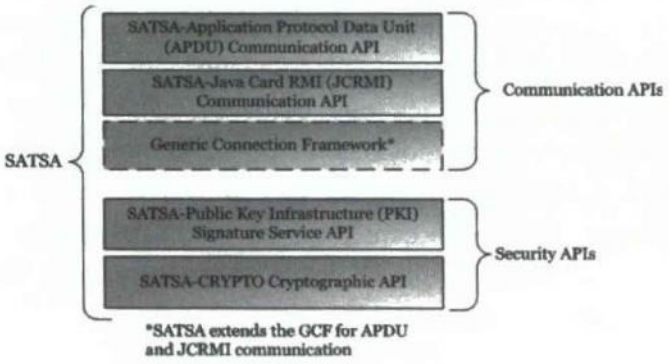


Figura 7. Entorno de operación de SATSA

La SATSA está compuesta por 5 paquetes como se ve en la siguiente gráfica:



El paquete SATSA Application Data Unit (APDU) se encuentra en el paquete javax.microedition.apdu y permite a las aplicaciones desarrolladas bajo el marco de trabajo J2ME establecer un vinculo con las aplicaciones de la Smart Card o también conocido como elemento de seguridad.

El paquete **SATSA Java Card RMI (JCRMI)** permite la comunicación con una Smart Card haciendo uso del protocolo del mismo nombre. Este protocolo ayuda a las aplicaciones que no están almacenadas en esta a utilizar objetos que si lo están. Estas aplicaciones utilizan una interfaz remota para interactuar con los objetos presentes en la Smart Card. El funcionamiento de este paquete es basado en el protocolo RPC y para su uso lo hace a través del la tecnología RMI.

El paquete **SATSA Public Key Infraestructure (PKI)** Este paquete permite a las aplicaciones J2ME solicitar a la Smart Card operaciones de firma digital, y también incluye algunos métodos de gestión básica de certificados digitales.

El paquete **SATSA CRYPTO** provee a las aplicaciones J2ME de una serie de herramientas criptográficas que les permiten realizar operaciones como hash, firma digital o cifrado de datos simétricos y asimétricos.

Podemos observar en la composición de SATSA que esta agrupa los paquetes en dos grupos, las API's de Comunicaciones compuestas por la SATSA Application Data Unit (APDU), la SATSA Java Card RMI (JCRMI) y el Generic Connection Framework, y las API's de Seguridad que incluye los paquetes, la SATSA Public Key Infraestructure (PKI) y la SATSA CRYPTO.

Cada uno de estos grupos poseen funciones distintas dentro del proceso de aseguramiento de la aplicación y las comunicaciones en un entorno móvil, el API de comunicaciones se encarga de establecer un vinculo entre las aplicaciones J2ME con los mecanismos de seguridad brindados por el SATSA, y el API de seguridad se encarga de prestar servicios de aseguramiento de las aplicaciones y las comunicaciones a través de la implementación de

firmas digitales, credenciales de usuarios o certificados digitales y compendios criptográficos simétricos y asimétricos aplicados a la información almacenada y las comunicaciones.

En la siguiente grafica podemos observar el modelo de comunicaciones de SATSA, donde interviene una aplicación J2ME y un Security Element representado por una Smart card (Sim Card), de esta forma la SIM Card realiza operaciones de seguridad como firma y autenticación de usuarios en una aplicación J2ME.

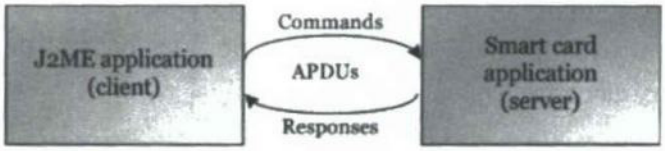


Figura 9. Modelo de comunicaciones SATSA [8]

El API de seguridad de SATSA, posee dos mecanismos de seguridad que son los que de cierta manera permiten garantizar la seguridad de la información manejada por las aplicaciones J2ME, estos mecanismos son:

- The Public Key Infrastructure (PKI)
- The cryptographic (CRYPTO)

A través de estos dos mecanismos se le permite a las aplicaciones J2ME:

- ***Trabajar con certificados públicos digitales, llaves públicas y privadas, Hash y firmas digitales.***
- ***Crear, almacenar y utilizar las credenciales de usuarios basadas en certificados digitales X.509.***
- ***Cifrar datos usando para ellos algoritmos criptográficos simétricos y asimétricos[9].***

4. CONCLUSIONES

Después de realizar un recorrido a través de los modelos y mecanismos de seguridad que las tecnologías .Net Compact Framework y MSA ofrecen a los programadores que desarrollan aplicaciones móviles se puede llegar a las siguientes conclusiones:

- La arquitectura MSA aunque posee un API bien definido para la implementacion de seguridad en aplicaciones móviles, este está ligado estrechamente con el elemento de seguridad (Smart Card) lo que hace que su implementación dependa de si la terminal móvil posee o soporta el tipo de Smart Cart que maneja el API.
- El marco de trabajo .Net Compact Framework posee una estructura para el manejo de la seguridad muy completa y a la vez simple, aunque la implementación de muchos de estos mecanismos dependa de software comerciales desarrollados por terceros.
- Es interesante como .Net Compact Framework tiene en cuenta como un factor de riesgo las entradas de datos de los usuarios, creando para ello mecanismos que permitan evitar ataques como SQL Inyection o XSS típicamente provocados por entrada de caracteres especiales o cadena de caracteres por parte del usuario, característica que no posee dentro de sus API de seguridad MSA.
- MSA denota mucha fortaleza en aplicaciones donde la seguridad en la transferencia de los datos es critica, ya que cuenta con mecanismos para el encriptamiento de la información y la autenticación de clientes sin tener que usar herramientas de terceros fabricantes.

- Aunque ambos marcos de trabajo usan lenguajes orientados a objetos y debido a esto la generación de código para las soluciones se hace un poco complicado debido al nivel de abstracción y que .Net Compact Framework presenta de forma dispersa sus mecanismos de seguridad, el desarrollo e implantación de esquemas de seguridad es mucho más sencillo en .Net Compact Framework que en MSA.

- Otro punto importante es la aceptación que tiene .Net Compact Framework en la comunidad de programadores para el desarrollo de aplicaciones móviles donde la seguridad de la información es un factor crítico, debido a la sencillez del modelo de seguridad y mecanismos que esta maneja.

- A diferencia de .Net Compact Framework que presenta un modelo de comunicaciones entre sus componentes de seguridad bastante simple aunque un poco disperso, MSA presenta un modelo de comunicaciones entre los componentes del API SATSA complejo y muy sesgado al uso de Smart Cart como elemento de seguridad central.

- Un punto importante también es la cantidad y calidad del material cognitivo que facilitan los fabricantes de estos marcos de trabajo, donde Sun Microsystem suministra al publico mucho material acerca de su API SATSA y de excelente calidad a diferencia de Microsoft que presenta pobresa en la cantidad de material disponible al público y de muy baja calidad, para encontrar fuentes de información que permitan estudiar este marco de trabajo en particular nos debemos dirigir a terceros diferentes al fabricante para encontrar material relevante.

Referencias bibliográficas

[1] Sitio Web www.microsoft.com

[2] Sitio Web java.sun.com

[3]Dan Fox, Jon Box, "Building solutions with the Microsoft .Net compact framework"

[4]Andy Wigley, Stephen Wheelwright, "Microsoft .Net compact framework – Core reference"

[5]Sitio Web http://espegizmo.com/2009/02/02/general/mofiria-el-sistema-biometrico-de-sony-capaz-de-leer-las-venas-de-tu-dedo/

[6]Sitio Web http://www.kaspersky.com/sp/kaspersky_mobile_security

[7] Sun Microsystem, "Mobile Service Architecture - Meeting the Growing Needs of Mobile Handsets", marzo del 2007.

[8] Sun Microsystem, Enrique Ortiz, "The Security and Trust Services API for J2ME, Part 1", marzo del 2005.

[9] Sun Microsystem, Enrique Ortiz, "The Security and Trust Services API for J2ME, Part 2", septiembre del 2005.

[10]Sitio Web: http://www.ipsec-howto.org/spanish/x161.html

[11]Sitio Web http://www.ietf.org/rfc/rfc2637.txt

[12]Sitio Web http://msdn.microsoft.com/es-es/library/esw638yk(VS.80).aspx