UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

IMPLEMENTACIÓN DE LA CRIPTOGRAFÍA DE CURVAS ELÍPTICAS A UNA APLICACIÓN DE MENSAJERÍA INSTANTÁNEA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

PEDRO ARTURO TREJO SÁNCHEZ

DIRECTOR DE TESIS: M.C. MARÍA JAQUELINA LÓPEZ BARRIENTOS

2013

Agradecimientos

A la Universidad Nacional Autónoma de México que por medio de la Facultad de Ingeniería me brindó la oportunidad de cursar mi educación superior, obteniendo una esmerada formación académica y para la vida culminando con el presente trabajo.

Agradezco a mi querida directora de tesis, M.C. Ma. Jaquelina López Barrientos por compartir conmigo su amplia experiencia, conocimientos y amistad durante mi tiempo como su alumno y asesorado, por orientarme en algunas de las decisiones más importantes de mi vida y por ser un ejemplo a seguir como profesionista y persona.

Agradezco a mis profesores, Ing. Carlos Alberto Zmitiz y el Ing. Aldo Jiménez Arteaga por brindarme su apoyo en la realización de esta tesis.

Agradezco a todos mis profesores que con su cátedra nutrieron mi aprendizaje tanto en conocimientos profesionales como en experiencia.

Agradezco a mi mamá Patricia Sánchez Martínez y a mi hermana Itztli Graciela Trejo Sánchez, por brindarme siempre total apoyo y cariño, por estar presentes durante mi formación académica y compartir tanto mis tristezas y decepciones como mis triunfos y alegrías. Gracias por ayudarme a ser la persona que soy, pues sin ustedes no sería nada.

Agradezco a mis abuelos Pedro Sánchez Castillo (q.e.p.d) y a mi abuela Angela Martínez Negrete por haberme alentado en los momentos difíciles de mi vida, por su amor y confianza, por su ejemplo de trabajo, tenacidad y entrega que ha alimentado mi esencia.

Por último y no menos importante, agradezco a mis amigos y compañeros que estuvieron conmigo durante mi tiempo en la Facultad de Ingeniería y compartieron conmigo toda clase de experiencias y aventuras.

Índice

Introducción

Hoy en día las tecnologías de la información (TIC), han aportado una gran cantidad de soluciones a muchos problemas de la Humanidad en el ámbito del tratamiento y transmisión de la información; principalmente la informática, Internet y las telecomunicaciones, ya que han aumentado su eficiencia, reducido sus costos y por lo tanto se han convertido en parte del día a día de muchas personas alrededor del mundo.

El éxito que ha tenido el uso de las TIC en la sociedad se debe en gran medida al avance que ha tenido la computación en paralelo con ésta. A tal grado que en la actualidad ya prácticamente todo aparato electrónico tiene integrado algún tipo de computadora, es decir tienen algún microprocesador o micro controlador en su interior.

Las computadoras han cambiado el rumbo de la Humanidad radicalmente y hoy en día es muy común encontrarlas en todos los lugares a los que volteamos y más aún por la tendencia de hacer compacto y portátil cualquier objeto que pueda facilitarles la vida a las personas. Sin embargo, la mayoría de la gente usuaria de estas tecnologías de vanguardia carece del conocimiento técnico y científico necesario para conocer tanto los beneficios totales de la tecnología, como los riesgos que implica el manejo de las mismas.

No es obligación del usuario el conocer a detalle el funcionamiento científico de algún dispositivo tecnológico, sino el brindar solución a una necesidad o problema proveniente de cualquier persona. Pero esto no conlleva a que los usuarios deban permanecer ignorantes de los riesgos que implica hacer uso de la tecnología. Es necesario que se fomente una cultura de la seguridad informática en la sociedad, concientizar a todos los individuos de todas las edades y en todo el mundo respecto a la obligación del usuario a ejercer el uso de la tecnología de forma responsable y consciente de los peligros y riesgos a los que se enfrentan, ya que la tecnología crece exponencialmente y por ende el uso de la misma.

Por otra parte los responsables de la creación y desarrollo de tecnología también tienen obligaciones, siendo su principal tarea la de ofrecer a las personas recursos tecnológicos cada vez más confiables, más eficientes y más seguros. Esta tarea no es nada fácil, pues siempre existen riesgos en el uso de la tecnología y jamás se podrán eliminar en su totalidad; pero sí es posible mitigarlos y prevenirlos de distintas maneras.

Para ello existen una serie de estándares, protocolos, métodos, reglas, herramientas y leyes concebidas para minimizar los riesgos de la infraestructura o de la información.

El activo informático más importante y que más se busca proteger al realizar un análisis de riesgos en una organización es la información, su importancia radica en que su pérdida representaría la pérdida de dinero y significaría la quiebra segura de dicha organización. Existen diferentes tipos de información y se clasifican de acuerdo a su importancia, dicha importancia puede ser asignada por diferentes factores, por ejemplo en una organización la establecen sus políticas, sus objetivos, sus intereses, su misión y su visión principalmente. Pero también la importancia de la información puede ser clasificada de forma muy subjetiva y particular por cada individuo, cada persona considera de mayor importancia algunas cosas sobre otras y todos tenemos derecho a proteger aquello que consideramos importante, en este caso la información y para ser más precisos la información que se guarda y envía por medio de dispositivos electrónicos. Para alcanzar esto la Seguridad de la Información brinda distintos servicios, los cuales son: la confidencialidad, la autenticación, la integridad, el no repudio, el control de acceso y la disponibilidad.

Una de las principales herramientas que se usa para brindar los servicios de la confidencialidad, la integridad y la autenticación a la información es la criptografía. La RAE (Real Academia de la Lengua Española) define la criptografía como *"El Arte de escribir con clave secreta o de un modo enigmático".*

Vista como una ciencia aplicada, se fundamenta en teorías y ciencias matemáticas para dar apoyo a las tecnologías de información y comunicación de datos en general.

Existen diversos tipos de criptografía, de acuerdo a las técnicas, métodos o ciencias que utiliza como sustento, por ejemplo, existe la criptografía simétrica o convencional, la criptografía cuántica, la criptografía asimétrica o de clave pública, la criptografía de curva elíptica, la criptografía híbrida, entre otras; todas utilizadas de alguna u otra forma para brindar alguno o algunos de los servicios de seguridad.

Cada una de las ramas de la criptografía se caracteriza por su gran diversidad en algoritmos criptográficos que se han desarrollado a partir de las teorías que los fundamentan, a los cuales se les han encontrado diversas aplicaciones, entre las más destacadas está el de reforzar la seguridad de protocolos y servicios de comunicación en las redes de computadoras. Un ejemplo claro y cotidiano es el servicio de mensajería instantánea o en tiempo real que utilizan páginas y aplicaciones como *Messenger, Skype,* entre otros y que carecen de seguridad en los servicios que ofrecen como el envío y recibo de información, pues viajan en el internet de forma clara y legible para cualquier entidad ajena a la comunicación establecida. Algunas aplicaciones tratan de solucionar este problema cifrando las comunicaciones y garantizando el servicio de la confidencialidad de la conversación, pero lamentablemente no han solucionado en su totalidad el problema de la seguridad, ya que no se garantiza la autenticación y en especial, la integridad de los mensajes que se

envían, dejando una vulnerabilidad visible y fácil de ser explotada por cualquier atacante.

Hoy en día, las herramientas matemáticas brindan a la criptografía la oportunidad de optimizar algoritmos de cifrado existentes como son ElGamal y el algoritmo de firma digital o en inglés *Digital Signature Algorithm* (DSA), que pueden mejorar su eficiencia y robustez sin necesidad de utilizar más recursos computacionales, un ejemplo de estas herramientas matemáticas son la teoría de curvas elípticas. La criptografía de curva elíptica (CCE), es una variante de la criptografía asimétrica o de clave pública basada en las matemáticas de las curvas elípticas; sus autores argumentan que la CCE puede ser más rápida y usar claves más cortas que los métodos como el *Rivest, Shamir y Adleman* (RSA), al tiempo que proporcionan un nivel de seguridad equivalente.

Así, el objetivo del presente proyecto de tesis es:

"Desarrollar una aplicación de mensajería instantánea que brinde los servicios de seguridad, autenticación, integridad, confidencialidad y no repudio, por medio de la teoría de curvas elípticas".

Y como objetivos particulares se busca:

- Autenticar mensajes mediante el uso de DSA elíptico.
- Cifrar mensajes mediante ELGamal elíptico.
- Brindar a los usuarios una alternativa práctica de mensajería instantánea segura.

- Proporcionar a la aplicación las características mínimas que debe tener un software seguro

De manera que para alcanzar los objetivos planteados es que en el capítulo 1 se presenta un breve análisis de la importancia que implica el hacer uso de los servicios de seguridad (integridad, confidencialidad, autenticación y no repudio), en el uso de las redes de datos. Además, se presentan los fundamentos básicos y definiciones sobre los sistemas de mensajería instantánea; por lo tanto se explican sus características, los protocolos seguros que utilizan y la arquitectura en la que se fundamentan.

Por otra parte, partiendo de la necesidad de proteger la información que se transmite por medio de la mensajería instantánea y sabiendo las amenazas que existen en ella, en el capítulo 2 se presenta a la criptografía de curvas elípticas como una posible herramienta que pueda auxiliar a los desarrolladores y empresas que brinden servicios de mensajería instantánea, partiendo de los fundamentos matemáticos que la respaldan, sus aplicaciones en la criptografía, y el desarrollo de ejemplos paso a paso que ayuden a entender su funcionamiento.

En el capítulo 3 se realiza una propuesta de implementación para los algoritmos de cifrado ElGamal elíptico y ECDSA (*Elliptic Curve Digital Signature Algorithm*) en el lenguaje de programación Java, de acuerdo con los estándares internacionales. Se plantea por pasos el proceso de cifrado y descifrado así como el algoritmo, incluyendo un ejemplo que ayude a comprender el procedimiento.

Además se realiza una comparativa de los algoritmos basados en el problema del logaritmo discreto elíptico y de los que se basan en el problema de logaritmo discreto convencional.

Por último en el capítulo 4 se finaliza con el diseño y la implementación de una aplicación de mensajería instantánea segura que utilice los algoritmos de cifrado y firma digital (ElGamal elíptico y ECDSA) para llevar a cabo no sólo la autenticación de los usuarios, sino que también la transmisión de mensajes de un usuario a otro, es decir, cifrando y firmando digitalmente cada mensaje antes de enviarlo para garantizar la confidencialidad de la comunicación y también la integridad de cada mensaje. Aunado a la autenticación de los usuarios y buenas prácticas en el diseño y desarrollo del *software* como la modularidad del código, se tiene una aplicación de mensajería instantánea segura que funciona como alternativa segura y sencilla de comunicación.

Finalmente, se presentan las conclusiones en las que se explica cómo esta aplicación de mensajería instantánea segura, logra brindar los servicios de seguridad (integridad, confidencialidad, autenticación y no repudio), y cómo resulta ser una alternativa viable y sencilla para los usuarios que estén interesados en establecer conversaciones a través del internet de manera segura y confiable.

Capítulo 1
Mensajería instantánea

El ser humano es por naturaleza un ser sociable, es decir, es naturalmente inclinado al trato y relación con las personas o que gusta de ello, por lo tanto la forma en la que se comunica con sus semejantes se ha convertido en parte de la historia de la humanidad.

La comunicación es uno de los pilares básicos en los que se apoya cualquier tipo de relación humana y es provechosa en prácticamente todas las esferas de la actividad humana. Es crucial para el bienestar personal, para las relaciones interpersonales, nos ayuda a superar situaciones delicadas, a resolver conflictos, a expresar sentimientos, a defender nuestros intereses y a evitar malas interpretaciones, entre otras.

1.1 La seguridad en las Redes de datos.

La infraestructura de red, los servicios y los datos contenidos en las computadoras conectadas a la red son activos comerciales y personales muy importantes. Comprometer la integridad de estos activos puede ocasionar serias repercusiones financieras y comerciales. De acuerdo con la información obtenida del primer tomo de la currícula de "CISCO CCNA *Exploration"*; algunas de las consecuencias de la ruptura en la seguridad de la red son:

- ✓ Interrupciones de red que impiden la realización de comunicaciones y de transacciones, con la consecuente pérdida de negocios.
- ✓ Mal direccionamiento y pérdida de fondos personales o comerciales.
- ✓ Propiedad intelectual de la empresa (ideas de investigación, patentes o diseños) que son robados y utilizados por la competencia.
- ✓ Detalles de contratos con clientes que se divulgan a los competidores o son hechos públicos, generando una pérdida de confianza del mercado de la industria.

La falta de confianza pública en la privacidad, confidencialidad y niveles de integridad de los negocios puede derivar en la pérdida de ventas y, finalmente, en la quiebra de una empresa.

CISCO propone que existen dos tipos de cuestiones de seguridad de la red que se deben tratar a fin de evitar serias consecuencias; las cuales son seguridad de la infraestructura de la red y seguridad del contenido.

- ✓ Asegurar la infraestructura de la red incluye la protección física de los dispositivos que proporcionan conectividad de red y evitan el acceso no autorizado al software de administración que reside en ellos.

- ✓ La seguridad del contenido se refiere a la protección de la información contenida en los paquetes que se transmiten en la red y la información almacenada en los dispositivos conectados a ésta.

Se deben implementar herramientas para proporcionar seguridad al contenido de los mensajes individuales sobre los protocolos subyacentes que rigen la forma en que los paquetes se formatean, direccionan y envían. Debido a que el reensamblaje y la interpretación del contenido se delegan a programas que se ejecutan en sistemas individuales de origen y destino, muchos de los protocolos y herramientas de seguridad deben implementarse también en esos sistemas. Las medidas de seguridad que se deben tomar en una red son:

- ✓ Evitar la divulgación no autorizada o el robo de información.
- ✓ Evitar la modificación no autorizada de información.
- ✓ Evitar la denegación de servicio.

Por otra parte tenemos los medios para lograr estos objetivos, los cuales incluyen:

- ✓ Garantizar la confidencialidad.
- ✓ Mantener la integridad de la comunicación.
- ✓ Garantizar la disponibilidad.

Garantizar la confidencialidad

Un sistema seguro de autenticación de usuarios consiste en el cumplimiento de las contraseñas difíciles de adivinar y el requerimiento a los usuarios para que las cambien frecuentemente ya que ayudan a restringir el acceso a las comunicaciones y a los datos almacenados en los dispositivos adjuntos de la red. Cuando corresponda, el contenido cifrado asegura la confidencialidad y reduce las posibilidades de divulgación no autorizada o robo de información.

Mantener la integridad de las comunicaciones

La integridad de datos significa que la información no se alteró durante la transmisión de origen a destino. La integridad de datos puede verse comprometida cuando al dañarse la información, ya sea en forma intencional o accidental, antes de que el receptor correspondiente la reciba.

El uso de firmas digitales, algoritmos de hash y mecanismos de *checksum*[1] son formas de proporcionar integridad de origen y de datos a través de la red para evitar la modificación no autorizada de información.

Garantizar disponibilidad

Disponibilidad significa tener la seguridad de acceder en forma confiable y oportuna a los servicios de datos para usuarios autorizados. Los dispositivos firewall de red, junto con los *software* antivirus de los equipos de escritorio y de los servidores pueden asegurar la confiabilidad y solidez del sistema para detectar, repeler y resolver esos ataques. .

1.2 Fundamentos de la Mensajería Instantánea

Entre todos los elementos esenciales para la existencia humana, la necesidad de interactuar está por debajo de la necesidad de sustentar la vida. La comunicación es casi tan importante para nosotros como el aire, el agua, los alimentos y un lugar para vivir.

[1] Nota: Según el libro Seguridad en la informática de empresa: riesgos, amenazas, prevención y soluciones de Jean-Marc Roye, "*es una función hash que tienen como propósito principal detectar cambios accidentales en una secuencia de datos para proteger la integridad de estos, verificando que no haya discrepancias entre los valores obtenidos al hacer una comprobación inicial y otra final tras la transmisión".*

Los métodos que utilizamos para compartir ideas e información están en constante cambio y evolución. Mientras la red humana estuvo limitada a conversaciones cara a cara, el avance de los medios ha ampliado el alcance de nuestras comunicaciones. Desde la prensa escrita hasta la televisión, cada nuevo desarrollo ha mejorado la comunicación.

1.2.1 Definición y características

En el ámbito de las redes de ratos, la tecnología para establecer comunicación entre sistemas computacionales ha evolucionado a tal grado que inclusive ya no es necesario tener o proveer servicios informáticos de manera manual, o mejor dicho operada por alguna persona. Gracias a los avances en lenguajes de programación y al surgimiento de nuevos paradigmas en este ámbito, se han logrado programar aplicaciones que puedan brindar servicios informáticos de manera automática, a partir de un nodo o equipo de cómputo en una red. A este equipo en específico se le da el nombre de servidor, y así como hay servidores de servicios también hay clientes que usen esos servicios, dando paso a lo que se conoce en informática como arquitectura cliente-servidor.

Existen muchas definiciones en diccionarios técnicos y computacionales sobre la mensajería instantánea, por ejemplo en el diccionario virtual "**Internetglosario**" la definen como "*un sistema de intercambio de mensajes entre personas, escritos en tiempo real a través de redes".*

Para fines de este trabajo de tesis, se manejará el término de mensajería instantánea, conocida también en inglés como IM (*Instant Messaging*), en una forma de comunicación en tiempo real entre dos personas basada en texto. El texto es enviado a través de dispositivos conectados a una red como Internet.

Chat y mensajería instantánea son términos muy confundidos y son servicios que aunque son muy parecidos tienen diferencias estructurales importantes:

Chat: Es una charla en tiempo real. Se trata de una red de servidores cuyo servicio principal es la interconexión de usuarios para que puedan tener conversaciones.

Mensajería instantánea: A parte de la descripción del chat, tiene además, mantener una conversación de forma exclusiva con los usuarios que se deseé. Es decir, es más exclusivo este sistema que el chat

A continuación se propone una tabla de las diferencias e igualdades más importantes entre un chat y la mensajería instantánea. Tabla 1.1.

Tabla 1.1 Cuadro comparativo entre mensajería instantánea y chat

Chat	**Mensajería instantánea**
Se envían los mensajes en tiempo real.	Se envían los mensajes en tiempo real.
Los mensajes no se borran, sino que aparecen los más recientes y los antiguos se quedan atrás.	Los mensajes son temporales.
No se puede tener conversaciones de voz.	Se puede tener conversación de voz.
Los contactos no los puedes agrupar en ningún grupo, tienes a todos en la misma lista.	Los contactos los puedes agrupar en diferentes grupos.
Se pueden tener conversaciones pero con todos los usuarios del chat.	Se puede tener varias conversaciones a la vez pero cada una de forma individual.
No se pueden enviar archivos.	Se pueden enviar archivos.
No se pueden mostrar estados.	Se pueden mostrar varios estados.
No se pueden borrar contactos de la lista.	Se pueden borrar contactos de la lista.

1.2.3 Descripción del RFC 2778

Este RFC (*Request For Comments*) estableció un modelo básico para el desarrollo de un protocolo, no teniendo ninguna relación con alguna implementación de *software*. Aclara que los elementos presentes aquí pueden estar o no en las implementaciones, y que las combinaciones de las entidades aquí nombradas pueden sufrir modificaciones. Se considera que es útil describirlo para tener un mejor entendimiento de la arquitectura subyacente en estos sistemas. El modelo define dos servicios:

1. *Servicio de Presencia*[2] y un *Servicio de Mensajería Instantánea*. El *servicio de presencia* acepta, almacena y distribuye *Información de Presencia*.

2. El *Servicio de Mensajería Instantánea* acepta y entrega *Mensajes Instantáneos* a las *Casillas de Mensajes Instantáneos*.

[2] Nota: La letra en itálica indica que en el modelo este término fue descrito como un elemento del modelo. En el RFC existe un apartado con la definición técnica de cada uno. Los términos son traducidos del inglés para su entendimiento, aunque algunas son palabras sin traducción formal al castellano.

El Servicio de Presencia

El *Servicio de Presencia* tiene dos tipos de clientes, llamados:

1. *Entidad Presentadora de Datos* (*Presentity* en inglés), que provee *Información de Presencia* para ser almacenada y distribuida.

2. *Observador* (*Watcher* en inglés), que recibe *Información de Presencia* del *Servicio de Presencia*.

Hay dos tipos de *Observadores*: llamados

1. *Trae* (*Fetcher* en Inglés)

2. *Suscriptor* (*Suscriber* en Inglés).

Un *Trae* pide *Información de Presencia* al *Servicio de Presencia*, y un *Suscriptor* tiene algún medio de "avisar" al servicio de presencia que notifique el cambio (futuro) de la *Información de Presencia* de alguna *Entidad Presentadora de Datos*. Un tipo especial de *Trae* es uno que trae información en un intervalo regular de tiempo (también llamado polling o encuesta). Es por esto que se llama *Encuestador*. Véase figura 1.1

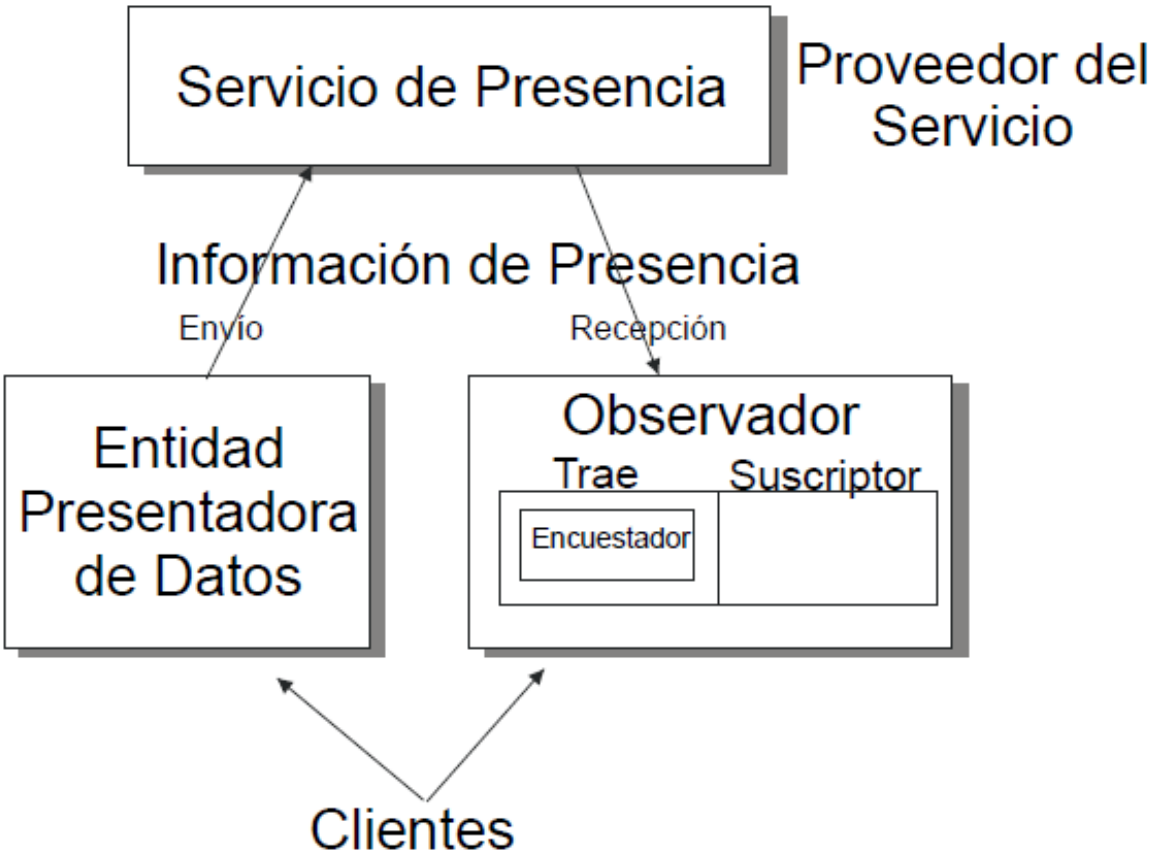


Figura 1.1 Modelo del funcionamiento de un Servicio de Presencia.

Cabe mencionar que el *Servicio de Presencia* tiene *Información de Observadores* acerca de los *Observadores* y sus actividades, en términos de sí traen o se suscriben a la *Información de Presencia*. Los cambios de la *Información de Presencia* son distribuidos a los *Suscriptores* vía lo que se llaman *Notificaciones*.

1.3 Protocolos seguros de Mensajería instantánea

El uso de la mensajería instantánea va en ascenso permanente. Existen muchos protocolos y aplicaciones que se usan actualmente, cada vez con más funciones que facilitan la vida diaria de las personas alrededor del mundo y de todas las edades. A pesar de tildarse como una "pérdida de tiempo productivo" en las empresas, "*e-Marketeer"* reveló los resultados de una encuesta realizada por el "*META Group*", donde las empresas indicaban los beneficios de usar la mensajería instantánea, con los siguientes resultados:

- ✓ 78% Respuesta más rápida que al correo electrónico.
- ✓ 74% Mayor rapidez en la resolución de problemas.
- ✓ 71% Posibilidad de efectuar multitareas.
- ✓ 63% Habilidad para determinar si alguien está en línea, disponible o libre.
- ✓ 62% Habilidad para obtener la atención de otra persona.
- ✓ 37% Contactar a alguien que no ha contestado el correo electrónico o mensaje de voz.
- ✓ 37% Reducción de costos evitando usar llamadas de larga distancia.
- ✓ 32% Obtención eficiente de información.

Inclusive a todas las edades y a todos los niveles sociales, se utiliza para comunicarse más la mensajería instantánea que los correos electrónicos, llamadas de voz y cualquier otro medio digital. En la medida que la tecnología avance, habrá mayor necesidad de abrir la posibilidad de comunicación por este medio.

Existe una gran cantidad de protocolos de mensajería instantánea, sin embargo no todos ofrecen seguridad en su uso. Por este motivo se han ido proponiendo una serie de alternativas que cumplían con los requisitos de seguridad necesarios y que no tengan un impacto en el rendimiento. A continuación se describen algunas de las propuestas.

Protocolo SILC

El protocolo SILC (*Secure Internet Live Conferencing*) fue propuesto por Pekka Riikonen en 1997 para hacer seguros a los sistemas de mensajería instantánea e IRC.

El protocolo de intercambio de llaves de SILC, es decir, SKE está basado en el algoritmo criptográfico de Diffie-Hellman. Tras una ejecución correcta de este protocolo se procede con el protocolo de autenticación "*SILC Connection Authentication Protocol*" haciendo uso o de una *passphrase*[3] o de una clave pública.

[3] Nota: Significa "Frase de contraseña", es una secuencia de palabras o de otro tipo de texto se utiliza para controlar el acceso a un sistema informático, programa o datos. Es una palabra de acceso es similar a una contraseña en el uso, pero es generalmente más larga para mayor seguridad.

La implementación del protocolo SKE es vulnerable a un ataque de tipo "*Man in the Middle*"[4] aprovechando que el certificado utilizado durante el protocolo SKE no se verifica correctamente.

Protocolo IMKE

En el año 2005, M. Mannan y P. C. van Oorschot propusieron el protocolo IMKE (*Instant Messaging Key Exchange*). Este protocolo puede descomponerse en tres estados: el intercambio de claves, es decir, fase inicial de autenticación de los usuarios en el servidor, la comunicación cliente-servidor y las conexiones cliente-cliente. Se entiende que el certificado digital del servidor está incluido en el propio cliente o se ha instalado previamente. Durante la fase de intercambio de claves:

1. El cliente genera dinámicamente un par de claves, pública y privada, (RSA 1024/2048-bit) y una clave simétrica aleatoria AES (*Advanced Encryption Standard*)-128.

2. Con la clave simétrica recién generada cifra los datos de autenticación, es decir, incluye el hash de la contraseña SHA (*Secure Hash Algorithm*)-1 y la clave pública del cliente, posteriormente envía los datos cifrados al servidor junto con la clave utilizada cifrada haciendo uso de la clave pública del servidor.

[4] Nota: Es un ataque en el que el enemigo adquiere la capacidad de leer, insertar y modificar a voluntad, los mensajes entre dos partes sin que ninguna de ellas conozca que el enlace entre ellos ha sido violado.

3. Si en el servidor concuerda la contraseña del usuario con la recibida, envía al cliente un paquete cifrado con la clave pública del cliente y el hash de la contraseña cifrado con la clave simétrica.

4. El cliente comprueba que la clave recibida es la correcta y en caso de que concuerden descifra el paquete recibido y lo reenvía al servidor haciendo uso de SHA-1.

5. El servidor comprueba que el paquete recibido es el correcto y en caso de que concuerden tanto el cliente como el servidor calculan el identificador de sesión y la MAC (*Media Access Control*), (HMAC utilizando SHA-1), a través de una nueva función hash conocida por ambos, en la que intervienen la clave simétrica inicial y el paquete.

Atendiendo a la seguridad del protocolo, cabe destacar que esta propuesta permite que, si la comunicación es interceptada en cualquiera de las fases de la misma, incluso en el intercambio de claves, el atacante no podrá descifrar la información por no disponer de la clave privada del cliente. Aun así, es sensible a ataques de denegación de servicio al poderse replicar los paquetes del cliente. Además, dado que se producen conexiones P2P (*Peer-to-Peer*) sin necesidad de enviar archivos, es posible acceder a la dirección IP de un contacto simplemente iniciando una sesión con él.

Capítulo 2
Criptografía de Curvas elípticas

Los seres humanos siempre han sentido la necesidad de ocultar información, mucho antes de que existieran las primeras computadoras y equipos informáticos que hoy en día es difícil no hacer uso de ellos para la vida diaria. Sin embargo, durante la historia se han utilizado diferentes estrategias que han ayudado al ser humano a ocultar aquello que no quiere que sea visto por personas no autorizadas, pero conforme avanza la tecnología, el mundo se ha ido globalizando cada vez más, y por lo tanto, cualquier clase de información puede estar siendo vista en el otro lado del mundo, en un instante.

Desde su creación, internet ha evolucionado hasta convertirse en una herramienta esencial de la comunicación. Sin embargo, esta comunicación implica un número creciente de problemas y riesgos estratégicos relacionados con las actividades de las empresas en la web y de prácticamente cualquier usuario que haga uso de este recurso tecnológico. Toda clase de interacciones que se realizan a través de la red, al igual que toda clase de información que se ingresa, es pública y visible para cualquier entidad que lo deseé. Esto desgraciadamente no lo conocen todos los usuarios que interactúan con el Internet y en muchas ocasiones piensan que su información esta a salvo de cualquier alteración y que es confidencial para todo el mundo.

Como se vio en el capítulo anterior, los sistemas de mensajería instantánea no están exentos de ataques informáticos ni brindan absoluta seguridad. Extrañas son las empresas o desarrolladores que tomen en serio la importancia de la seguridad en las comunicaciones, ya que lamentablemente y en muchas ocasiones el dinero es el único interés de estas entidades y dejan a un lado el brindar un servicio de calidad y sobre todo, seguro para los usuarios.

2.1 Criptografía asimétrica

Este sistema de cifrado usa dos claves diferentes. Una es la clave pública y se puede enviar a cualquier persona y otra que se llama clave privada, que debe guardarse para que nadie tenga acceso a ella. Para enviar un mensaje, el remitente usa la clave pública del destinatario para cifrar el mensaje. Una vez que lo ha cifrado, solamente con la clave privada del destinatario se puede descifrar, ni siquiera el que ha cifrado el mensaje puede volver a descifrarlo. Por ello, se puede dar a conocer perfectamente la clave pública para que todo aquel que se quiera comunicar con el destinatario lo pueda hacer. En la figura 2.1 se muestra el proceso criptográfico de cifrado y descifrado de un sistema asimétrico.

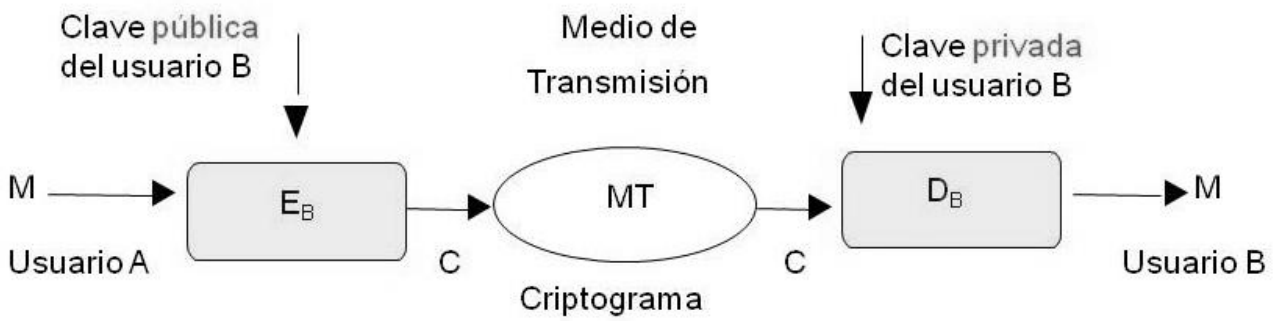


Figura 2.1 Sistema criptográfico asimétrico

Un sistema de cifrado de clave pública basado en la factorización de números primos se basa en que la clave pública contiene un número compuesto de dos números primos muy grandes. Para cifrar un mensaje, el algoritmo de cifrado usa ese compuesto para cifrar el mensaje. Para descifrar el mensaje, el algoritmo de descifrado requiere conocer los factores primos, y la clave privada tiene uno de esos factores, con lo que puede fácilmente descifrar el mensaje.

Según el segundo principio de Kerckhoffs, toda la seguridad debe descansar en la clave y no en el algoritmo. Por lo tanto, el tamaño de la clave es una medida de la seguridad del sistema, pero no se puede comparar el tamaño de la clave del cifrado simétrico con el del cifrado de clave pública para medir la seguridad. En un ataque de fuerza bruta sobre un cifrado simétrico con una clave del tamaño de 80 bits, el atacante debe probar hasta 2^{80}-1 claves para encontrar la clave correcta. En un ataque de fuerza bruta sobre un cifrado de clave pública con una clave del tamaño de 512 bits, el atacante debe factorizar un número compuesto codificado en 512 bits, es decir, hasta 155 dígitos decimales.

La cantidad de trabajo para el atacante será diferente dependiendo del cifrado que esté atacando. Mientras 128 bits son suficientes para cifrados simétricos, dada la tecnología de factorización de hoy en día, se recomienda por el NIST el uso de claves públicas de 1024 bits para la mayoría de los casos.

La mayor ventaja de la criptografía asimétrica es que la distribución de claves es más fácil y segura, ya que la clave que se distribuye es la pública manteniéndose la privada para el uso exclusivo del propietario, pero este sistema tiene bastantes desventajas:

- ✓ Para una misma longitud de clave y mensaje se necesita mayor tiempo de proceso.

- ✓ Las claves deben ser de mayor tamaño que las simétricas.
- ✓ El mensaje cifrado ocupa más espacio que el original.

Los nuevos sistemas de clave asimétrica basado en curvas elípticas tienen características menos costosas en cuanto a recursos de la computadora.

2.1.1 Conceptos matemáticos de la Criptografía asimétrica

Campos finitos. Los campos finitos son un importante prerrequisito para utilizar sistemas basados en curvas elípticas y para comprender algoritmos como RSA ya que las operaciones utilizadas para implementar éstos funcionan bajo las leyes del álgebra moderna.

Grupos, anillos y campos.

Son los elementos fundamentales de la parte de las matemáticas conocida como álgebra moderna o abstracta. En este tipo de álgebra se puede combinar dos elementos con características similares de diferentes maneras para obtener un tercer elemento que también contenga dichas características y que por lo tanto pertenezcan al álgebra moderna. Así, es importante que en esta área no estemos limitados con operaciones aritméticas ordinarias.

Grupo. Es una dupla ordenada {G, *}, donde G es un conjunto y el * es una operación binaria, que cumple con las siguientes propiedades:

1. **Cerradura:** Si un elemento a y uno b pertenecen a G, entonces $a*b$ también pertenece a G.

2. **Asociativa:** $a*(b*c)=(a*b)*c$ para todos los a, b y c pertenecientes a G.
3. **Elemento idéntico:** Existe un elemento e en G tal que $a*e=e*a=a$ para toda a en G.
4. **Elemento inverso:** Para cada a en G existe un elemento a' en G tal que $a*a'=a'*a=e$.

Si además se tiene la propiedad de conmutatividad ($a*b=b*a$ para todo a, b en G), el grupo recibe el nombre de grupo conmutativo o grupo abeliano.

Los grupos se definen para una operación sin embargo en ocasiones es útil tener estructuras con propiedades que involucren en forma simultánea dos operaciones sobre un mismo conjunto a esto se le llama anillo.

Anillo. Es una colección de elementos con dos operaciones binarias, llamadas adición y multiplicación, denotada por {R, +, *}. Además para todo a, b, c pertenecientes al conjunto R se deben seguir los siguientes axiomas:

1. (R, +) forman un grupo abeliano, esto quiere decir que cumple con las propiedades anteriores.
2. **Cerrada bajo la multiplicación:** Si a y b pertenecen a R, entonces $a*b$ también pertenece a R.
3. **Asociativa en la multiplicación:** $a*(b*c)=(a*b)*c$ para toda a, b, c en R.
4. **Ley distributiva:** $a*(b+c)=a*b+a*c$ para toda a, b, c en R, $(a+b)*c=a*c+b*c$ para toda a, b, c en R.

Un anillo cuya multiplicación es conmutativa se denomina anillo conmutativo ($a*b=b*a$ para todo a, b en R).

En el dominio de los enteros, al cual se le considera un anillo conmutativo, se obedecen los siguientes axiomas:

5. **Idéntico multiplicativo:** Existe un elemento 1 en R tal que $a*1=1*a=a$ para toda a en R.
6. **No existe división entre cero:** Si a, b en R y $a*b=0$, entonces $a=0$ ó $b=0$.

Los números primos son anillos que presentan propiedades que lo hacen un tipo de anillo más especial llamado campo.

Campo. Es una estructura algebraica en la cual las operaciones de adicción, multiplicación y división se pueden efectuar y cumplen con las propiedades anteriormente vistas como lo son las propiedades asociativa, conmutativa y distributiva, las cuales nos son familiares de la aritmética de los números ordinarios (los números racionales, reales y complejos).

Los cuerpos o campos eran llamados dominios racionales. En general un campo es una abstracción de algún sistema numérico y de sus propiedades esenciales. Un campo F es un anillo conmutativo, denotado por {F,+,*}, en el que cada elemento que sea distinto de cero, es decir todo elemento que no sea nulo, tiene un inverso multiplicativo, esto es:

Para cada a en F, excepto 0, existe un elemento a^{-1} en F tal que $a*a^{-1}=a^{-1}*a=1$. La división se define como: $a/b=a*b^{-1}$. En un campo podemos sumar, restar, multiplicar y dividir.

Si un conjunto de elementos de un campo es finito, se dice que es un campo finito. El orden de un campo finito es el número de elementos en el campo.

Campo finito. Es un cuerpo que contiene un número finito de elementos. Los cuerpos finitos son importantes en teoría de números, geometría algebraica, teoría de Galois, y en especial para el estudio en criptografía.

Dado que todo cuerpo de característica 0 contiene a los números racionales y es por lo tanto infinito, todos los campos finitos tienen característica prima, y por lo tanto existe un campo finito de orden q si y sólo si q es una potencia prima $q = p^m$ donde p es un número primo y m es un entero positivo. Si m=1, entonces el campo finito es llamado un campo primo. Si $m \geq 2$, entonces el campo finito es llamado campo extendido. No es en general cierto, sin embargo, que todo cuerpo de característica prima sea finito.

Para un primo p, los enteros módulo p forman un cuerpo de p elementos, denotado por GF(p), en algunos casos se usa Z_p.

Campos Primos

Para un número primo p, el conjunto de los números enteros módulo p es un cuerpo finito con los p elementos: esto se escribe a menudo como GF(p) ={0, 1, … , p-1} donde las operaciones son definidas realizando la operación en GF(p), dividiendo por p y tomando únicamente el residuo.

Campos Binarios

Campos finitos de orden 2^m son llamados campos binarios o campos finitos de característica dos. Una forma de construir estos campos (denotados por GF(2^m)) es utilizar la representación de base polinomial, en la cual los elementos de GF(2^m) son los polinomios binarios (polinomios cuyos coeficientes corresponden al campo GF(2) = {0,1}) de grado m-1.

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1 x + a_0 = \sum_{i=0}^{n-1} a_i x^i$$

Un polinomio binario irreducible f(x) de grado m es seleccionado. Irreducible significa que f(x) no puede ser factorizado como un producto de polinomios binarios cada uno de grado menor a m. La adición de los elementos del campo es la adición de los polinomios, con aritmética de los coeficientes módulo 2. La multiplicación de los elementos del campo se efectúa módulo el polinomio de reducción f(x).

2.2 Curvas Elípticas

Las curvas elípticas como entidades algebraicas y geométricas han sido estudiadas extensivamente por los últimos 150 años, y a través de algunos de estos estudios ha surgido una teoría profunda y extensa. Los primeros sistemas de curvas elípticas como aplicaciones a la criptografía fueron propuestos en 1985 de forma independiente por Neal Koblitz de la Universidad de Washington y por Victor Miller de IBM.

Muchos criptosistemas requieren el uso de grupos algebraicos, las curvas elípticas pueden utilizar los que provienen de los grupos de curvas elípticas.

La gran mayoría de criptosistemas de clave pública están basados en el uso de grupos abelianos, y estos demuestran ciertas propiedades de las operaciones que definen un grupo y son aplicables a un sistema, esto es, para criptografía de curvas elípticas, por ejemplo, se utiliza una operación de adición sobre el grupo de curvas elípticas y la multiplicación se define como la repetición de la adicción.

Por ejemplo para la multiplicación a*k se suma k veces a, $a*k=(a_1+a_2+\ldots+a_k)$, donde la adicción se presenta sobre una curva elíptica.

Una curva elíptica es definida por una ecuación con dos variables con sus respectivos coeficientes. Primero se va a ver las curvas elípticas cuando las variables y los coeficientes tienen números reales porque en este caso se puede visualizar mejor.

2.2.1 Curvas elípticas sobre los números reales

Las curvas elípticas no son elipses, se les nombra de esta forma porque se describen a través de una ecuación cúbica. En general, ecuaciones cúbicas para curvas elípticas tienen la forma:

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

Donde "a", "b", "c", "d" y "e" son números reales, "x" y "y" tienen valores sobre los números reales. En general, una curva elíptica sobre los números reales puede ser definida por los puntos (x,y), que satisfacen la ecuación de una curva elíptica de la forma:

$$y^2 = x^3 + ax + b$$

La ecuación anterior se dice que es cúbica, o de grado 3, porque el valor exponencial mayor es 3. Se puede ver que un grupo definido en su base E(a, b) que provee la curva $x^3+ ax + b$ no tiene factores repetidos. Esto es equivalente a la condición

$$\Delta = \Delta(a,b) = 4a^3 + 27b^2 \neq 0$$

La expresión anterior se llama discriminante. Si Δ no se anula no se tienen raíces múltiples lo que equivale a que la curva representada por $x^3+ ax + b$ no tiene puntos singulares. En el caso de que a=0 queda incluido ya que si a=0 tenemos que $x^3+ ax + b = x^3+ b$.

Cada uno de los números "a" y "b" requiere una curva elíptica diferente. Por ejemplo a=-4 yb=0.67 se asignan a una ecuación de una curva elíptica $y^2 = x^3 - 4x + 0.67$, tal como se muestra en la figura 2.2.

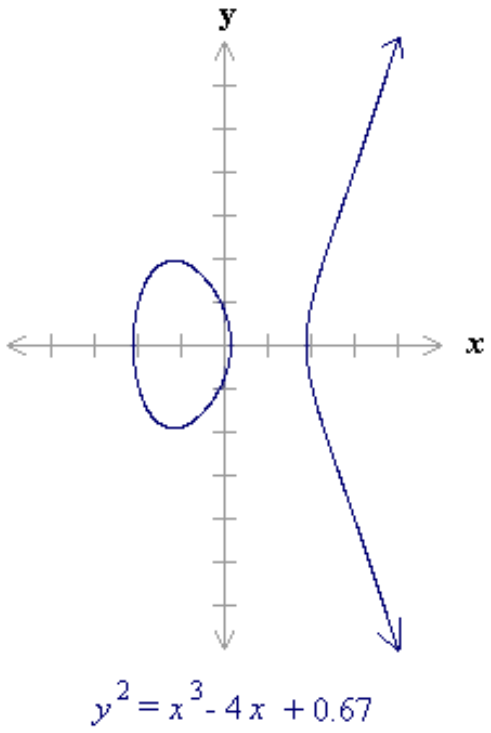


Figura 2.2 Curva elíptica con a=-4 y b=0.67

Si $x^3 + ax + b$ no contienen factores repetidos, o si $4a^3 + 27b^2$ es diferente de 0, entonces la curva elíptica $y^2 = x^3 + ax + b$ puede ser utilizada para formar un grupo.

Un grupo de curva elíptica sobre los números reales consiste de los puntos que corresponden a esta curva, junto con un punto especial llamado el punto al infinito o punto cero.

Descripción geométrica de la adición

Los puntos sobre una curva elíptica forman un grupo abeliano respecto a la operación que se va a introducir en forma geométrica. Se asume por el momento que se está en el caso de los números reales, con el punto (x, y) perteneciente a los mismos números reales.

Entonces se va a definir una operación llamada adición para un conjunto E(a,b), donde a y b satisfacen la ecuación que define una curva elíptica. En términos geométricos, las reglas de adición pueden ser establecidas como sigue: Si tres puntos se encuentran sobre la curva elíptica su suma es igual a 0. Se va a definir las reglas de la adición sobre una curva elíptica:

1. El punto 0 sirve como elemento aditivo idéntico. Desde que 0= -0, para cualquier punto P en la curva elíptica, P + 0 = P. Se asume que P≠0 y Q≠0. En la figura 2.3 se puede ver un ejemplo de esta regla.

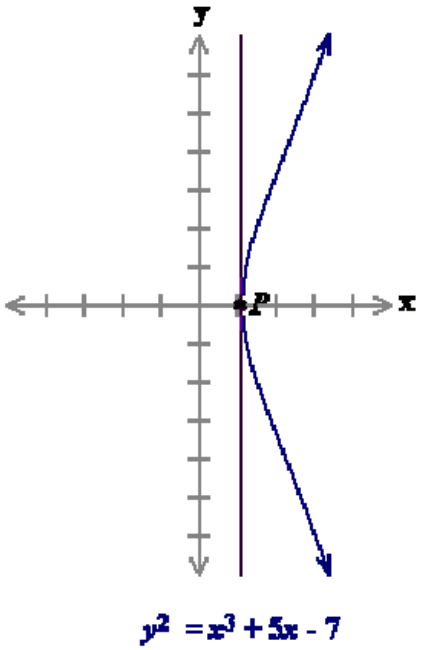


Figura 2.3 Se tiene el punto P = (1, 0), por lo tanto P + 0 = P, es decir la intersección con la curva es en un solo punto, la recta es tangencial a la curva en el punto P.

2. El negativo de un punto P es el punto con la misma coordenada en x, abscisa, pero con coordenada y, ordenada, negativa (P = (x, y) entonces –P = (x, -y)). Se debe notar que estos 2 puntos se representan como una línea vertical y además que P + (-P) = P –P = 0. Recordando que a 0 se le llama punto cero o punto en el infinito.

3. Para sumar dos puntos distintos P y Q se traza una línea que pase por ambos puntos y se tiene que intercepta a la curva en un tercer punto denominado R. Se puede apreciar que R es el único punto de intersección a menos que la línea trazada sea tangente a la curva en cuyo caso se tiene que P o Q son tangentes a dicha curva, entonces se tiene que R=P o R=Q respectivamente. Para formar una estructura de grupo, se necesita definir la adición de esos tres puntos como: P + Q = -R. Esto es P+Q va ha ser la imagen reflejada sobre el eje de las abscisas del tercer punto de intersección. La figura 2.4 ilustra esta construcción.

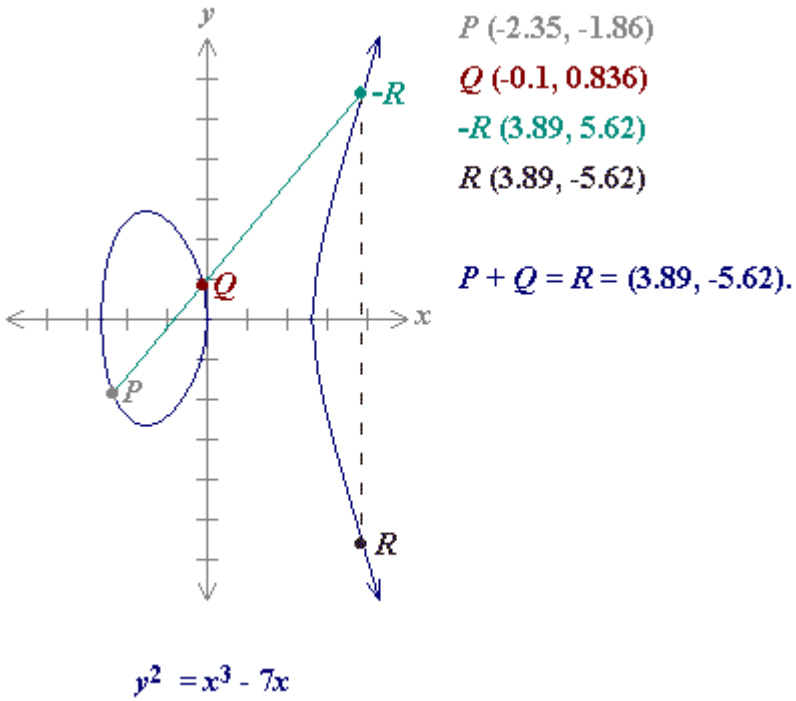


Figura 2.4 Para sumar los puntos P y Q, una línea se dibujara a través de los dos puntos. Esta línea intercepta a la curva elíptica en exactamente un punto –R. El punto –R se refleja en el eje de las x al punto R.

4. La interpretación geométrica anterior también se puede aplicar a dos puntos con la misma coordenada x como por ejemplo P y –P. Los dos puntos muestran una línea vertical la cual considera que la intersección con el tercer punto de la curva es con el punto infinito. Se tiene que P+ (-P)=0. La figura 2.5 muestra lo referente a la presente regla.

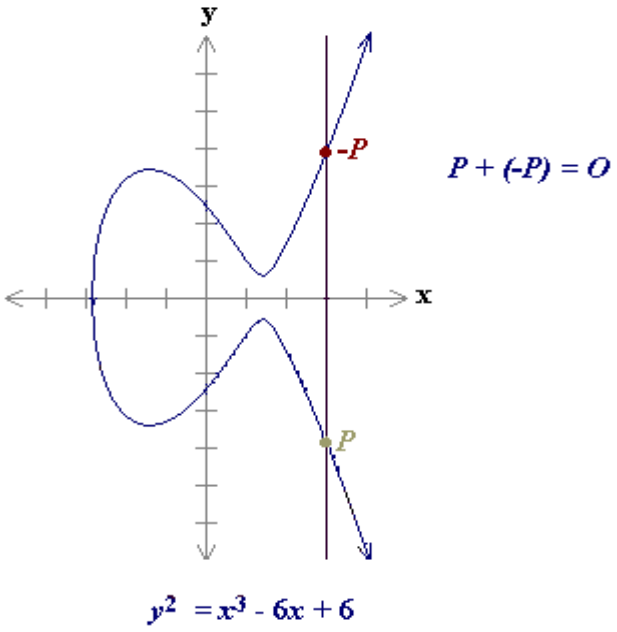


Figura 2.5 Suma de P+ (-P) = O

5. El punto doble P, traza una línea tangente a la curva y encuentra otro punto de intersección -R. Entonces P+P=2P=R. En la figura 2.6 se ilustra esto.

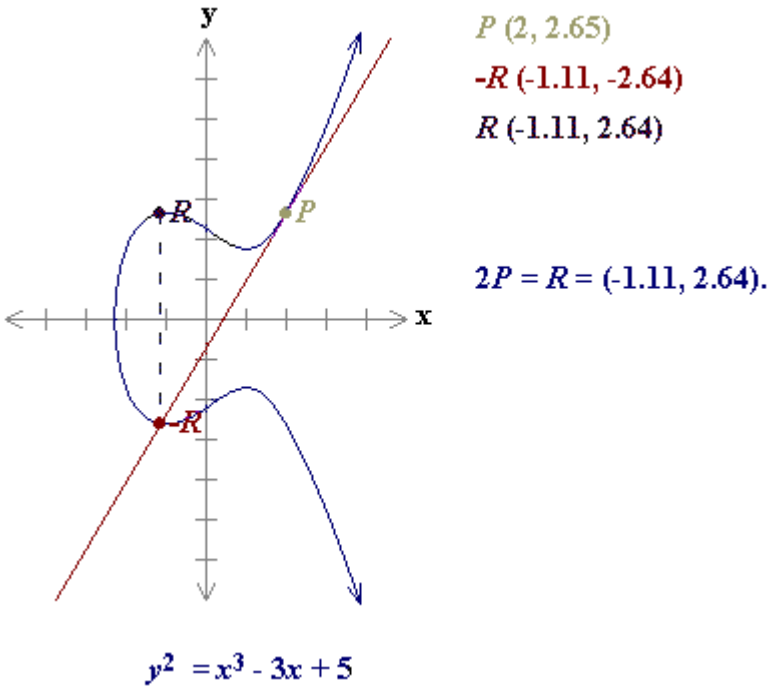


Figura 2.6 Suma de P+P

Con las cinco reglas anteriores se puede mostrar como operan básicamente las curvas elípticas de una forma geométrica.

Descripción algebraica de la adición

Aunque la descripción geométrica anterior de una curva elíptica proviene de un excelente método de ilustrar la aritmética de las curvas elípticas, ésta no es una manera adecuada para la aritmética computacional. Las fórmulas algebraicas que se muestran a continuación se construyen para procesar la información en la computadora de manera eficientemente.

Para sumar dos puntos distintos $P = (x_P, y_P)$ y $Q = (x_Q, y_Q)$ los cuales no son negativos uno del otro, se tiene la inclinación de la línea a través de P y Q, de la forma $\Delta = (y_P - y_Q) / (x_P - x_Q)$, y se puede expresar la suma P+Q=R como sigue:

$$x_R = \Delta^2 - x_P - x_Q$$

$$y_R = -y_P + \Delta(x_P - x_R)$$

Además, se requiere sumar un punto consigo mismo: P + P = 2P = R. Cuando $y_P \neq 0$.

$$\Delta = (3x_P^2 + a) / (2\, y_P)$$

$$x_R = \Delta^2 - 2\, x_P$$

$$y_R = -\, y_P + \Delta(x_P - x_R)$$

Ejemplos de la adición

En el grupo de la curva elíptica definida por $y^2 = x^3 -17x + 16$ sobre los números reales ¿Cuál sería la suma P+Q si P = (0,-4) y Q = (1,0)? Utilizando las ecuaciones correspondientes para la adición se tiene que:

$\Delta = (y_P - y_Q) / (x_P - x_Q) = (-4-0) / (0-1) = 4$

$x_R = \Delta^2 - x_P - x_Q = 16 - 0 - 1 = 15$

$y_R = -y_P + \Delta(x_P - x_R) = 4 + 4(0 - 15) = -56$

Por lo tanto P+Q = R = (15, -56).

En el grupo de la curva elíptica definida por $y^2 = x^3 -17x + 16$ sobre los números reales ¿Cuál sería la suma 2P si P= (4, 3.464)? De las fórmulas para sumar puntos consigo mismo:

$\Delta = (3x_P^2 + a) / (2\, y_P) = (3*(4)^2 - (17))/ (2*3.464) = 31/6.928 = 4.475$

$x_R = \Delta^2 - 2\, x_P = (4.475)^2 - (2*4) = 20.022 - 8 = 12.022$

$y_R = - y_P + \Delta(x_P - x_R) = -3.464 + 4.475(4-12.022) = 3.464 - 35.898 = -39.362$

Por lo tanto 2P= (12.022, -39.362).

2.2.2 Curvas elípticas sobre los números primos

Calcular sobre el campo de los números reales es lento e inexacto debido al error de redondeo. Las aplicaciones criptográficas requieren rapidez y precisión algebraica; así el grupo de curvas elípticas sobre el campo finito de GF(p) pertenecientes a los campos primos y GF(2^m) pertenecientes a los campos binarios son usadas en la práctica. No existe una interpretación geométrica de la aritmética de curvas elípticas sobre los campos finitos.

Recordando que el campo de GF(p) usa los números del 0 al p-1 y en cómputo final se obtiene el módulo dep. Por ejemplo, en GF(23) el campo compuesto de enteros es de 0 a 22 y cualquier operación dentro de este campo dará lugar a un número entero también entre 0 y 22.

Una curva elíptica con su subyacente campo de GF(p) puede estar formado por las variables a y b dentro del campo de GF(p). Las curvas elíptica incluyen todos los puntos de (x,y)que satisface la ecuación de una curva elíptica de un modulo p.

Por ejemplo: $y^2 \bmod p = (x^3 + ax + b) \bmod p$ tiene un campo subyacente de GF(p) si a y b estánen GF(p).

Si $x^3 + ax + b$ contiene factores no repetidos (o equivalentemente si $4a^3 + 27b^2$ mod p no es0), entonces la curva elíptica se puede utilizar para formar a un grupo. Una curva elíptica sobre el grupo de GF(p)tiene los puntos correspondientes en la curva elíptica, junto con un punto especial 0, el cual se le llama punto en infinito o punto cero. Son limitados los muchos puntos en las curvas elípticas.

Como ejemplo, considérese una curva elíptica en el campo GF(23). Con a=1 y b=0,la ecuación de la curva elíptica es $y^2 = x^3 + x$. El punto (9, 5) satisface la ecuación:

$y^2 \pmod p = x^3 + x \pmod p$

$25 \pmod{23} = 729 + 9 \pmod{23}$

$25 \pmod{23} = 738 \pmod{23}$

$2 = 2$

Los 23 puntos que satisfacen esta ecuación son:(0,0) (1,5) (1,18) (9,5) (9,18) (11,10) (11,13) (13,5)(13,18) (15,3) (15,20) (16,8) (16,15) (17,10) (17,13) (18,10)(18,13) (19,1) (19,22) (20,4) (20,19) (21,6) (21,17).Estos puntos se pueden representar gráficamente en la figura 2.7:

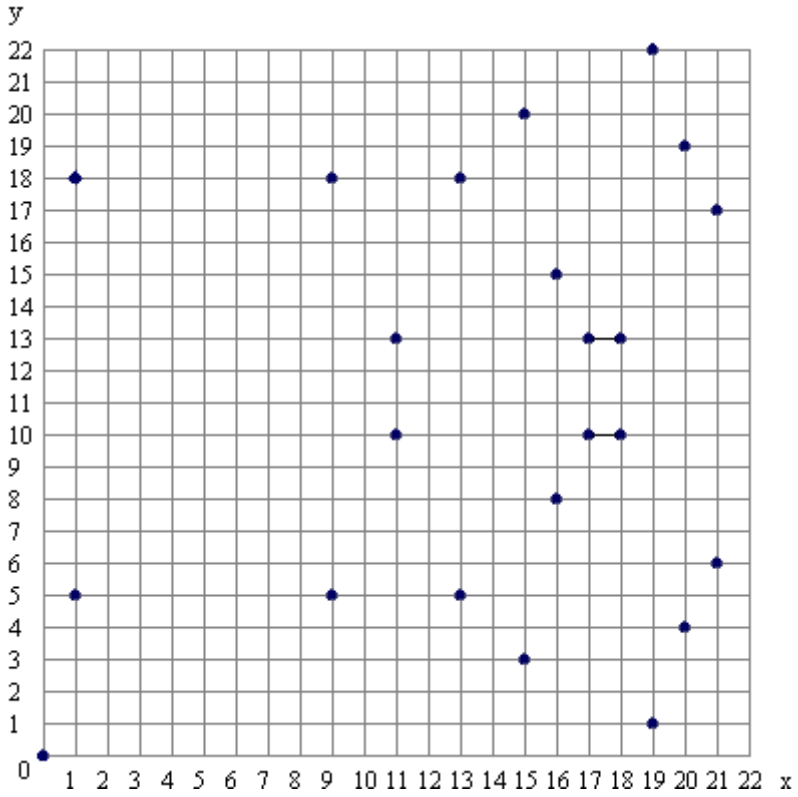


Figura 2.7 Curva elíptica definida por $y^2 = x^3 + x$ en el GF(23).

Se puede observar que hay dos puntos por cada valor de x. Aún cuando el gráfico se parece al azar, allí es simetría inmóvil sobre y = 11.5. Hay que recordar que las curvas elípticas sobre el campo de los números reales, cuentan con un punto negativo por cada punto que es el reflejo a través del eje x. Sobre el campo de GF(23) la componente negativa valuada en y se toma el módulo de 23, dando por resultado un número positivo diferente de 23.

Se encuentran muchas diferencias impotantes entre el grupo de curvas elípticas sobre GF(p) y sobre los números reales. El grupo de curvas elípticas sobre GF(p) tiene un número finito de puntos, lo cual es una propiedad deseable para los objetivos de la criptografía. Puesto que estas curvas consisten en algunos puntos discretos, no está claro cómo se relacionan estos puntos para hacer que su gráfica parezca una curva y tampoco está claro cómo las relaciones geométricas pueden ser aplicadas. Consecuentemente, la geometría usada en los grupos de curvas elípticas sobre números reales no puede usarse para grupos de curvas elípticas sobre GF(p).Sin embargo, las reglas algebraicas para la aritmética se pueden adaptar para curvas elípticas sobre GF(p). A diferencia de las curvas elípticas sobre números reales, el cálculo sobre el campo de GF(p) no involucra un error de redondeo una propiedad esencial que se requiere para su utilización en un criptosistema.

Las reglas para la adición sobre $E_p(a, b)$ equivalente a la técnica algebraica que se describió para las curvas elípticas definidas para los números reales:

1. P+0=P

2. Si P = (x_P, y_P), entonces P + $(x_P,-y_P)$=0. El punto $(x_P, -y_P)$ es el negativo de P, se denota como -P. Por ejemplo, en $E_{23}(1,0)$, para P=(15, 3) tenemos que el punto negativo es - P=(15, -3). Pero -3 mod 23= 20. Entonces - P=(15, 20), también se encuentra en $E_{23}(1,0)$.

3. Si $P=(x_P, y_P)$ y $Q=(x_Q, y_Q)$ con $P \neq -Q$, entonces $R=P+Q$ es determinada por las siguientes reglas:

   Si $P \neq Q$

   $\lambda = [(y_P - y_Q) / (x_P - x_Q)] \bmod p$

   $x_R = [\lambda^2 - x_P - x_Q] \bmod p$

   $y_R = [-y_P + \lambda (x_P - x_R)] \bmod p$

   Si $P=Q$ o se quiere sumar $R=2P=P+P$

   $\lambda = [(3x_P^2+a) / (2y_P)] \bmod p$

   $x_R = [\lambda^2 - 2x_P] \bmod p$

   $y_R = [-y_P + \lambda (x_P - x_R)] \bmod p$

4. La multiplicación se define como la repetición de la adición, por ejemplo $5P = P + P + P + P + P$.

Ejemplos

En el grupo de la curva elíptica definida por $y^2 = x^3 + x + 7$ sobre GF(17) ¿Cuál sería la suma P + Q si P = (2,0) y Q = (1,3)?Utilizando las fórmulas de adición se tiene:

$\lambda = [(y_P - y_Q) / (x_P - x_Q)] \bmod p = [(-3) / (2-1)] \bmod 17 = [(-3) / (1)] \bmod 17 = (-3*1^{-1}) \bmod 17$.

$\lambda = (-3*1^{-1}) \bmod 17 = (-3*1) \bmod 17 = -3 \bmod 17 = 14$.

$x_R = (\lambda^2 - x_P - x_Q) \bmod p = (196–2–1) \bmod 17 = 193 \bmod 17 = 6$

$y_R = [-y_P + \lambda (x_P - x_R)] \bmod p = [0+14*(2–6)] \bmod 17 = -56 \bmod 17 = 12$

Por lo tanto $P + Q = R = (6, 12)$.

En el grupo de la curva elíptica definida por $y^2 = x^3 + x + 7$ sobre el GF(17) ¿Cuál sería la suma 2P si P= (1, 3)?De las ecuaciones para sumar puntos consigo mismo:

$\lambda = [(3x_P^2 + a) / (2y_P)] \bmod p = [(3+1) / (2*3)] \bmod 17 = [(4)*6^{-1}] \bmod 17$

$\lambda = (4*6^{-1}) \bmod 17 = (4*3) \bmod 17 = 12 \bmod 17 = 12$.

$x_R = (\lambda^2 - 2x_P) \bmod p = (144–2) \bmod 17 = 142 \bmod 17 = 6$

$y_R = [-y_P + \lambda (x_P - x_R)] \bmod p = [-3+12*(1–6)] \bmod 17 = -63 \bmod 17 = 5$

Por lo tanto $2P = (6, 5)$.

2.2.3 Curvas elípticas sobre grupos de la forma GF(2^m)

Elementos del grupo GF(2^m) son cadenas de m-bit, las reglas para la aritmética en GF(2^m) pueden ser definidas por una representación polinominal o representación normal óptima de las bases. Ya que GF(2^m) funciona con cadenas de bits, las computadoras pueden formar aritméticas del campo muy eficientes.

Una curva elíptica con un campo subyacente se forma escogiendo los elementos de "a" y "b" de GF(2^m) (sólo en las condiciones en que el parámetro b de la curva no es 0),entonces el campo GF(2^m) tiene una base de 2, la ecuación de la curva elíptica se puede ajuntar a una representación binaria:

$$y^2 + xy = x^3 + ax^2 + b$$

La curva elíptica incluye todos los puntos (x,y) que satisfacen a la ecuación de la curva sobre el campo GF(2^m) (donde "x" y "y" son elementos de GF(2^m)). Un grupo de curvas elípticas sobre GF(2^m) consiste en los puntos que están sobre la curva, junto con el ya conocido punto al infinito o punto cero0.

Un ejemplo de una curva elíptica sobre el campo de GF(2^m).

Para el ejemplo se considera el campo GF(2^4), definido usando la representación polinómica con el polinomio irreducible f (x) = x^4 + x + 1.El elemento g = (0010) es un generador para el campo.

g^0= (0001), g^1= (0010), g^2= (0100), g^3= (1000), g^4= (0011), g^5= (0110), g^6= (1100), g^7= (1011), g^8= (0101), g^9= (1010), g^{10}= (0111), g^{11}= (1110), g^{12}= (1111), g^{13}= (1101), g^{14}= (1001), g^{15}= (0001).

En un uso criptográfico verdadero, el parámetro m debe ser bastante grande para imposibilitar la generación de la tabla que haga que el criptosistema pueda ser roto. Hoy en día, m = 160 es una opción conveniente. Las tablas permiten el uso de generadores de notación (g^e) las cuales son usadas en el siguiente ejemplo. Se utiliza la notación del generador que permite la multiplicación sin referencia al polinomio irreducible:

$$f(x) = x^4 + x + 1$$

Considere la curva elíptica $y^2 + xy = x^3 + g^4x^2 + 1$. Aquí a= g^4y b= g^0=1. El punto (g^5,g^3) satisface la ecuación sobre GF(2^m):

$$y^2 + xy = x^3 + g^4x^2 + 1$$

$$(g^3)^2 + g^5g^3 = (g^5)^3 + g^4g^{10} + 1$$

$$g^6 + g^8 = g^{15} + g^{14} + 1$$

$$(1100) + (0101) = (0001) + (1001) + (0001)$$

$$(1001) = (1001)$$

Los 15 puntos que satisfacen esta ecuación son:(1, g^{13}), (g^3, g^{13}), (g^5, g^{11}), (g^6, g^{14}), (g^9, g^{13}), (g^{10}, g^8), (g^{12}, g^{12}), (1, g^6), (g^3, g^8), (g^5, g^3), (g^6, g^8), (g^9, g^{10}), (g^{10}, g),(g^{12},

0) (0, 1). Esos puntos son graficados y se muestran en la figura 2.8de la siguiente forma:

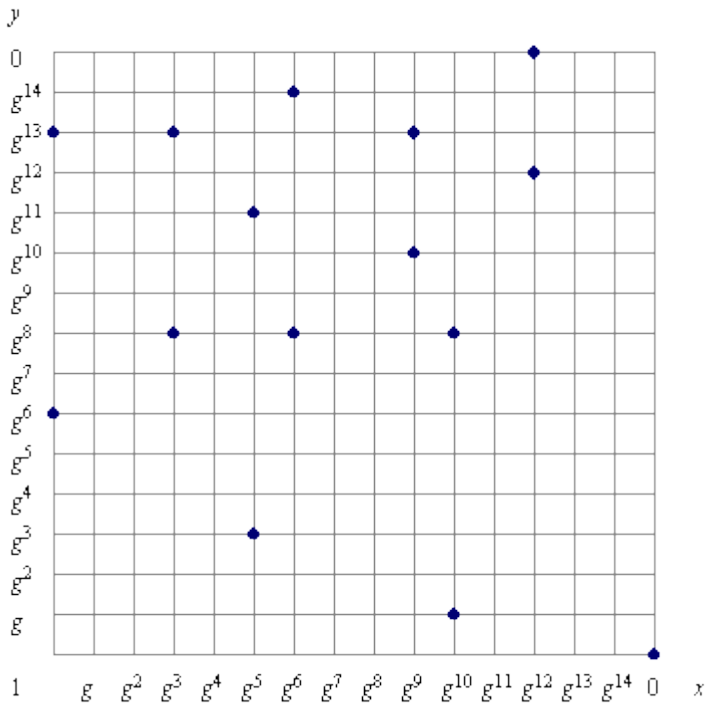


Figura 2.8 Puntos que satisfacen la ecuación $y^2 + xy = x^3 + g^4x^2 + 1$ sobre GF(2^m)

El grupo de curvas elípticas sobre el campo de GF(2^m) tienen un número finito de puntos y su aritmética involucra que no haya errores de redondeo, gracias a esto los campos que se acaban de definir permiten su utilización en aplicaciones de hardware, en donde se instala el algoritmo al diseñar algún dispositivo.

Las reglas algebraicas para la adición se muestran a continuación. Para todos los puntos P, Q pertenecientes a $E_2{}^m(a, b)$:

P + 0 = P

Si P = (x_P, y_P), entonces P + (x_P,x_P +y_P) = 0. El punto (x_P, x_P +y_P) es el negativo de P, se denota como –P.

Si $P = (x_P, y_P)$ y $Q = (x_Q, y_Q)$ con $P \neq -Q$, y $P \neq Q$ entonces $R=P+Q$ es determinada por las siguientes ecuaciones:

$$\lambda = (y_P - y_Q) / (x_P + x_Q)$$

$$x_R = \lambda^2 + \lambda + x_P + x_Q + a$$

$$y_R = \lambda (x_P + x_R) + x_R + y_P$$

Si $P = (x_P, y_P)$, entonces $R=2P=P+P$ se determina bajo las siguientes ecuaciones:

$$\lambda = x_P + (y_P / x_P)$$

$$x_R = \lambda^2 + \lambda + a$$

$$y_R = x_p^2 + (\lambda + 1) * x_R$$

2.2.4 El problema del logaritmo discreto en Curvas elípticas

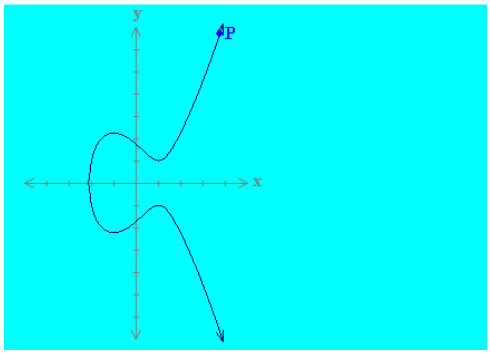
En la construcción de cada uno de los criptosistema esta un problema matemático fuerte que aún utilizando herramientas computacionales es difícil de resolver. El problema del logaritmo discreto es la base de seguridad de muchos criptosistemas incluido el algoritmo de criptografía de curvas elípticas. Siendo más específicos, el cifrado de curvas elípticas confía en la dificultad del Problema del Logaritmo Discreto para Curvas Elípticas (ECDLP por sus siglas en inglés).

Como se ha podido observar existen dos operaciones básicas que son la adición de puntos y la suma de un punto consigo mismo. Si se selecciona un punto en un grupo elíptico de una curva, se puede sumar consigo mismo para obtener el punto 2P.Después de esto, se puede sumar el punto P al punto 2P para obtener el punto 3P.

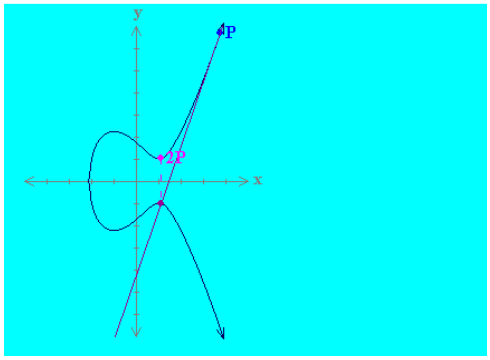
La determinación de un kP del punto de este modo se refiere como multiplicación escalar de un punto. El problema del logaritmo discreto para curvas elípticas se basa sobre la interactividad de los productos escalares de la multiplicación.

Multiplicación escalar.

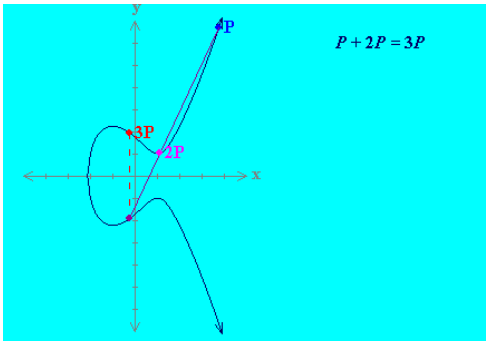
Comúnmente se utiliza la notación de suma para describir un grupo elíptico de la curva, una cierta excepción es proporcionada cuando se utiliza la notación de multiplicación. Específicamente, se considera la operación llamada "multiplicación escalar" debajo de la notación de suma, es decir, kP que se calcula agregando junto a las copias de k del punto P. Usando la notación de multiplicación escalar, la cual consiste en el multiplicar juntas las copias de k del punto P, obteniendo el punto P+P+P+P+…+P = kP. La figura 2.9 muestra una representación gráfica de sumar un mismo punto k veces.



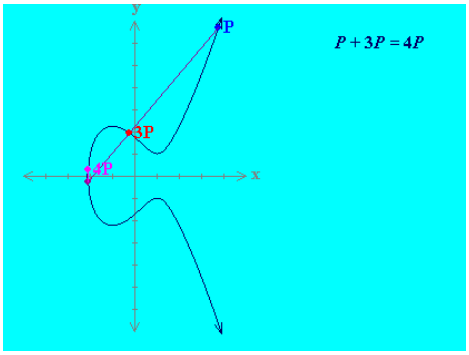
(a) Punto P



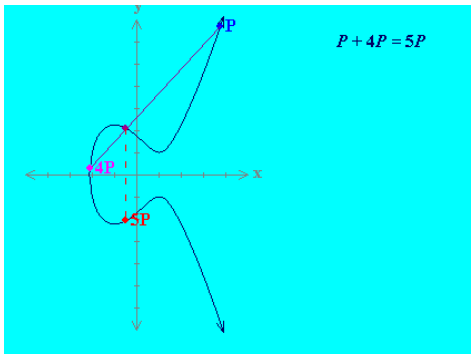
(b) Suma de P+P=2P



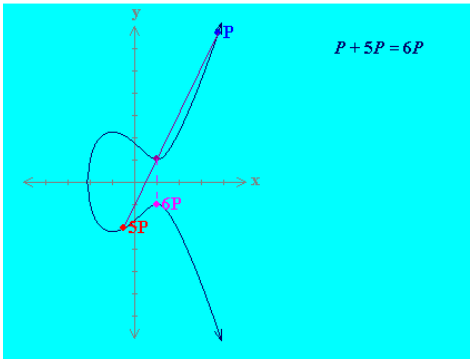
(c)Suma de 2P+P=3P



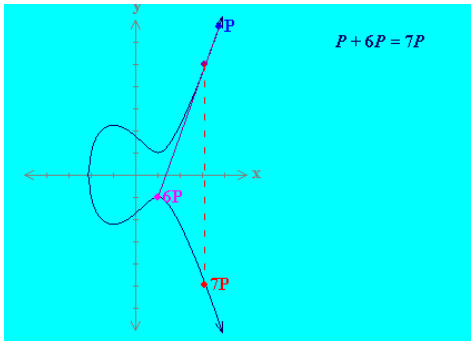
(d) Suma de 3P+P=4P



(e) Suma de 4P+P=5P



(f) Suma de 5P+P=6P



(g) Suma de 6P+P=7P

Figura 2.9 Con una curva elíptica de ecuación y2 = x3 – 3x +3 se observa como se multiplica un punto, de tal forma que la operación 7*P es equivalente a hacer la suma de P+P+P+P+P+P+P.

2.3 Aplicación de las Curvas Elípticas a la Criptografía

Para formar un sistema criptográfico utilizando curvas elípticas, se necesita encontrar como factorizar el producto de 2 primos o de obtener el logaritmo discreto.

En el grupo multiplicativo de(Gp, *), el problema del logaritmo discreto es obtener los elementos r y q de los grupos, y un primo de p, para encontrar un número k talque r=k*q mod p. Si, por otro lado, el grupo de curvas elípticas se describe utilizando la notación multiplicativa, entonces el problema del logaritmo discreto para curvas elípticas es dar los puntos P y Q en un grupo, encontrando un número k tal que k*P=Q. A k se la llama el logaritmo discreto Q para la base P. Cuando el grupo de curva elíptica es descrito por la notación aditiva entonces el problema del logaritmo discreto para curvas elípticas es: dados los puntos P y Q en el grupo, encontrar un número k tal que k*P = Q.

Ejemplo

En el grupo de curvas elípticas definido por$y^2 = x^3 + 9x + 17$ sobreGF(23), ¿cuál es el logaritmo discreto k de Q = (4,5) en la base P = (16,5)?

Un camino para encontrar k es hacer una multiplicación con ayuda de la computadora del punto P hasta que sea igual a Q. Las primeras multiplicaciones de P serían:

P = (16,5), 2P = (20,20), 3P = (14,14), 4P = (19,20), 5P = (13,10), 6P = (7,3), 7P = (8,7), 8P = (12,17) y 9P = (4,5).

Pero se tiene que 9P = (4,5) = Q, el logaritmo discreto de Q en la base P es k = 9.En aplicaciones reales k es demasiado grande que seria imposible calcularla de esta forma.

2.3.1 Obtención de múltiplos de puntos

Se puede ver que sumar un mismo punto k veces sería lo mismo que hacer un ataque de fuerza bruta, es decir, el tener un generador y multiplicarlo k veces consumiría muchos recursos y no se tendría un proceso a tiempo, por esto se describe a continuación un procedimiento para el cálculo de múltiplos puntos en una curva. Por lo cual se utiliza curvas elípticas del tipo:

$$y^2 = x^3 + ax + b$$

Definidas sobre un campo GF(p) con p un número primo. Con esto se tiene un punto G que pertenece a la curva elíptica sobre este campo y que puede generar otros puntos que también pertenezcan a dicha curva, esto siguiendo la definición de campo vista al principio del capítulo, se tiene que un punto P que es un múltiplo del mismo dado P=kG. Se tiene que k es un entero que pertenece al GF(p).

Con lo anterior se puede utilizar el procedimiento de izquierda a derecha, el cual realiza el cálculo del punto P=k*G partiendo de la representación binaria del entero k. Al ser k un entero se procede a representarlo en su forma binaria de la forma:

$$k = (k_{n-1}, k_{n-2}, ..., k_1, k_0)$$

Con esto se utiliza el punto cero P = (0,0) como inicio y para cada i = n-1, n-2, …, 1, 0 se hace lo siguiente:

1. Haciendo P = 2P, es decir, se obtiene la multiplicación escalar por dos del punto.

2. Si k_i =1entonces se hace P = P + G.

3. Si i es igual diferente de cero, se regresa al paso 1, pero si i es igual a uno se tiene la suma del punto k veces.

Ejemplo

Considerando la curva elíptica $y^2= x^3 + 56x + 74$ definida sobre GF(83) la cual también se puede representar con la notación $E_{83}(56, 74)$ la que indica el valor de p=83 y seleccionando a G = (19, 64), un punto de la misma. Se desea obtener el punto P = 75*G; se tiene que el número en base decimal 75 es igual a 1001011 en base decimal. En la tabla 2.10 se anotan los resultados.

I	n_i	P	G
7	-	(0, 0)	0G
6	1	2G+G=2(0, 0)+(19, 64)=(19, 64)	G=(19, 64)
5	0	2(G)=2(19, 64)=(56, 25)	2G=(56, 25)
4	0	2(2G)=2(56, 25)=(71, 74)	4G=(71, 74)
3	1	2(4G)+G=2(71, 74)+(19, 64)=(47, 17)	9G=(47, 17)
2	0	2(9G)=2(47, 17)=(82, 73)	18G=(82, 73)
1	1	2(18G)+G=2(82, 73)+(19, 64)=(57, 76)	37G=(57, 76)
0	1	2(37G)+G=2(57, 76)+(19, 64)=(13, 29)	75G=(13, 29)

Tabla 2.10 Resultado de sumar 75 veces el punto G = (19, 64) sobre la curva elíptica $y^2=x^3+56x+74$ utilizando el método de izquierda a derecha.

Por lo tanto el punto P = 75*(19, 64) = (13, 29). Con este método se ahorra cómputo y hace que sea factible el encontrar un punto multiplicado por un escalar mucho mayor.

3.3.2 Cálculo del orden de la curva

En algunos de los problemas prácticos que se plantean a la hora de trabajar con curvas elípticas en criptografía, es necesario saber el orden de la curva con la que se está trabajando, es decir el número de puntos que contiene una curva elíptica en un campo de Galois determinado.

El orden de una curva elíptica es un parámetro importante en el cifrado de los mensajes, ya que es precisamente el orden de la curva el que determina la estructura del grupo abeliano formado por los puntos de la misma curva. Por lo tanto, el saber el orden de una curva elíptica es necesario para la implementación de la mayoría de los algoritmos de cifrado que se basen en este sistema. Su obtención es laboriosa, sobre todo cuando se trabaja en campos de dimensiones elevadas, aunque es necesario para garantizar su seguridad.

Así el número de puntos de una curva elíptica E(x, y) definida sobre un cuerpo finito GF(p) con un primo mayor a 3 es:

$$N = 1 + \sum_{x \in GF(p)} \{[x^3 + Ax + B) / p] + 1\}$$

Como puede observarse el cálculo de la ecuación es viable siempre y cuando p no sea un primo demasiado grande (considérese números de al menos 20 dígitos), en caso contrario, el cálculo del orden de la curva es una tarea muy laboriosa. El teorema de Weil permite el cálculo del número de puntos N de una curva elíptica E(x, y) sobre GF(p) a partir del número de puntos N' de la misma curva definida sobre GF(p) siendo $q=p^m$ con p primo y m entero. Esto se puede llevar a cabo ya que:

$$N = p^m + 1 - \alpha^m - \beta^m$$

Aquí α y β son dos números complejos, son raíces de la ecuación de segundo grado dada por:

$$x^2 + (N' - p - 1)x + p = 0$$

Este teorema se utiliza por ejemplo para el cálculo del número de puntos de curvas elípticas definidas en cuerpos de base 2 de la forma GF(2^m) a partir del número de puntos de la misma curva definida en GF($2^{m'}$) con m' mucho menor que m.

La curva $y^2+xy = x^3 + x^2 +1$ tiene orden N'=2 si se define en GF(2) es decir p=2, ya que sus únicos puntos en este campo son (0, 1) y el punto en el infinito O=(0, 0). En consecuencia se tiene que (N'–p–1)=-1 con estas condiciones suponiendo que se desea calcular en orden de la curva definida sobre el campo GF(2^8) = GF(256), para esto, nos planteamos la ecuación:

$$x^2 - x + 2 = 0$$

Y se obtienen las raíces dadas por:

X1=α=.5+1.322287**i**

X2=β=.5-1.322287**i**

Y se tiene como resultado:

$$N = 2^8 + 1 - (.5 + 1.322287i)^8 - (.5 - 1.322287)^8 = 288$$

Entonces la curva $y^2+xy= x^3 + x^2 +1$ definida sobre GF(256) tiene 288 puntos.

Es igualmente necesario que, una vez obtenido el orden de la curva, se calcule la factorización del mismo, para de este modo podamos determinar la estructura del grupo de puntos de la curva. La factorización del orden de la curva permitirá efectuar la búsqueda de los posibles puntos generadores del grupo.

3.3.3 Obtención de generadores

Un generador de una curva elíptica es un punto de la misma que al ser multiplicado por números enteros se obtienen todos los puntos que pertenecen a dicha curva. Los generadores son muy importantes para los algoritmos basados en curvas elípticas, ya que gracias a éstos se puede encontrar otros puntos sobre una curva elíptica que se halla definido. Para facilitar la obtención de éstos es conveniente que el número de puntos de la curva sea de la forma N=k*s, con s un número primo muy grande y k un valor entero pequeño mucho menor que s. El grupo finito E(x, y) en GF(p) con p primo de los puntos de la curva es cíclico y tiene φ(N) puntos generadores de orden N. Por lo tanto, en este caso la proporción τ de generadores (sin contar el punto cero) en relación al número total de elementos del grupo viene dada por:

$$\tau = \frac{\varphi(N)}{N-1} = \frac{\varphi(k)\varphi(s)}{ks-1} = \frac{\varphi(k)(s-1)}{ks-1} \cong \frac{\varphi(k)}{k}$$

Puesto que k es un valor entero pequeño, el valor φ(k) es tan sólo un poco menor a k, lo que significa que existe un número suficientemente elevado de generadores como para que su búsqueda no resulte demasiado larga.

Si el número de puntos de la curva contiene un factor primo grande s, se asegura la existencia de un subgrupo de puntos de trabajo lo suficientemente grande como para garantizar la dificultad de resolución del problema del logaritmo elíptico, o lo que es lo mismo, para garantizar la seguridad del sistema de cifrado.

3.3.4 Intercambio de claves secretas

El logaritmo elíptico puede ser utilizado como función de una sola dirección, también llamados unidireccionales, para el intercambio de claves a través de canales inseguros, de tal forma que es análogo al procedimiento de intercambio propuesto por Diffie-Hellman.

Para realizar el intercambio de claves utilizando las curvas elípticas, se parte de una curva elíptica definida sobre GF(p) como $E_q(a, b)$ con q un entero largo el cual es un número primo relativo de p y de un punto base $G = (x_1, y_1)$ de la misma curva que sea un generador de grupo, es decir, un punto cuyo orden sea igual al número de puntos de la curva. Los parámetros $E_q(a, b)$ y G son públicos. Con estas condiciones, considérese que dos usuarios A y B desean intercambiar una clave secreta a través de un canal inseguro. Para ello, A y B realizan lo siguiente:

1. A selecciona un valor entero aleatorio secreto n_A perteneciente a GF(p) y envía a B el punto de la curva $P_A = n_A * G$.

2. De la misma forma B selecciona un valor entero aleatorio secreto n_B perteneciente a GF(p) y envía a A el punto de la curva $P_B = n_B * G$.

3. A genera la clave secreta $K = n_A * P_B$.

4. Por su parte B calcula la clave secreta $K = n_B * P_A$.

Puesto que el grupo de puntos de la curva elíptica es abeliano se verifica que las claves secretas calculadas por A y B son iguales utilizando K para posteriores comunicaciones cifradas. De esta forma, en este procedimiento de intercambio de claves, los puntos P_Ay P_B actúan como claves públicas mientras que los valores enteros n_A y n_B son sus respectivas claves secretas asociadas.

Para romper este esquema, el atacante debe ser capaz de calcular K teniendo G y K*G, lo cual es prácticamente imposible de lograr.

Ejemplo

Considerando la curva elíptica $y^2= x^3 – 4$ definida sobre GF(211) la cual también se puede representar con la notación $E_{211}(0, -4)$ la que indica el valor de p=211 y seleccionando a G=(2, 2); se tiene que:

El usuario A escoge la clave privada n_A=121, la cual es un número primo relativo de 211 ya que su mcd(211, 121) = 1 lo cual es una característica deseada para trabajar en criptografía con los números primos.

- Entonces se tiene que la clave pública de A es P_A= 121(2, 2)= (115, 48). Como se puede ver en la tabla 2.11.

I	n_i	P	G
7	-	(0, 0)	0G
6	1	2G+G=2(0, 0)+(2, 2)=(2, 2)	G=(2, 2)
5	1	2(G)+G=2(2, 2)+(2, 2)=(129, 56)	3G=(129, 56)
4	1	2(3G)+G=2(129, 56)+(2, 2)=(179, 199)	7G=(179, 199)
3	1	2(7G)+G=2(179, 199)+(2, 2)=(28, 2)	15G=(28, 2)
2	0	2(15G)=2(28, 2)=(70, 200)	30G=(70, 200)
1	0	2(30G)=2(70, 200)=(116, 114)	60G=(116, 114)
0	1	2(60G)+G=2(116, 114)+(2, 2)=(115, 48)	121G=(115, 48)

Tabla 2.11 El cálculo de la clave pública del usuario A, la cual es 121 veces el punto G = (2, 2) sobre la curva elíptica $y^2=x^3- 4$.

De igual forma se hace el cálculo de la clave pública para el usuario B.

- La clave privada que selecciona B es $n_B=203$, la cual de la misma forma que A es primo relativo de 211 ya que su mcd(211, 203) = 1. Se muestran los cálculos en la tabla 2.12.

I	n_i	P	G
8	-	(0, 0)	0G
7	1	2G+G=2(0, 0)+(2, 2)=(2, 2)	G=(2, 2)
6	1	2(G)+G=2(2, 2)+(2, 2)=(129, 56)	3G=(129, 56)
5	0	2(3G)=2(129, 56)=(125, 152)	6G=(125, 152)
4	0	2(6G)=2(125, 152)=(155, 96)	12G=(155, 96)
3	1	2(12G)+G=2(155, 96)+(2, 2)=(69, 20)	25G=(69, 20)
2	0	2(25G)=2(69, 20)=(13, 100)	50G=(13, 100)
1	1	2(50G)+G=2(13, 100)+(2, 2)=(1, 182)	101G=(1, 182)
0	1	2(101G)+G=2(1, 182)+(2, 2)=(130, 203)	203G=(130, 203)

Tabla 2.12 El cálculo de la clave pública del usuario B, la cual es 203 veces el punto G = (2, 2) sobre la curva elíptica $y^2=x^3- 4$.

- Se tiene que la clave pública de B es P_B= 203(2, 2)= (130, 203).

Después se calcula la clave secreta o privada que utilizarán ambos usuarios para cifrar y compartir información por medio de un algoritmo simétrico:

- Se tiene que K= 121(130, 203)= 203(115, 48)= (161, 69).

Se puede apreciar que la clave pública está compuesta por un par de números. Si ésta va a ser utilizada como clave convencional de cifrado, entonces se puede usar simplemente la coordenada x o una función simple de la misma.

3.3.5 Codificación y decodificación en curvas elípticas

Antes de que se presenten los algoritmos para cifrar con curvas elípticas, se debe resolver el problema de codificar el mensaje en claro de forma tal que a cada elemento de dicho mensaje le corresponda un punto dentro de la curva que se esta considerando para llevar acabo el cifrado. Además considerando que trabajar con las unidades del mensaje en claro resultaría un poco molesto, primero se realiza una asociación entre cada unidad del mensaje en claro con algún número, el cual debe pertenecer al campo sobre el que se define la curva, considerando que el número de caracteres del alfabeto debe ser menor al número primo sobre el que se define el campo, para de esta forma poder trabajar con números en vez de con unidades o caracteres del mensaje.

Para codificar un mensaje en claro de modo que se obtengan puntos de una curva elíptica determinada, se hace lo siguiente:

Si para cada unidad del mensaje m se verifica que $0<m<M$, se considera un entero h de modo que $q > M*h$, siendo q un número primo o la potencia de un primo y GF(q) el campo finito sobre el que se llevan a cabo las operaciones. Los enteros entre 1 y M*h se escriben de la forma siguiente $m*h + j$, para cada $j= 1, 2, 3,\ldots, h-1$, y así se obtiene una correspondencia uno a uno entre estos enteros y elementos de "x" que pertenecen a la curva elíptica que se escoja la cual esta definida sobre GF(q). Para cada x que se obtenga se calcula el valor de la ecuación de la curva elíptica que se escoja que pertenece a GF(q). Se busca un valor entero para el parámetro "y" que verifique la igualdad de la ecuación que define la misma curva elíptica y si tal valor existe, se tienen las coordenadas del punto sobre la curva $E_m= (x, y)$ que pertenece a la unidad del mensaje m. Si tal valor de "y" no existe, entonces se incrementa de uno en uno el valor de "x" y se repite la búsqueda de "y".

Para decodificar el mensaje cifrado formado por los puntos (x, y), se hace el cálculo para cada uno de los puntos recibidos:

$$m = \frac{x-1}{h}$$

Donde "x" es el entero que corresponde a que m tome un valor entero.

Ejemplo

Supongamos que la curva elíptica es $y^2 = x^3 - x + 188$, y que está definida sobre GF(751). Para codificar el mensaje se utiliza el alfabeto mostrado en la tabla 2.13.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27

Tabla 2.13 Alfabeto para codificar sistemas de Curvas Elípticas.

A continuación se seleccionan M = 36 y h = 20 tomando en cuenta que se debe cumplir q > M*h y considerando que q es el número sobre el cual se define la curva, se tiene q = 751 > M*h = 720.

Ahora, considérese que se desea enviar el mensaje m= "MAR". A continuación se codifica cada una de las unidades del mensaje:
M:13 → con j=1 x= 13*20+1= 261 sustituyendo en $y^2=x^3$ -x +188 da como resultado en el segundo miembro de la igualdad un valor de 334 entonces como 334 =

y^2 para un y=288 entonces el punto correspondiente a la curva para la parte del mensaje "M" será P_M = (261, 288).

A:1 → con j=1 x= 1*20+1= 21 sustituyendo en $y^2=x^3$ -x +188 da como resultado en el segundo miembro de la igualdad un valor de 416 entonces como 416 = y^2 para un y=133 entonces el punto correspondiente a la curva para la parte del mensaje "A" será P_A = (21, 133).

R:19 → con j=1 x= 19*20+1= 381 sustituyendo en $y^2=x^3$ -x +188 da como resultado en el segundo miembro de la igualdad un valor de 255 entonces como 255 = y^2 para un y=69 entonces el punto correspondiente a la curva para la parte del mensaje "R" será P_R = (381, 69).

Por lo tanto se tiene que el mensaje codificado es:

(261, 288), (21, 133), (381, 69)

Por consiguiente para decodificar el mensaje anterior, se llevan a cabo las siguientes operaciones:

$$\frac{261-1}{20}=13\rightarrow M \qquad \frac{21-1}{20}=1\rightarrow A \; \frac{381-1}{20}=19\rightarrow R$$

3.3.6 Ejemplo de cifrado ElGamal con curvas elípticas

Los parámetros del procedimiento son una curva elíptica E definida sobre un campo GF(p) con p un número primo y un punto $G(x_G,y_G)$ de la misma curva que sea generador de grupo. La curva E y el punto $G(x_G,y_G)$ son públicos.

Con tales condiciones, si los usuarios A y B desean intercambiar un mensaje confidencial representado por un punto P de la curva cada usuario debe poseer un sistema de clave pública. El usuario A posee una pareja de claves (a, P_a), con "a" que pertenece a GF(p) que es un valor aleatorio y secreto y $P_a=a*G(x_G,y_G)$ un punto público de la curva E. Del mismo modo el usuario B tiene su pareja de claves (b, P_b), también con "b" que pertenece a GF(p) el cual se considera secreto y $P_b=b*G(x_G,y_G)$ el cual se considera público. Con esto A puede transmitir a B el mensaje confidencial P eligiendo un valor entero aleatorio k que también pertenece a GF(p) y enviándole la pareja de puntos (M, N) con:

$$(M, N) = (kG, P+kP_b) = (kG, P+kbG)$$

Por su lado, el usuario B recupera el punto P utilizando su clave secreta b con la que calcula:

$$P = N–bM$$

Se observa que:

$$P = N–bM = P + kbG – bkG$$

Es fácil darse cuenta de que si un atacante pretendiese vulnerar este sistema de cifrado de clave pública intentando obtener la clave secreta de descifrado "b" a partir de la clave pública de cifrado P_b entonces debería ser capaz de resolver el problema del logaritmo elíptico, ya que ambas claves están relacionadas mediante la ecuación:

$P_b = b*G(x_G, y_G)$

Una de las características más destacables de este procedimiento de cifrado de clave pública es que los cifrados de un mismo mensaje pueden ser diferentes sin más que calcularlos a partir de valores enteros aleatorios k igualmente diferentes.

Ejemplo

Se va a cifrar el mensaje "O" utilizado el ejemplo sobre como codificar un mensaje en curvas elípticas es decir nuestro punto a cifrar seráP_O = (324, 7). Además se considera la curva elíptica $E_{751}(-1, 188)$:$y^2 = x^3 - x + 188$ sobre GF(751) y el punto base G = (680, 94).

Se tiene que el usuario A escoge la clave secreta a=3 por tanto su clave pública será 3G = 3(680, 94) = (697, 279). Su par de claves es:

A={3, (697, 279)}

Por otro lado se tiene que B escoge la clave secreta b=7 por tanto la clave pública para B será 7G = 7(680, 94) = (607, 18). Su par de claves es:

B={7, (607, 18)}

Si el usuario A quiere enviar el mensaje "O" codificado como P_O = (324, 7), elige un numero aleatorio k=11 y calculara el punto de la curva $P_O + kP_b$, es decir:

11G = (393, 710)

y

(324, 7) + 11(607, 18)

o

(324, 7) + (299, 183)

y por tanto envía a B la pareja

{(393, 710), (657, 595)}.

Para recuperar el mensaje el usuario B multiplica el primero de los puntos que recibió de A por su clave privada 7(393,710) = (299, 183) y a continuación resta el punto que se obtuvo al segundo punto recibido:

(657, 595) – (299, 183) = (657, 595) + (299, -183) = (657, 595) + (299, 568) = (324, 7)

Una vez que B sabe cuál es el punto de la curva elíptica se procede a decodificar éste, utilizando lo visto anteriormente se tiene:

$$\frac{324-1}{20} = 16.15 \approx 16 \rightarrow O$$

Ahora B sabe que el usuario A le envía una letra “O”.

Capítulo 3
Implementación de Curvas elípticas

Es evidente ver que el uso de la tecnología hoy en día es algo cotidiano, ha llegado a todos los rincones del mundo, desde un obrero hasta altos ejecutivos; todos ellos se benefician de los avances tecnológicos y ayudan a sus labores diarias, sin embargo, ignoran los riesgos que existen en su uso. Por ello, es el deber de los especialistas en el desarrollo de las nuevas tecnologías el mitigar los riesgos para garantizar seguridad a los usuarios y sus intereses. Existen muchas herramientas que se usan actualmente para brindar seguridad de la tecnología, una de ellas, y ya mencionada, es la criptografía.

El objetivo de la criptografía es el de proporcionar comunicaciones seguras y secretas sobre canales inseguros. Ahora bien, la criptografía no es sinónimo de seguridad. No es más que una herramienta que es utilizada sobre la base de mecanismos de cierta complejidad para proporcionar no solamente protección, sino también garantizar que haya confidencialidad.

3.1 La Criptografía de Curvas elípticas en la vida diaria

Como se ha visto existen muchas teorías matemáticas en las que se basan los algoritmos de cifrado que en la actualidad se usan, como por ejemplo las matemáticas de curvas elípticas. Ahora que se conoce y se entiende la teoría en la que están fundamentadas, y es sencillo poder comprender su funcionamiento, sin embargo el saber cómo funcionan y poder desarrollarlas de forma matemática en una hoja de papel, no reflejan su importancia en la seguridad de las TIC, ni mucho menos beneficia a la sociedad de ninguna manera. Es por esto que es fundamental implementarlas para poder darles un uso positivo, es decir, llevarlas del papel a la realidad.

Para poder hacer la transmisión de datos de manera segura entre dos o más usuarios por medio de dispositivos electrónicos como las computadoras, *laptop*, *netbook* y los hoy tan populares *smartphone*, a través de un medio inseguro, que en este caso es el internet, es conveniente hacer uso de la criptografía de clave pública, ya que está diseñada para este fin.

Existen muchos algoritmos de cifrado de clave pública, que se usan para diversas tareas, sin embargo la cuestión es saber cuál usar y para qué, ya que a pesar de que todos se basan en un mismo concepto, son tan diferentes como un ser humano de otro. Para poder determinar el algoritmo de cifrado que se necesita utilizar, depende mucho de lo que se quiere proteger o mantener secreto, y con qué recursos se cuenta para poder hacerlo.

Por ejemplo, si se quiere mantener secreta una fecha de cumpleaños de un grupo de personas no autorizadas, no es conveniente hacer uso de un algoritmo de cifrado muy complejo y, que requiera el uso de claves extensas, ya que dicha información no es de vital importancia o sensitiva para algún proceso. Por otra parte, si se quiere mantener secreto los códigos de activación para armas nucleares es fundamental hacer uso de los algoritmos de cifrado más robustos y poco vulnerables para garantizar la seguridad. Algo similar ocurre con los requerimientos necesarios para aplicar algunos algoritmos de cifrado, ya que no es lo mismo cifrar una palabra con un algoritmo robusto como AES (*Advanced Encryption Standard*) en una computadora gubernamental a cifrarla en un *smartphone*, los impedimentos físicos o de *Hardware* son un factor evidente al momento de cifrar información.

Muchas personas pueden considerar que no es factible brindar la misma seguridad que se tendría en una potente computadora, que en un dispositivo pequeño como *smartphone* o PDA a causa de sus limitantes en *Hardware*, sin embargo sí es posible hacerlo, gracias a los avances en nuevas teorías matemáticas como son las ya mencionadas curvas elípticas.

Los primeros sistemas de clave pública son seguros si se asume que es difícil factorizar un entero grande compuesto de dos o más factores primos grandes. Por otra parte, para los protocolos basados en la teoría de curvas elípticas, se supone que la búsqueda del logaritmo discreto de un elemento de curva elíptica aleatoria con respecto a un punto base de conocimiento público es inviable.

El tamaño de la curva elíptica determina la dificultad del problema. El principal beneficio prometido por la criptografía de curvas elípticas es un tamaño de la clave más pequeña, lo que reduce los requisitos de almacenamiento y la transmisión, es decir, que un grupo de curvas elípticas podría proporcionar el mismo grado de la seguridad proporcionada por un sistema basado en RSA (*Rivest, Shamir y Adleman*) con un gran módulo y correspondientemente a una mayor clave por ejemplo, una clave pública ECC (*Eliptic Curve Cryptography*) de 256 bits debe proporcionar una seguridad comparable a una clave pública RSA de 3.072 bits.

Para poder apreciar los beneficios que tiene el hacer uso de algoritmos de cifrado basados en la teoría de curvas elípticas, es necesario aplicarlo de alguna manera a la vida diaria, es decir, implementarlo ya sea en *Software* o en hardware para poder ver su funcionamiento y así brindar confidencialidad a la información.

Un ejemplo de implementación de la criptografía de curvas elípticas en hardware es en las muy populares "*smart card*"[1] o tarjetas inteligentes, que son utilizadas para los "cuentahabientes" de los bancos. Estas tarjeas contienen un pequeño chip que está compuesto de una circuitería encargada de generar el algoritmo cada vez que se desliza sobre un lector de tarjetas inteligentes para poder comprobar la autenticidad de la tarjeta y del usuario.

[1] Nota: Según la "*Smart Card Allience*", una tarjeta inteligente (*smart card*), o tarjeta con circuito integrado (TCI), es cualquier tarjeta del tamaño de un bolsillo con circuitos integrados que permiten la ejecución de cierta lógica programada.

Esto surgió como resultado del gran número de usuarios que eran víctimas de la clonación de sus tarjetas de crédito.

Por otra parte, también se puede hacer una implementación de la criptografía de curvas elípticas por medio del *Software*, de hecho, todos los algoritmos criptográficos tienen su mayor usabilidad en el *Software*, ya que gracias a la gran gama de lenguajes de programación que existe en la actualidad, es posible implementarlos sencillamente.

Para poder hacer una implementación de un algoritmo criptográfico que va a ser utilizado en algún tipo de aplicación de *Software* para dar seguridad, se deben de seguir estándares y metodologías que sugieren el uso apropiado de dicho algoritmo, ya que al depender de él, la seguridad de algún tipo de información es fundamental la buena implementación de cualquier algoritmo, pues como se ha visto hasta ahora no todos los criptosistemas[2] ofrecen la misma seguridad y no todos nos vulnerables a ciertos ataques.

[2] Nota: "Segu-Info" lo define como la quíntupla (m,C,K,E,D), donde:

- ✓ m representa el conjunto de todos los mensajes sin cifrar (texto plano).
- ✓ C Representa el conjunto de todos los posibles mensajes cifrados.
- ✓ K representa el conjunto de claves que se pueden emplear en el Criptosistema.
- ✓ E es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de m para obtener un elemento de C.
- ✓ D es el conjunto de transformaciones de descifrado.

3.2 Curvas elípticas recomendadas por estándares internacionales

En julio de 1999 el gobierno de los Estados Unidos de América publicó un artículo en el cual se recomienda cierta colección de curvas elípticas para uso del gobierno federal y contiene opciones de longitud de la clave privada y campos subyacentes.

- Un primer campo es el campo GF (p) que contiene un número primo p de elementos. Los elementos de este campo son los enteros módulo p, y el campo de la aritmética se implementa en términos de la aritmética de enteros módulo p.

- Un campo binario es el campo GF (2m), que contiene algunos elementos para 2m, siendo m el grado del campo. Los elementos de este campo son las cadenas de bits de longitud m, y la media aritmética de campo se aplica en términos de las operaciones en los bits.

En la tabla 3.1 se muestra la comparación de los campos con los tamaños de claves de algunos algoritmos de cifrado de acuerdo con el Gobierno de Estados Unidos.

Tabla 3.1 Cuadro comparativo de campos Binario y Primo (NIST 2001)

Clave	Algoritmo	Primo	Binario
80	SKIPJACK	$\|\|p\|\| = 192$	$m = 163$
112	Triple-DES	$\|\|p\|\| = 224$	$m = 233$
128	AES Small	$\|\|p\|\| = 256$	$m = 283$
192	AES Medium	$\|\|p\|\| = 384$	$m = 409$
256	AES Large	$\|\|p\|\| = 521$	$m = 571$

Para utilizar criptografía de curvas elípticas, todas las partes deben ponerse de acuerdo sobre todos los elementos que definen la curva elíptica, es decir, los parámetros de dominio del sistema. El campo se define por “p” en el campo de los primos y el par de “m” y “f” en el campo binario. La curva elíptica se define por las constantes “A” y “B” utilizados en su ecuación de definición. Por último, el subgrupo cíclico se define por su generador de “G”. Para aplicación criptográfica el orden de G, que es el menor número no negativo “n” tal que, normalmente es primordial. Dado que “n” es el tamaño de un subgrupo de la misma sigue del teorema de Lagrange que el número es un número entero. En aplicaciones criptográficas este número “h”, llamado el cofactor, debe ser pequeño preferentemente.

A menos que haya una garantía de que los parámetros de dominio se generaron por una parte de confianza con respecto a su uso, los parámetros de dominio deben ser validados antes de su uso.

La generación de los parámetros de dominio no se hace generalmente por cada participante ya que este consiste en contar el número de puntos de una curva, que es mucho tiempo y es problemático para poner en práctica.

Como resultado de varios organismos de normalización publicaron parámetros de dominio de las curvas elípticas de varios tamaños de los campos comunes. Tales parámetros de dominio son comúnmente conocidos como “Curvas estándar" o "Curvas denominadas"; una curva con nombre puede hacer referencia por su nombre o por el identificador de objeto único definido en los documentos estándar:

- ✓ NIST, curvas elípticas recomendadas para Uso Gubernamental.

- ✓ SECG, SEC 2: Parámetros de dominio curvas elípticas recomendadas.

- ✓ Brainpool ECC, ECC Brainpool Curvas estándar y generación de curvas.

Si se quiere construir los propios parámetros de dominio se debe seleccionar el campo base y utilizar una de las siguientes estrategias para encontrar una curva con el número apropiado de puntos utilizando uno de los siguientes métodos:

1. Seleccionar una curva al azar y el uso de un algoritmo general de conteo de punto, por ejemplo, el algoritmo de Schoof o algoritmo Schoof-Elkies-Atkin,

2. Seleccionar una curva al azar de una familia, que permite un fácil cálculo del número de puntos de dicha curva, por ejemplo las curvas de Koblitz.

3. Seleccionar el número de puntos y generar una curva con este número de puntos utilizando la técnica de la multiplicación compleja.

Varias clases de curvas son deficientes y deben evitarse:

1. Las curvas sobre $\mathbb{F}_{2^m}$ con no primos "m", son vulnerables a los ataques de descenso de Weil.

2. Las curvas tales que "n" divide $p^B - 1$ (dónde p es el campo característico -q para un campo primo ó 2 para un campo binario) por un B suficientemente pequeño; son vulnerables a los ataques de MOV (*Menezes-Okamoto-Vanstone*) que se aplica siempre al DLP (*Discrete Logarithm Problem*) en un pequeño grado de extensión del campo $\mathbb{F}_p$ para resolver ECDLP (*Elliptic Curve Discrete Logarithm Problem*). El límite B debe ser elegido de modo que los logaritmos discretos en el campo $\mathbb{F}_{p^B}$ son al menos tan difícil de calcular como los logaritmos discretos sobre la curva elíptica.

3. Las curvas tales que $|E(\mathbb{F}_q)| = q$ son vulnerables al ataque que mapea los puntos en la curva para el grupo aditivo de $\mathbb{F}_q$.

El NIST recomienda quince curvas elípticas. Específicamente, FIPS 186-3 tiene diez campos finitos recomendados:

1. Cinco campos principales para ciertos números primos "p" de tamaños de 192, 224, 256, 384, y 521 bits. En cada uno de los campos principales se recomienda una curva elíptica.

2. Cinco campos binarios para m igual 163, 233, 283, 409, y 571 para cada uno de los campos binarios, una curva elíptica y una curva de Koblitz fue seleccionada.

La recomendación del NIST contiene, pues, un total de cinco curvas principales y diez curvas binarios. Se escogieron las curvas de la seguridad y la eficiencia óptima aplicación.

3.2.1 Curvas sobre el campo de los primos

De acuerdo con el gobierno de Estados Unidos de América en Julio de 1999, para cada primo p, hay una curva pseudo aleatoria **E: y2 ≡ x3 - 3x + b (mod p)** de primer orden "r". Por lo tanto, para estas curvas el cofactor siempre es f=1, y se dan los siguientes parámetros:

- ✓ El módulo primo p
- ✓ El orden de r
- ✓ La semilla de entrada "s" de 160 bits para el algoritmo base SHA-1
- ✓ La salida c del algoritmo base SHA-1
- ✓ El coeficiente b (satisfaciendo b2, es decir, c ≡ -27 (mod p))
- ✓ La base del punto de coordenadas x Gx
- ✓ La base de punto y coordenadas Gy

Los números enteros p y r se dan en forma decimal; cadenas de bits y los elementos de campo se dan en hexadecimal. Como ejemplo se tiene la curva P-192.

p = 6277101735386680763835789423207666416083908700390324961279

r = 6277101735386680763835789423176059013767194773182842284081

s = 3045ae6f c8422f64 ed579528 d38120ea e12196d5

c = 3099d2bbbfcb2538 542dcd5f b078b6ef 5f3d6fe2 c745de65

b = 64210519e59c80e7 0fa7e9ab 72243049 feb8deec c146b9b1

G_x = 188da80eb03090f6 7cbf20eb 43a18800 f4ff0afd 82ff1012

G_y = 07192b95ffc8da78 631011ed 6b24cdd5 73f977a1 1e794811

3.2.2 Curvas sobre el campo de los binarios

Al igual que el campo de los números primos, el gobierno de Estados Unidos de América, propone para el campo de los números binarios que para cada campo de grado m, se da una curva pseudo aleatoria, junto con una curva de Koblitz. La curva pseudo aleatoria tiene la forma:

$$E: y^2 + xy = x^3 + x^2 + b$$

La curva de Koblitz tiene la forma:

$$E_a: y^2 + xy = x^3 + ax^2 + 1$$

donde a = 0 ó 1.

Para cada curva, el cofactor es f=2. El cofactor de cada curva de Koblitz es:

f=2, si a=1 y f=4, si a=0.

Los coeficientes de las curvas pseudo aleatorias, y las coordenadas de los puntos de base de los dos tipos de curvas, se dan en términos tanto del polinomio como de las representaciones bases normales. Para cada m, se dan los siguientes parámetros:

Campo de representación:

- El tipo de base normal T.
- El polinomio de campo (un trinomio o pentanomio).

La curva de Koblitz:

- El coeficiente a.
- El punto base de orden r.
- El punto base x en la coordenada Gx.
- El punto base y en la coordenada Gy.

La curva pseudo aleatoria:

- El orden de punto de base r.

Curva pseudo aleatoria (representación Base polinómica):

- El coeficiente b.
- El punto base x en la coordenada Gx.
- El punto base y en la coordenada Gy.

Curva pseudo aleatoria (representación en Base normal):

- La entrada s de la semilla de 160-bit para el algoritmo base SHA-1.
- El coeficiente b (es decir, la salida del algoritmo base SHA-1).
- El punto base x en la coordenada Gx.
- El punto base y en la coordenada Gy.

Los números enteros (como T, m, y r) se dan en forma decimal; cadenas de bits y elementos de campo se dan en hexadecimal. Como ejemplo se tiene la curva de grado 163 en el Campo Binario.

$T = 4$

$p(t) = t^{163} + t^{7} + t^{6} + t^{3} + 1$

Curva K-163

$a = 1$

$r = 5846006549323611672814741753598448348329118574063$

Base polinómica:

$G_x =$ 2 fe13c053 7bbc11ac aa07d793 de4e6d5e 5c94eee8

$G_y =$ 2 89070fb0 5d38ff58 321f2e80 0536d538 ccdaa3d9

Base normal:

$G_x =$ 0 5679b353 caa46825 fea2d371 3ba450da 0c2a4541

$G_y =$ 2 35b7c671 00506899 06bac3d9 dec76a83 5591edb2

Curva B-163

$r = 5846006549323611672814742442876390689256843201587$

Base polinómica:

$b =$ 2 0a601907 b8c953ca 1481eb10 512f7874 4a3205fd

$G_x =$ 3 f0eba162 86a2d57e a0991168 d4994637 e8343e36

$G_y =$ 0 d51fbc6c 71a0094f a2cdd545 b11c5c0c 797324f1

Base normal:

$s =$ 85e25bfe 5c86226c db12016f 7553f9d0 e693a268

$b =$ 6 645f3cac f1638e13 9c6cd13e f61734fb c9e3d9fb

$G_x =$ 0 311103c1 7167564a ce77ccb0 9c681f88 6ba54ee8

$G_y =$ 3 33ac13c6 447f2e67 613bf700 9daf98c8 7bb50c7f

3.3 Programación de Curvas elípticas

Como se mencionó anteriormente, la implementación de la criptografía se puede realizar por medio del *Software*, es decir, en algún lenguaje de programación.

Entre todos los lenguajes de programación existentes, se utilizó Java para este trabajo de tesis, el cual es un lenguaje orientado a objetos y que para la implementación de algoritmos de cifrado puede ser muy práctico y didáctico.

En la página de Oracle, se puede encontrar que Java es la base de casi todos los tipos de aplicaciones en red y el estándar global para el desarrollo y suministro de aplicaciones móviles, juegos, contenido basado en web y software de empresa. Con más de nueve millones de desarrolladores en todo el mundo, Java permite desarrollar y desplegar de un modo eficiente interesantes aplicaciones y servicios. Con un conjunto integral de herramientas, un ecosistema maduro y un sólido rendimiento, Java ofrece portabilidad de aplicaciones incluso entre los entornos informáticos más dispares. Las características y ventajas más importantes y destacadas se presentan en la tabla 3.2, según la página oficial de Oracle.

Tabla 3.2 Cuadro comparativo de características y ventajas de Java (página web Oracle).

Característica	**Ventaja**
Independencia de la plataforma	Java funciona con las principales plataformas de hardware y sistemas operativos, o bien con el software JVM directamente desde Oracle, a través de uno de los muchos *partners* del ecosistema de Java, o como parte de la comunidad OpenJDK.
Alto rendimiento	*HotSpot y JRockit* son ejemplos de tecnologías de equipos virtuales de interpretación dinámica (JIT) y de eficacia probada que hacen de Java uno de los entornos de programación más rápidos. Las optimizaciones integradas para entornos multiproceso lo hacen aún más rápido.
Fácil de aprender	El modelo de Java para la gestión de la memoria, los procesos múltiples y la gestión de excepciones lo convierte en un lenguaje eficaz para los desarrolladores nuevos y para los más experimentados.
Basado en estándares	El lenguaje Java y la tecnología relacionada evolucionan a través de Java *Community Process*, un mecanismo que permite desarrollar especificaciones técnicas para la tecnología Java.

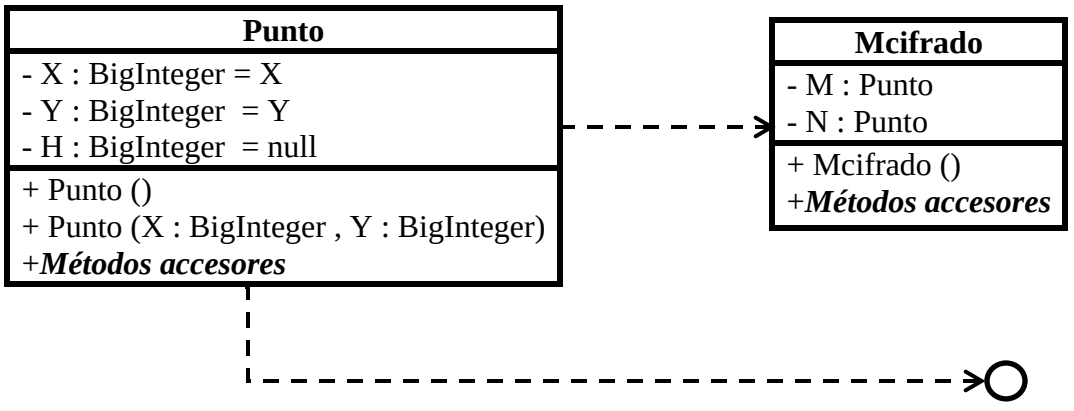
Prevalencia mundial	Java es la plataforma de aplicaciones más popular del planeta y proporciona un interesante ecosistema de desarrolladores impulsado por herramientas eficaces, libros, bibliotecas, muestras de código y mucho más.
Entornos de ejecución coherentes	Java permite realizar despliegues con confianza con entornos de tiempo de ejecución que van de Java SE en equipos de sobremesa a Java SE *for Embedded Devices* y Oracle Java *Micro Edition Embedded Client.*
Optimizado para los dispositivos integrados	Java SE *for Embedded Devices* incluye compatibilidad con requisitos clave, como la compatibilidad con procesadores integrados, la gestión de potencia, los despliegues con huella pequeña y mucho más. Oracle Java ME *Embedded Client* se basa en *Connected Device Configuration* (CDC), un subconjunto de la plataforma Java SE, y proporciona rendimiento Java para los dispositivos con recursos restringidos.
Aplicaciones portátiles con alto rendimiento	Java alcanza un rendimiento nativo y proporciona portabilidad en una amplia gama de procesadores y sistemas operativos integrados.
Modelo con seguridad probada	Java ofrece un entorno de aplicaciones avanzado con un alto nivel de seguridad que es idóneo para las aplicaciones de red.

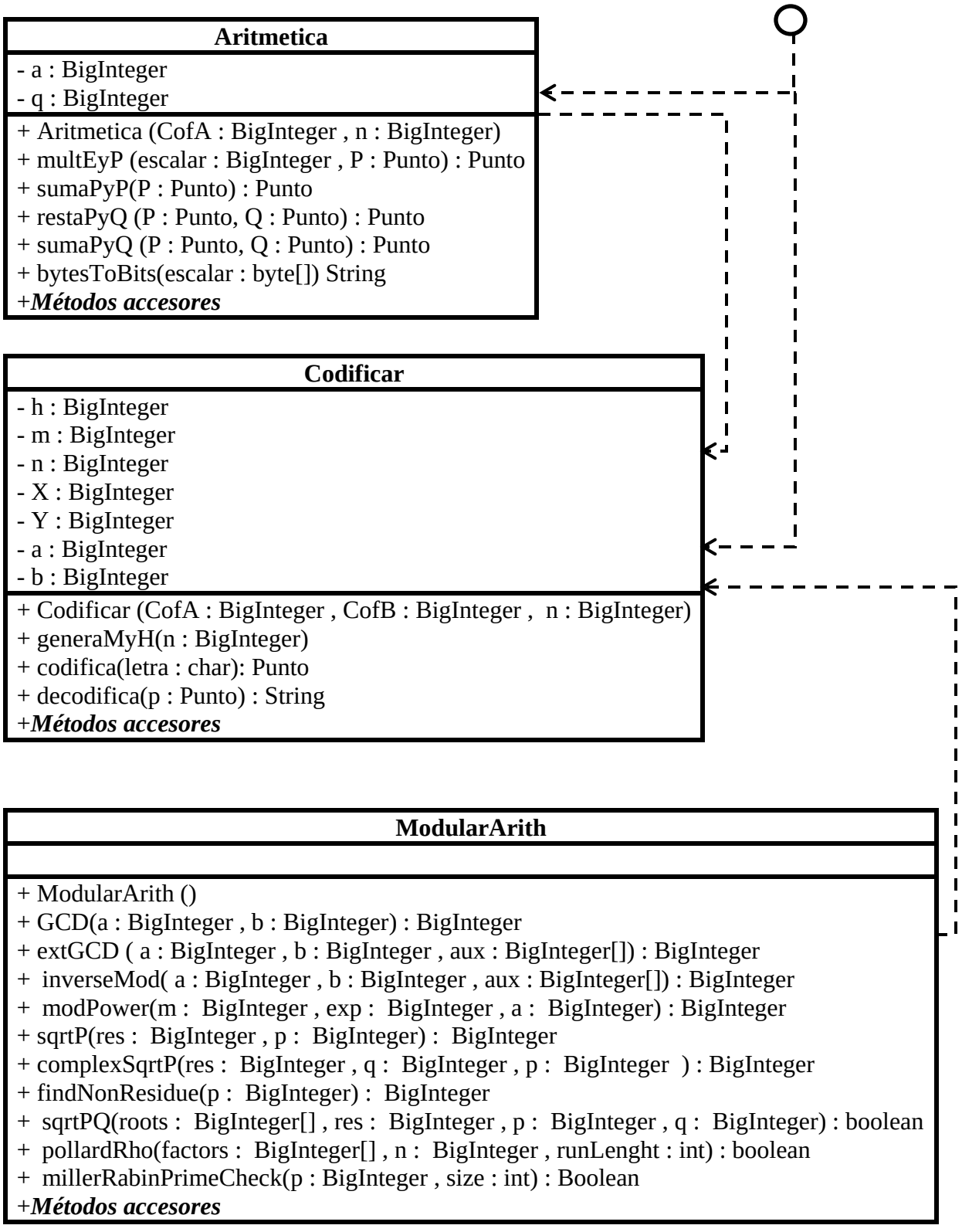
3.3.1 Fundamentos de implementación

Java, al ser un lenguaje de programación orientado a objetos es posible hacer uso de clases de programación y crear objetos a partir de esas clases. Esto resulta ser una gran ventaja al programador ya que existe una gran versatilidad en el diseño del algoritmo, es decir cada quien puede proponer una forma totalmente distinta de implementar un algoritmo y hacerla tan personal y original como se necesite.

Para poder programar algoritmos de cifrado basados en curvas elípticas se necesita el uso de operaciones de aritmética modular y operaciones de suma y multiplicación basadas en matemáticas de curvas elípticas, ya que no son las operaciones de suma y multiplicación convencionales que todos conocen. Por lo tanto es conveniente hacer clases que contengan éstas operaciones y que puedan utilizarse para cualquier algoritmo de cifrado y firma digital que se necesite implementar sin la necesidad de empezar desde cero, ya que todos los algoritmos basados en criptografía de curvas elípticas las necesitan. De esta forma se garantiza la reusabilidad del código generado.

Se propone el siguiente diagrama de clases:




Aritmetica
- a : BigInteger
- q : BigInteger
+ Aritmetica (CofA : BigInteger , n : BigInteger)
+ multEyP (escalar : BigInteger , P : Punto) : Punto
+ sumaPyP(P : Punto) : Punto
+ restaPyQ (P : Punto, Q : Punto) : Punto
+ sumaPyQ (P : Punto, Q : Punto) : Punto
+ bytesToBits(escalar : byte[]) String
+Métodos accesores
Codificar
- h : BigInteger
- m : BigInteger
- n : BigInteger
- X : BigInteger
- Y : BigInteger
- a : BigInteger
- b : BigInteger
+ Codificar (CofA : BigInteger , CofB : BigInteger , n : BigInteger)
+ generaMyH(n : BigInteger)
+ codifica(letra : char): Punto
+ decodifica(p : Punto) : String
+Métodos accesores
ModularArith
+ ModularArith ()
+ GCD(a : BigInteger , b : BigInteger) : BigInteger
+ extGCD (a : BigInteger , b : BigInteger , aux : BigInteger[]) : BigInteger
+ inverseMod(a : BigInteger , b : BigInteger , aux : BigInteger[]) : BigInteger
+ modPower(m : BigInteger , exp : BigInteger , a : BigInteger) : BigInteger
+ sqrtP(res : BigInteger , p : BigInteger) : BigInteger
+ complexSqrtP(res : BigInteger , q : BigInteger , p : BigInteger) : BigInteger
+ findNonResidue(p : BigInteger) : BigInteger
+ sqrtPQ(roots : BigInteger[] , res : BigInteger , p : BigInteger , q : BigInteger) : boolean
+ pollardRho(factors : BigInteger[] , n : BigInteger , runLenght : int) : boolean
+ millerRabinPrimeCheck(p : BigInteger , size : int) : Boolean
+Métodos accesores

Clase: *"Punto"*

Clase que crea objetos de tipo punto; éstos objetos simulan un punto en coordenadas elípticas ya que cada carácter es representado por un punto antes y después de cifrarlo en la criptografía de curvas elípticas. Contiene las coordenadas X y Y, así como el elemento H que se utiliza al momento de codificar y decodificar los caracteres. Todos los atributos son de tipo *BigInteger* por las operaciones modulares que se realizan durante el algoritmo.

Clase: "*Mcifrado"*

Clase que crea objetos de tipo Mcifrado; éstos simulan el mensaje confidencial P compuesto por los elementos M y N, es decir, M=k*GF y N=Mcod+k*KpublicaB, los cuales son atributos de tipo Punto.

$$(M, N) = (kG, P+kP_b) = (kG, P+kbG)$$

Clase: "*Codificar*"

Clase que implementa los métodos para codificar y decodificar un mensaje con base en la teoría de curvas elípticas. Véase figura 3.1 y figura 3.2.

```
/*Método que codifica un caracter y regresa un punto
 *de la curva elíptica E codificado*/
public Punto codifica(char letra)
{
    int i=1;
    /*Variable temporal que guarda y^2*/
    BigInteger YCuad=null;
    /*Punto que regresa el método ya codificado*/
    Punto pCod=new Punto();
    //Convierte la letra a punto
    do{
        X=BigInteger.valueOf(letra).multiply(h).add(BigInteger.valueOf(i));
        YCuad=X.pow(3).add(a.negate().multiply(X)).add(b).mod(n);
        i++;
    }while(ModularArith.sqrtP(YCuad,n)==null);
    //Regresa el elemento raiz cuadrada en matemáticas modulares
    Y=ModularArith.sqrtP(YCuad,n);
    pCod.setX(X);
    pCod.setY(Y);
    pCod.setH(h);
    return pCod;
}
```

Figura 3.1 Método que codifica un carácter.

```
/*Método que decodifica un punto P y regresa un elemento String
 *decodificado: Mdecod=CoordXpCod-1/h
 **/
public String decodifica(Punto p)
{
    BigInteger tmp=p.getX().add(BigInteger.valueOf(-1)).divide(p.getH());
    char[] charArray = new char[1];
    for(int i=0; i<charArray.length; i++)
        charArray[i] = (char) (tmp.intValue());
    return(new String(charArray));
}
```

Figura 3.2 Método que decodifica un punto en curvas elípticas.

Clase: "*Aritmetica*"

Clase que implementa los métodos para realizar las matemáticas de Curvas elípticas sobre puntos de una curva, es decir los métodos de suma y resta de 2 puntos, la multiplicación de un número escalar por un punto y la suma de dos puntos iguales.

El principal motivo que las curvas elípticas sean usadas en criptografía, radica que sobre el grupo $GF(p)$ puede definirse el problema del logaritmo discreto, y no hay una manera eficiente de resolverlo. Más aún el algoritmo conocido hasta hoy para este problema corre a un tiempo totalmente exponencial.

El Problema del Logaritmo Discreto Elíptico (PLDE), sobre el grupo $GF(p)$ se define de la siguiente manera: sea P un punto en $GF(p)$, y considérese al subgrupo cíclico generado por P, (P). Entonces el PLDE para Q en (P), es encontrar un número entero x tal que $xP=Q$. Por lo tanto, el problema del logaritmo discreto en curvas elípticas se refiera una multiplicación, la cual es de vital importancia para cualquier implementación, en ésta clase se propuso como un método. Véase figura 3.3.

```
/*Método que realiza la multiplicación de un Escalar por un Punto,
 * usando el algoritmo de izquierda a derecha, en Curvas Elípticas*/
public Punto multEyP(BigInteger escalar,Punto P)
{
    String binario=new String(bytesToBits(escalar.toByteArray()));
    Punto multEP=new Punto(BigInteger.ZERO,BigInteger.ZERO);
    if(escalar.equals(BigInteger.ZERO))
        return multEP;
    else{
        Punto tmp=new Punto(BigInteger.ZERO,BigInteger.ZERO);
        for(int i=0;i<binario.length();i++)
        {
            multEP=sumaPyP(multEP);
            if(binario.charAt(i)=='1')
                multEP=sumaPyQ(multEP,P);
        }
    return multEP;
    }
}
```

Figura 3.3 Método que realiza la multiplicación entre dos puntos en Curvas elípticas.

Clase: "*ModularArith*"

El contenido de esta clase son los métodos necesarios en aritmética modular para poder realizar las operaciones necesarias durante los procesos de cifrado en curvas elípticas. Por ejemplo, es necesario al momento de codificar un mensaje el obtener la raíz cuadrada en matemáticas modulares. Véase Figura 3.4.

```
public static BigInteger sqrtP(BigInteger res, BigInteger p)
{
    BigInteger zero = BigInteger.valueOf(0);
    BigInteger one = BigInteger.valueOf(1);
    BigInteger two = BigInteger.valueOf(2);
    BigInteger three = BigInteger.valueOf(3);
    BigInteger four = BigInteger.valueOf(4);
    if (p.mod(two).compareTo(zero) == 0) return null;
    BigInteger q = (p.subtract(one)).divide(two);

    if (modPower(p,q,res).compareTo(one)!=0) return null;

    while (q.mod(two).compareTo(zero) == 0)
    {
        q = q.divide(two);

        if (modPower(p,q,res).compareTo(one)!=0)
        {
            return complexSqrtP(res, q, p) ;

        }

    }

    q = (q.add(one)).divide(two);
    return modPower(p,q,res);
}
```

Figura 3.4 Método que realiza la raíz cuadrada en matemáticas modulares.

Las clases y métodos descritos anteriormente son la base para poder implementar algoritmos basados en curvas elípticas, además de que nunca cambiaran independientemente del algoritmo que se desee implementar, ya sean algoritmos de cifrado como ElGamal o de firma digital como DSA, los cuales tienen su equivalencia en curvas elípticas.

3.3.2 ElGamal con Curvas elípticas (ECCElGamal)

Como se mostró en el tercer capítulo, el algoritmo ElGamal es un ejemplo muy práctico y claro para hacer notar las diferencias entre la criptografía de curvas elípticas y la criptografía convencional, ya que la modificación de las curvas elípticas en el algoritmo original de ElGamal es muy notoria, incluso se puede llegar a pensar que es otro algoritmo y que por lo tanto debería llevar otro nombre totalmente distinto, sin embargo en esencia el concepto es el mismo, simplemente las matemáticas que se utilizan son diferentes, es decir, el problema del logaritmo discreto cambia.

El algoritmo de ElGamal implementado con curvas elípticas, para cifrar y descifrar información, se presenta a continuación:

Algoritmo: (Cifrado ElGamal elíptico).

INPUT: Los parámetros (p, a, b, G, n), la clave pública KpB y el mensaje en claro “m”.

OUTPUT: El mensaje cifrado (M, N).

1. Representar el mensaje m como un punto M de E (p).
2. Escoger un entero aleatorio “k” en [1, n - 1].
3. Calcular los puntos M = k* G y N = m + k* KpB en E (p).
4. Devolver (M, N).

Algoritmo: (Descifrado ElGamal elíptico)

INPUT: Los parámetros (p, a, b, G, n), la clave privada "b", y el mensaje cifrado (M, N).

OUTPUT: El mensaje en claro "m".

1. Calcular los puntos: b* M = b * k *, G = k*KpB y N = m + k * KpB en E (p).
2. Obtener el mensaje en claro "m" del punto M.
3. Devolver "m".

Un caso práctico para mostrar el algoritmo: Alicia quiere mandarle un mensaje a Beto. Primero, Beto establece su clave pública de la siguiente manera. Escoge una curva elíptica sobre un cuerpo finito GF(p). También escoge un punto G sobre E. Después escoge un entero "a" que mantiene secreto y calcula KpB=a*G. Entonces la clave pública de Beto la conforman la curva elíptica E, el cuerpo finito GF, y los puntos G y Kp sobre E. La clave privada de Beto es el entero "a". Para mandarle un mensaje a Beto, Alicia debe hacer lo siguiente:

Paso 1. Obtener la clave pública de Beto.

Paso 2. Expresar su mensaje como un punto $m \epsilon GF(p)$. (*Procedimiento de codificación*).

Paso 3. Escoge un entero aleatorio k, que mantiene secreto, y calcula el punto M = kG.

Paso 4. Calcula N= m + k*KpB.

Paso 5. Le envía M y N a Beto.

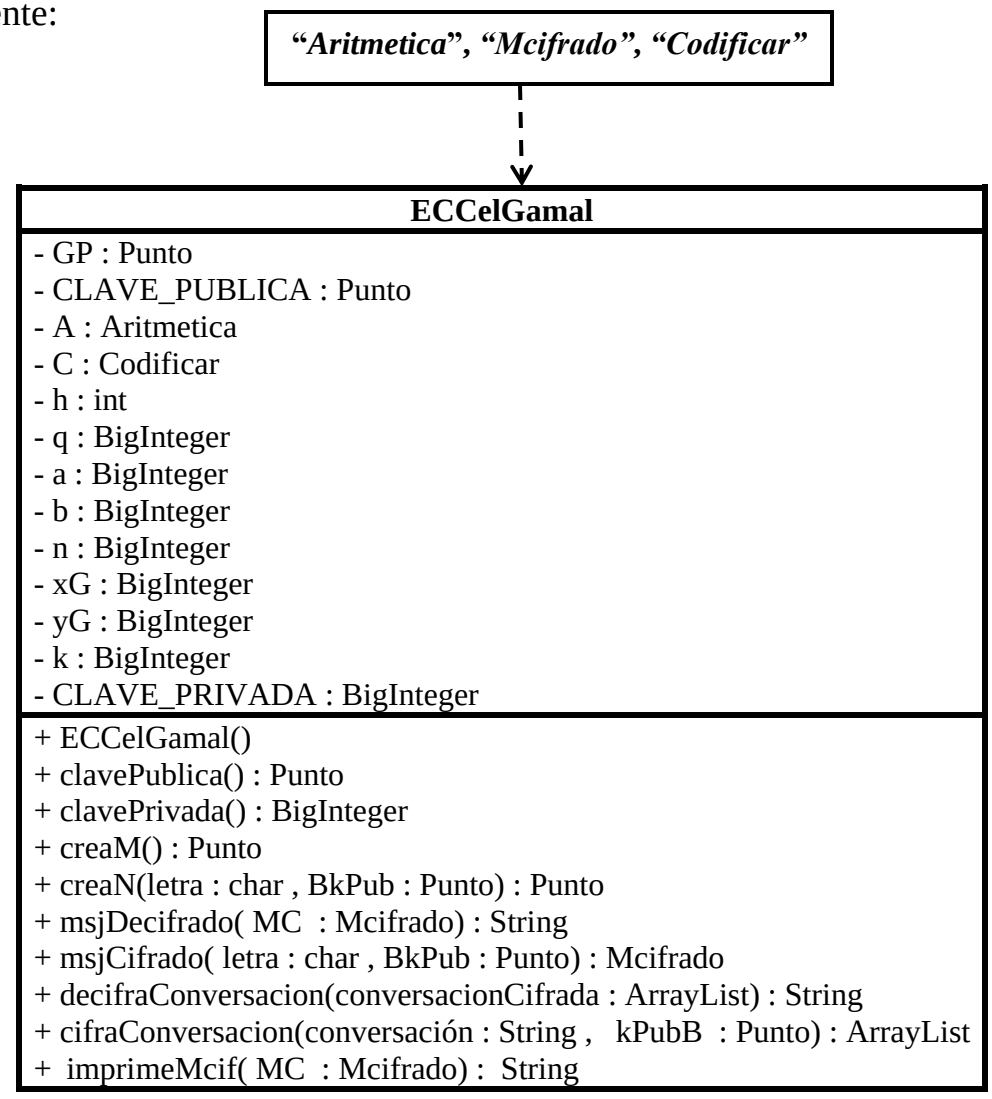
Luego, Beto puede recuperar el mensaje de Alicia calculando:

$$m = N – (b*M)$$

Nótese que este cálculo precisamente nos da ‘‘m’’ pues:

$N - b*M = (m + k*KpB) - b\,(k*G) = m + k\,(b*G) - b\,(k*G) = m + b\,(k*G) – b*k*G = N$

El diagrama de clase que se propone para este criptosistema es el siguiente:



+ Métodos Accesores

El resultado de ejecutar el algoritmo de cifrado es el que se muestra en la siguiente figura. Véase Figura 3.5.

```
--------------------Configuration: <Default>--------------------

Mensaje en claro: Marisol!

Mensaje Cifrado:

D65DCB0AD3CE9A1CD37DC9B32BBD2FEB6ADF7C8EAFCD230E85A635C8B2A036E37615BB8BF1671FA640CA0743E2A37889
7887E6A08B42A94BB45941EC10E1F4B0248632389D2DA816D7A094E7479B9EC5A48B2C5C9402C479E50293EBAC609730
4F15A5DDD505F6516CB602C4C7743005FA14020ECD17CD7F42F3F3151C9530D0952594B75FE7FF5B4030DA2F3F14E8DA
C4F9368699B68ABBCEB6B2CF08E0459A098CF5B276CB9DCC6605F83014F623C5F02F9A0000CCDD62B64F3E1D06341FCE
1A93596CE2751025F191AC500F690090022CF0B234865F16D36939DCD99D3F296BA3FCB1CAA72BA449F972EE943855D6
B6B416ABDF8E7D7C710BAD39416A771312407F3B9D8BE65E340CF5B8FECF06944E586FCFF6AD0ECBC7D684191E73CD53
E2930FC872968AF1C87712479D28992C3E3F33DC5B671AFE52F1EA6161FB68C9E24BABBB3CF4682F0BE8F6F76E4F5CB6
C9EB8BCDE5DA6D457B737814C588A072530B49207CF51F6355F6344E9C0AF691A426FD42D4DD7194D9F5042E8E421BDC

Mensaje Descifrado: Marisol!

Process completed.
```

Figura 3.5 Ejecución del algoritmo ElGamal elíptico.

4.3.3 DSA con Curvas elípticas (ECDSA)

Por otro lado, no solamente la criptografía de curvas elípticas se puede aplicar a criptosistemas de cifrado de clave pública, si no que también a criptosistemas de firmas digitales. El ejemplo mas ilustrativo es el del algoritmo DSA, el cual es muy utilizado actualmente incluso su versión en curvas elípticas.

Como se ha visto hasta ahora, la principal ventaja de este esquema es que requiere números de tamaños menores para brindar la misma seguridad que DSA (*Digital Signature Algorithm)* normal o RSA. De igual manera que en todos los algoritmos basados en Curvas elípticas, existen dos tipos de curvas dependiendo del campo finito en el que se definan que pueden ser GF(P) o GF(2^m), es decir el campo de los primos y el campo de los binarios. Además de hacer uso de las dos primitivas básicas para puntos en Curvas elípticas, las cuales son la Suma de puntos y la Multiplicación escalar.

El algoritmo DSA es una variante de la firma de ElGamal, que es la base del estándar de firma digital DSS (*Digital Signature Standard*). El algoritmo ECDSA (*Elliptic Curve Digital Signature Algorithm*) es el análogo al algoritmo DSA con curvas elípticas. Los procesos de generación y verificación de firma, considerando los mismos parámetros que en la configuración del criptosistema ElGamal, son los siguientes:

Algoritmo (Generación de firma digital del ECDSA)

INPUT: Los parámetros (p; a; b; G; n), la clave pública Kp, la clave privada "d" y el mensaje en claro "m".

OUTPUT: El mensaje "m" con la firma (r; s).

1. Calcular el Hash del mensaje: h = H (m).
2. Escoger un entero aleatorio k en [1, n - 1].
3. Calcular el punto k*P = (x, y) en E (p).
4. Calcular r = x (mod n) (si r = 0 ir al inicio).
5. Calcular s = k^{-1}(h + d *r) (mod n) (si s = 0 ir al inicio).
6. Devolver "m" y (r, s).

Para verificar la firma a partir del Hash del mensaje hay que calcular el inverso "w" de "s" módulo n. Entonces con la clave pública basta calcular el punto R = (w * h) * P + (w *r) *Q y comprobar, dado que k = (w*h) + (w* d *r) y Q = d*P, que las abscisas de los puntos R y k*P coinciden.

El criptosistema ECDSA, como se mencionó con anterioridad es muy utilizado para una gran cantidad de aplicaciones y protocolos, incluso Java en su versión 6 ya implementa el algoritmo ECDSA como parte de su API, además de contar con clase referidas exclusivamente a la implementación de algoritmos de curvas elípticas como es el paquete ***java.security.spec.*** Por lo tanto, la propuesta de implementación de este algoritmo es diferente a la propuesta para el criptosistema ElGamal. El diagrama de clase que se propone para este criptosistema de firma digital es el siguiente:

ECDSA
- keyGen : KeyPairGenerator - random : SecureRandom - pair : KeyPair - pub : PrivateKey - dsa : Signature - dsaVerify : Signature
+ ECDSA() + getKpub() : PublicKey + CreaFirma(Mensaje : String) : byte[] + ValidaFirma(Kpub : PublicKey , Mensaje : String , Firma : byte []) : String + ***Métodos accesores***

El resultado de ejecutar el algoritmo de firma digital es el que se muestra en la siguiente figura. Véase Figura 3.6.

```
--------------------Configuration: <Default>--------------------

Mensaje 1: Marisol!

Firma Mensaje 1:

3036021900E1622DE1BC1CB193DA1CF000E27358515618229E5920718C021900AFF43CD8E1464F712C6C58BE6AA41BB1A3727D363FDE463A

Mensaje 2 (sin firmar): Mar y Sol!

Validando Firma Mensaje 1: Aceptada

Validando Firma Mensaje 2: Rechazada

Process completed.
```

Figura 3.6 Ejecución del algoritmo ECDSA.

3.3.4 Conclusiones

Después de haber implementado dos de los algoritmos más significativos para la Criptografía de Curvas elípticas, es indispensable el hacer la comparación entre estos algoritmos y sus versiones basadas en el problema del logaritmo discreto convencional, ya que ambas vertientes son utilizadas actualmente y cumplen con el mismo fin, sin embargo es necesario ver la eficiencia que proporciona cada criptosistema. Existe una ventaja fundamental entre los criptosistemas basados en el ECDLP sobre los criptosistemas basados en DLP que ya se ha mencionado en varias ocasiones a lo largo de este documento, y es sin duda la reducción del tamaño de las claves en proporción a la seguridad y a los recursos físicos de una computadora, es decir el hardware.

Un ejemplo muy claro se puede apreciar en el primer algoritmo de cifrado basado en Curvas elípticas implementado, el cual es ElGamal elíptico. Véase Tabla 3.3.

Tabla 3.3 Relación de la longitud de la clave para obtener un nivel de seguridad similar dado por NIST.

Clave Privada (bits)	ELGamal (bits)	ElGamal elíptico (bits)
80	1024	161-233
112	2048	224-255
128	3072	256-383
192	7680	384-511
256	15360	+512

Comparación Tamaño de Claves en bits para criptografía basada en DLP y ECDLP

La ventaja de la Criptografía de Curvas elípticas radica en la reducción del tamaño de las claves, sin pérdida de seguridad, respecto a los algoritmos basados en el problema del logaritmo discreto clásico. El siguiente cuadro compara el tamaño de claves entre DSA y ECDSA. Véase Tabla 3.4

Tabla 3.4 Cuadro comparativo de tamaños de clave entre DSA y ECDSA (NIST 2001).

DSA (bits)	ECDSA (bits)	Reducción
1024	163	1:6
3072	256	1:12
7680	384	1:20
15360	512	1:30

La diferencia es tan interesante que, DSA se logra romper en *18* segundos para un tamaño de claves de *160 bits*, mientras que, con el mismo tamaño, usando curvas elípticas, se requieren $8 * 107$ años, usando máquinas con capacidad de cálculo normales.

Con lo demostrado anteriormente, se puede comprobar que la criptografía de curvas elípticas es una vertiente de la criptografía moderna con muchas ventajas y beneficios para la seguridad informática que puede ser aprovechada por los especialistas en la seguridad y desarrolladores de software que quieran dar garantía de confidencialidad con bajo costo de *Hardware* a sus usuarios.

Capítulo 4
Aplicación de Mensajería instantánea segura

En los capítulos anteriores se habló de la importancia que implica el proceso de comunicación para el ser humano, de cómo han ido evolucionando las formas que el ser humano utiliza para comunicarse por medio de las TIC y más particularmente de los sistemas de mensajería instantánea, los cuales son sin lugar a duda los medio modernos más utilizados y funcionales que el ser humano tiene para comunicarse; pero además se ha hecho hincapié en el funcionamiento de estos servicios y sobre todo, en la falta de seguridad que poseen para brindar un servicio confiable y eficiente a los usuarios.

Por otro lado, también se han abordado temas acerca de la teoría de curvas elípticas, desde el ubicarlas dentro de la Criptografía asimétrica pasando por los fundamentos matemáticos en los cuales se basan sus algoritmos, hasta llegar a algunas de sus aplicaciones que explotan la principal ventaja que tienen con respecto a otros sistemas de cifrado de clave pública, es decir, ofrecer el mismo nivel de seguridad que otros algoritmos asimétricos como RSA pero con claves de menor tamaño.

Para respaldar estas aseveraciones, se desarrollaron en el capítulo anterior dos algoritmos basados en la teoría de curvas elípticas. Uno de ellos fue ElGamal elíptico el cual es un algoritmo de cifrado y ECDSA el cual es un algoritmo de firma digital, ambos muy utilizados en la actualidad para diversas aplicaciones y protocolos.

Al haber analizado, en el primer capítulo, las vulnerabilidades y carencias de seguridad que tienen los sistemas de mensajería instantánea actuales, y además el haber contemplado los beneficios y ventajas que implica el uso de la criptografía de curvas elípticas como herramienta para proveer seguridad a la tecnología, se ha propuesto una solución que implica el hacer uso de los algoritmos de cifrado y firma digital basados en curvas elípticas para mitigar las carencias de seguridad en los sistemas de mensajería instantánea.

4.1 Análisis e identificación del problema

Retomando la investigación hecha sobre los sistemas de Mensajería instantánea utilizados actualmente, se sabe que los protocolos que brindan este servicio, dedican todo o la mayoría de sus recursos en garantizar la autenticación de los usuarios por medio de cifrado y firma digital en la validación del usuario y contraseña, pero dejan al descubierto una importante vulnerabilidad al momento de establecer la comunicación entre los usuarios después de que se ha validado su respectiva identidad, la cual es que la comunicación que se establece entre ellos viaja en claro a través del internet, siendo susceptible a ser interceptada, vista y alterada.

La seguridad en un proceso de comunicación no es solamente validar al inicio de la comunicación la identidad de los usuarios y garantizar ante un proveedor de servicios, en este caso de Mensajería instantánea, que un usuario es quién dice ser; sino que también se deben tener medidas que garanticen seguridad durante todo el proceso de comunicación entre dos o más usuarios. Si no se mantienen medidas adecuadas de seguridad que garanticen integridad, confidencialidad y disponibilidad durante el proceso de intercambio de mensajes, los usuarios pueden ser víctimas de espionaje, suplantación de identidad (incluso después de haber sido autenticados ante un servidor), denegación de servicio, entre otros.

Hoy en día hay muchas empresas y desarrolladores que se dedican a ofrecer aplicaciones y servicios de Mensajería instantánea, tratan de brindar seguridad en el envío de mensajes cifrando la información. En principio la propuesta es muy buena, pero la forma en que la implementan no es muy conveniente, ya que como se ha visto anteriormente (capítulo 2), las aplicaciones de mensajería instantánea seguras que cifran la conversación lo hacen, por ejemplo, con "AES 256" que es un algoritmo de clave privada o basado en criptografía simétrica; la cual es por naturaleza utilizada para cifrar grandes volúmenes de información por medio de una clave única. Sin lugar a duda, existe una gran seguridad en este sistema, sin embargo implementarlo es verdaderamente costoso monetariamente y en recursos computacionales pues los algoritmos de clave privada son muy "caros" computacionalmente, aunado a la necesidad de contar con un "tercero de confianza" o un centro de distribución de claves, KDC (*Key Distribution Center*), ya que es Criptografía simétrica.

Todo este sistema está implementado para el envío de "cadenas" de texto, los cuales no son datos muy grandes pero que se hacen enormes al momento de cifrarlos.

Se puede tener una perspectiva distinta si se visualiza que un sistema de mensajería instantánea se utiliza para el intercambio de mensajes a través de un medio inseguro, y para esta situación es adecuado utilizar la Criptografía asimétrica o de clave pública pues en origen fue creada para esta labor, sin embargo los algoritmos de cifrado de clave pública son incluso más costosos que los algoritmos de cifrado de clave privada, por lo tanto resulta preferible utilizar un centro de distribución de claves y solucionar ese problema que hacer uso de la Criptografía asimétrica. Por otra parte, aquí es donde la criptografía de curvas elípticas puede mitigar la dificultad en el uso de los algoritmos de cifrado de clave pública por sus características descritas anteriormente.

Al realizar el intercambio de mensajes directamente desde un usuario a otro sin la necesidad de un "tercero de confianza", se puede ahorrar dinero y recursos computacionales en un sistema de Mensajería instantánea, pues solamente la labor de autenticación sería realizada por un servidor que administre a los usuarios y la comunicación se llevaría a cabo de manera independiente entre los clientes que requieran comunicarse.

Si se utilizan algoritmos de clave pública o de Criptografía asimétrica para los sistemas de Mensajería instantánea, se puede garantizar la confidencialidad de la comunicación junto con la autenticación de los usuarios en cada mensaje cifrado, sin embargo el cifrar información protege de gente no autorizada para ver la información, pero no brinda integridad a la misma.

Por esto, es necesario poder asegurarse de que los mensajes que se envíen a través de un medio inseguro, no solamente conserven la confidencialidad de la información sino también la integridad de cada mensaje; para lo cual es posible, gracias a la criptografía, firmar digitalmente cada mensaje antes de cifrarlo y tener la confianza de que no fue alterado durante el envío.

4.2 Diseño y planteamiento

Después de haber analizado las vulnerabilidades que se presentan en los sistemas de mensajería instantánea, es conveniente presentar una alternativa que solucione estas carencias y pueda mitigar en lo mayor posible los riesgos de la comunicación vía mensajería instantánea por medio de herramientas de vanguardia como la Criptografía de Curvas elípticas.

Como posible solución a las deficiencias de seguridad en los sistemas de Mensajería instantánea, ahora se puede fijar un objetivo claro y establecido, el cual es hacer la propuesta de una aplicación de Mensajería instantánea segura.

Para lograr una aplicación de Mensajería instantánea que brinde seguridad a las comunicaciones de los usuarios, es necesario que además de cifrar y firmar digitalmente la autenticación del usuario con el servidor, se cifre y firme digitalmente cada mensaje que se mande entre los clientes, sin embargo, este procedimiento puede resultar contraproducente, ya que cifrar y firmar digitalmente cada mensaje con algoritmos robustos y seguros, requiere una gran capacidad de recursos de *Hardware* del dispositivo en el que se lleva a cabo la ejecución, además de que el tiempo en la transmisión de información se vería seriamente afectado por el aumento de tamaño en los datos cifrados. Por esto, es indispensable el uso de la Criptografía de Curvas elípticas, que por los beneficios anteriormente argumentados, puede cumplir con los requerimientos de seguridad sin perder eficiencia.

El procedimiento que se espera es el siguiente. Véase Figura 4.1.

1. Ejecutar la aplicación y realizar la autenticación del usuario y su contraseña con el servidor. En esta primera conexión de los clientes con el servidor, la información de usuario y contraseña debe enviarse de forma cifrada y firmada digitalmente (Como se hace en todos los sistemas de mensajería instantánea), sin embargo en esta aplicación no se realizará cifrado y firma digital de las credenciales por requerir una mayor infraestructura de implementación.

2. Posteriormente, y después de haber autenticado a los usuarios con éxito, se propone realizar una segunda conexión directamente entre los dos clientes que quieran comunicarse sin pasar por el servidor.

3. Con esto, se realiza un intercambio de claves entre clientes y se garantiza que solamente la información será visible entre ellos.

4. Por último, se cifra y firma digitalmente cada mensaje con los algoritmos correspondientes y se realiza la comunicación de forma segura.

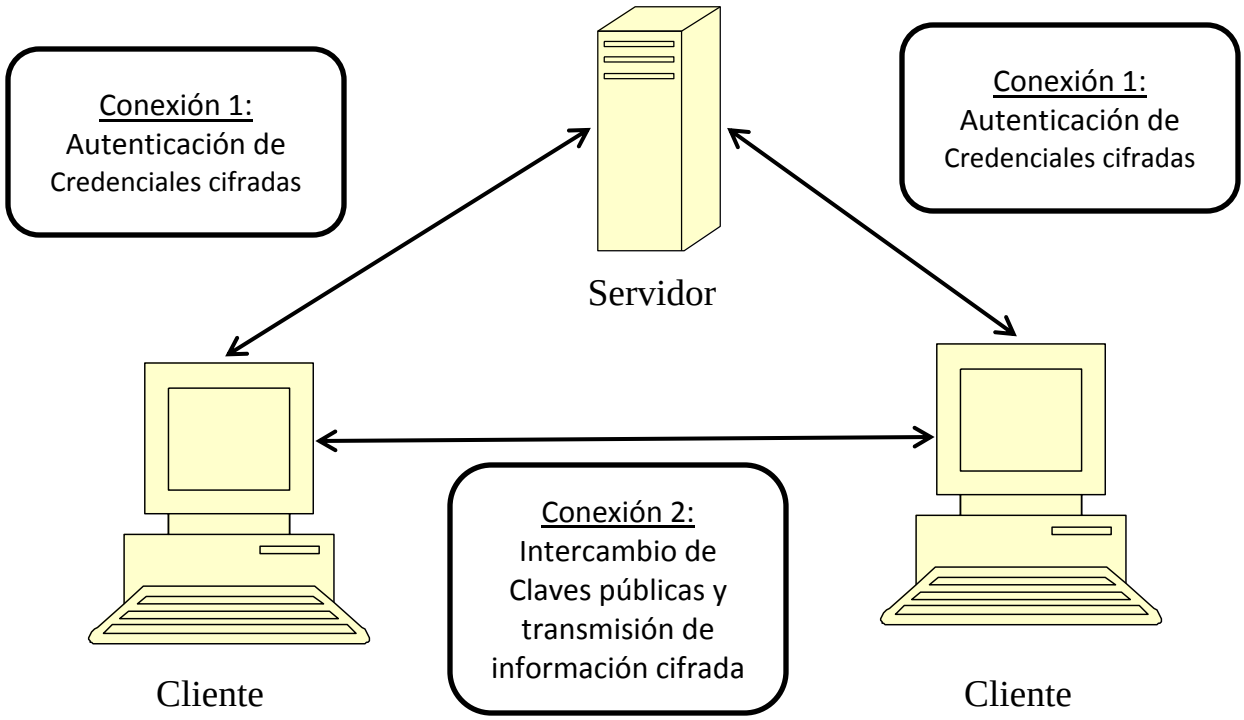


Figura 4.1 Esquema de un Sistema de Mensajería instantánea segura.

Para realizar esta aplicación, es indispensable utilizar algoritmos de cifrado y firmas digitales basadas en curvas elípticas, por lo tanto se pueden utilizar cualquiera de los algoritmos de clave pública que tengan su versión con ECDLP, como puede ser RSA, DSA, ElGamal, entre otros.

En el capítulo anterior se mostró y ejemplificó el funcionamiento de los algoritmos ElGamal y DSA elípticos, por lo tanto la aplicación de Mensajería instantánea segura utiliza estos algoritmos para fines prácticos y didácticos.

Se ha propuesto un diseño de la aplicación utilizando las clases de los algoritmos implementados en el capítulo cuatro como se mencionó anteriormente, por lo tanto sus métodos de cifrado y firma digital, serán fundamentales en el planteamiento de esta aplicación. El diagrama de clases que está diseñado para la aplicación de mensajería instantánea segura consta de dos programas, las cuales son los clientes y el servidor de autenticación de usuarios:

1. Aplicación Cliente

Clase: *"HiloLeerMensjae"*

Clase que realiza la lectura de los mensajes recibidos en la conversación, es decir se encarga de descifrar y comprobar la firma de cada uno de los mensajes que recibe. Véase Figura 4.2.

HiloLeerMensaje
- conexion : ConexionSocket - texto : JTextArea - seguir : boolean - ecc : ECCelGamal - ecdsa : ECDSA
+ HiloLeerMensaje (conexión : ConexionSocket, texto : JTextArea, ecc : ECCelGamal, ecdsa : ECDSA) + run()

Figura 4.2 Diagrama de la clase "*HiloLeerMensaje"*.

Clase: *"ConexionSocket"*

Clase implementada para realizar la transmisión de datos por medio de sockets. Está implementada de tal forma en que pueda enviar no sólo objetos de tipo "*String*", sino también Objetos de todo tipo. Para el caso de esta aplicación de mensajería instantánea, envía objetos de tipo "*ArrayList*".

Con esta clase se puede realizar una conexión de manera normal, como cliente o como servidor según se requiera. Fue diseñada de esta forma para poder reutilizar código y sea una clase que se pueda utilizar en otras aplicaciones, es decir no esta limitada a funcionar en esta aplicación solamente. De igual forma que realiza las conexiones correspondientes, también implementa métodos para la desconexión y la obtención de los datos.

Por último realiza la lectura y el envío de datos serializados, que en este caso los datos son los mensajes cifrados enviados y recibidos por los usuarios. Véase Figura 4.3.

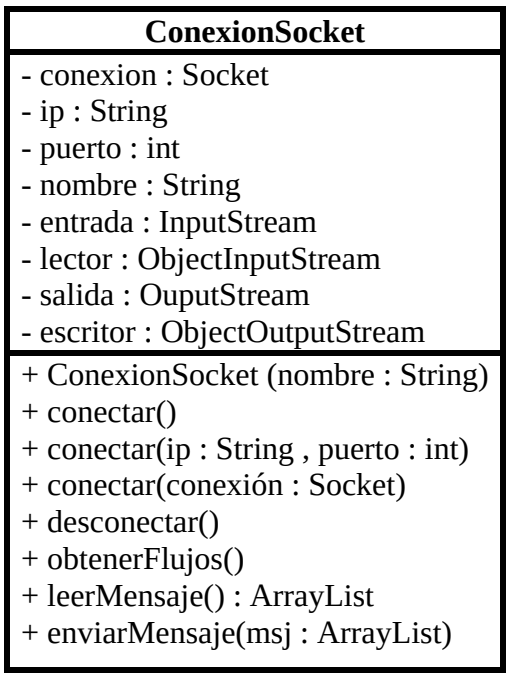


Figura 4.3 Diagrama de la clase *"ConexionSoket"*.

Clase: *"VistaServidor"*

Clase que implementa de manera formal la aplicación de chat, es decir, contiene los atributos para la interfaz gráfica, junto con los métodos necesarios para poder enviar a la pantalla los mensajes recibidos y regresar la respuesta. El diseño le permite funcionar ya sea como Servidor o Cliente. La cantidad máxima de caracteres que se pueden enviar son 140, los mismos que los SMS ya que contiene la escalabilidad para poder implementarse como aplicación móvil con algunas modificaciones. Véase figura 4.4.

Figura 4.4 Diagrama de la clase *"VistaServidor"*.

Clase: *"Ventana"*

Clase que funciona como la salida estándar de cualquier IDE. Implementada para poder observar el proceso que sigue la conversación de manera interna al Chat, es decir, muestra los mensajes cifrados y descifrados, así como la validación de la firma digital. Su implementación es meramente demostrativa y es absolutamente ajena a la aplicación de chat seguro. Véase figura 4.5.

Ventana
- eManda : JLabel - eRecibe : JLabel - eTitulo : JLabel - jScrollPane1 : JScrollPane - jScrollPane2 : JScrollPane - tManda : JTextArea - tRecibe : JTextArea
+ Ventana () + imprimeCifrado(A : ArrayList, kpunB : Punto, kpunA : Punto, msj : String)

Figura 4.5 Diagrama de la clase *"Ventana"*.

2. Aplicación Servidor de autenticación y autenticación de la aplicación.

Clase: *"Autenticacion"*

Clase que se implementa en el chat seguro descrito anteriormente, que tiene la función de un sistema de autenticación, donde el usuario ingresa su nombre de usuario y contraseña para ser enviada a un servidor que autentica las credenciales. Cabe mencionar que las cadenas que tienen el nombre de usuario y contraseña se envían cifradas y firmadas digitalmente con los algoritmos ElGamal y DSA elípticos. Imprime el mensaje de *"Bienvenido al sistema"* o *"ERROR! Usuario o contraseña incorrectos"* según la respuesta de validación del servidor. Véase figura 4.6.

Autenticacion
- jLabel1 : JLabel - jLabel2 : JLabel - jButton1 : JButton - jButton2 : JButton - jPasswordField1 : JPasswordField - jTextField1 : JTextField
+ Autenticacion () - jButton1ActionPerformed(evt : ActionEvent) - jButton2ActionPerformed(evt : ActionEvent) + cliente(s1 : String, s2 : String)

Figura 4.6 Diagrama de la clase *"Autenticacion"*.

Clase: *"Servidor"*

Clase que simula un servidor de mensajería instantánea, el cual se encarga de autenticar las credenciales de los usuarios. En este caso, recibe las credenciales de manera cifrada, las descifra y las valida con ayuda de un archivo de texto, es decir, recibe un *String* del cliente que es su nombre de usuario y contraseña concatenados y busca en un archivo "txt" si existen las credenciales. Véase figura 4.7.

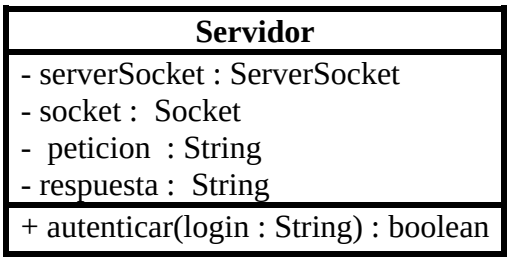


Figura 4.7 Diagrama de la clase *"Servidor"*.

4.3 Implementación

Una vez diseñada la aplicación de mensajería instantánea segura, se procede a implementarla en un lenguaje de programación; el lenguaje de programación seleccionado es Java, ya que por las características presentadas en el capítulo cuatro resulta ser ideal para esta aplicación, además que los algoritmos de cifrado utilizados son implementados en Java.

En cuanto a los requerimientos necesarios para la correcta ejecución de la aplicación, solamente es necesario tener instalada la versión más reciente de JDK en el equipo. Al ser una aplicación implementada en Java, es multiplataforma, es decir puede ejecutarse en sistemas Linux, UNIX y Windows.

La aplicación esta contenida en un archivo JAR (*Java ARchive*) llamado ***"MIS"***, el cual es un tipo de archivo que permite ejecutar aplicaciones escritas en el lenguaje Java, por lo tanto basta con dar un doble "*click*" sobre el archivo para poder ejecutarla en un sistema Windows o para sistemas UNIX-Linux con el comando: ***java -jar MIS.jar*** desde una terminal.

A continuación se muestra a lo largo de diferentes escenarios la ejecución del programa y la interacción que tendría el usuario con la aplicación. Esta demostración se realizó en un sistema operativo Windows 7 profesional y un JDK 7.

Escenario 1. Ejecución del programa

Antes de que los usuarios comiencen a utilizar la aplicación, es necesario poner en marcha el programa que simula el servidor de autenticación, el cual no tiene interfaz gráfica. Posteriormente se procede a que el usuario ejecute el archivo JAR localizado en la parte superior izquierda de la pantalla. Véase Figura 4.8.

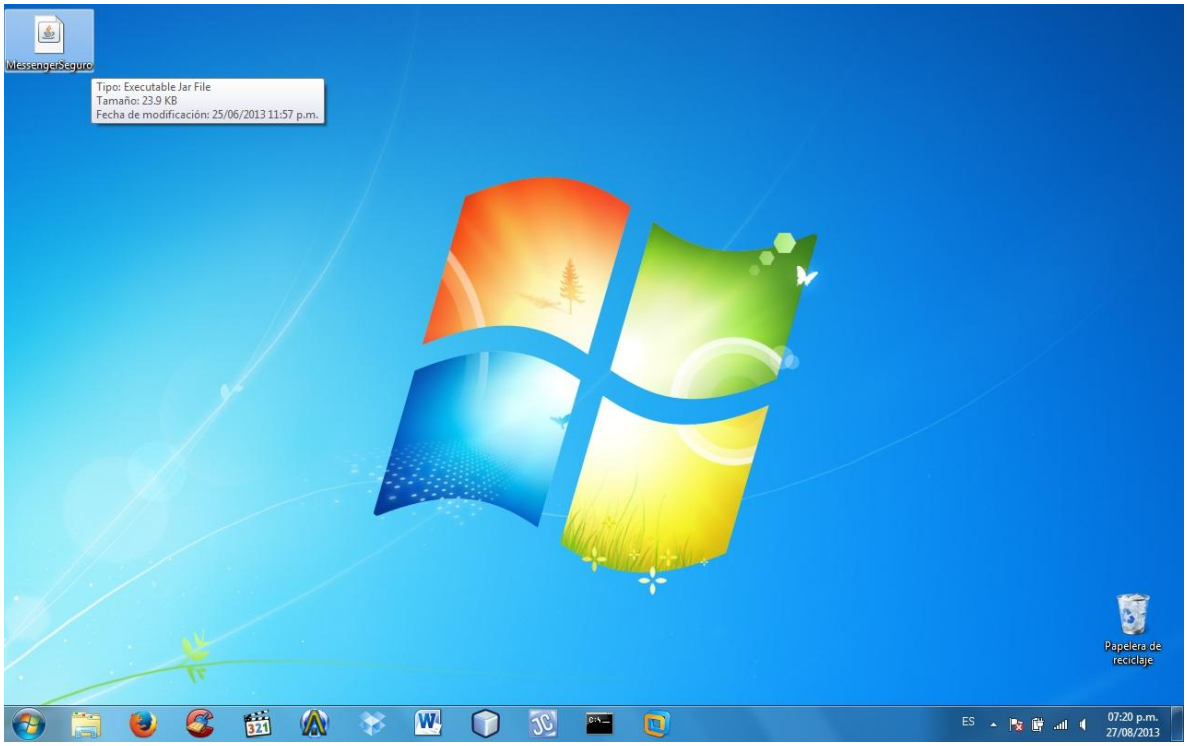


Figura 4.8 El usuario ejecuta la aplicación.

Escenario 2. Autenticación del usuario

Una vez iniciado el programa servidor y después de haber ejecutado la aplicación, el usuario procede a ingresar sus credenciales, es decir su dirección de correo electrónico y su contraseña para corroborar su identidad ante el servidor de autenticación. Véase Figura 4.9. Cabe mencionar que la comunicación entre los clientes y el servidor de autenticación se realiza mediante el puerto 8000.

De ser corroborada la identidad del usuario muestra un mensaje de autenticidad de las credenciales. Véase Figura 4.10. En caso contrario muestra un mensaje de error. Véase Figura 4.11.

Por otra parte, al ser una aplicación segura debe poder brindar seguridad en todos los aspectos, en este caso mitigar los ataques de autenticación por fuerza bruta, por lo tato solamente permite tres oportunidades de autenticación fallida antes de cerrar la aplicación. Véase Figura 4.12.

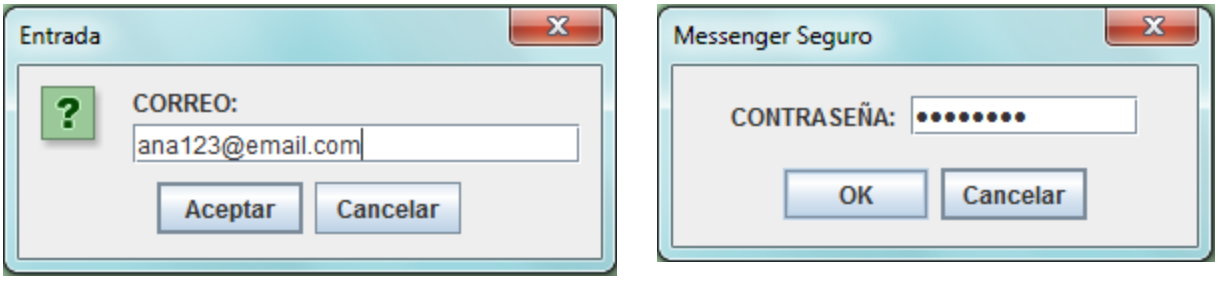


Figura 4.9 Ingreso de las credenciales de usuario a la aplicación.

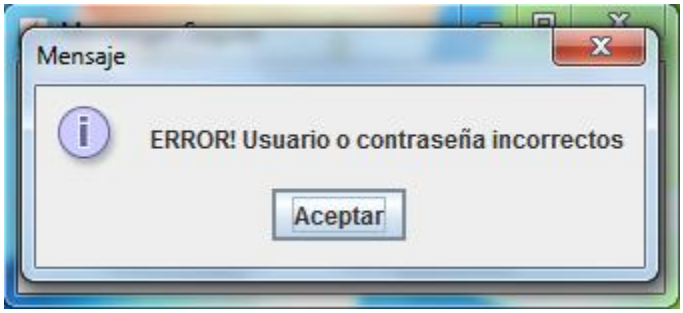


Figura 4.10 Mensaje de error en caso de autenticación fallida.

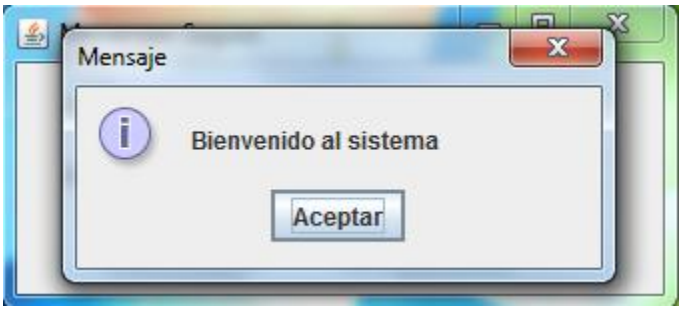


Figura 4.11 Mensaje de bienvenida en caso de autenticación exitosa.

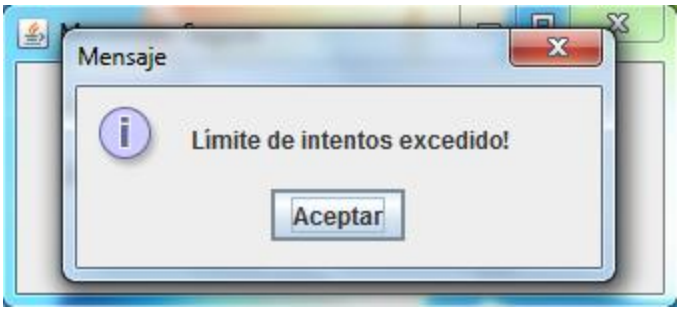


Figura 4.12 Mensaje de límite de intentos permitidos antes de cerrar la aplicación.

Escenario 3. Preparando la comunicación entre usuarios

Posteriormente y en caso de que la autenticación haya sido exitosa, el usuario realiza los preparativos para poder comenzar a comunicarse, es decir selecciona el nombre que desea tener durante la conversación. Véase Figura 4.13.

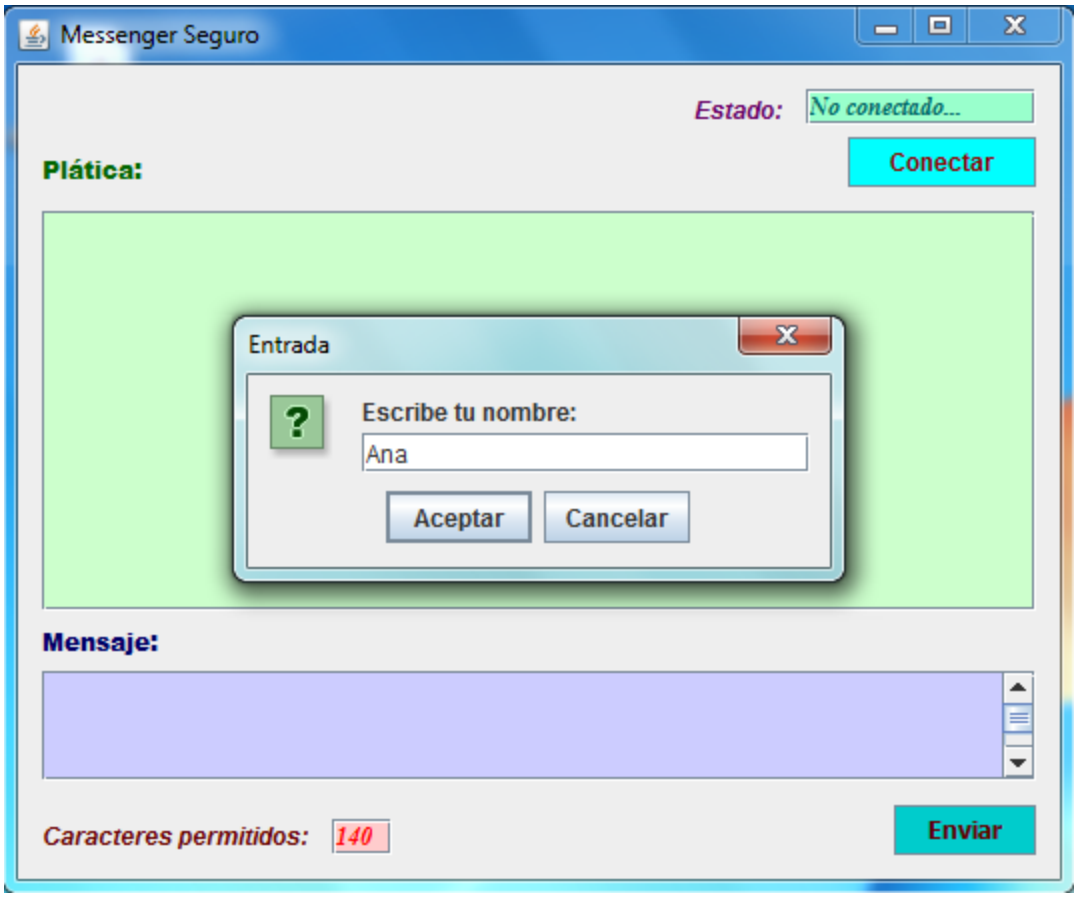


Figura 4.13. El usuario establece el nombre que tendrá durante la conversación.

Escenario 4. Estableciendo una sesión para comunicarse

Cabe mencionar que en esta versión de la aplicación solamente es posible comunicarse con un usuario a la vez y en cada sesión. La comunicación se establece por medio de direcciones IP, es decir si el usuario Ana quiere comunicarse con el usuario Beto, debe ingresar la dirección IP de la computadora de Beto para poder enviar mensajes.

Con la ayuda de un servidor más completo, es posible que esta aplicación pueda realizar la conexión de usuarios por medio de su dirección de correo electrónico, la cual estaría asociada a su IP como era el caso del extinto Microsoft Messenger.

Para que un Ana pueda conectarse con Beto, basta con oprimir el botón conectar para que ingrese la IP del usuario con quien quiera comunicarse. Nótese que el estado de la conexión aparece como **"No conectado…"**. Véase Figura 4.14. Si no se desea conectarse con otro cliente, la aplicación permanece en estado de "escucha", es decir, puede recibir mensajes de otro usuario. Véase Figura 4.15.

Cabe mencionar que el ingreso de la dirección IP está validado por una expresión regular que se encuentra en el método **"validarIP()"** de la clase **"VistaSeervidor"**, todo esto para evitar la entrada inadecuada de datos, en dado caso que se ingrese una IP inválida, aparece el mensaje: ***"DIRECCIÓN IP NO VÁLIDA"*** y solicita el ingreso nuevamente de una IP correcta. Por otra parte, la comunicación entre usuarios se realiza mediante el puerto 9595.

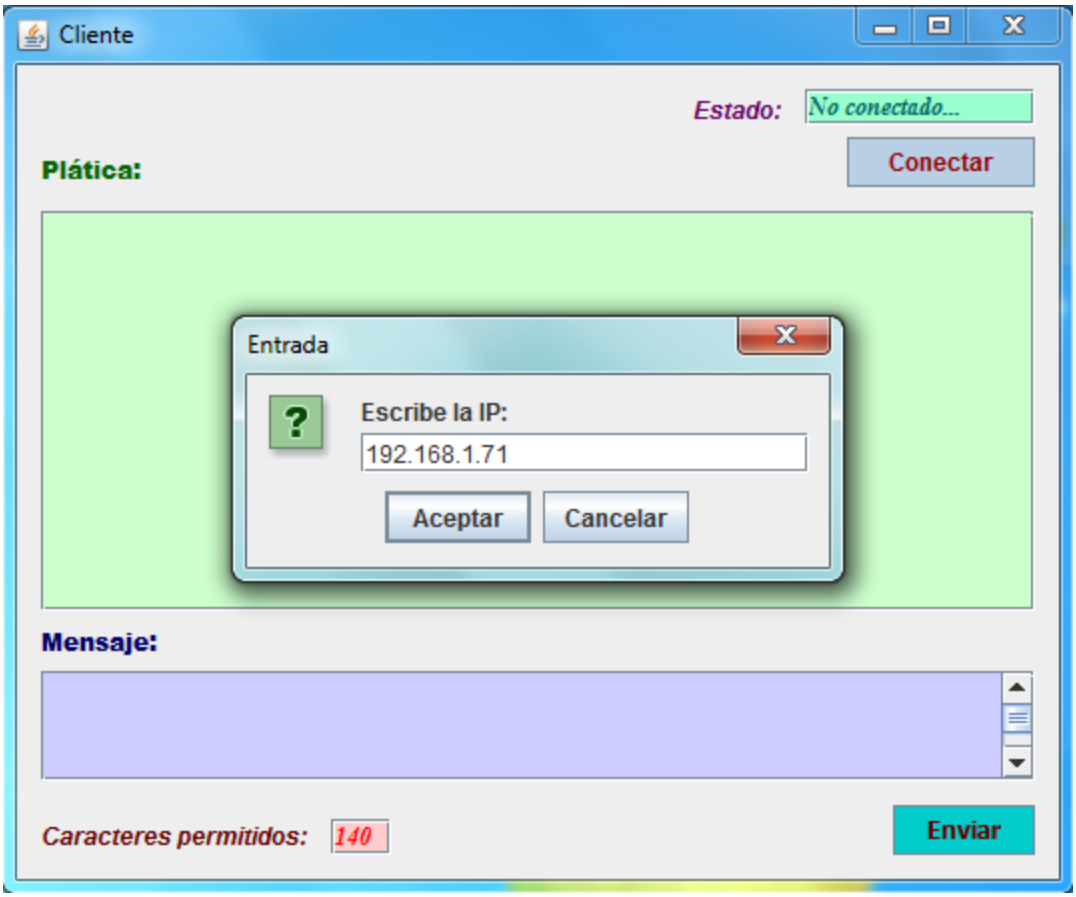


Figura 4.14. El usuario ingresa la dirección IP del cliente con quien quiere conectarse, posterior al haber oprimido el botón conectar.

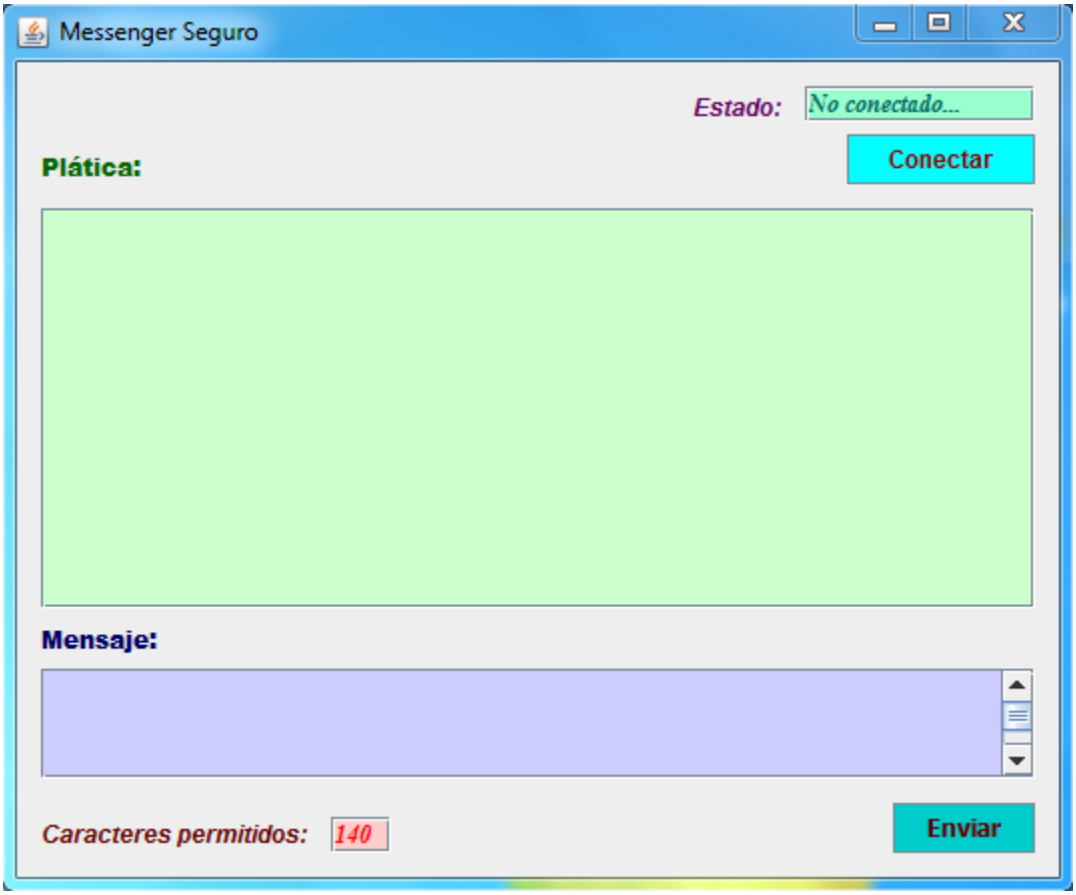


Figura 4.15. Estado en "escucha" de la aplicación, en caso de no querer conectarse con algún cliente.

Escenario 5. Conversación entre usuario

Después de haber sido autenticado por el servidor que guarda las credenciales de los usuarios y después de establecer la comunicación con quien se desee conversar, finalmente el usuario puede comunicarse con otro cliente por medio de la aplicación, solamente basta con comenzar a escribir los mensajes que se deseen enviar y presionar el botón **"Enviar"**, para lograr con éxito el intercambio de información entre ellos. Nótese que el estado de la conexión cambia a **"Conectado…"**. Véase Figura 4.16.

Al tratarse de una aplicación de mensajería instantánea segura, ya que cifra y firma digitalmente los mensajes enviados y recibidos, se debe limitar el número de caracteres listos a ser enviados en cada mensaje, pues los caracteres originales aumentan considerablemente su tamaño en Bits antes de ser transmitidos, por lo tanto la información enviada a través de los sockets puede demorar en demasía y la aplicación perdería eficiencia, ya que además de tardar en enviar un volumen grande de información, tardaría considerablemente el proceso de cifrado y firma digital. Por tal motivo se limitó el número de caracteres por mensaje a 140, tomando como referencia el número de caracteres permitidos en los mensajes SMS. Nótese como disminuye el contador que aparece en la parte inferior izquierda con la leyenda **"Caracteres permitidos"** mientras se escribe, al llegar a 0, la aplicación no le permite al usuario continuar escribiendo. Véase Figura 4.17.

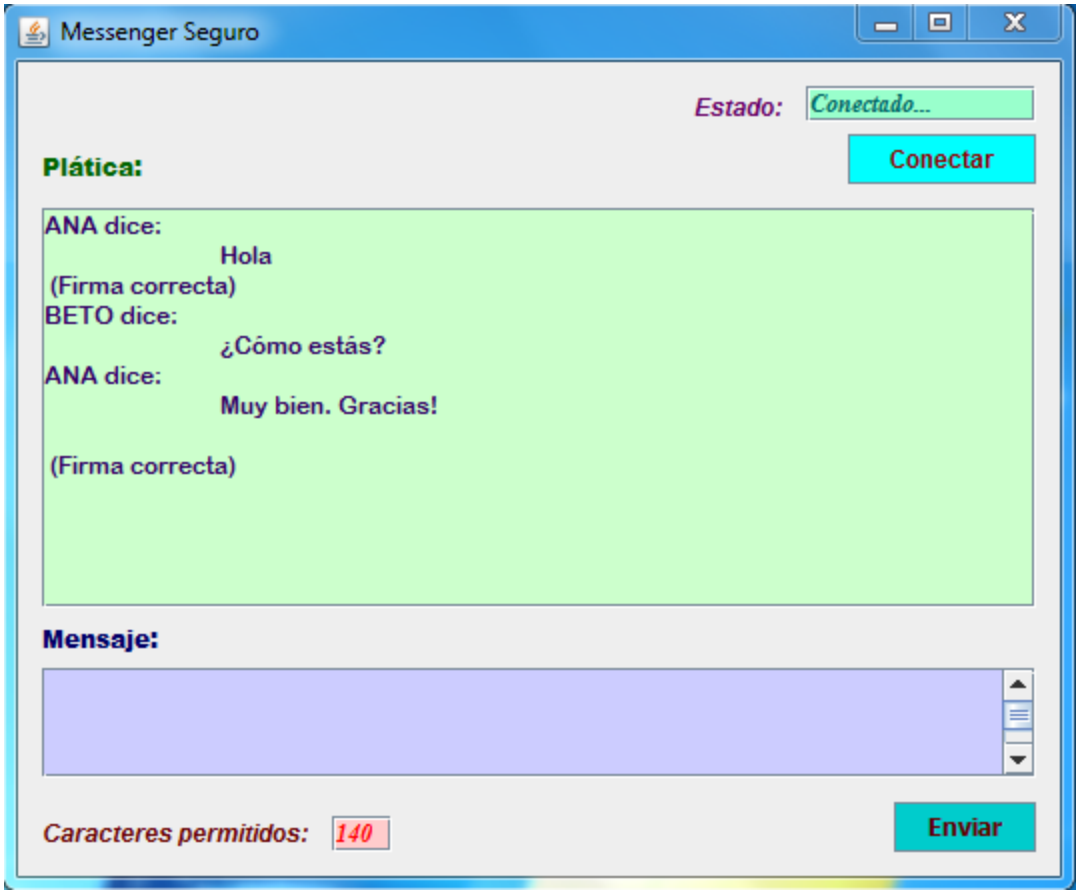


Figura 4.16. Conversación llevada a cabo entre dos usuarios por medio de la aplicación.

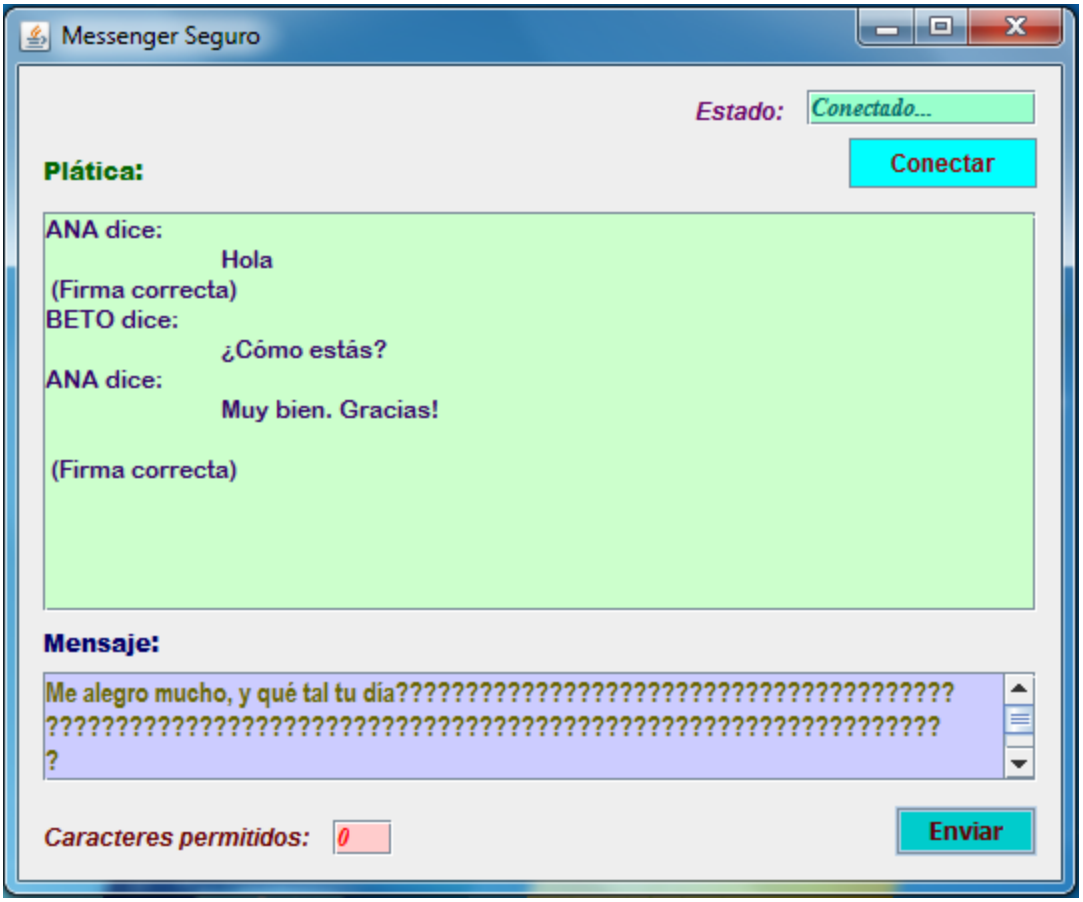


Figura 4.17.El contador de caracteres solamente permite 140 caracteres y disminuye mientras se escribe.

Cuando se desea terminar la conversación, basta con cerrar la aplicación, esto con la intención de evitar que algún usuario ajeno a la plática pueda enviar información mientras la aplicación permanece abierta y el usuario original no se encuentre. Por otra parte, si alguno de los usuarios cierra su aplicación, en la otra computadora el estado de la conexión en la aplicación cambia nuevamente a desconectado.

4.4 Resultados

Como se puede observar durante la conversación que se tiene entre dos usuarios, aparece debajo del mensaje enviado por el emisor una leyenda que dice **"(Firma correcta)"**, esto significa que el mensaje recibido no fue alterado durante el envío del mismo, incluso después de haber sido cifrado y que la información es la correcta.

Aparentemente la aplicación funciona sin problema alguno, ya que ambos usuarios, en el caso demostrativo Ana y Beto, pueden establecer una comunicación entre ellos. Sin embargo es necesario demostrar que sucede internamente durante el proceso de transmisión de la información, y si verdaderamente funcionan los algoritmos de cifrado y firma digital, en este caso ElGamal y DSA elípticos, para garantizar que la información viaje segura.

Para realizar la demostración del proceso interno que se lleva acabo en la aplicación “MIS”, fue necesario implementar una ventana doble que muestre los pasos establecidos para hacer segura la transmisión de información. En la ventana superior se muestran cuatro pasos que sigue cada mensaje antes de ser enviado.

1) En el paso 1 se generan e intercambian las claves públicas de ambas entidades para cifrar la información. Se muestran la clave pública propia que envió y la clave pública recibida.

2) En el paso 2 se firma el mensaje en claro, se muestra la llave pública y la firma digital del mensaje en claro.

3) En el paso 3 se cifra la información antes de enviarse.

4) En el paso 4 finalmente se manda la información cifrada y la firma digital.

Por otro lado, en la aplicación “MIS” del receptor, se muestra el proceso que se sigue para recuperar la información y pueda llegar al usuario final de manera clara. En la ventana inferior aparecen tres pasos posteriores a recibir el mensaje.

1) En el paso 1 se descifra el mensaje cifrado.

2) En el paso 2 se valida la firma digital recibida, en caso de ser correcta la firma se muestra el mensaje de "Aceptada" o "Rechazada" en caso contrario. Además se muestra la clave pública recibida de la firma.

3) En el paso 3 se muestra el mensaje validado y en claro al usuario.

Las pruebas se realizaron en tres equipos conectados inalámbricamente en una red LAN, por un *Access Point Infinitum* 2Wire con un Ancho de Banda de 2Mbps. Las características de ambos equipos son:

Equipo 1.
Usuario: Beto
Sistema operativo: Windows 7 Professional x64
JDK: Versión 7
RAM: 4Gb.
Procesador: Intel Centrino.
IP: 192.168.1.71

Equipo 2.
Usuario: Ana
Sistema operativo: Windows XP Servicie pack 2 x32
JDK: Versión 7
RAM: 1Gb.
Procesador: Intel Pentium 4.

IP: 192.168.1.66

Equipo 3.

Usuario: Servidor de autenticación

Sistema operativo: Windows Vista Servicie pack 2 x32

JDK: Versión 7

RAM: 1.5Gb.

Procesador: Intel Centrino.

IP: 192.168.1.66

Una vez explicadas las características de cómo se lleva a cabo la prueba, finalmente se puede comprobar que la comunicación entre usuarios se realiza de una forma segura con cifrado y firma digital de cada uno de los mensajes, se envía un mensaje del usuario Ana hacia el usuario Beto, donde el mensaje es: *"Hola cómo estás?*". El proceso de transmisión del mensaje se muestra a continuación. Véase Figuras 4.18 y 4.19.

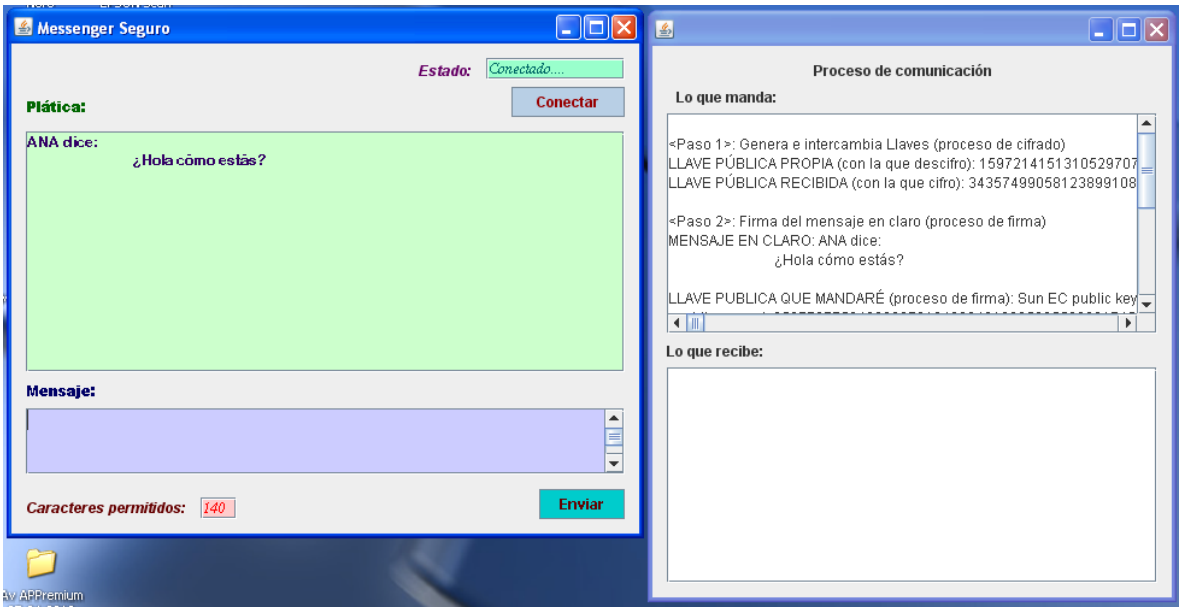


Figura 4.18.Proceso de envío del mensaje de Ana.

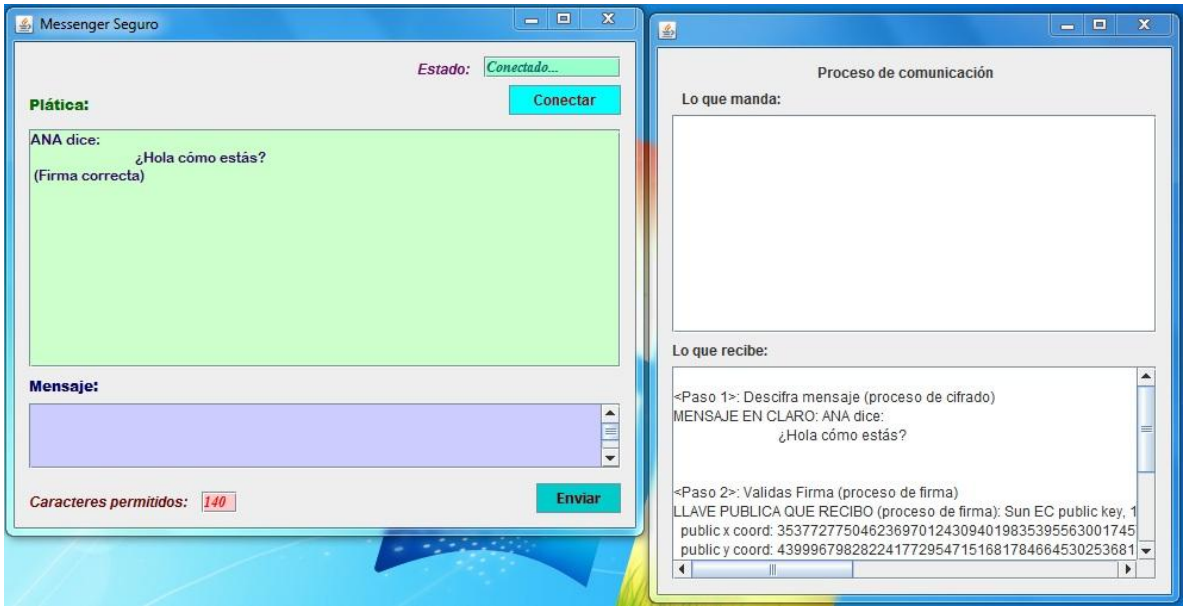


Figura 4.19.Proceso de recibimiento del mensaje por parte de Beto.

Lo que aparece en la ventana superior de Ana son los cuatro pasos descritos anteriormente del proceso de envío del mensaje, es decir la firma digital y el cifrado del mensaje.

<Paso 1>: Genera e intercambia Llaves (proceso de cifrado)

LLAVE PÚBLICA PROPIA (con la que descifro):
159721415131052970704010024794586499952004339641518087390236932075478916506251581080682349176696507356591948495719 16

LLAVE PÚBLICA RECIBIDA (con la que cifro):
343574990581238991087538358177275771292548822728118192641358151974457589880460759605615567247437349259381306640372 1

<Paso 2>: Firma del mensaje en claro (proceso de firma)

MENSAJE EN CLARO: ANA dice:

¿Hola cómo estás?

LLAVE PUBLICA QUE MANDARÉ (proceso de firma): Sun EC public key, 192 bits

public x coord: 353772775046236970124309401983539556300174576236726065642 1

public y coord: 43999679828224177295471516817846645302536813913802913361 25

parameters: secp192r1 [NIST P-192, X9.62 prime192v1] (1.2.840.10045.3.1.1)

FIRMA DIGITAL DEL TEXTO:
30340218770D76AC43C98196D135E2D64CDE0A2FD3E1483DBFB64E6A02183D6DFD0DA2DB3010B09AD07E405EFDE3143991E50F3C4668

<Paso 3>: Cifrado del mensaje (proceso de cifrado)

MENSAJE CIFRADO:
7056A8EB5B1A26F9D9EE47E0B59873513C3AD101641F29A76D6C6257AC4638659A
18904A627C55C43BFDC780B50EE4116372626A5B1AD7573AEF21D2913512BA987D
900BD576435D7F97E421FE6FA8EC46C5DC829534F9BF6102CC2F8B0858B17056A8
EB5B1A26F9D9EE47E0B59873513C3AD101641F29A76D6C6257AC4638659A18904A
627C55C43BFDC780B50EE411F076A3BE066F7BDC668BC221E94DD1076A6F3E5E0
61E75A460415A6B3CCE811A0362C5EF8FA208C86D4733675DCC42261A94C5D71F1
D11BB84C2E4B7F98A6A0FA7C2ABC26180D4FA347404104D735507EF820B3BBF89
2098B4BBD3CF376CA36735A7D54729D92F7404B037D4878F65EFEC31A239C83F82
BBC0D6FED5808CCE38A47DA0B142FB8E472DD4D21E720FE8C4DE48F0542A691D
CB6B2E062D34A0541243F391F4C5A19C00A5F3957D13B67D1AB43626F7DC657C68
009BFDFB76862093337426AA38A61CD709C235BA41D58B81E8C72B2DA01DC9876
E01FD9DFD6B43192DCE9B074B6A2A7B9514EDB88C7FE0B435A676B5BC7525FC6
9FDE53CBC5365846319A3446C74876098A11D99D2BFFBD76734255740BE7899D120
23B23B9F7E539FF537B3F043404FC62F9823F880922A0DF80F95BF417E1A3C4D0729
0C85F1DAE5E30E5B1F2FF89E4A6AE7397AD73ED3AEF0E931F3F5BAA10709E5396
25820EC22D7141B6BB753BB4429EBAC9E4AFDB9409DAE7BD3C08E623489650400
919E16B9C61E82F2CD588FCB802DCDB11C23C548D1BC10C310CA12328FDAE0F7
BC38F38D9FA0AB915D19C583446C76021FDCA6D3A6A89C1FE290184596E0D993D
0CAF3020EDAB12C1D59BF2AA6C396A7452E64FD50A60464507A6B8086900DF66B
949BDEB00A85C06C9B029325477B78F9C9274D0B093281D6C91B3F8FD27FF951758
B5D2C00A468B2FF89C1F4BA53C618A34DF73C0E9F0A7159E9D63610C41089361F6
4CBB1A028BE035E9013BDE2FAAB9898F772E796A84173257452E60E245A7A5C294
B77ED7C485424B389B4C7A26067E632C8F1FD6811CD007E3F15EAB1368E9B75E0E
76CD96871B51F6741CD8E4AF076A3BE066F7BDC668BC221E94DD1076A6F3E5E06
1E75A460415A6B3CCE811A0362C5EF8FA208C86D4733675DCC4226DE48F0542A69
1DCB6B2E062D34A0541243F391F4C5A19C00A5F3957D13B67D1AB43626F7DC657
C68009BFDFB76862093456C1D13B2354B3B3935902FAE006DE2AA2624E3D7629F9
CA83BB8D6D75A792213E1001BA9AD94C358F5BD3D667F7D0F2EF2CC32F840B644
1466076B2D4B4061B68A300A27BE466D13E5118A1DD3A765FCBCAA27DC66E3706
A3C7F89DD65764C49BDEB00A85C06C9B029325477B78F9C9274D0B093281D6C91B
3F8FD27FF951758B5D2C00A468B2FF89C1F4BA53C618AF076A3BE066F7BDC668B
C221E94DD1076A6F3E5E061E75A460415A6B3CCE811A0362C5EF8FA208C86D4733
675DCC4226337426AA38A61CD709C235BA41D58B81E8C72B2DA01DC9876E01FD9
DFD6B43192DCE9B074B6A2A7B9514EDB88C7FE0BA56D7CBF56D00C5E10D4BF7
EC6B0872F73A1519DB1B3C17E5AF23250F34979579289BB5F483AFB50D80D888290
8B25D6C05A17EDF3BCB2D37B7A407914A1D76CE725544BC322CF04E4AA228FC4

C17EF329D4BE16BA0377B047E377457282348F1180C1B079843465F0D70464B0BCE2
3629170151989B8198FDB20C2BDE10EF3935232E6141499176BD2765E4A3487150A5
6D7CBF56D00C5E10D4BF7EC6B0872F73A1519DB1B3C17E5AF23250F34979579289
BB5F483AFB50D80D8882908B25D66C227DFA57EFC35830FB390BA2D42BE4CD192
BCFBACB98D03961247C0475590F8640DB5CCD7719CE1FECD3A0A3C3FB5D539FF
537B3F043404FC62F9823F880922A0DF80F95BF417E1A3C4D07290C85F1DAE5E30E
5B1F2FF89E4A6AE7397AD73E

<Paso 4>: Se manda al receptor Firma digital y Mensaje cifrado

Por otra parte en la ventana inferior de Beto se muestran los tres pasos descritos anteriormente del proceso de recibimiento del mensaje, es decir la comprobación de la firma digital del mensaje y el descifrado:

<Paso 1>: Descifra mensaje (proceso de cifrado)

MENSAJE EN CLARO: ANA dice:

¿Hola cómo estás?

<Paso 2>: Validas Firma (proceso de firma)

LLAVE PÚBLICA QUE RECIBO (proceso de firma): Sun EC public key, 192 bits

public x coord: 353772775046236970124309401983539556300174576236726065642 1

public y coord: 43999679828224177295471516817846645302536813913802913361 25

parameters: secp192r1 [NIST P-192, X9.62 prime192v1] (1.2.840.10045.3.1.1)

FIRMA QUE RECIBO (proceso de firma):
30340218770D76AC43C98196D135E2D64CDE0A2FD3E1483DBFB64E6A02183D6DF
D0DA2DB3010B09AD07E405EFDE3143991E50F3C4668

ESTADO DE LA FIRMA (proceso de firma): Aceptada

<Paso 3>: Se lee el mensaje validado y en claro

Ahora se realiza el mismo proceso para la respuesta de Beto hacia Ana. El mensaje que envía Beto para Ana es: *"Muy bien gracias, ¿y tú?* Véase Figuras 4.20 y 4.21.

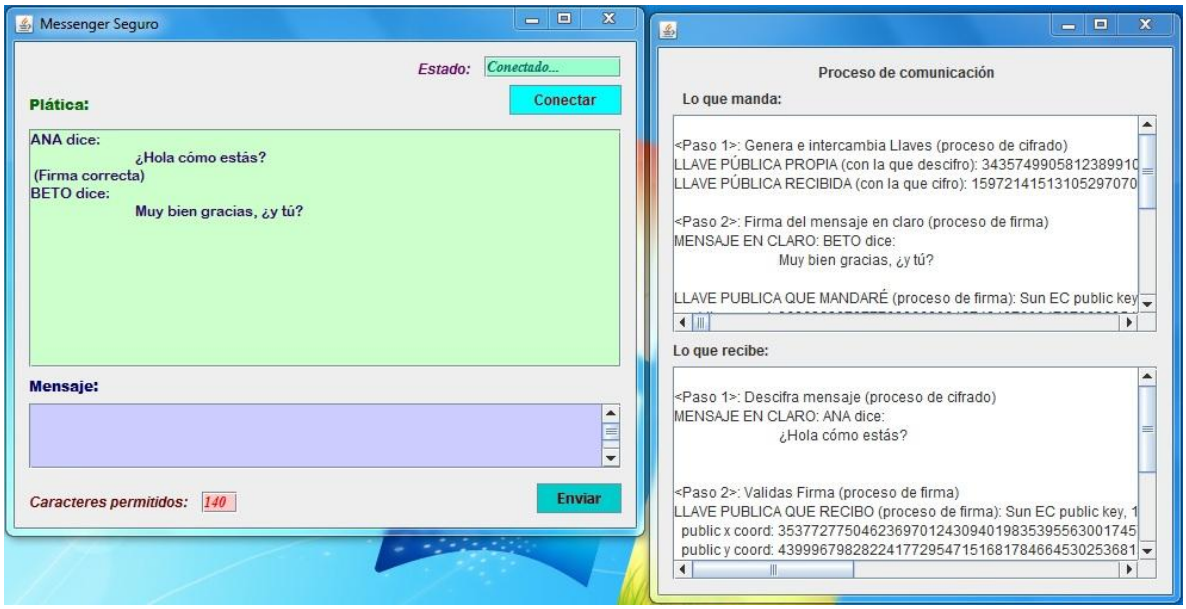


Figura 4.20.Proceso de envío de respuesta de Beto.

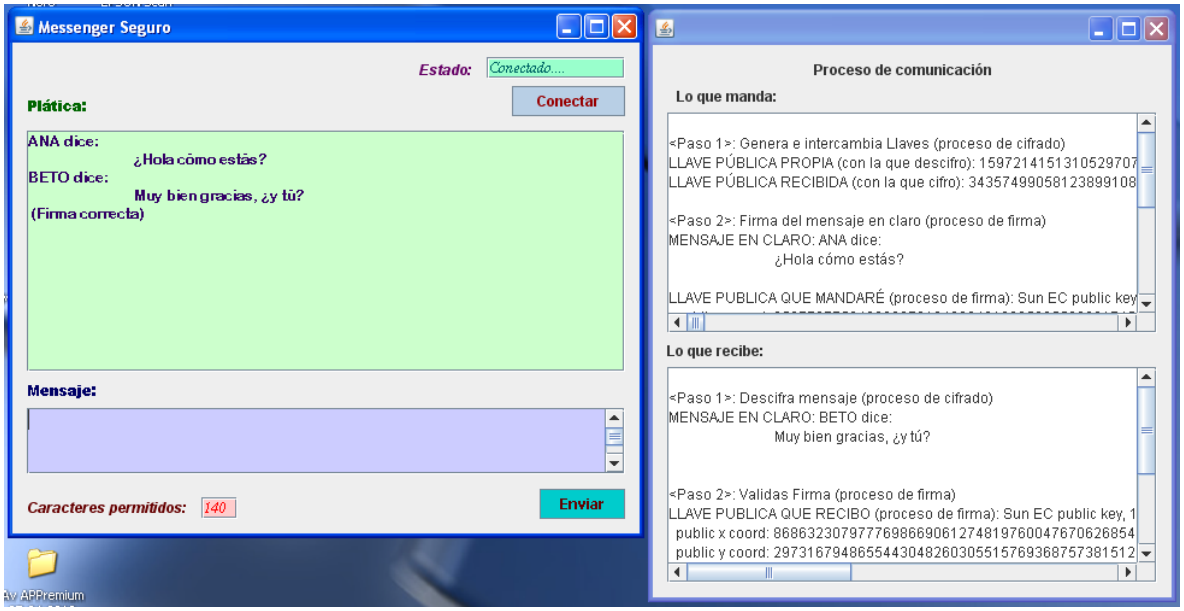


Figura 4.21.Proceso de recibimiento de la contestación por parte de Ana.

Ahora, lo que Beto le contestó a Ana se muestra en la ventana superior de Beto, y fue lo siguiente:

<Paso 1>: Genera e intercambia Llaves (proceso de cifrado)

LLAVE PÚBLICA PROPIA (con la que descifro):
343574990581238991087538358177275771292548822728118192641358151974457589880460759605615567247437349259383130664037 21

LLAVE PÚBLICA RECIBIDA (con la que cifro):
159721415131052970704010024794586499952004339641518087390236932075478916506251581080682349176696507356591948495719 16

<Paso 2>: Firma del mensaje en claro (proceso de firma)

MENSAJE EN CLARO: BETO dice:

Muy bien gracias, ¿y tú?

LLAVE PÚBLICA QUE MANDARÉ (proceso de firma): Sun EC public key, 192 bits

public x coord: 868632307977769866906127481976004767062685474223632966426

public y coord: 29731679486554430482603055157693687573815128410734458593 92

parameters: secp192r1 [NIST P-192, X9.62 prime192v1] (1.2.840.10045.3.1.1)

FIRMA DIGITAL DEL TEXTO:
3034021830003046A5AA53EDDFE91AD89571072047DF9AC984AF958402187CC98E8
C84D9D2FC849912B35D9C54B13BE2926CF9AF728E

<Paso 3>: Cifrado del mensaje (proceso de cifrado)

MENSAJE CIFRADO:
63A686301D9CE62983EE716580835123E7C645464DEE1A4B56B1B1370C865F9793F7
418B809C8AAE63524AB26C1990E7F78C0B3B112C4E33F4768D838471A7073307B00
E361FE4CA94A7A89FE56B292A3ED5C6D05F490C26B23350FC12653DAEEAC2B156
08A815B89818B8251F11EE4DB1802845C36A13597560DFA15EE8402716A327AC8E5
5389E5DFE48892B343CD3CE294B7B4B7198D274107BC967027460DCD5341F8FD07
B867CA98F15C9C1E218EF9A466E09AC5C532017E4EB113256E37A50DA8A1CB2D8
7C218DB799FB064AFDEF7ED44EEC6699E53778037BE4B649776C33F5E005C2CBA1
C4AA003C0DA1D15B7C1E809681A1F4E065DFC107B4F501097B0327D063F73B7528
18A286C17900311155B9900067309B4D4D11851D34DFB4437EC6FD8308C5471B5E7
34D591F46528D82A0D37FCFAB67BC7D077919C3E5D9D8F0A69506CD26B9EC2783
F3C717C287BAC27324914B3A7EF26120183263F6A0676A05294B9E50F2BC3C8489D
A86A4B68485675E06808E80BF93413918E44FB8DA3AA16F99CFD074ACB7636AAC

DE65DDFC51C59B412F37760159E603ADD68BF0FA534391C58441B6CA3EFD9C4C7
E6C7DDE278182945876ADA8DC5AF5BBD1EB26B3B40E1D18D01142393F0846D729
3D7A19BF0CD3928203A9544EEB80846853E51051B763F1AF0D93BB80E9446726100
41D6E99286CF9BABDEC3A4E62C355E17C6F44FC733FC1DFBDF63308DA17E9D53
5E1A09FC0036CE3A44BF32571D404A8236E12924548E627FC0A98875F84CE2DBB18
307ECBFB11A1CFA5E82BB3844C9CD32C4D0CA90CAD8DE2A5DA98EBC93CD2E0
E622438E09A3DC6AAB7B51C54F3CA3988BCBEB7A9A8D0F204C7985D1448195E93
5E1BF2F6D0F99991B9E4D4153317DCA9D00CFACE88EE8F5901242B7210802EBE58
7958F76B378E5FD74D2DC57AF6CCB99E2F65A335F544E720796424FE0BC54FAE65
1B8B6F67D44516945AC3D4530455F4E6D51A0DFF9E294E647585AA77037A50DA8A
1CB2D87C218DB799FB064AFDEF7ED44EEC6699E53778037BE4B649776C33F5E005
C2CBA1C4AA003C0DA1D15BE4E2B0C107D22F49B7E4A69A6F2E8AE60D7DF36C0
358B14CB74FD67B0EA600243D35484D40BFCDA522320BAFCA08CFD4437EC6FD8
308C5471B5E734D591F46528D82A0D37FCFAB67BC7D077919C3E5D9D8F0A69506C
D26B9EC2783F3C717C2B8DA3AA16F99CFD074ACB7636AACDE65DDFC51C59B41
2F37760159E603ADD68BF0FA534391C58441B6CA3EFD9C4C7E6CF2C79FB0DAA97
E8A005FF2681618EDDA2E4977FEB4391D7986A242BA9ADC9C6B460D596E0A80D0
FF26206E7D2685721037A50DA8A1CB2D87C218DB799FB064AFDEF7ED44EEC6699
E53778037BE4B649776C33F5E005C2CBA1C4AA003C0DA1D15DF559EDE67D19ED
6BE3B9B23B4AED8BAFBFD756A9EE0D279CB3CA95228108BCB107B2BD7FAB19C
81C6C9D4172D15F1522323C432BAEE4D24805FE6D459B8610B9913143891030D071
C36E74400DC9CF983D91FE5CAE6C453831412B50E0F8A19A52D053310B38170DF5
154173AF32377BCA355F77C051600B44256556491C31F79A050E67E9D47D03C7713D
D1D207FBB87BAC27324914B3A7EF26120183263F6A0676A05294B9E50F2BC3C8489
DA86A4B68485675E06808E80BF93413918E44F4437EC6FD8308C5471B5E734D591F4
6528D82A0D37FCFAB67BC7D077919C3E5D9D8F0A69506CD26B9EC2783F3C717C2
A52D053310B38170DF5154173AF32377BCA355F77C051600B44256556491C31F79A0
50E67E9D47D03C7713DD1D207FBBBE60808172C0DE2DB4BBDAE80133C15F96CE
8B297FABBE65966FFD3D4FF9A8A97FA470D7D074CFAE739BDA762EB7DE34915C
4B96403DD1923FFE767ADB0307587A71DF703AC095A02EE203D1B5C0B644EFBA3

EE81CB746F8394DAEB032BA915637A50DA8A1CB2D87C218DB799FB064AFDEF7E
D44EEC6699E53778037BE4B649776C33F5E005C2CBA1C4AA003C0DA1D15E2F7BA
18571AA272D6820F8684BDF681FD86905C583EF75CF560A0004040045E17CD86917
EB78DB80CA308E7D9DBB45ACB99E2F65A335F544E720796424FE0BC54FAE651B8
B6F67D44516945AC3D4530455F4E6D51A0DFF9E294E647585AA77037A50DA8A1C
B2D87C218DB799FB064AFDEF7ED44EEC6699E53778037BE4B649776C33F5E005C2
CBA1C4AA003C0DA1D154798663335628527FE361EC7D07B6247487317329748BEA
3EAD6C989FC3DE0191BC609558E509E01798EBCA5F05284CAAFEA023E65C354E0
423D2D67B7FDD9A1B3F6DC6F8FCFF69D74A5FA115C96C740CE1576A487C1C7FA
C845795B4C8304FE312C7062A995C9BBA5BC41BE38A13169ABB9E030621D602F6E
872489C3E9F40F4C9317127B80FA164827D786A77563E51051B763F1AF0D93BB80E9
44672610041D6E99286CF9BABDEC3A4E62C355E17C6F44FC733FC1DFBDF63308D
A17E9D5

<Paso 4>: Se manda al receptor Firma digital y Mensaje cifrado

Y ahora en la ventana inferior de Ana se muestran los tres pasos descritos anteriormente del proceso de recibimiento del mensaje, es decir la comprobación de la firma digital del mensaje y el descifrado.

<Paso 1>: Descifra mensaje (proceso de cifrado)

MENSAJE EN CLARO: BETO dice:

Muy bien gracias, ¿y tú?

<Paso 2>: Validas Firma (proceso de firma)

LLAVE PÚBLICA QUE RECIBO (proceso de firma): Sun EC public key, 192 bits

public x coord: 868632307977769866906127481976004767062685474223632966426

public y coord: 297316794865544304826030551576936875738151284107344585939 2

parameters: secp192r1 [NIST P-192, X9.62 prime192v1] (1.2.840.10045.3.1.1)

FIRMA QUE RECIBO (proceso de firma):
3034021830003046A5AA53EDDFE91AD89571072047DF9AC984AF958402187CC98E8C84D9D2FC849912B35D9C54B13BE2926CF9AF728E

ESTADO DE LA FIRMA (proceso de firma): Aceptada

<Paso 3>: Se lee el mensaje validado y en claro

De esta manera se comprueba que la comunicación entre los usuarios Ana y Beto se realiza de forma segura, donde cada uno de los mensajes es firmado digitalmente y cifrado por medio de criptografía asimétrica de curvas elípticas.

La prueba anterior se realizó también con un sistema operativo Linux, desde la computadora del usuario Beto. La versión de Linux utilizada fue un Fedora 15 x64. Por lo tanto se comprueba que esta aplicación es multiplataforma y que puede comunicar a usuarios con diferentes computadoras, sin importar la arquitectura ni el sistema operativo.

Se siguieron varios de los procedimientos y buenas prácticas del desarrollo de software seguro, para garantizar su correcto funcionamiento y mitigar posibles vulnerabilidades. Por ejemplo se hizo hincapié en la modularidad del código, no generar código redundante, hacer uso de los *"Java Beans*", validar las entradas de datos por parte del usuario con expresiones regulares, se brinda trazabilidad y el manejo de excepciones durante la ejecución, entre otras buenas prácticas. Por otra parte, no existe un software cien por ciento seguro, y por lo tanto, se puede encontrar alguna vulnerabilidad de día cero en esta aplicación.

Al ser una aplicación implementada en Java, puede ser incluso modificada para convertirse en una aplicación móvil, ya que además los algoritmos utilizados fueron diseñados de tal manera en que solamente se necesite crear la instancia de su correspondiente clase para su utilización.

Cabe mencionar, que este trabajo de tesis puede dejar pauta a la posibilidad de implementación de este sistema de mensajería instantánea segura a gran escala. Ya que el programa que simula el servidor de autenticación se debe firmar y cifrar las credenciales, además podría guardar los datos de más de dos usuarios mediante una conexión a una base de datos, y de que en la aplicación de los clientes se implementaría una agenda con los contactos disponibles y que estén asociados a su IP. Para alcanzar estos objetivos es necesaria una infraestructura más robusta que pueda soportar todas estas modificaciones.

Conclusiones

Retomando los objetivos de este proyecto de tesis, los cuales tienen como fin el de desarrollar una aplicación de mensajería instantánea como segura alternativa, que brinde los servicios de integridad, confidencialidad, autenticación y no repudio, haciendo uso de la criptografía basada en curvas elípticas, se puede concluir que se alcanzó el objetivo, ya que se logró implementar una aplicación de mensajería instantánea que además de cifrar la información por medio del algoritmo "ElGamal elíptico", también firma digitalmente los mensajes transmitidos entre dos usuarios por medio del algoritmo criptográfico "DSA elíptico".

Brinda autenticación por medio de un programa servidor que se encarga exclusivamente de la validación de las credenciales presentadas por los usuarios que deseen establecer una comunicación a través de la red. Dichas credenciales son la dirección de correo electrónico y la contraseña del usuario en turno.

Esta aplicación de mensajería instantánea segura, resulta ser una posible alternativa para mitigar los riesgos que se presentan en la transmisión de información por medio de la red, como son el alterar o destruir información sin autorización, la revelación no autorizada de la información, el acceso no permitido a la información, y la denegación del servicio.

Además de proporcionar seguridad a la transmisión de información entre usuarios, esta aplicación se caracteriza por ser un software robusto en su diseño e implementación, ya que está diseñado de acuerdo con algunas prácticas del desarrollo de software seguro, como son la modularidad del código, el uso de encapsulamiento, no hacer uso de código redundante y programación en capas. Además de utilizar java, el cual es un lenguaje seguro, multiplataforma y de alto nivel, como lenguaje de programación, para implementar la aplicación de mensajería instantánea segura y los algoritmos de cifrado y firma digital utilizados en ella.

Referencias

Libros

Bauer, Friedrich, *"Decrypted secrets, Methods and Maxims of Cryptology"*, Tercera edición, Springer, Alemania, 2002.

Bishop, David, *"Introduction to Cryptography with Java Applets"*, primera edición, Jones and Bartleit, Estados Unidos, 2003.

CISCO, "CCNA Exploration 4.0 Aspectos básicos de networking", Cisco Networking Academy.

Delfs, Hans, *"Introduction to Criptography, Principles and Applications"*, Springer, Alemania, 2002.

Fúster Sabater, Amparo, *"Técnicas Criptográficas de protección de datos"*, Alfa omega, segunda edición actualizada, México, 2001.

Joseph Schmuller, "Aprendiendo UML en 24 horas", México, 1997

Lopez Barrientos María. Jaquelina. *"Criptografía, México"*, Primera edición, México, 2009.

Menezes, Alfred, *"Handbook of applied cryptography"*, CRC Press series, Estados Unidos, 1997.

Miret Biosca M. Josep, "Criptografía con Curvas elípticas", Departamento de Matemáticas, Escuela politécnica superior, Universidad de Lleida.

Pastor Franco, José, *"Criptografía digital, Fundamentos y aplicaciones"*, primera edición, Prensas Universitarias de Zaragoza, España, 1998.

Pino Caballero, Gil, *"Introducción a la Criptografía"*, Alfa omega, segunda edición actualizada, México, 2003.

Pressman, Roger s. "Ingeniería del Software, Un enfoque práctico", Mc Graw Hill, quinta edición, España, 2002.

Rodríguez, Amador, *"Protección de la información, Diseño de criptosistemas informáticos"*, Paraninfo, España, 1985.

Santiago Pey y Juan Ruiz Calonja. "Diccionario de sinónimos y contrarios", Sexta edición, México, 1990

Stallings, William, *"Cryptography and network security, Principles and Practice"*, third edition, Prentice-Hall, Estados Unidos, 2003.

Tesis y trabajos de investigación

Barquero Sánchez Adrian, "Curvas elípticas y Criptografía"

ECC Brainpool, "ECC Brainpool Standard Curves and Curve Generation", Version 1.0, Octubre 19 2005

Gabriel Belingueres, "Introducción a los Criptosistemas de Curva elíptica", Buenos Aires Argentina.

NIST, "Recommended Elliptic Curves for Government Use", Julio 1999

Silva Sarabia, Christopher Román "Criptografía y Curvas Elípticas", Facultad de Ciencias, Universidad Nacional Autónoma de México, México, 2006.

Soto Ocaña, Omar. "*Análisis, estudio y desarrollo de criptografía de curvas elípticas",* México, 2008, Tesis (Licenciatura), Universidad Nacional Autónoma de México.

Standards for Efficient Cryptography Group (SECG), "SEC 1: Elliptic Curve Cryptography", Version 1.0, September 20, 2000.

Standards for Efficient Cryptography Group (SECG), "SEC 2: Recommended Elliptic Domain Parameters", Version 1.0, September 20, 2000.

Mabel Hernández Molina, Milton Jesús Vera Contreras, "Criptografía de curvas elípticas" Seminario en Seguridad Informática, Universidad de los Andes-Mérida,

Revistas y boletines

Revista "SG, software gurú conocimiento en práctica", no.5, septiembre-octubre, México, 2006.

Revista "Revista de ingeniería eléctrica, electrónica y computación", Vol. 10, No. 1, Diciembre 2012

Internet

http://www.soyborderline.com/documentacion-y-articulos/documentacion-variada-trastornos/3135-la-importancia-de-la-comunicacion.html

http://www.siliconweek.es/noticias/cuales-son-los-lenguajes-de-programacion-mas-seguros-35504

http://redesdedatosinfo.galeon.com/enlaces2128619.html

http://wikitel.info/wiki/Redes_de_datos

http://es.kioskea.net/contents/200-introduccion-al-comercio-electronico

http://www.wikiteka.com/apuntes/intercambio-de-informacion-entre-sistemas/

http://blog.zerial.org/seguridad/la-importancia-de-la-seguridad-en-la-informacion/

http://blancayjaviinfor.blogspot.mx/2011/05/diferencias-entre-chat-y-mensajeria.html

http://www.cypherpunks.ca/otr/

http://www.criptored.upm.es/crypt4you/temas/privacidad-proteccion/leccion3/leccion3.html

http://computacion.cs.cinvestav.mx/~jjangel/Pagina_cce_es.html#PLDE

http://centrodeartigos.com/articulos-noticias-consejos/article_129175.html

http://en.wikipedia.org/wiki/Elliptic_curve_cryptography

http://www.ecured.cu/index.php/Mensajer%C3%ADa_instant%C3%A1nea

http://platea.pntic.mec.es/vgonzale/trabcolab_0910/archivos/_110/Tema_4.2.htm

http://es.wikipedia.org/wiki/Mensajer%C3%ADa_instant%C3%A1nea

http://www.deltaasesores.com/estadisticas/tecnologia/3156-beneficios-de-la-mensajeria-instantanea

http://www.ecured.cu/index.php/Protocolo_para_la_mensajer%C3%ADa_instant%C3%A1nea

http://www.ecured.cu/index.php/Chat