VÍNCULOS
JULIO - DICIEMBRE DE 2011
VOLUMEN 8 ■ NÚMERO 2

Implementación del criptosistema de curva elíptica en entornos móviles

Implementation of cryptosystem of elliptic curve in mobile environments

Seguridad en la transmisión de información crítica entre aplicaciones móviles y aplicaciones Web

Gerardo Castang Montiel*
Camilo Andrés Barbosa Hernández**
Yesid Alberto Tibaquira Cortes***

Fecha de recepción: Julio 11 de 2011
Fecha de aceptación: Octubre 15 de 2011

Resumen

Este artículo presenta el trabajo realizado por los autores para implementar el criptosistema de curva elíptica para las transacciones que se presentan entre un dispositivo móvil y una aplicación Web. La propuesta de este artículo, es explicar el esquema ECIES que ofrece Bouncy Castle para incorporar a nivel de aplicación el algoritmo de cifrado de curva elíptica. En el desarrollo de la aplicación móvil se utiliza J2ME y la aplicación Web, que en este caso es un servicio Web, se construye bajo la plataforma de Java JEE. Por último se presentan imágenes del análisis de tráfico en el servidor donde se encuentra la aplicación Web, con información cifrada y sin cifrar.

Palabras clave: Curva Elíptica, criptografía, clave y móviles.

* Ingeniero electrónico, magister en Teleinformática de la Universidad Distrital Francisco José de Caldas, miembro del grupo de investigación en Telemática ORION, docente de planta adscrito a la Facultad Tecnológica. Correo electrónico: gacastangm@udistrital.edu.co.

** Tecnólogo en Sistematización de Datos, ingeniero en Telemática de la Universidad Distrital Francisco José de Caldas. Correo electrónico: camiloandb@gmail.com.

*** Técnico en Análisis y Diseño de Sistemas, tecnólogo en Sistematización de Datos, ingeniero en Telemática de la Universidad Distrital Francisco José de Caldas. Correo electrónico: yatibaquirac@correo.udistrital.edu.co.

Abstract

This paper presents the work undertaken by the authors to implement the elliptic curve cryptosystem for transactions that occur between a mobile device and a Web application. The goal of this article is to explain the ECIES scheme that offers Bouncy Castle to incorporate application-level algorithm using elliptic curve. For the mobile application development, using J2ME and Web application, which in this case is a Web service is built on JEE Java platform. Lastly the pictures of the traffic analysis in the server, where the Web application is inside, are presented with encrypted information and without encrypt.

Key words: Elliptic curve, cryptography, key and mobiles

Introducción

El extenso uso de los dispositivos móviles ha provocado que las aplicaciones que hasta hace unos años estuvieran dedicadas para computadores de escritorio o portátiles, tales como juegos, paquetes de ofimática, agendas, calculadoras, etc., puedan ahora implementarse con limitados recursos en estos dispositivos. Las aplicaciones orientadas a entornos Web, no son ajenas a esta nueva plataforma, es por eso que ya se pueden encontrar micro-navegadores, o aplicaciones que se comunican con servidores ubicados en otra posición geográfica, utilizando Internet como entorno de comunicación.

Algunas de estas aplicaciones que utiliza Internet (en un entorno no seguro) para la transmisión de datos, requieren confidencialidad en la información que es transportada. Para cumplir este objetivo, el protocolo WAP (Wireless Applications Protocol – Protocolo de Aplicaciones Inalámbricas), basado en estándares de Internet, implementa algoritmos criptográficos. Dentro de las funciones que exige la seguridad de la información, donde hace parte la criptografía (Arte de ocultar la información), se encuentran la firma digital, el intercambio de claves de forma segura y el cifrado y descifrado de la información.

Para los procesos de firma digital y de intercambio de claves, WAP hace uso de criptosistemas asimétricos. Estos criptosistemas constan de dos claves, una de carácter público, que se utiliza para cifrar la información y otra de carácter privado, con la que se descifra. Los criptosistemas de este esquema utilizados en WAP son RSA (Rivest-Shamir-Adleman) y el criptosistema de curva elíptica, específicamente el acuerdo de intercambio de claves de Diffie-Hellman bajo curva elíptica.

En el proceso de cifrado y descifrado de la información, WAP utiliza criptosistemas simétricos, que a diferencia de los asimétricos, sólo utilizan una clave, tanto para cifrar como para descifrar.AES (Advanced Encryption Standard) es el esquema criptográfico que WAP implementa para estos procesos.

El principal riesgo de los criptosistemas simétricos, es que la clave sea descubierta por terceros y poder así tener acceso a información confidencial, o, alterar esta información con fines malintencionados. Es por esta razón que en el presente trabajo se presenta la alternativa de utilizar el criptosistema de curva elíptica para el cifrado y descifrado de información transmitida entre un Servicio Web y una aplicación móvil.

Inicialmente, se presentan los conceptos necesarios para comprender el funcionamiento del Esquema Integrado de Encriptación de Curva Elíptica (ECIES – Elliptic Curve Integrated Encryption Scheme) en campos finitos primarios, incluyendo la teoría de curvas elípticas. Posteriormente se explican los procesos de cifrado y descifrado de ECIES, y por último, los resultados del análisis de tráfico en el servidor de aplicaciones Web de cómo es la entrada y la salida de la información cifrada con ECIES.

Campos Finitos

Toda la teoría de la criptografía moderna se basa en operaciones, funciones y teorías matemáticas, es por eso que para comprender el criptosistema de curva elíptica es necesario definir y entender algunos conceptos base. El primero de ellos es el concepto de *campo*.

Sin embargo, antes de definir que es un campo, se debe conocer que es un grupo abeliano y que es un anillo. Un grupo abeliano es un conjunto de elementos sobre los cuales se pueden realizar operaciones para obtener un tercer elemento. A estas operaciones se les denomina *operaciones binarias* y se identifican con el símbolo ⊕. Los elementos de estos grupos deben cumplir la ley conmutativa en las operaciones:

$$1.\quad a \oplus b = b \oplus a$$

Un anillo es un grupo abeliano de mínimo dos elementos, con dos operaciones binarias, la adición y la multiplicación [12], y se denomina conmutativo si cumple la ley conmutativa de la multiplicación para estos dos elementos:

$$2.\quad a \times b = b \times a$$

Otro nombre con el que se conocen los anillos conmutativos es de *campo* o en otros textos como *cuerpo*. Un campo es finito cuando contiene un número fijo de elementos que deben cumplir con dos operaciones (suma y multiplicación) [3]. En el esquema de encriptación de curva elíptica se puede trabajar con dos tipos de campos finitos:

- F_p: En este campo finito "p" identifica el número de elementos y es un número primo impar. Se le denomina campo finito primo o primario.
- F_2m):: El número de elementos de este campo es 2^m y "m" es un número entero mayor o igual que 1. Se le denomina campo finito de característica 2 o binario.

Curvas Elípticas

Una curva elíptica E definida sobre un campo F_p(Para este caso un campo finito primario) es el conjunto de soluciones (x,y) donde $x,y \in F_p$, de la ecuación de Weierstrass [11] y su módulo p, definen:

$$3.\quad y^2 \equiv x^3 + ax + b \ (mod\ p)$$

Además de este conjunto de puntos o soluciones existe un punto extra o punto en el infinito (∞). La anterior ecuación es llamada la definición de la curva elíptica sobre un campo F_p. Las variables a y b también pertenecen al campo F_p y deben satisfacer la ecuación:

$$4.\quad 4a^3 + 27b^2 \neq 0 \ (mod\ p)$$

Las reglas de adición de puntos de una curva elíptica sobre un campo finito primo son:

- La adición del punto en el infinito más él mismo, tiene como resultado es el mismo punto en el infinito.
- La suma de un punto cualquiera más el punto en el infinito, tiene como resultado el mismo punto.
- La suma de dos puntos donde la coordenada "x" de ambos es igual y la coordenada "y" difiere o tiene el valor de cero (0), tiene como resultado el punto en el infinito.

- La suma de dos puntos con diferente coordenada "*x*". Este resultado da un punto con coordenadas diferentes a las coordenadas de los puntos sumados. Los puntos originales pertenecen a la curva.
- La suma de un punto por sí mismo, teniendo en cuenta que la coordenada "*y*" sea diferente de cero (0), tiene como resultado otro punto.

Los resultados de estas reglas de adición forman también un grupo. Este grupo es abeliano ya que cumple que $P1 + P2 = P2 + P1$, y estos puntos pertenecen a la curva.

Los esquemas de criptografía basados en curva elíptica depende de la multiplicación de un escalar por los puntos de la curva. Dado un entero y un punto de la curva, el proceso de multiplicación es la adición de ese punto tantas veces como lo indique el escalar. Esto es kP.

Parámetros de dominio

Los parámetros de dominio establecen el contexto para realizar las tareas de cifrado y descifrado utilizando basándose en la teoría de curvas elípticas. Estos parámetros en curvas elípticas sobre F_p se componen de un séxtuple:

$$T=(p,a,b,G,n,h) \quad (5)$$

Este séxtuple contiene:

- p: Indica el número de elementos del campo finito primario.
- a y b: Elementos que pertenecen al campo finito primario y hacen parte de la ecuación 3.
- G: Un punto base con coordenadas (G_x, G_y).
- n: Es el orden de G. Al multiplicar el punto por este número da el punto en el infinito.
- h: Es el número de puntos del campo finito dividido por el orden del punto G.

Para la generación de estos parámetros de dominio se debe establecer un nivel de seguridad. Este nivel de seguridad puede ser 56, 40, 80, 96, 112, 128, 192 o 256. A partir de valor se deduce p teniendo en cuenta las siguientes condiciones:

- Si el valor del nivel de seguridad es diferente de 256 entonces p se deduce de la ecuación:

$$[log2\ p]=2t \quad (6)$$

- Si el valor del nivel de seguridad es igual a 256, el valor de p resulta de la ecuación:

$$[log_2\ p]=521 \quad (7)$$

La variable t hace referencia al nivel de seguridad en la ecuación 6.

Par de claves en curvas elípticas

El criptosistema de curva elíptica es de carácter asimétrico. Esto significa que para las operaciones de cifrado y descifrado necesita generar dos claves, una pública y una privada. La clave privada es un entero d que se encuentra en el intervalo $[1,n-1]$, teniendo en cuenta que n es el grado del punto G; éste es seleccionado de forma pseudo-aleatorio. Estos valores, n y G, son lo que hacen parte de los parámetros de dominio de la curva. La clave pública es un punto Q que resulta del producto de un escalar por un punto de la curva, este escalar es la clave privada y el punto es G, en otras palabras:

$$Q=dG \quad (8)$$

El nuevo punto Q o ahora la clave pública debe cumplir las siguientes características:

- La clave pública debe ser un punto diferente al punto en el infinito.
- Las coordenadas del punto G deben ser número enteros que se encuentren en el rango $[1, p-1]$ y que cumpla la siguiente ecuación:

$$Qy^2 \equiv Qx^3 + aQx + b \ (mod\ p) \quad (9)$$

- El punto Q multiplicado por debe tener como resultado el punto en el infinito.

Con las dos primeras condiciones aprueba que Q sea un punto de la curva y con la tercera, que sea un múltiplo escalar del punto G. De esta forma es generado el par de claves (d,Q), que se encuentra asociado a los parámetros de dominio de curva elíptica.

ECIES

ECIES (Elliptic Curve Integrated Encryption Scheme – Esquema Integrado de Encriptación de Curva Elíptica) fue propuesto por Mihir Bellare y Philip Rogaway como una versión mejorada del esquema de encriptación El-Gamal. Este esquema fue estandarizado por la norma IEEE 1363a y por la ISO/IEC 15946-3 [8]. Como lo indica su nombre ECIES es un esquema de encriptación integrado que tiene los siguientes componentes funcionales:

- Función de acuerdo de claves.
- Función hash.
- Función de descubrimiento de claves.
- Código de autenticación de mensaje.
- Algoritmo de encriptación simétrico.

Función de acuerdo de claves

En ECIES, la función de acuerdo de claves se utiliza para generar un secreto compartido por las dos partes que harán parte de la comunicación. Hay dos variaciones del acuerdo de claves de Diffie-Hellman basados en curva elíptica, una es la primitiva de la curva elíptica de Diffie-Hellman y la otra es la primitiva por cofactores de Diffie-Hellman. Sin embargo, la idea básica de ambas variaciones es la misma, generar un valor secreto compartido desde la clave privada de B y la clave pública de B, de tal forma que si ambas partes, A y B, ejecutan este proceso, con las claves correspondientes como entrada recuperan el mismo secreto compartido [3].

El mecanismo de *acuerdo de claves* es el siguiente: El receptor y el emisor acuerdan unos parámetros de dominio de una curva elíptica. Tanto el emisor y el receptor generan, cada uno por separado un número entero d, el cual es la clave secreta, y a partir de este, ejecutan la operación de multiplicación de este valor por el punto G que hace parte de los parámetros de dominio, obteniendo como resultado un nuevo punto Q. Este nuevo punto es la clave pública del receptor y del emisor, eso significa que hay un punto Q para el emisor, que es compartido con el receptor, y otro punto Q para el receptor que es compartido por el emisor. El emisor con el punto Q receptor calcula un nuevo punto R, al multiplicarlo por su clave secreta d. Este mismo procedimiento lo realiza el receptor, multiplica la clave pública del emisor por su clave secreta, y también genera un punto R. Este punto R es el mismo punto para tanto para el emisor como para el receptor, cumpliendo con:

$$10. \quad R = d_e \times Q_r = d_r \times Q_e$$

d_e: Clave secreta del emisor.

Q_r: Clave publica del receptor.

d_r: Clave secreta del receptor.

Q_e: Clave publica del emisor.

De este punto "R" el emisor y el receptor acuerdan que cantidad de bits son definidos para convertirlos en su clave secreta, en el caso de utilizar un criptosistema simétrico para el intercambio de información entre el emisor y el receptor.

Función hash

Una función hash o función de una sola vía permite computar su resultado de manera relativamente rápida, pero en cambio, calcular la entrada de la función a partir del resultado, es un proceso prácticamente inviable [10]. Estas funciones reciben como parámetro un mensaje de longitud variable y retornan el valor hash con una longitud fija

del mensaje. Esto quiere decir que independientemente de la longitud del mensaje, la longitud del valor hash siempre es el mismo. MD5 y SHA son las funciones hash más implementadas en criptografía.

ECIES trabaja con la función SHA-1. Esta se utiliza para la autenticación de mensajes y para el descubrimiento de claves con las funciones de derivación de claves KDF. Esta función mapea octetos (cadenas de 8 bits) de tamaños menores a 2^{61} a valores hash con una longitud de 160 bits o 20 octetos. El funcionamiento es sencillo; si se tiene un mensaje *M*, primero se verifica que el tamaño de *M*, sea menor que el tamaño máximo permitido, luego se convierte *M* a una cadena de bits. A esta cadena de bits se le ejecuta la función hash, obteniendo una cadena de bits de longitud fija, la cual por último se convierte en una cadena de octetos, encontrando de esta forma el valor hash del mensaje *M*.

Función de descubrimiento de claves

El acrónimo de estas funciones es KDF (Key Derivation Functions). El objetivo de estas funciones es obtener una o más claves a partir de un valor secreto compartido utilizando una función hash. La función KDF que utiliza ECIES es ANSI-X9.63-KDF.

El mecanismo de esta función es primero seleccionar o identificar la función hash con la que se va a trabajar, que para este esquema es SHA-1. Luego es necesario definir un secreto compartido y el tamaño de las claves que generara la función KDF. Para calcular las claves se utiliza la función hash para recuperar el valor hash de la combinación entre el secreto compartido, el contador de las veces que se realiza este proceso y alguna información extra. Este proceso se realiza desde una vez hasta *n* número de veces, esta cantidad de iteraciones es determinada por la división entre el tamaño de las claves generadas y el tamaño del valor hash.

En el esquema de curva elíptica por lo general el tamaño de las claves, resultantes de aplicar una función de derivación de clave es el doble del tamaño del valor hash generado, lo que significa que el resultado serán dos claves.

Código de autenticación del mensaje

El código de autenticación del mensaje (MAC – Message Authentication Code) es la información usada para comprobar la autenticidad de un mensaje transmitido entre dos partes. Para la generación de este código se emplean las funciones hash. Para ECIES hay dos opciones de esquemas MAC. El primero, HMAC–SHA-1–160 con 20 octetos o claves de 160 bits; y, HMAC–SHA-1–80 con 10 octetos o claves de 80 bits.

Una función MAC es una función hash que necesita una clave secreta para generar el nuevo valor MAC, independientemente del esquema seleccionado, el proceso de autenticación es el siguiente:

- Definir el esquema MAC, la longitud de la clave secreta y la longitud de las etiquetas resultantes de aplicar este esquema.
- Compartir de forma segura la clave secreta entre las dos partes que conocerán el mensaje.
- El emisor aplica el esquema MAC al mensaje, junto con la clave secreta compartida.
- El resultado de la implementación del esquema MAC, es la etiqueta del mensaje, que se envía junto con el mensaje.
- El receptor realiza la operación de verificación de marcado del mensaje, primero aplicando al mensaje el esquema MAC, junto con la clave previamente compartida.
- El resultado de este proceso se compara con la etiquete enviada con el mensaje, y en el caso de ser la misma, se autentica el

mensaje, de lo contrario significa que el mensaje de llegada no es el mismo mensaje original.

Algoritmo de encriptación simétrico

La finalidad de este algoritmo es mantener la confidencialidad de la transmisión de la información entre dos partes. ECIES trabaja con uno de los dos siguientes algoritmos:

- Modo CBC de 3-key TDES: Se implementa para compartir claves de 24 octetos o 192 bits. De esta clave, pueden definirse 3 claves, cada una de 64 bits. Los bits de cada clave son reemplazados con bits de paridad.
- Esquema de encriptación XOR: Ejecuta la operación XOR a la clave y el mensaje en el momento de cifrar del lado del emisor, y, en recepción, se ejecuta de nuevo esta operación sobre el mensaje recibido y la clave para la recuperación del texto original.

Cuando dos partes utilizan un esquema simétrico para transmitir información cifrada, primero, definen que algoritmo de encriptación simétrica van a trabajar, luego comparten la clave que se utilizara para cifrar y descifrar la información. El proceso que sique es que la parte que desea enviar, cifra la información con la clave compartida, y el resultado lo transmite a la otra parte. En el momento de recibir el mensaje cifrado, éste se descifra con la clave, para así recuperar el mensaje original.

Operación de cifrado

Antes de iniciar la operación de cifrado de la información en ECIES, es necesario que las dos partes definan los parámetros de dominio de la curva elíptica, la función KDF, la función MAC, el esquema de encriptación simétrico, la función de acuerdo de claves; además, las dos partes deben compartir su clave pública. Los pasos para la operación de cifrado se describen a continuación, teniendo en cuenta que el mensaje a cifrar es M:

- El emisor debe generar un par de claves, d como clave secreta y Q como clave pública, teniendo como base la información de los parámetros de dominio de la curva. La clave pública hará parte del mensaje que es transmitido.
- Utilizar el acuerdo de claves de Diffie-Hellman para descubrir un elemento que pertenezca a F_p, que será el secreto compartido, a partir de la clave secreta del emisor y de la clave pública del receptor.
- Usar la función KDF para generar una clave de longitud fija. Esta longitud es el tamaño de la clave de la encriptación simétrico, más, el tamaño de la clave MAC de autenticación, teniendo como base el elemento o secreto compartido de Diffie-Hellman.
- Extraer del resultado de la función KDF una parte como clave de encriptación simétrica y la otra parte como clave para generar el código de autenticación del mensaje.
- Utilizar el esquema de cifrado simétrico y cifrar el mensaje.
- Utilizar la operación de marcado de la función MAC con la clave MAC generada, para marcar el mensaje cifrado.
- Unificar la clave pública del emisor, con el mensaje cifrado y el resultado de la función MAC. El resultado de esta unión es el mensaje transmitido.

Operación de descifrado

Los pasos para descifrar el mensaje son los siguientes:

- En recepción el mensaje recibido es primero divido en cada una de las partes que fue estructurado, se separa la clave pública Q, el mensaje cifrado y la etiqueta del mensaje.
- Comprobar que la clave pública Q es un punto de la curva elíptica.
- Utilizar el acuerdo de claves de Diffie-Hellman para descubrir un elemento que pertenezca a F_p, que es el secreto

Tabla 1. Tabla de clases Bouncy Castle

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, descifra) bloques de bits de los mensajes.
KDF2BytesGenerator	Parámetros de la Key Derivation Functions (KDF), 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 de las 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 codificar los datos de entrada produciendo una matriz de bytes codificados hexadecimal.
Base64	Clase que codificar los datos de entrada produciendo una matriz de bytes codificados en base 64.

compartido, a partir de la clave secreta del receptor y de la clave pública Q.

- Usar la función KDF para generar una clave de longitud fija a partir del secreto compartido.
- De la clave de longitud fija resultante de aplicar la función KDF, seleccionar la clave de cifrado simétrico y la clave para la autenticación del mensaje.
- Con la clave MAC, realizar la verificación de marcado del mensaje.
- Luego de comprobar la autenticidad del mensaje se descifra el mensaje con el esquema simétrico acordado.
- El resultado de ese proceso de descifrado es el mensaje original.

API Bouncy Castle

Bouncy Castle es una API (Application Programming Interface – Interfaz de Programación de Aplicaciones) que contiene distintas implementaciones de esquemas criptográficas desarrolladas en Java y en C#. El grupo de desarrolladores voluntarios que trabajan en esta API se hace llamar *The Legion of The Bouncy Castle* [9]. Esta API contiene dos paquetes criptográficos para Java, uno para ser implementado en entornos móviles y otro para el desarrollo sobre aplicaciones autó-

nomas o Web. Las clases que se trabajaron de Bouncy Castle para este proyecto se presentan en tabla 1.

Algunas características generales de esta API son:

- API ligera para Java y C#.
- Un proveedor para la Extensión Criptográfica de Java (JCE) y la Arquitectura Criptográfica de Java (JCA).
- Librería para la lectura y escritura de objetos ASN.1.
- Generadores de certificados X.509 versión 1 y 3, versión 2CRLs y archivos PKCS12.
- Generadores de atributos de certificados X.509.
- Soporte para SSL y TLS.

Aplicación

Para la implementación de ECIES se diseñó una aplicación móvil con J2ME (Plataforma de Java para el desarrollo de aplicaciones móviles) que simula operaciones bancarias como consulta de saldo, consulta de productos y consulta de movimientos. Esta aplicación accede a un servicio Web, elaborado bajo la especificación de Java, JEE5, para el desarrollo de aplicaciones de varias capas que trabajan sobre un servidor de aplicaciones, que para este caso es Glassfish.

Una de estas capas es llamada la capa de persistencia, que implementa JPA (Java Persistence API), que permite la gestión de datos relacionales en Java. La finalidad de esta API es mapear el modelo entidad-relación de la base de datos diseñada en PostgreSQL (Un gestor de bases de datos orientado a objetos de libre distribución), y utilizar JPQL (Java Persistence Query Language – Lenguaje de Consultas de la capa de Persistencia de Java) como lenguaje para realizar las operaciones de inserción, actualización y consulta sobre la base de datos.

Resultados

Con el fin de evidenciar que la información transmitida entre la aplicación móvil y el servicio Web, en el equipo que funciona como servidor de aplicaciones se instaló el programa Wireshark, un analizador de protocolos y puertos. Este programa analiza todo el tráfico que se presenta sobre una in-

Figura 1. Traza validar usuario – Solicitud

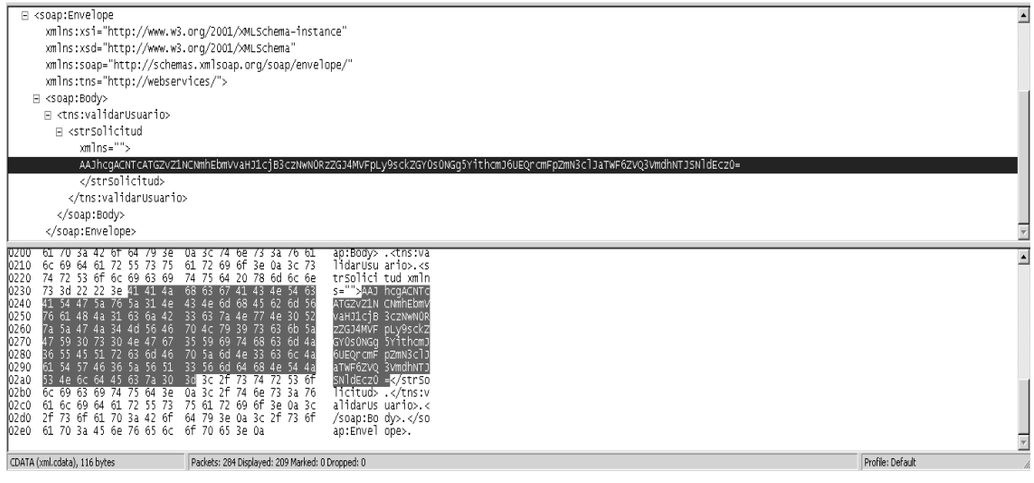
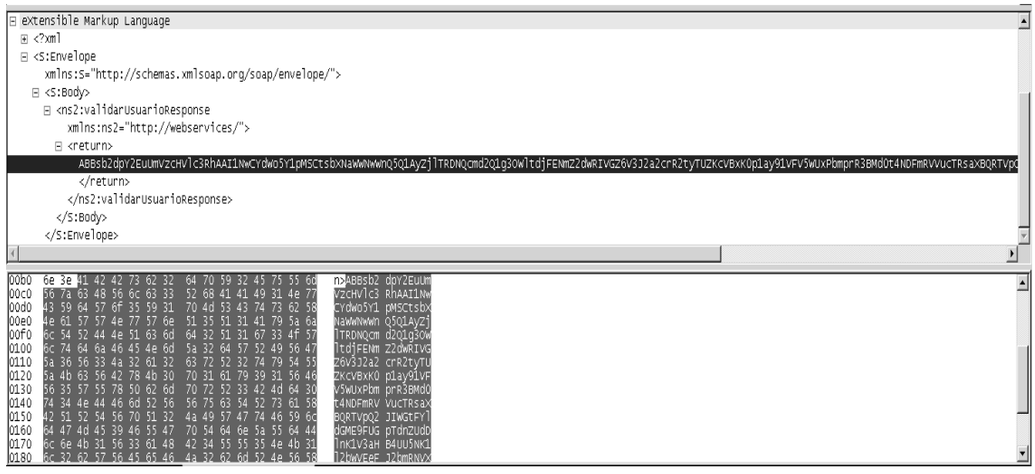


Figura 2. Traza validar usuario – Respuesta



terfaz de red. Las figuras 1 y 2, visualizan la información cifrada de la solicitud que realiza la aplicación móvil al servicio web y la respuesta de esta solicitud. Cada imagen se encuentra dividida en dos paneles, el primer panel muestra la información contenida en el paquete, como el encabezado y el cuerpo del paquete, el segundo panel discrimina en bytes el paquete, específicamente la porción que se selecciona en el primer panel. En este panel los bytes son mostrados en formato hexadecimal y ASCII. Se resalta en cada imagen la información cifrada con ECIES, las imágenes corresponden a la operación

Figura 3. Traza validar usuario – Solicitud – Sin Encriptación

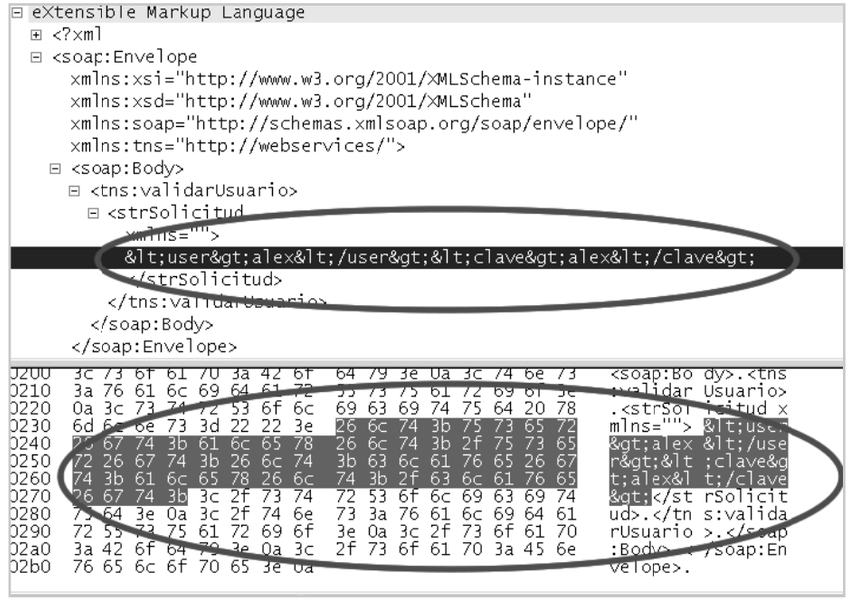
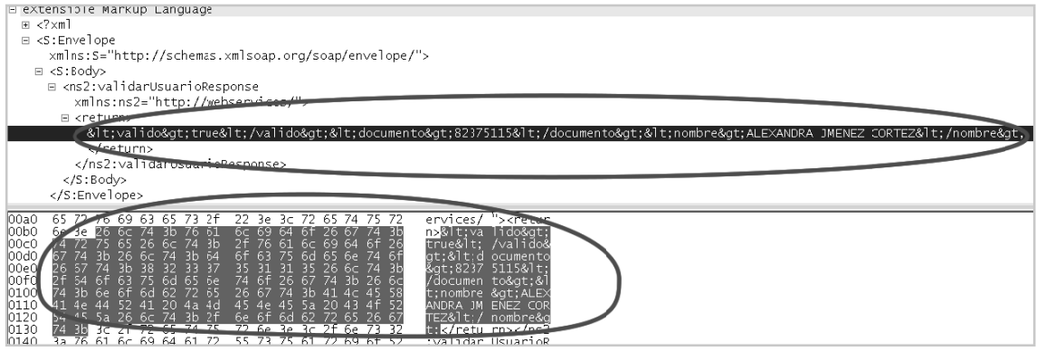


Figura 4. Traza validar usuario – Respuesta – Sin Encriptación



de validación de un usuario al inicio de la aplicación móvil.

Las figuras 3 y 4 visualizan como es el tráfico de red sin utilizar ECIES para el cifrado de la información. En la figura 3 se pueden identificar, en la parte resaltada, el nombre de usuario y la clave, para el proceso de validación de usuario, y en la figura 4, la respuesta del servicio Web, la cual comprende información que solo el usuario debería conocer. Por lo general esta es la forma de como transmiten la información los servicios Web sin implementar algún algoritmo criptográfico.

Conclusiones

- En este trabajo se presentó la opción de implementar un esquema de cifrado asimétrico basado en la teoría matemática de curvas elípticas para la transmisión de la información de forma confiable en un entorno inseguro, como es el caso de Internet, en aplicaciones móviles, donde el protocolo WAP usa esquemas simétricos para esta misma tarea.

- ECIES fue el esquema utilizado para este desarrollo, encontrado en la API Bouncy Castle, el cual permite incorporar el criptosistema de curvas elípticas en el diseño de aplicaciones para entornos móviles y en servicios Web.

- Por último, se demostró la diferencia que existe entre la transmisión de información entre dos partes sin utilizar ECIES, y como es esta misma transmisión haciendo uso de las clases que ofrece Bouncy Castle para ECIES, con el programa Wireshark para el análisis de tráfico en el servidor de aplicaciones.

Bibliografía

[1] Barco, C. y Isaza, G. (2006). Bases matemáticas y aplicaciones de la criptografía de curvas elípticas. Recuperado del sitio http://vector.ucaldas.edu.co/downloads/Vector1_4.pdf

[2] Castillo, C. Curvas Elípticas. Recuperado del sitio http://www.uv.es/ivorra/Libros/Elipticas.pdf

[3] Certicom Corp. (2000). Sec1: Elliptic Curve Cryptography. Recuperado del sitio Web http://www.secg.org/collateral/sec1_final.pdf.

[4] Gayoso, V., Sánchez, C., Espinosa, J. y Hernández, L. (2005). Estado del arte de las implementaciones Java en criptografía de curva elíptica. I Congreso Español de Informática (CEDI).

[5] Gayoso, V., Hernández, F., Hernández, L. y Sánchez, C. (2010). A comparison of the standardized versions of ECIES. Sixth

International Conference on Information Assurance and Security

[6] Gómes, R. (2009). Campos Finitos. Recuperado del sitio Web http://homepage.cem.itesm.mx/rogomez/SlidesCripto/CamposFinitos.pdf

[7] Goya, Denise. (2009). Criptografía de Curva Elíptica. Recuperado del sitio http://www.ime.usp.br/~dhgoya/ecc.pdf

[8] Hankerson, D., Menezes, A. y Vanstone, S. (2004). Guide to Elliptic Curve Cryptography. Springer-Verlag.

[9] http://www.bouncycastle.org/

[10] Maiorano, A. (2009). Criptografía. Técnicas de desarrollo para profesionales. Alfaomega Grupo Editor.

[11] Sklavos, N. y Zhang, X. (2007). Wireless Security and Cryptography. CRC Press.

[12] Wagstaff, S. (2003). Cryptanalysis of Number Theoretic Ciphers. Chapman & Hall/CRC, CRC Press.