Revista Tekhnê
2011, Vol. 8, 5–12

ISSN 1692-8407

Implementación del criptosistema de curva elíptica en un prototipo de aplicación móvil para E-Commerce

Implementation of elliptic curve cryptosystem in a mobile application prototype for E-Commerce

Camilo A. Barbosa
Directv Colombia
camiloandb@gmail.com

Yesid A. Tibaquira
Avianca
yatibaquirac@correo.udistrital.edu.co

Gerardo Castang Montiel
Universidad Distrital Francisco José de Caldas
gcastangm@udistrital.edu.co

El prototipo consiste en la implementación del criptosistema de curva elíptica en una aplicación móvil y en una aplicación Web. Se desarrolló para el cifrado y descifrado del flujo de datos entre las dos aplicaciones, y opera bajo un entorno *e-commerce* en el contexto de transacciones bancarias. El criptosistema de curva elíptica fue definido como opción de añadir seguridad a las aplicaciones dadas sus ventajas frente a otros criptosistemas de clave pública como RSA. Se trabajó la especificación de java J2ME y la API correspondiente de Bouncycastle para entornos móviles. En la aplicación Web, se diseñó un Servicio Web, accesible desde la aplicación móvil, basado en la especificación JEE5. Para el envío de objetos entre el dispositivo móvil y el servicio Web se diseñó un esquema de serialización (conversión de un objeto a flujo de bytes) y deserialización (proceso inverso) para facilitar la transmisión de la información.

Palabras clave: aplicación web, criptosistema, curva elíptica, entornos móviles

The prototype is the implementation of elliptic curve cryptosystem in a mobile application and Web application for encryption and decryption of data stream between the two applications under e-commerce environment in the context of banking transactions. The elliptic curve cryptosystem was defined as an option to add security to applications, because of its advantages over other public key cryptosystems such as RSA. The java J2ME specification and corresponding API Bouncycastle for mobile environments was used. In the Web application, a based java JEE5 Web service was designed, accessible from the mobile application. For sending objects between the mobile device and Web service a scheme for serialization (object to bytes stream) and deserialization (reverse process) was designed for easy information transmission.

Keywords: cryptosystem, elliptic curve, mobile environments, web application

Introducción

El comercio por medios electrónicos es una las actividades que un usuario puede realizar desde cualquier computador de escritorio o portátil, que tenga acceso a Internet. Sin embargo, esta actividad, ha traspasado las fronteras de los equipos de cómputo para ser parte también de los dispositivos móviles, gracias al crecimiento de la demanda de estos dispositivos como celulares y PDA's.

El auge de estos dispositivos y de las aplicaciones para comercio electrónico móvil, ha generado la necesidad de implementar un entorno seguro que garantice un alto grado de confidencialidad en el transporte de la información. Los protocolos para entornos móviles WEP y WAP hacen uso de criptosistemas en su capa de transporte para el envío y recepción de información en un entorno inseguro. Sin embargo, estos protocolos han demostrado vulnerabilidades debidas a ataques criptográficos tales como el ataque de hombre en el medio o ataques por fuerza bruta, que dan cuenta de las falencias en los algoritmos de encriptación. Ellos, por lo general, son de criptosistema simétrico.

Este proyecto surge con la finalidad de agregar un nivel de seguridad en la transmisión de información entre dispositivos móviles, implementando el criptosistema de curva elíptica, y una aplicación web, bajo un entorno en el cual se requiera la confidencialidad de la información en el momento de su transmisión. El entorno simulado adopta algunos procesos de transacciones bancarias, en los cuales el carácter de los datos transportados es crítico.

Metodología

Protocolo de aplicaciones inalámbricas WAP 2.0

El protocolo WAP es el actual estándar para el envío, recepción, presentación de información y servicios de telefonía desde terminales inalámbricas y teléfonos celulares. Esto significa que mediante el protocolo WAP, adicionalmente a los tradicionales servicios de voz, es posible manejar contenidos de comercio electrónico, Internet y servicios avanzados de datos desde un teléfono celular. Aunque es posible hacer dicho manejo con otras tecnologías, hasta el momento, WAP, es el protocolo que ha mostrado un camino estándar, definido y estable para hacerlo (Fuquene, 2008).

Modelo WAP. El protocolo WAP (Figura 1) utiliza la tecnología de proxy para optimizar y mejorar la conexión inalámbrica entre el dominio y el servidor proxy WAP, proporcionando una variedad de funciones que incluyen:

1. El protocolo de puerta de enlace (Gateway), que traduce las peticiones de una pila de protocolos inalámbricos (por ejemplo, la pila WAP 1.x- WSP, WTP, WTLS, y el PDC) para los protocolos de WWW (HTTP y TCP / IP).

2. Los codificadores y decodificadores de contenidos, que puede ser utilizado para traducir el contenido WAP en un formato compacto. Este permite una mejor utilización de la relación subyacente debido a su reducido tamaño.

3. El agente de usuario, necesario para la administración de perfiles

4. El almacenamiento en caché de proxy, que puede mejorar el rendimiento percibido y la utilización de la red mediante el mantenimiento de una memoria caché de los recursos de acceso frecuente.

Fecha recepción del manuscrito: Mayo 17, 2011
Fecha aceptación del manuscrito: Agosto 10, 2011

Camilo A. Barbosa, Directv Colombia; Yesid A. Tibaquira, Avianca; Gerardo Castang Montiel, Facultad Tecnológica, Universidad Distrital Francisco José de Caldas.

Esta investigación fue financiada por: Universidad Distrital Francisco José de Caldas.

Correspondencia en relación al artículo debe ser enviada a: Yesid A. Tibaquira. Email: yatibaquirac@correo.udistrital.edu.co

5. El proxy WAP, que permite que el contenido y las aplicaciones sean almacenadas en servidores Web, y que sean desarrolladas utilizando tecnologías el nivel de la Web (OMA, 2001).

Arquitectura WAP. El protocolo WAP en su versión 2.0 fue optimizado con respecto a la versión 1.0 para redes con un ancho de banda bajo para el dispositivo móvil. Esto con el fin de mejorar la latencia en los dispositivos. Para ello se incorporaron las siguientes características.

1. Capa de Aplicación (WAE) (Figura 2), que prevé la interacción entre el protocolo WAP y aplicaciones Web para dispositivos inalámbricos. Estas aplicaciones Web son desarrolladas para el micronavegador WAP que posee las siguientes funcionalidades:

- Un lenguaje denominado WML6 similar al HTML, pero optimizado para su uso en terminales móviles.
- Un lenguaje denominado WMLScript. similar al JavaScript (esto es, un lenguaje para su uso en forma de Script).
- Un conjunto de formatos de contenido, que son un conjunto de formatos de datos bien definidos entre los que se encuentran imágenes, entradas en la agenda de teléfonos e información de calendario.

2. Capa de sesión. El protocolo WSP (*WirelessSessionProtocol*) proporciona la transferencia de hipertexto HTTP en su versión 1.1, incorporando nuevas características tales como sesiones de larga duración y un período de suspender / reanudar. Este protocolo proporciona la capa de aplicación de nivel superior del protocolo WAP con una interfaz consistente para dos servicios de sesión. El primero es un servicio de conexión, de modo que opera por encima del protocolo de la capa de transacción. El segundo es un servicio de conexión, que funciona por encima de un servicio de transporte de datagramas seguro o no seguro.

3. Capa de transacción. El protocolo WTP (*WirelessTransactionProtocol*) se ha definido como un protocolo orientado a transacciones de poco peso, que se ha adecuado para aplicaciones de estaciones móviles y funciona de manera eficiente a través de redes inalámbricas por medio de datagramas. Los beneficios de usar el protocolo WTP incluyen:

- Mayor fiabilidad en los servicios de datagrama, ya que el protocolo WTP alivia la capa superior de las retransmisiones y reconocimientos que son necesarios cuando los servicios de datagrama se utilizan.
- Mejora de la eficiencia en los servicios orientados a conexión. Debido a que no tiene ninguna conexión explícita para crear o derribar las fases.
- Se tiene la ventaja de utilizar un protocolo orientado a mensajes, diseñados para servicios orientados a transacciones.

4. Capa de Seguridad. La capa WTLS (*WirelessTransportLayer Security*) está diseñada para proporcionar privacidad, integridad de datos y la autenticación entre dos aplicaciones que se comunican. Proporciona la

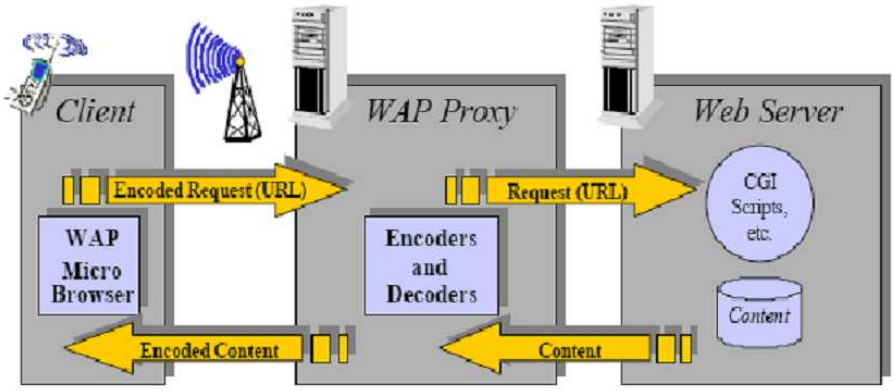

Figura 1. El protocolo WAP (OMA, 2001).

capa de nivel superior del protocolo WAP, con una interfaz de servicio de transporte seguro que conserva la interfaz de servicio de transporte por debajo de ella. Además, WTLS proporciona una interfaz para la gestión (por ejemplo, la creación y terminación) de las conexiones seguras. Así mismo, proporciona una funcionalidad similar a TLS 1.0 e incorpora funciones adicionales como el soporte de datagramas, y la optimización del intercambio dinámico de claves.

5. Capa de Transporte. El protocolo WDP (*WirelessDatagramProtocol*) ofrece un servicio equivalente al de la capa de transporte de datagrama de Internet UDP. Ofrece a las capas superiores un servicio transparente a la forma como el protocolo WAP transmite los datos en una red móvil.

Criptografía de Curva Elíptica (CCE)

Una curva elíptica está definida por la ecuación 1:

$$y^2 = x^3 + Ax + B \tag{1}$$

Donde A y B son constantes. Estas constantes deben cumplir con la siguiente desigualdad (ecuación 2):

$$4A^3 + 27B^2 \neq 0 \tag{2}$$

La ecuación de la curva es una versión corta de la ecuación generalizada de Weierstrass. Esta ecuación, puede ser estudiada para varias estructuras algebraicas, pero para las curvas elípticas solo se considerará en un campo finito (Huguet, Rifá, y Tena, 2009). La Figura 3 muestra dos opciones de cómo sería la gráfica de una curva elíptica.

Suma de puntos en curvas elípticas. La suma de dos puntos en una curva elíptica, P y Q, gráficamente se puede distinguir, trazando una línea que pase por los dos puntos. Por lo general, esta línea intercepta la curva en un punto. Este punto es denominado como $–R$, de forma que se puede hallar el punto opuesto a este tercer punto. Esto significa que si $–R = (x, y)$, el punto opuesto es $R = (x, -y)$. Se puede dar el caso de que se realice la suma del mismo punto y que el resultado de esta suma sea otro punto de la curva, o que no se intercepte la línea trazada sobre un punto de la curva, como resultado la suma de un punto más el punto en el infinito, que genera el punto original. Para ilustrar los diferentes casos de sumas de puntos en una curva elíptica, se presenta la Figura 4.

A continuación se presentan las fórmulas de los diferentes casos de la suma de puntos en una curva elíptica (ecuaciones 3, 4 y 5). El punto en el infinito es definido con la letra O:

$$P + O = P \tag{3}$$

Si $Q = -P$ entonces:

$$P + Q = O \tag{4}$$

Si $P = Q$ entonces:

$$P + Q = -R \tag{5}$$

También se puede decir que si se quiere doblar el punto P (ecuaciones 6 y 7):

$$P + (-P) = O \tag{6}$$

$$P + P = 2P = R \tag{7}$$

Multiplicación de un punto de una curva elíptica por un escalar. Teniendo como base la suma de puntos en una curva elíptica, otra operación que se puede realizar con los puntos es la multiplicación de estos puntos por un escalar k. Por ejemplo, si se supone que k tiene un valor de 17, para calcular K_p, se puede realizar un doblado de puntos de la siguiente forma (ecuación 8):

$$\begin{gathered} P \\ 2P = P + P \\ 4P = 2P + 2P \\ 8P = 4P + 4P \\ 17P = 8P + 8P + P \end{gathered} \tag{8}$$

De esta forma, se obtiene el punto resultante de k multiplicado por el punto P. Ello puede generar gran cantidad

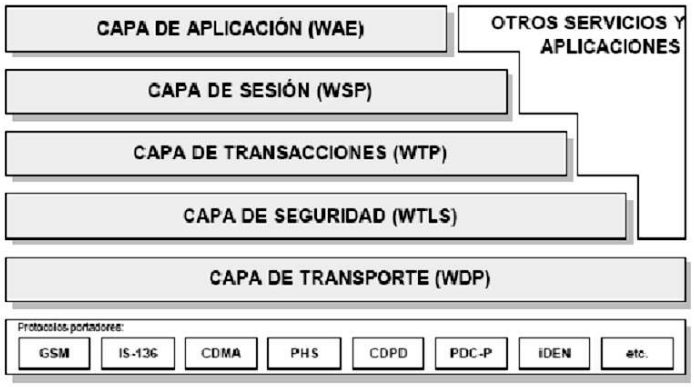

Figura 2. Arquitectura WAP 2.0 (OMA, 2001).

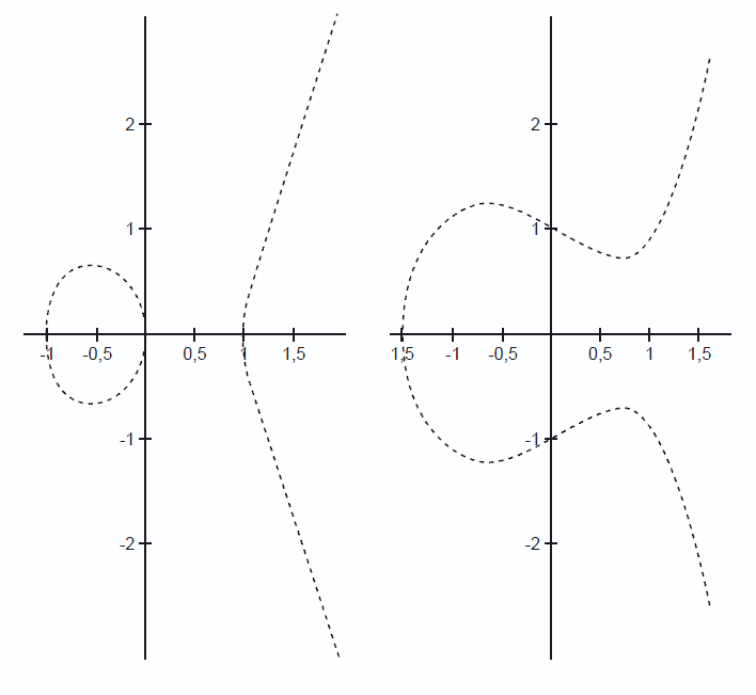

Figura 3. Diagrama de dos curvas elípticas (Lerch, 2007).

de puntos, con coordenadas de grandes tamaños. Sin embargo, para limitar este cálculo de puntos se suele trabajar con campos finitos, en especial cuando se implementa la teoría de curva elíptica en la criptografía.

Criptosistema de curva elíptica

El criptosistema de curva elíptica en su asimetría basa su seguridad en lo difícil que resulta calcular logaritmos discretos en curvas elípticas. El problema del logaritmo discreto para curvas elípticas (conocido como ECDLP) es la base de los criptosistemas de curva elíptica. Este problema establece que teniendo dos puntos dentro de un campo finito, que pertenecen a una curva elíptica, se necesita encontrar un valor x que cumpla con la siguiente relación (ecuación 9):

$$Q = xP \tag{9}$$

Esta operación es computacionalmente fácil, sin embargo, obtener x a partir de P y Q es un problema difícil, incluso para un ordenador. De hecho si se utilizan valores de k lo suficientemente grandes, la tarea se vuelve computacionalmente imposible. Al menos con los algoritmos y máquinas actuales (Lerch, 2007).

La forma cómo funciona el criptosistema de curva elíptica se explica a continuación de forma general:

1. Tanto el emisor como el receptor acuerdan el uso de una curva E sobre un campo finito F_n de tal forma que sea complejo resolver el problema del logaritmo discreto en $E(F_n)$. Además de la definición de la curva se establece un punto P

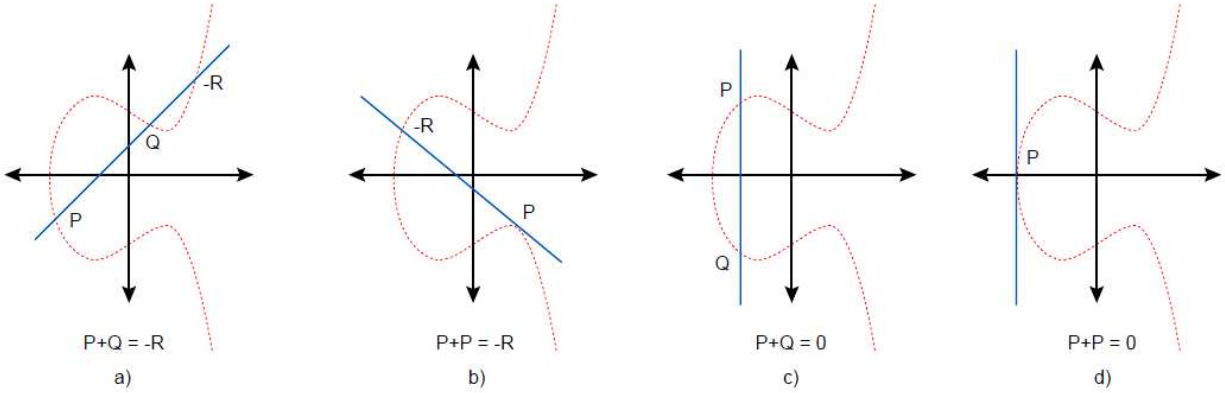

Figura 4. Suma de puntos en una curva elíptica (Lerch, 2007).

que pertenezca a la curva elíptica de modo que su orden sea un número primo demasiado grande.

2. El emisor escoge un número entero secreto a y calcula el punto P_a con la operación $P_a = aP$, para luego enviar al receptor el punto calculado.

3. El receptor de la misma manera genera un entero secreto b y calcula el punto P_b con $P_b = bP$, para luego enviarlo al emisor.

4. El usuario emisor calcula $aP_b = abP$.

5. El usuario receptor calcula $bP_a = baP$.

6. Ahora tanto el emisor como el receptor pueden establecer una comunicación utilizando una clave de cifrado que solo ellos conocen extraiéndola de abP.

API curva elíptica

La Legión del Castillo que Rebota (*The Legion of the Bouncy Castle*), es un proyecto de software libre que reúne un conjunto de librerías en el lenguaje Java y C#, para el desarrollo de trabajos que incluyan criptografía en estos dos lenguajes. Esta API (Interfaz de Programación de Aplicaciones - *Application Programming Interface*) funciona con todos los entornos de Java, desde el JDK 1.4 hasta la versión actual, JDK 1.6, incluyendo librerías que pueden trabajar con el framework de Java para el desarrollo de aplicaciones móviles J2ME. Algunas características de esta API son (Maiorano, 2009):

- API Criptográfica liviana (pretende mantener el soporte para las máquinas virtuales de java en J2ME).
- Proveedor para la Java CryptographyExtension (JCE) y la Java CryptographyArchitecture.
- Una librería para la lectura de objetos codificados en ANSI.1 (Notación Sintáctica Abstracta 1. Forma para representar datos independientemente del equipo).
- API de cliente liviana TLS *liviana*.
- Una versión firmada de un archivo jar para ser utilizado con la JDK 1.4 a 1.6 y la JCE de Sun.
- Implementación de la JCE 1.2.1.
- Soporte para algoritmos de curva elíptica como Diffie-Hellman y ElGammal.

Resultados

Diseño de la aplicación

La aplicación móvil se divide en varios módulos. El módulo inicial de autenticación valida el acceso de un usuario a las diferentes opciones de la aplicación. Dentro de la aplicación móvil se encuentran 3 módulos; el módulo de consulta de productos lista los productos activos con los que cuenta el usuario dentro de la entidad bancaria. Dentro de este mismo módulo el usuario puede consultar el saldo de algún producto específico, el cual se cuenta como otro módulo de la aplicación. En el módulo de consultar movimientos, el usuario de la lista de productos selecciona uno, al que requiera conocer el listado de movimientos dentro de un rango de fechas, no superior a una diferencia de 3 días entre estas.

Identificación de actores. Para este prototipo de la aplicación sólo se definió un actor, que es el usuario final, quién posee el dispositivo móvil y realizará las consultas (Figura 5).

Diagrama de clases. Para la implementación del cifrado y descifrado de los datos se utilizó el Esquema Integrado de Encriptado con Curva Elíptica (ECIES), haciendo uso de la librería criptográfica *BouncyCastle*, que será implementada tanto en la aplicación del servidor como en la aplicación móvil dentro de un paquete de clases que se ha denominado con el nombre *CurvaEliptica* (Figura 6).

Herramientas de desarrollo

Este prototipo se fundamenta en dos aplicaciones, una aplicación web, específicamente un servicio web, implementado en un servidor de aplicaciones que interactúa con un servidor de base de datos PostgreSQL 9.0 o posterior. La versión de Glassfish recomendada es 3 y del JDK (Java Development Kit) es la 1.6. Además, utiliza la API de Bouncycastle en su

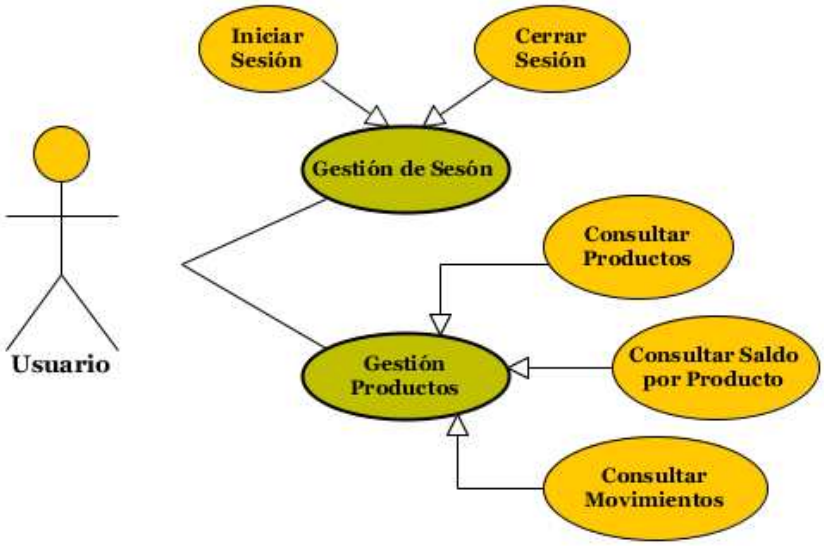

Figura 5. Diagrama de casos de uso - Diagrama general.

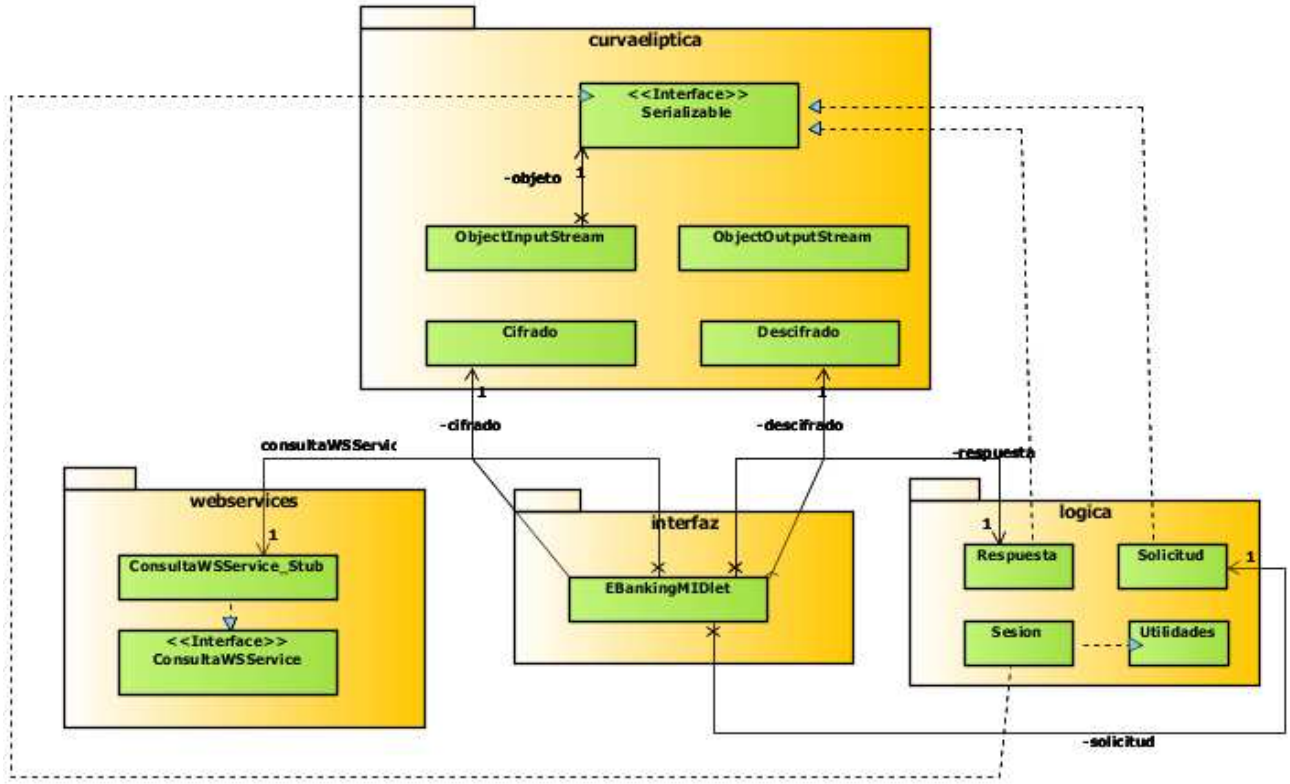

Figura 6. Diagrama de Clases - Cliente móvil.

versión 1.45, compatible con la versión de java para el esquema de encriptación.

Por otro lado, se encuentra la aplicación móvil, que establece el puente entre el usuario final y el Servicio Web. Esta es compatible con la especificación de Java para dispositivos móviles J2ME y también implementa una versión de la API de Bouncycastle para este tipo de dispositivos. Cualquier dispositivo compatible con aplicaciones java, puede tener la aplicación móvil.

API Bouncycastle

La API de Bouncycastle que se implementó dentro de la aplicación móvil y la aplicación Web, para las tareas de cifrado y descifrado bajo el esquema ECIES, consta de las siguientes clases (ver Tabla 1).

Análisis de tráfico. Inicialmente, en el equipo servidor se instaló el programa Wireshark (analizador de protocolos y puertos mantenido bajo la licencia General Public License o GPL), para analizar y visualizar la forma cómo la información entraba y salía en cada solicitud que se realizaba desde un dispositivo móvil. La imagen que se muestra a continuación (Figura 7) consta de dos paneles; el primer panel es llamado el *Panel de Detalles del Paquete*. En este panel se muestra el encabezado y el acuerdo de cada paquete junto con la información que es transmitida (ya sea de entrada o de salida). El segundo panel se llama *Panel de Bytes del Paquete*, el cual permite visualizar los Bytes de todo el paquete, o de alguna parte del paquete que se selecciona en el panel de detalles. En este panel se puede ver los Bytes en forma hexadecimal y en formato ASCII. Se resalta en las imágenes el contenido de la solicitud y de la respuesta del servidor, tanto en el panel de detalles como en el panel de Bytes.

Conclusiones

El protocolo WAP2 utiliza medidas de seguridad dentro de su arquitectura para el intercambio de información entre aplicaciones para dispositivos móviles y aplicaciones Web basadas en los criptosistemas simétricos o de una sola clave. Un criptosistema asimétrico o de clave pública es implementado en el momento de compartir la clave única para cifrar y descifrar la información entre ellas, demostrando con este hecho el alto nivel de seguridad que tiene la implementación de un criptosistema asimétrico a comparación de un criptosistema simétrico, sin demeritar el nivel de este.

Un criptosistema basado en curva elíptica utiliza claves de longitudes cortas para el cifrado y descifrado de los datos pero con un alto nivel de seguridad, lo cual hace que sea apropiado utilizarlo en entornos donde la información requiera un alto nivel de confidencialidad en el momento de ser transmitida. En entornos móviles es utilizado el algoritmo de Diffie-Hellman de curva elíptica para el intercambio de la clave, con la cual se va a cifrar y a descifrar la información antes y después de su transmisión. Sin embargo, el esquema ECIES (*Elliptic Curve Integrated Encryption Scheme*) implementado en la API de Bouncycastle, permite utilizar la curva elíptica tanto en entornos móviles como en aplicaciones Web, que para este caso es un servicio Web.

Referencias

Fuquene, H. (2008). M-commerce: el nuevo protagonista del comercio electrónico. *Vínculos*, *4*(1), 62-77.

Huguet, L., Rifá, J., y Tena, J. (2009). *Criptografía con curvas elípticas*. Universitat Oberta de Catalunya.

Lerch, D. (2007). Criptografía de curva elíptica: Ataque de rho de pollard. *Hakin9*.

Maiorano, A. (2009). *Criptografía técnicas de desarrollo para profesionales* (1ed. ed.). Alfaomega Grupo Editor.

OMA. (2001, Julio). *Wap 2.0 specifications. architecture.* On line. Descargado de `www.wapforum.com`

Tabla 1
Tabla de clases Bouncycastle

Clase	Descripción
ECDHBasicAgreement	Clase que obtiene un valor secreto a partir de una clave privada y otra pública.
AsymmetricCipherKeyPair	Una clase de mantenimiento para los pares de parámetros públicos/privados.
SHA1Digest	Implementación del algoritmo SHA-1.
IESEngine	Clase de apoyo para construir cifradores para el intercambio básico de mensajes en la aplicación. En resumen, procesa (cifra y descifra) bloques de bits de los mensajes.
KDF2BytesGenerator	Parámetros de la Key DerivationFunctions (KDF) que construye un generador de KDF2 bytes.
InvalidCipherTextException	Excepción que se lanza cuando se encuentra algo anormal en un mensaje.
Hmac	Clase para la implementación de los códigos de autenticación de los mensajes.
ECDomainParameters	Clase para la creación de parámetros delas claves de curva elíptica.
ECPublicKeyParameters	Parámetros de la clave pública de la curva elíptica.
ECPrivateKeyParameters	Parámetros de la clave privada de la curva elíptica.
IESParameters	Parámetros para usar cifrado en un modo streaming. Es la clave del acuerdo (Diffie-Hellman) usada como base para la encriptación.
ECCurve	Clase base para implementar la curva elíptica.
ECPoint	Clase base para implementar los puntos de la curva elíptica.
Hex	Clase que codifica los datos de entrada, produciendo una matriz debytes hexadecimal codificados.
Base64	Clase que codifica los datos de entrada, produciendo una matriz de bytes codificados en base 64.

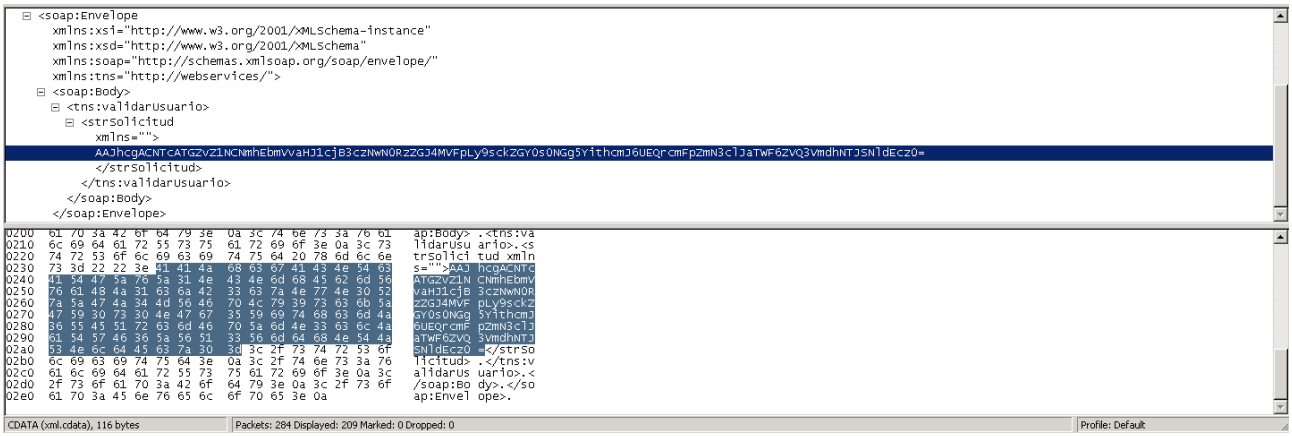

Figura 7. Traza validar usuario - Solicitud.