UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

ANÁLISIS, ESTUDIO Y DESARROLLO DE CRIPTOGRAFÍA DE CURVAS ELÍPTICAS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

OMAR SOTO OCAÑA

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

2008

Agradecimientos

En toda mi experiencia universitaria y la conclusión del trabajo de tesis, conviví con personas que merecen las gracias porque sin su valiosa aportación no hubiera sido posible este trabajo, también hay quienes las merecen por haber plasmado su huella en mi vida.

A mi mamá que me ha guiado con su ejemplo a través de la vida y en todo momento ha estado a mi lado impulsándome a seguir y a alcanzar mis sueños. Gracias “Chini”.

A mi papá que me enseñó el valor del trabajo diario, que la vida no se debe tomar demasiado en serio, que me cuida y me espera en un lugar muy especial. Gracias “Güerito”.

A mi hermana Mary por su apoyo, compañía y por ese orden con el que siempre hace las cosas. Gracias “Chuy”.

A mi hermana Claudia por mostrarme que nuestros errores nos hacen crecer y ser lo que somos hoy. Gracias “Cata”.

A la M.C. Cintia Quezada Reyes por su apoyo en la parte final del presente trabajo.

A mí querida Directora de Tesis M.C. María Jaquelina López Barrientos por su paciencia, sus asesorías, el estímulo para seguir adelante y sobretodo por enseñarme a disfrutar el trabajar con mi tema.

A mis profesores, que compartieron conmigo sus conocimientos y su pasión por la carrera.

A todos mis amigos y familiares que han aportado grandes experiencias a lo largo de mi formación como profesionista.

De corazón, les doy mi más profundo agradecimiento por ayudarme a llegar a esta parte de mi proceso en la vida.

Índice

Prólogo

En la actualidad gracias al surgimiento de la computación comercial las computadoras tienen gran importancia en los procesos administrativos, productivos y de comunicación de los individuos, las empresas, el gobierno y la industria en general. Los sistemas de información se han convertido en un componente indispensable en prácticamente cualquier lugar.

La razón por la cual la computación ha formado parte tan importante en la sociedad actual es porque resulta de gran utilidad en el manejo de toda la información que se tiene disponible en un sistema y que debe ser procesada y utilizada en un tiempo muy breve. Mucha de esta información es indispensable para el manejo o puesta en marcha de muchos negocios o sistemas de gobierno y por lo tanto es de vital importancia que la información se encuentre disponible sólo para las personas autorizadas ya que lo valioso de la misma depende de quién la posee, por ejemplo los ejecutivos de una empresa o negocio que se dedican a tomar decisiones dentro de la organización necesitarán la información que haga referencia a su estado actual dentro del mercado o relacionado con su contabilidad. Por lo cual, con base en el análisis de la información que tengan a la mano se tomará una decisión. Por otro lado, si ésta misma información estuviera en manos de alguna persona que no esté enterado de absolutamente nada en lo referente al negocio en cuestión no le dará el valor que se merece. Pero si la información estuviera en manos de algún competidor comercial, ésta le sería de gran utilidad.

Por lo anterior, la información se puede considerar como un activo muy importante ya que después de los recursos humanos es el elemento más valioso.

Además, con el aumento en el uso de las computadoras y de los elementos que interactúan con ella y que además almacenan información tales como ipod's memorias USB, palm's, tarjetas de crédito o teléfonos celulares también ha surgido la necesidad de compartir información, utilizar mejores equipos, compartir recursos, hacer uso del trabajo conjunto y sobretodo comunicarse, ya que la mayoría de los sistemas modernos prácticamente no tendrían razón de ser sin la capacidad de intercambiar información con otros sistemas.

Los hackers o piratas informáticos son usuarios muy avanzados que por su elevado nivel de conocimientos técnicos son capaces de superar determinadas medidas de protección utilizando diferentes tipos de programas (gusanos, troyanos y bombas lógicas entre otros). Su motivación abarca desde el espionaje industrial hasta el mero desafío personal.

La información que se encuentra registrada en un sistema que se compone de hardware y software se encuentra de forma no palpable por lo que se debe ser muy cuidadoso y prestarle mayor importancia, ya que existen algunas organizaciones que se han visto en peligro debido a ataques que han llegado a sufrir.

En el mundo del ciberespacio la posibilidad de que el fraude o la estafa existan es mucho mayor. La capacidad de tener acceso a información las 24 horas del día, desde cualquier lugar del mundo, es para muchos un beneficio que brinda Internet. Sin embargo, esto plantea algunos inconvenientes prácticos, ya que las redes y sistemas informáticos se han convertido en un nuevo escenario para el delito porque se pueden interceptar comunicaciones electrónicas entre dos personas u organizaciones, introducirse en los sistemas informáticos de empresas, difundir y vender ciertos datos industriales, destruir, modificar o alterar datos, programas o documentos electrónicos.

Por lo tanto es necesario hacer conciencia de la importancia de proteger la información ya que la tecnología avanza cada día más y también las amenazas y los ataques son cada día más sofisticados.

La seguridad de la información puede incluso afectar la vida privada de las personas de ahí el interés que se tiene en protegerla.

Podemos notar que a pesar de la implantación de políticas de seguridad en las empresas, en la restricción en el acceso a las instalaciones que contienen los equipos de cómputo, de las publicaciones elaboradas por diversos organismos referentes a la seguridad de la información acerca de posibles prevenciones, se siguen suscitando innumerables delitos por parte de personas que pueden tener acceso a programas y equipos dañándolos total o parcialmente. Es por esta razón que se debe analizar la

posibilidad de implantar nuevos mecanismos que deben incluir tanto la parte física como la parte lógica.

De ahí, que en años recientes el estudio de las bases teóricas y de las implementaciones prácticas para asegurar la confidencialidad, la integridad y disponibilidad en el uso de la información y sobretodo en el intercambio de ésta para que siga teniendo un gran auge. La ciencia que se encarga de estos aspectos y en general del diseño de procedimientos para cifrar información de carácter confidencial es la Criptología.

La principal forma de protección va a venir de la mano de la Criptología. El cifrado de los datos va a permitir desde proteger el correo personal para que ningún curioso lo pueda leer, hasta controlar el acceso a los archivos de forma que sólo personas autorizadas puedan examinar o modificar su contenido, pasando por proteger las claves cuando se necesita conectarse a un sistema remoto o los datos bancarios cuando se realiza una compra a través de Internet.

La Criptología es una de las ciencias consideradas como más antiguas, ya que sus orígenes se remontan al nacimiento de nuestra civilización. Su uso original era el proteger la confidencialidad de informaciones militares y políticas, pero en la actualidad es una ciencia interesante no sólo en esos círculos cerrados, sino para cualquiera que esté interesado en la confidencialidad de los datos. Actualmente existe gran cantidad de software y hardware destinado a analizar y monitorizar el tráfico de datos en redes de computadoras, si bien estas herramientas constituyen un avance en técnicas de seguridad y protección, su uso indebido es al mismo tiempo un grave problema y una enorme fuente de ataques a la intimidad de los usuarios y a la integridad de los propios sistemas.

Se puede decir que la Criptología tiene dos objetivos fundamentales:

- Busca ocultar la información de carácter confidencial para protegerla cuando es transferida a través de las comunicaciones que se efectúan por medio de los denominados canales inseguros, como lo son el teléfono, tarjetas electrónicas o computadoras. A esta rama se le denomina Criptografía.
- Busca descubrir la información que se encuentra oculta sin ser el usuario autorizado para conocerla. Llamado Criptoanálisis.

La criptografía responde a la necesidad de codificar mensajes que sólo pueda descifrar el destinatario y se ha aplicado tanto a defensa, como a secretos industriales y en los últimos años al comercio electrónico.

Existen dos tipos principales de criptografía de uso común hoy día. La más antigua y simple se conoce como criptografía simétrica o de clave secreta, que resulta útil en muchos casos, aunque tiene limitaciones significativas.

Los algoritmos simétricos, o de clave secreta, se caracterizan por ser altamente eficientes y robustos. Se les llama así porque se emplea la misma clave para cifrar y para descifrar. Se basan en el uso de claves secretas que previamente hay que intercambiar mediante canales seguros, con los riesgos que ello supone. Todas las partes deben conocerse y confiar totalmente la una en la otra. Cada una de ellas debe poseer una copia de la clave que haya sido protegida y mantenida fuera del alcance de los demás. Además, dichas claves no se deben utilizar para varios mensajes, ya que si se interceptaran algunos de ellos, se podrían encontrar métodos para descodificarlos. Por sí solo, este tipo de cifrado no es suficiente para desarrollar el pleno potencial del comercio electrónico o de cualquier otro sistema que requiera seguridad. De un lado, resulta poco práctico que una gran corporación intercambie claves con miles o incluso millones de clientes o con personas con los que nunca ha tratado.

La solución a la seguridad en toda red abierta es una forma de codificación más novedosa y sofisticada, desarrollada en los años setenta, y conocida como clave pública o criptografía asimétrica. Al contrario que los anteriores, los algoritmos asimétricos tienen claves distintas para cifrado y descifrado. Por ello, también se les llama algoritmos de clave pública. Permiten eliminar el gran inconveniente de cómo hacer llegar al remitente la clave de cifrado. En el caso de los algoritmos asimétricos se usan una clave pública y una clave secreta. La primera se publica en un tipo de directorio al que el público en general tiene acceso, mientras que la privada se mantiene en secreto. Las dos claves funcionan conjuntamente. De esa manera, una intercepción de la clave pública es inútil para descifrar un mensaje, puesto que para ello se requiere la clave secreta. Cualquier tipo de datos o información que una de las claves cierre, sólo podrá abrirse con la otra. De forma tal que si se quiere enviar a un amigo un mensaje sin que ningún intruso lo lea se busca la clave pública del amigo y se utiliza para realizar el cifrado del texto. Luego, cuando él lo recibe, se utiliza su clave privada para revertir el cifrado del mensaje en la pantalla de su computadora y aparece el mensaje en forma de texto claro. Si un extraño interceptara este mensaje, no podría descifrarlo porque no tendría la clave privada de esta persona.

Como desventaja, las claves han de ser de mayor tamaño para ofrecer una seguridad comparable a la de los algoritmos simétricos. También resultan más lentos y producen mensajes cifrados de mayor tamaño.

Por ejemplo en el sistema criptográfico con clave pública RSA, el cual es el más popular actualmente, los mensajes enviados usando el algoritmo RSA se

representan mediante números y el funcionamiento se basa en el producto de dos números primos grandes (mayores que 10^{100}) elegidos al azar para conformar la clave de descifrado. Esto como se puede apreciar con este algoritmo se consume muchos recursos computacionales, además de que la seguridad de este algoritmo radica en que no hay maneras rápidas conocidas de factorizar un número grande en sus factores primos utilizando computadoras tradicionales pero la computadora cuántica podría darle una solución a este problema.

Una de las técnicas recientemente utilizadas dentro de los sistemas de clave pública es el denominado criptosistema de curvas elípticas (ECC por sus siglas en inglés), propuesta de forma independiente por Neal Koblitz y Victor Miller en 1985. Se han demostrado sus capacidades para cifrar información y además muestra mejores condiciones de seguridad, eficiencia en el uso de los recursos computacionales y menor uso de la memoria.

Los sistemas basados en curvas elípticas son los criptosistemas más recientes dentro del campo de los sistemas de clave pública y representan sólo otra forma de implementar métodos de logaritmo discreto.

Además de presentar algunas ventajas teóricas son muy prácticas. No existe ningún algoritmo rápido de cálculo de un logaritmo, lo cual supone que el tamaño de la clave, así como las firmas digitales y mensajes cifrados obtenidos son pequeños. De hecho, los criptosistemas basados en curvas elípticas proporcionan la misma seguridad que los basados en la factorización del logaritmo discreto, tal como RSA, reduciendo considerablemente el número de dígitos utilizados.

Las curvas elípticas pueden ser implementadas con gran eficiencia en hardware y software, y son capaces de competir en velocidad con sistemas como RSA. En general se cree que son bastante seguros, pero no ha sido demostrado.

La seguridad de los sistemas de criptografía con curvas elípticas es buena a priori y, pese al esfuerzo realizado para intentar atacarlos, hasta el momento no ha habido ninguna sorpresa.

Los factores que promueven la adopción de la criptografía de curvas elípticas son claves más pequeñas, cómputos más rápidos, menor consumo de energía, menos memoria, incluso cuando la memoria es barata, la diferencia entre cientos de bits y miles de bits es notoria cuando existen billones de billones de claves en el mundo.

Una de las aplicaciones especialmente atractivas se relaciona con los dispositivos diminutos. El problema es que, sin importar que tan pequeño es el dispositivo que se tiene que proteger, los oponentes van a atacarlo con las computadoras más grandes que pueden conseguir. Básicamente, independientemente

de lo pequeño sea su sistema de cómputo, será necesario protegerlo con los criptosistemas más sólidos. Allí es donde interviene la criptografía de curvas elípticas.

Como cada vez se conectan a Internet dispositivos más pequeños y a medida que el comercio electrónico y otras comunicaciones Web seguras continúan creciendo, la criptografía de curvas elípticas se vuelve cada vez más atractiva.

La empresa Certicom edita un boletín de seguridad y criptografía en el que se encuentran algunos trabajos realizados actualmente, como lo son la adopción dentro de estándares globales dentro del ancho mundo del comercio electrónico ya que la Criptografía de Curvas Elípticas provee aplicaciones igual de buenas para la industria como para el sector financiero y el sistema postal.

Por ejemplo el gobierno Chino esta considerando a la Criptografía de Curvas Elípticas como el camino para arreglar la seguridad en el uso de redes inalámbricas usadas en su país. Los niveles de seguridad mostrados hacen que sea ideal para resolver problemas como éste.

Una aplicación importante es la de pagos electrónicos y verificación o difusión de cheques, éstos se vuelven cada vez más atractivos para la industria financiera ya que los beneficios los pueden llevar a la automatización, lo cual reducirá los costos de oficina pero además mejorará los servicios a los clientes.

Las firmas pueden ser tan pequeñas como de 20 bytes para un mensaje original, lo cual es seis veces más pequeño que con RSA lo cual lo hace más eficiente.

Otro rasgo específico para la criptografía de curvas elípticas es la habilidad para ajustarse a los niveles de seguridad, dependiendo de los requerimientos.

Además es más conveniente para ambientes rudos como aquellos en los que las tarjetas y las computadoras personales típicamente trabajan.

La criptografía de curvas elípticas puede mejorar las comunicaciones en Internet ya que es la mejor opción para cuando el funcionamiento nos preocupa. Además utiliza menos ancho de banda que sistemas criptográficos alternativos.

Desde el planteamiento de la criptografía basada en curvas elípticas, se han implementado numerosas arquitecturas en ambientes como software de alto nivel en diferentes sistemas operativos y en dispositivos de lógica programable.

En general la criptografía de curvas elípticas produce puestas en práctica más eficientes que otros sistemas de la llave pública debido a su fuerza adicional:

Eficacias de almacenaje, menor ancho de banda y eficiencia de cómputo. Esto conlleva a velocidades más altas, un consumo de energía más bajo y reducciones de tamaño de código.

Lo anterior permite por mucho que los algoritmos de la Criptografía de Curvas Elípticas sean idóneos para ser implementados en donde el poder de cómputo y el espacio del circuito sea reducido, donde sea requerida una alta velocidad de procesamiento o grandes volúmenes de transacciones, donde el espacio de almacenamiento, la memoria o el ancho de banda sea limitado. Lo que permite su uso en teléfonos celulares, fax, organizadores de palma, PCs y en memorias USB.

Por todo lo anterior es necesario que las nuevas generaciones de especialistas del cómputo y la información conozcan y aprovechen las bondades de la Criptografía de Curvas Elípticas por lo que se vuelve importante desarrollar herramientas de apoyo para el aprendizaje de este conocimiento que ya se considera en los nuevos planes de estudio de la carrera de Ingeniería en Computación dentro de la Facultad de Ingeniería; así, los objetivos del presente trabajo de tesis son:

- Establecer las bases teóricas referentes a la criptografía de curvas elípticas, de tal forma que puedan servir de plataforma para futuras aplicaciones que requieran protección de la información, en las cuales la transferencia de ésta se lleve a cabo bajo ambientes rudos, además de que el poder de cómputo esté reducido y el ancho de banda sea limitado.

- Comparar la criptografía de curvas elípticas con el sistema de clave pública más utilizado actualmente tal como lo es RSA, estableciendo las ventajas y desventajas de utilizar uno u otro sistema de cifrado y acentuando las principales diferencias.

- Clasificar las aplicaciones que sean factibles de implementar tanto en software como en hardware.

- Desarrollar un sistema que permita a los interesados en criptografía familiarizarse con la Criptografía de Curvas Elípticas, programando los principales algoritmos que utilizan este tipo de sistema.

Para llevar a cabo el desarrollo de la tesis, el capítulo 1 “Estado del arte de la criptología”, se diseñó de forma tal que se inicia con una introducción a los conceptos básicos de criptología y las características más sobresalientes de las dos ciencias que la componen, la criptografía y el criptoanálisis. También se mencionan las características de los principales algoritmos de la criptografía simétrica y asimétrica, así como las herramientas utilizadas por el criptoanálisis.

En el capítulo 2, “Fundamentos matemáticos de la criptografía asimétrica”, se trabaja con los campos finitos, los cuales son un importante prerrequisito para trabajar con criptografía asimétrica y especialmente para trabajar con algoritmos basados en curvas elípticas, se menciona el algoritmo para intercambiar claves de Diffie-Hellman, el algoritmo asimétrico más utilizado RSA y finalmente los algoritmos fundamentales en la criptografía de curvas elípticas.

El capítulo 3, “Importancia de la CCE en el aprovechamiento de los recursos computacionales y la seguridad en elementos reducidos”, se hace una comparación entre los algoritmos basados en curvas elípticas y los criptosistemas RSA, también se estudian algunas de las aplicaciones de la criptografía de curvas elípticas y posibles implementaciones de ésta en el futuro basados en que ofrecen el mismo nivel de seguridad que otros sistemas de cifrado asimétrico pero con claves de menor tamaño.

En el capítulo 4, “Sistema de aprendizaje de criptografía de curvas elípticas”, se habla precisamente del software desarrollado en forma de tutorial para que todo aquel interesado en criptografía pueda comprender conceptos y algoritmos fundamentales de criptografía de curvas elípticas. Aquí se programaron los algoritmos principales y se hace mención de detalles importantes para la criptografía basada en curvas elípticas.

Finalmente, se presentan las conclusiones.

Capítulo 1

Estado del arte de la criptología

La Criptología es la ciencia encargada de las comunicaciones en forma segura y usualmente en forma secreta. Engloba dos disciplinas opuestas y a la vez complementarias éstas son la Criptografía y el Criptoanálisis[1]. La primera se refiere al estudio y aplicación de los principios y técnicas por las cuales la información se presenta de forma ininteligible para todos excepto para el receptor previsto, para lo cual se hace uso de las transformaciones matemáticas. Mientras que el segundo es la ciencia y arte de revelar la información escondida por la criptografía usando técnicas analíticas y matemáticas para recobrar la información sin ser el usuario autorizado para conocerla.

En la actualidad esta definición necesita ser extendida para los propósitos de la criptología moderna, poniendo especial atención en el diseño y evaluación de un amplio rango de técnicas y métodos para la protección de la información. En esta protección se debe cubrir no solamente la confidencialidad, también es importante la autenticación, integridad, disponibilidad, el no repudio y el control de acceso, es decir, todos los servicios de seguridad que se le puede dar a la información. Para esto necesitamos conocer detalles específicos del sistema, entiéndase por esto el entorno de seguridad. Para llevar a cabo la detección de esto nos podemos ayudar enfocándonos a responder tres preguntas: ¿qué se quiere proteger?, ¿de qué se quiere proteger? y ¿cómo se va ha proteger?

[1] Nota: Un criptoanalista utiliza técnicas matemáticas para romper el cifrado. Una persona puede romper el cifrado, por ejemplo, espiando a los usuarios del sistema, robando la llave para descifrar, sobornando al criptógrafo y muchas otras técnicas. El criptoanálisis va más allá y se refiere a utilizar diferentes disciplinas conocidas como las matemáticas, la capacidad de procesamiento de la computadora o la inteligencia.

Es evidente que para dar respuesta a estas preguntas es necesario hacer un análisis detallado del sistema a proteger, pero a pesar de que la última pregunta puede responderse desde distintos puntos de vista, desde el punto de vista de la Criptología la solución se encuentra en la ya mencionada Criptografía.

La Criptografía provee a un sistema con herramientas que pueden encargarse de la protección de las comunicaciones consiguiendo así la seguridad de la información de carácter confidencial.

1.1 Criptografía

Como ya se mencionó la parte de la criptología que trata con el diseño de algoritmos, protocolos y los sistemas que son utilizados para la protección de la información contra amenazas específicas es la llamada Criptografía.

La Criptografía se ocupa del diseño de procedimientos para cifrar o enmascarar una determinada información de carácter confidencial, para esto utiliza una serie de herramientas básicas como algoritmos de cifrado, códigos de autenticación, esquemas de firma digital, distribución de claves, generación de bit aleatorio, etcétera. Gracias a estas herramientas elementales es posible crear nuevas herramientas y servicios más complejos tales como algoritmos de cifrado robustos, una gran variedad de protocolos orientados a las aplicaciones, como lo son sistemas de pago electrónico, elecciones electrónicas, protocolos de autenticación, de generación de claves y de comercio electrónico. Cada herramienta se caracteriza por sus especificaciones de seguridad las cuales usualmente indican la configuración recomendada y su resistencia contra amenazas específicas, tales como espionaje o modificación ilegal de la información. El diseño puede utilizar todas las herramientas previstas por la Criptografía para combinarlas y utilizarlas en una solución única.

Además la criptografía como medio de proteger la información es un arte tan antiguo como lo es la misma escritura. Permaneció durante mucho tiempo muy estrechamente ligada con el entorno militar y con fines diplomáticos, ya que éstos eran los que en un inicio tenían la necesidad de utilizarla. Claramente se ha podido ver que en la actualidad la situación ha cambiado de forma drástica ya que el desarrollo de las comunicaciones electrónicas y al surgimiento de la computación comercial ha hecho posible la transmisión y almacenamiento de enormes cantidades de información confidencial la cual es necesaria proteger. Por esta razón la criptología se ha convertido en una necesidad real del hombre, el cual ve en la falta de protección de sus datos una amenaza contra su propia intimidad.

En la siguiente figura[2] 1.1 se muestra el proceso criptográfico de cifrado y descifrado.

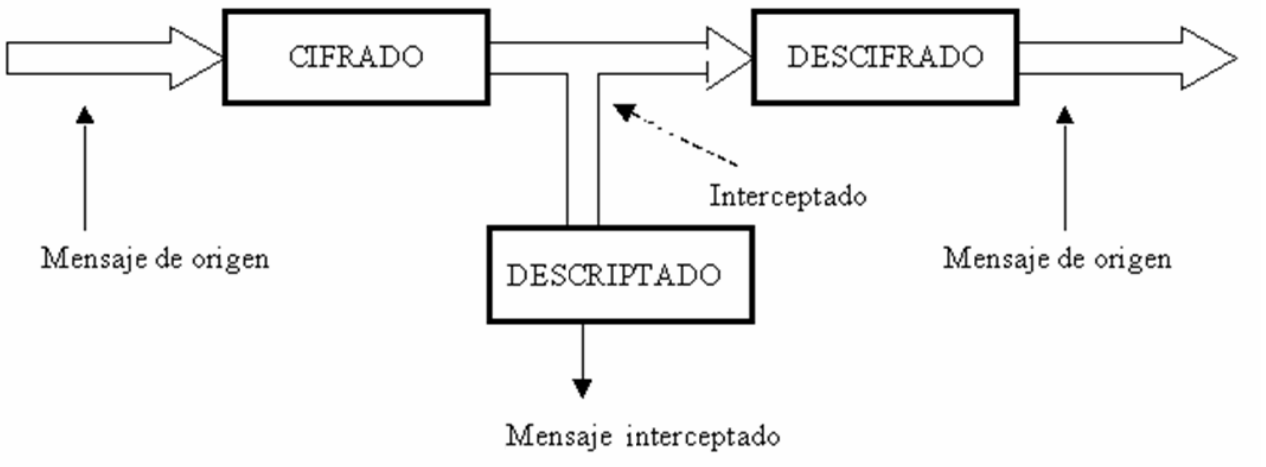


Figura 1.1 Sistema criptográfico

Teniendo el emisor y receptor de un determinado mensaje, se dice que el emisor transforma el mensaje original a través de un procedimiento de cifrado para lo cual utiliza una clave convirtiéndolo en un mensaje cifrado que será enviado por un canal inseguro. Cuando el receptor capta el mensaje utiliza la clave para transformar ese criptograma en el texto original, es decir, hace un proceso de descifrado recuperando así la información original. Un criptoanalista por su parte intenta hacer el descriptado para obtener la información original de manera ilícita.

La criptografía tiene tres principales finalidades, una es mantener la confidencialidad de la información, ya que el mensaje debe permanecer oculto, este aspecto fue la principal preocupación de la criptografía clásica. Pero la Criptografía de hoy en día debe cubrir la segunda y tercera finalidad principal que es garantizar la autenticidad tanto del criptograma como del remitente, esto quiere decir que el criptograma y el remitente comprueben que el mensaje es de quién dice ser. La siguiente finalidad de la criptografía moderna es la de mantener la integridad de la información de forma que ésta no sufra alteraciones.

Una de las diferencias fundamentales entre la Criptografía clásica y la Criptografía moderna es el concepto de seguridad, ya que en la primera los procedimientos de cifrado tenían una seguridad probable y hoy en día gracias a los

[2] El original es de: Alfao Fúster Sabater Amparao "Técnicas Criptográficas de protección de datos", mega, segunda edición, Mexico, D.F., 2001. La imagen que se presenta en el presente trabajo fue modificada del original.

modernos equipos de cómputo y a sus velocidades de procesamiento cada día más rápidos, los procedimientos de cifrado han de tener seguridad matemática demostrable.

Ya que las comunicaciones electrónicas se usan hoy en día para casi todas las actividades de interés están expuestas a todos los trucos, amenazas y manipulaciones que derivan de las debilidades del ser humano. Si tuviéramos honradez y confianza mutua, la Criptografía no tendría razón de ser, pero por falta de ellas, la Criptografía trata de suplirlas con protocolos y algoritmos matemáticos de seguridad demostrable.

1.1.1 Técnicas clásicas de cifrado

En la criptografía clásica tenemos dos técnicas básicas de cifrado que han sido utilizadas por muchos años, hasta llegar a nuestros días. Estas dos técnicas son la técnica de sustitución y la de transposición.

La técnica de sustitución básicamente consiste en establecer una correspondencia entre las letras del alfabeto en el cual se escribe el mensaje original con otros elementos pertenecientes a otro conjunto que pueden ser del mismo alfabeto o de uno distinto. Así cada letra del mensaje en claro se sustituye por su símbolo correspondiente en la elaboración del criptograma. Una vez que el legítimo receptor conoce previamente la correspondencia establecida, sustituye cada símbolo del criptograma por el símbolo correspondiente al alfabeto original, recuperando de esta forma el mensaje en claro.

Dentro de los métodos de sustitución podemos encontrar los siguientes:

Sustitución mono alfabética

El procedimiento de sustitución más antiguo que se conoce es el hoy llamado “cifrado de César” el cual fue utilizado por los romanos, en él la letra A se representa por la letra D, B por E, C por F, y así sucesivamente para cada letra del abecedario hasta sustituir Z por C. El algoritmo de César, llamado así porque era el que empleaba Julio César para enviar mensajes secretos, es uno de los algoritmos criptográficos más simples. Si asignamos a cada letra un número (A = 1, B = 2,. . .), y consideramos un alfabeto de 26 letras, la transformación criptográfica sería:

$$C = (\text{Mensaje} + 3) \bmod 26$$

Obsérvese que este algoritmo posee una clave única y constante y por ello es que la transformación siempre es la misma. Obviamente, para descifrar basta con restar 3 al número de orden de las letras del criptograma y aplicarle la operación módulo 26.

Ejemplo:

Asumiendo un alfabeto de 26 símbolos como el se presenta en la tabla 1.1.

Tabla 1.1 Alfabeto para cifrado del César

A	B	C	D	E	F	G	H	I	J	K	L	M	N	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

Vamos a cifrar el siguiente mensaje: OMAR

Podemos hacerlo manualmente o utilizado la expresión anteriormente dada:

1. Reemplazar **Mensaje** por el valor de la primera letra, en este caso Z equivale a 26.
2. Realizar la operación indicada: **O = (15 + 3) mód 26 = 18**.
3. Realizar la operación con las letras restantes.

Mensaje cifrado: RPDU

En el caso general del algoritmo de César. Su transformación sería:

$$\mathbf{E(a, b)\,(M) = (aM + b)\ mód\ N}$$

Siendo **a** y **b** dos números enteros menores que el cardinal **N** del alfabeto, y cumpliendo que **mcd(a, N) = 1**. La clave de cifrado **k** viene entonces dada por el par **(a, b)**. El algoritmo de César sería pues una transformación afín con **k = (1; 3)**.

A este sistema se le conoce como sustitución monoalfabética porque la única clave se escoge dentro de una cadena de 26 letras correspondiente al alfabeto completo.

En una sustitución monoalfabética, cada letra del texto original es cambiada por otra de acuerdo con una tabla y con su posición del texto. La sustitución de César es un ejemplo de sustitución monoalfabética, que consiste en cambiar cada letra por otra que está en orden alfabético 3 letras adelante.

Se pueden usar otros valores en vez de 3, lo que constituye una clave para cifrar. Existen apenas 26 claves, pero este método basta para proteger textos con pequeño grado de confidencialidad. Se tiene una clave que dice cuál de las tablas será usada para cada letra del texto original. Por lo tanto, cuanto mayor sea la clave, más seguro es el método. Entretanto, es suficiente descubrir el tamaño de la clave k y analizar bloques de k caracteres del texto, verificando la frecuencia de repetición de los caracteres.

Sustitución por desplazamiento

En una sustitución por desplazamiento una clave indica cuántas posiciones alfabéticas se deben avanzar para sustituir cada letra. Sería diferente a la sustitución de César, las letras no son cambiadas siempre por una letra cada n posiciones del alfabeto. Es decir, el desplazamiento es variable, mientras que en el método César es fijo.

Hay que resaltar que cada dos dígitos de la clave corresponde al desplazamiento hacia delante que se va a realizar con respecto a la letra del texto cifrado. Se toman los dígitos de dos en dos debido a que el alfabeto que se utiliza es de 26 letras, por lo tanto, se necesitan dos dígitos para formar el desplazamiento.

Sustitución poli alfabética

Otro sistema de sustitución que se conoce es el polialfabético, que es el resultado de introducir múltiples alfabetos de cifrado que se utilizan en rotación de acuerdo con un criterio o clave, su objetivo es adecuar las frecuencias del texto cifrado de tal forma que las letras con mayor frecuencia de aparición no sobresalen tan claramente.

Dentro de este sistema se tiene el cifrado Vigénere, en este tipo de sistema el principal elemento es la llamada tabla de Vigénere, una matriz de caracteres cuadrada que se muestra a continuación en la tabla 1.2.

Tabla 1.2 Tabla de Vigénere

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
A	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
B	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
C	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
D	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
E	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
F	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
G	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
H	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
I	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
J	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
K	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
L	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
M	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
N	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
O	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
P	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
Q	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
R	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
S	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
T	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
U	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
V	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
W	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
X	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
Y	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
Z	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

La clave del sistema de cifrado de Vigénere es una palabra de letras del alfabeto, esta palabra es un elemento del producto cartesiano, que es justamente el alfabeto del criptosistema de Vigénere. De esta forma, el mensaje a cifrar en texto claro ha de descomponerse en bloques de elementos, es decir, letras y aplicar sucesivamente la clave empleada a cada uno de estos bloques, utilizando la tabla anteriormente proporcionada.

Ejemplo:

Aplicación del criptosistema de Vigénere podemos cifrar la frase "La abrumadora soledad del programador" utilizando la clave "prueba". En primer lugar nos tenemos que fijar en la longitud de la clave, que es de seis caracteres, por lo que se debe descomponer la frase en bloques de longitud seis, aunque el último bloque es de longitud tres, esto no afecta para nada al proceso de cifrado:

laabru madora soleda ddelpr ograma dor

Ahora se aplica a cada bloque la clave "prueba" y buscamos los resultados como entradas de la tabla de Vigénere:

laabru	madora	soleda	ddelpr	ograma	dor
prueba	prueba	prueba	prueba	prueba	pru
arufsu	brxssa	hffiea	suypqr	dxlena	sfl

Por ejemplo, la primera "a" del texto cifrado corresponde a la entrada l y p la cual encuentra su equivalentemente en la tabla de Vigénere con la letra a. Finalmente, vemos que el texto cifrado ha quedado: "arufsu brxssa hffiea suypqr dxlena sfl".

Otra de las formas de dar mayor complejidad al cifrado es utilizar una clave que sea de mayor longitud que la del texto en claro. El cifrado Vernam representa el caso límite del cifrado de Vigénere pues emplea un alfabeto binario y escoge como clave una cadena de bits aleatoria. Después, se convierte el texto en claro en una cadena de bits. Posteriormente se aplica un OR EXCLUSIVO, bit por bit, con estas 2 cadenas; de este modo, el texto cifrado no puede desbaratarse puesto que todos los posibles textos en claro son candidatos, igualmente probables y no le proporcionará ninguna información al criptoanalista. Para recuperar el mensaje original se realiza la operación OR EXCLUSIVO de la secuencia aleatoria con la clave al criptograma.

El OR EXCLUSIVO sirve tanto para el cifrado como para el descifrado, ya que:

Clave: 101110…

Texto en claro: Texto cifrado Texto en claro

101101… 000011… 101101…

Para la operación de descifrado en el extremo receptor se precisa disponer en forma sincronizada de la misma secuencia de dígitos binarios de la clave.

El texto cifrado es el OR EXCLUSIVO del texto en claro y de la clave, pero la aplicación de otro OR EXCLUSIVO al texto cifrado y a la clave lleva de nuevo al texto en claro. Se observa también que el OR EXCLUSIVO de los textos cifrado y en claro da la clave.

Las ventajas que tiene este método, conocido como clave de una sola vez, son las siguientes:

- Debido a que la clave es arbitraria y de la misma longitud del texto en claro, cuando no se tiene la clave y se desea obtenerla será necesario probar con cada una de las combinaciones posibles que nos permita la longitud de la clave.
- No sólo se puede utilizar el código ASCII, sino que se puede establecer cualquier otro código de tal manera que cada letra no siempre tendrá la misma equivalencia del código ASCII.
- Con los puntos anteriores se puede decir que este cifrado es más completo y complejo en comparación a los cifrados descritos anteriormente.

La técnica de transposición consiste en revolver los símbolos del mensaje original colocándolos sólo en un orden distinto, de tal forma que el criptograma contenga los mismos elementos del mensaje en claro, pero ordenándolos de tal forma que resulten incomprensibles. La persona a la que va dirigido el mensaje sabe de la transposición y reacomoda todos los símbolos desordenados del criptograma para recuperar el mensaje original.

En las técnicas de transposición podemos encontrar los siguientes:

Transposición simple

En esta técnica el mensaje en claro se escribe como texto corrido en secuencia diagonal. Por ejemplo para cifrar el mensaje “a las ocho de la noche se oculta el sol”, se divide el texto en dos renglones y escribimos:

a a o h d l n c e e c l a l o

l s c o e a o h s o u t e s l

Nuestro mensaje cifrado es: aaohdlnceelalolscoeaohsoutesl

Transposición doble

Una transposición simple es fácil de reconocer porque tiene la misma frecuencia de letras que el mensaje original y puede leerse claramente el texto si se divide en 2 el mensaje cifrado, para hacer aún más segura la información se hace dos veces el procedimiento anterior:

a a o h d l n c e e c l a l o

l s c o e a o h s o u t e s l

La primera transposición había quedado: aaohdlnceelalolscoeaohsoutesl

a o d n e l l l c e o s u e l

a h l c e a o s o a h o t s

El mensaje cifrado quedará: aodnelllceosuelahlceaosoahots.

Se puede ver que con una transposición doble es un poco más complicado identificar el mensaje en claro.

Transposición por columnas

La forma más común es la transposición columnas. Para llevarla a cabo hay que elegir como clave una palabra que no contenga letras repetidas. En el ejemplo será NETWORK. La clave nos permite numerar las columnas, ordenando las letras según su proximidad al comienzo del alfabeto. El texto en claro se escribe horizontalmente, en filas. El texto cifrado se lee por columnas, comenzando por la columna de numeración más baja y en orden creciente.

N E T W O R K

3 1 6 7 4 5 2

Texto normal: esto es un ejemplo de cifrado por transposición

e s t o e s u

n e j e m p l

o d e c i f r

a d o p o r t

r a n s p o s

i c i o n

Texto cifrado: seddaculrtsenoariemiopnspfrotjeonioecpso.

Como se ha mencionado anteriormente, la seguridad de la Criptografía clásica era probable, pero en la actualidad los procedimientos de la Criptografía moderna han de tener una seguridad matemáticamente demostrable, es por esto que nace la necesidad de utilizar la Criptografía simétrica.

1.1.2 Criptografía simétrica

Los principios básicos utilizados en los primeros criptosistemas fueron la sustitución y transposición de la secuencia de caracteres. Pero esto ha cambiado y actualmente se utilizan algoritmos matemáticos más complejos.

En el cifrado de tipo simétrico, el emisario y el receptor tienen la misma clave para cifrar y descifrar. Dentro de los cifrados simétricos existen dos tipos: cifrado en flujo y el cifrado en bloque. La diferencia entre ambos es que en el cifrado en flujo se hace bit a bit tal y como se vio en las técnicas clásicas de cifrado analizadas en el apartado anterior y que dan una idea clara de lo que implica este tipo de cifrado ya que los usuarios tienen que mantener en secreto tanto la clave como el dispositivo a cifrar. Por otro lado en lo que se refiere al cifrado en bloque, lo único que debe mantenerse secreto es la clave. En ambos casos, la clave secreta debe ser compartida por los usuarios, con los riegos que esto implica como pérdida, robo, revelación de la misma, etcétera.

En esta sección se va a profundizar en el cifrado en bloque.

Se denomina cifrado en bloque aquel en el que se cifra el mensaje original agrupando los símbolos en grupos o bloques de dos o más elementos.

Todos los cifrados en bloque se componen de cuatro elementos, el primer elemento es la transformación inicial que puede tener una o dos funciones: la primera consiste en aleatorizar simplemente los datos de entrada, careciendo de significado criptográfico si no depende de la clave.

El segundo elemento son las iteraciones intermedias que consisten en una función no lineal complicada de los datos y la clave, que puede ser unidireccional o no. La función no lineal puede estar formada por una operación muy compleja o por la repetición de varias transformaciones simples.

Las iteraciones intermedias se enlazan por sumas módulo 2 bit a bit con los datos que vienen de la transformación inicial o de las vueltas anteriores, de ésta forma se hace posible que se produzca un retroceso cuando se repite el proceso de forma idéntica, obteniéndose de esta forma los datos de partida.

En las iteraciones intermedias no se han de formar un grupo, para que el conjunto de varias pasadas sucesivas con sus subclaves correspondientes no sean equivalentes a una pasada única con una subclave diferente lo que sería un desastre.

El tercer elemento de los componentes de cifrado en bloque es la transformación final y sirve para que las operaciones de cifrado y descifrado sean simétricas. Cuando las vueltas de cifrado son de una sola operación, separadas por sumas módulo 2 bit a bit, esta transformación se limita a realizar la operación inversa de la transformación inicial. Sin embargo, en los sistemas donde las vueltas de cifrado acaban con una operación que afecta a todos los bits del bloque, la transformación de salida debe realizar tanto la función inversa de esta operación como la inversa de la transformación inicial.

El último elemento es el algoritmo de expansión de la clave el cual tiene por objeto convertir la clave de usuario, normalmente de longitud limitada entre 32 y 256 bits, en un conjunto de subclaves anteriores o siguientes. Además, se ha de cuidar que las subclaves producidas no constituyan un pequeño subconjunto monótono de todas las posibles.

Los algoritmos simétricos, o de clave secreta, se caracterizan por ser altamente eficientes con relación al tamaño de su clave y robustos. Como se puede inferir se les llama así porque se emplea la misma clave para cifrar y para descifrar.

Entre los principales algoritmos de cifrado simétrico se pueden encontrar los que siguen:

DES (Data Encription Standard)

Es un algoritmo diseñado por IBM y utilizado habitualmente desde los años 70. Es un método de cifrado altamente resistente frente a ataques criptoanalíticos diferenciales. Por desgracia, su tamaño de clave (56 bits) la hace vulnerable a ataques de fuerza bruta. Un reciente ataque, "patrocinado" por la empresa de seguridad RSA Data Security Inc. contra un mensaje con cifrado DES requirió el uso de cientos de ordenadores durante 140 días. Sin embargo, hay diseños de máquinas que, con un costo de un millón de dólares, podrían descifrar mensajes DES en cuestión de minutos. Quizá por eso el gobierno de EEUU lo utiliza solamente para cifrar datos no clasificados. En la actualidad ofrece protección contra el pirata informático habitual, pero no contra un esfuerzo masivo por parte de un usuario con grandes recursos.

TDES (Triple DES).

Se le llama de esta forma al algoritmo que hace tres veces el cifrado del DES, fue desarrollado por IBM en 1978. En términos generales utiliza el mismo algoritmo que DES, sólo que se efectúa tres veces cada ocasión con claves diferentes. Cuando se descubrió que una clave de 56 bits no era suficiente para un ataque de fuerza bruta, TDES fue elegido como forma de agrandar el largo de la clave sin tener que cambiar el algoritmo de cifrado. La mayoría de las tarjetas de crédito y otros medios de pago electrónico tienen como estándar el algoritmo TDES.

AES (Advanced Encrytion Standard)

Es un esquema de cifrado por bloques adoptado como estándar de cifrado por el gobierno de Estados Unidos. El AES fue anunciado por el Instituto Nacional de Estándares y Tecnología (NIST) el 26 de noviembre de 2001 después de un proceso de estandarización que duró 5 años. Es uno de los algoritmos más populares usados en criptografía simétrica. AES es rápido tanto en software como en el hardware, es relativamente fácil de implementar y requiere poca memoria. Se está utilizando actualmente a gran escala ya que puede llegar a ser hasta 6 veces más rápido que TDES.

Blowfish

Fue creado por Bruce Schneier. Utiliza claves de hasta 448 bits y, hasta el momento, ha resistido con éxito todos los ataques. Por ello y por su estructura se le considera uno de los algoritmos más seguros, a pesar de lo cual no se utiliza

masivamente. Tal vez se deba a su relativa juventud. Su autor no ha patentado el método para que éste pueda ser empleado sin limitaciones.

CAST (Carlisle Adams y Stafford Tavares)

Tiene estructura similar a la de Blowfish. Parece ser un buen algoritmo, aunque tampoco lleva el tiempo suficiente como para haber sido atacado hasta la saciedad. De momento, sus posibilidades son buenas. Se conocen ataques criptoanalíticos contra la versión de clave 64 bits, aunque distan mucho de ser eficaces (requieren 2^{17} textos sin cifrar y 2^{48} cálculos diferentes). No se conocen ataques contra la versión de 128 bits. Ha sido patentado por Entrust Technologies, quienes permiten el uso libre de este algoritmo.

IDEA (International Data Encription Algorithm)

Ha sido desarrollado por Xuejia Lay y James Massey. A pesar de que solamente lleva unos años en uso, es probablemente el mejor algoritmo de bloques existente. Utiliza clave de 128 bits y se cree que es resistente al criptoanálisis. Se encuentra bajo patente de Ascom-Tech, aunque se permite su uso gratuito para aplicaciones no comerciales.

RC2

Es un código protegido bajo secreto comercial (aunque no patentado) por RSA Data Security Inc. Existen ataques criptoanalíticos que, aunque requieren de gran cantidad de texto cifrado, muestran las vulnerabilidades de RC-2. Existen versiones mejoradas, y hoy día RC2 tiende a utilizarse cada vez menos en beneficio de su "hermano mayor" RC4.

RC4

Es un intento de reemplazar RC2 por un algoritmo más sólido. También es un secreto comercial, aunque (al igual que RC2) su código fuente ha sido publicado en grupos de discusión. No se conocen ataques contra él. Forma una parte vital del sistema de cifrado en capas SSL, ampliamente utilizado en navegadores de Internet tales como Netscape Navigator y Microsoft Internet Explorer. Por desgracia, la versión exportable de RC4 tiene una clave de solamente 40 bits, lo que lo hace altamente vulnerable a ataques de fuerza bruta. La versión EEUU, con clave de 128 bits, es segura.

RC5

Fue diseñado por Ron Rivest y se encuentra bajo patente de RSA Data Security Inc. Es relativamente nuevo, y se conocen ciertos tipos de ataques contra él. Asimismo existe un cierto número (pequeño) de claves débiles que no deben utilizarse. A pesar de ello, se le considera un sistema seguro.

SAFER

Es un algoritmo diseñado por Robert Massey (uno de los creadores de IDEA). Tiene claves de hasta 128 bits y, a pesar de algunas debilidades en la primera versión y de ciertos ataques, parece un algoritmo seguro. Este programa fue desarrollado para la empresa Cylink, que algunos ligan a la no muy querida Agencia de Seguridad Nacional Norteamericana (NSA); por ello, hay quien no se fía.

Como se puede ver la criptografía simétrica en términos generales es confiable, sin embargo existe un gran riesgo inherente, el de transmitir la clave utilizada por el emisario y el receptor por un canal inseguro por esta razón hoy en día se necesita una técnica mediante la cual el uso y distribución de las claves sea seguro.

1.1.3 Criptografía asimétrica

La criptografía de clave pública fue inventada en 1976 por los matemáticos Whitfield Diffie y Martin Hellman y es la base de la criptografía moderna.

La criptografía asimétrica utiliza dos claves complementarias llamadas clave privada y clave pública. Lo que está codificado con una clave privada necesita su correspondiente clave pública para ser decodificado. Y viceversa, lo codificado con una clave pública sólo puede ser decodificado con su clave privada.

En los cifrados de tipo asimétricos o de clave pública, la característica fundamental es que la clave de descifrado no se puede calcular a partir de la de cifrado.

Las claves privadas deben ser conocidas únicamente por su propietario, mientras que la correspondiente clave pública puede ser dada a conocer abiertamente. Si un usuario quiere enviar a otro un mensaje de forma que sólo el receptor pueda entenderlo, lo codificará con la clave pública del receptor. Éste

utilizará su clave privada, que solo él tiene, para poder leerlo. Con un sistema asimétrico cualquier usuario puede enviar un mensaje cifrado a otro usando la clave pública de este último, pero sólo aquellos que conozcan la clave secreta correspondiente pueden descifrar el mensaje en claro correctamente.

La criptografía asimétrica está basada en la utilización de números primos muy grandes. Si multiplicamos entre sí dos números primos muy grandes, el resultado obtenido no puede descomponerse eficazmente, es decir, utilizando los métodos aritméticos más avanzados en las computadoras más avanzadas sería necesario utilizar durante miles de millones de años tantas computadoras como átomos existen en el universo. El proceso será más seguro cuanto mayor sea el tamaño de los números primos utilizados.

Al contrario que los anteriores, los algoritmos asimétricos tienen claves distintas para cifrado y descifrado. Se han hecho muy populares, entre otras cosas porque elimina un gran inconveniente: cómo hacer llegar al remitente la clave de cifrado. De esa manera, una intercepción de la clave pública es inútil para descifrar un mensaje, puesto que para ello se requiere la clave secreta. Como desventaja, las claves han de ser de mayor tamaño para ofrecer una seguridad comparable a la de los algoritmos simétricos. También resultan más lentos y producen mensajes cifrados de mayor tamaño.

En los algoritmos de clave pública se tienen como principales los siguientes:

RSA (Rivest, Shamir, Adleman)

Es el algoritmo de clave pública más utilizado en nuestros días y uno de los más populares. En principio utiliza claves de cualquier longitud; en la actualidad se emplean claves de 1024 bits, consideradas lo bastante largas como para resistir ataques de fuerza bruta. Su seguridad se basa en la dificultad de factorizar números primos de gran tamaño. En principio se puede deducir la clave secreta conocida la clave pública, pero solamente por medio de la factorización de números de gran longitud. Estuvo patentado en EEUU hasta el año 2000, en el resto del mundo es de uso libre. Es vulnerable a ciertos ataques, pero si se utiliza adecuadamente es un algoritmo seguro.

Hay un detalle que debe tenerse en cuenta. La dificultad intrínseca de factorizar grandes números es un tema abierto. Actualmente el método de factorización más rápido utilizado es la Criba Numérica Especial de Campo, pero no está demostrado que no haya otro mejor. Si se descubre un método de tiempo polinómico (esto es, cuyo tiempo de ejecución dependa del número N de cifras como N^a), cualquier producto de números primos podrá factorizarse con relativa facilidad. Es un tema sin

resolver dentro de la llamada teoría de la complejidad. Mientras tanto todo parece indicar que el algoritmo RSA continuará siendo poco menos que invulnerable.

Diffie-Hellman

Es un algoritmo de intercambio de claves. Una variante conocida como ElGamal funciona como algoritmo de clave pública, se suele conocer dicho algoritmo como Diffie-Hellman. Se basa en el llamado problema de los logaritmos discretos, que se cree es computacionalmente tan complejo como el de la factorización de números primos y al igual que RSA, no está demostrado que el problema de logaritmos discretos no se pueda resolver mediante herramientas matemáticas más poderosas en el futuro. Está siendo utilizado cada vez con más frecuencia.

CCE (Criptografía de Curvas Elípticas)

Son los algoritmos más recientes dentro del campo de los sistemas de clave pública. En general se cree que son bastante seguros, pero no ha sido demostrado. La valía de los sistemas de curvas elípticas permanece hoy por hoy bajo dudas.

En general la criptografía de clave pública es el tema principal del presente trabajo por lo que se profundizará en este tema y específicamente en los algoritmos RSA y Criptografía de Curvas Elípticas en posteriores capítulos.

Los esquemas de clave privada y de clave pública tienen varias ventajas y desventajas algunos son comunes para ambos. Se puede destacar las siguientes:

Ventajas de la criptografía simétrica.

Los algoritmos basados en criptografía simétrica pueden ser diseñados para soportar gran cantidad de datos de entrada, ya que algunas implementaciones de hardware alcanzan rangos de cifrado de miles de megabytes por segundo, mientras que las implementaciones de software pueden lograr rangos de entrada en megabytes por segundo. Se debe tomar en cuenta que las claves con la criptografía simétrica son relativamente cortas. Además puede estar compuesta para producir cifrados fuertes, ya que una simple transformación puede hacer que un sistema sea fácil de analizar pero puede ser utilizada junto con otras para construir un sistema de cifrado mucho más fuerte.

Desventajas de la criptografía simétrica.

En una comunicación de dos usuarios, la llave debe permanecer en secreto por ambas partes. También en una red muy grande pueden existir muchas claves de usuarios para poder administrarlas todas de manera eficiente. En la comunicación de dos partes se debe cambiar la clave frecuentemente y tal vez sea necesario cambiarla en cada sesión. Otra desventaja es que los mecanismos de firma digital que se están utilizando no pueden ser utilizados por la criptografía simétrica.

Ventajas de la criptografía asimétrica.

En los sistemas basados en criptografía asimétrica solamente la clave privada debe permanecer en secreto lo que significa una gran ventaja con respecto a los criptosistemas del tipo que se tiene intercambiar las claves por canales inseguros. También la administración de las claves en una red es mucho más sencilla. Además dependiendo del modo de uso, la llave privada puede permanecer igual por periodos relativamente largos. La clave pública también puede permanecer sin cambios por tiempo más prolongado, dependiendo la forma de utilizarla. Muchos esquemas de clave pública son relativamente eficientes en mecanismos de firma digital. En una red muy grande el número de claves necesarias debe ser considerablemente menor que en el cifrado simétrico.

Desventajas de la criptografía asimétrica.

Los tamaños de las claves son mucho más largos que aquellos utilizados por criptografía simétrica. El esquema de clave privada debe ser emitido de manera segura. Son mucho más lentas con respecto a los criptosistemas de clave privada. Además la criptografía asimétrica no tiene una historia como la criptografía simétrica ya que empezó en 1976 por lo que todavía no ha sido posible estudiarla exhaustivamente.

1.2 Criptoanálisis

Como ya se mencionó el Criptoanálisis es el estudio de los métodos para obtener el sentido de una información cifrada, sin ser el usuario autorizado para obtenerla. Típicamente, esto se traduce en conseguir la clave secreta. Se le conoce a esta práctica como romper o forzar el código.

Criptoanálisis también se utiliza para referirse a cualquier intento de sortear la seguridad de otros tipos de algoritmos y protocolos criptográficos en general, no solamente el cifrado. Cuando alguien diseña un criptosistema, debe de tener en mente todos los posibles ataques que éste puede sufrir, y cada mecanismo de ocultación que implementa está respondiendo a un hipotético procedimiento de Criptoanálisis. Sin embargo, el criptoanálisis suele excluir ataques que no tengan como objetivo primario los puntos débiles de la Criptografía utilizada por ejemplo ataques a la seguridad que se basen en el soborno, el ataque físico y el robo ya que aunque estos tipos de ataques son un riesgo creciente para la seguridad de la información se están haciendo gradualmente más efectivos que el criptoanálisis tradicional.

Los métodos y técnicas del criptoanálisis han cambiado drásticamente a través de la historia de la Criptología, adaptándose a una creciente complejidad criptográfica, que abarca desde los métodos de lápiz y papel del pasado, pasando por máquinas como Enigma, hasta llegar a los sistemas basados en computadoras del presente.

Los resultados del criptoanálisis han cambiado también ya que no es posible tener un éxito ilimitado al romper un código y existe una clasificación jerárquica de lo que constituye un ataque en la práctica. Los métodos utilizados para romper los sistemas asimétricos son radicalmente diferentes de los sistemas simétricos y usualmente implican resolver un problema cuidadosamente construido en el dominio de la matemática pura.

El criptoanálisis ha evolucionado conjuntamente con la criptografía, y la competición entre ambos puede ser rastreada a lo largo de toda la historia de la criptología. Las claves nuevas se diseñaban para reemplazar los esquemas ya rotos, y nuevas técnicas de criptoanálisis se desarrollaban para abrir las claves mejoradas. Hoy en día se suele invitar a la comunidad científica a que trate de romper las nuevas claves criptográficas, antes de considerar que un sistema es lo suficientemente seguro para su uso.

1.2.1 Criptoanálisis básico

El término criptoanálisis fue acuñado por William Friedman en 1920, aunque los métodos para romper códigos y cifrados son mucho más antiguos. La primera explicación conocida del criptoanálisis se debe al sabio árabe del siglo IX, Yusuf Yaqub Ibn Isaac Al-Sabbah Al-Kindi, en su manuscrito para descifrar mensajes criptográficos.

Ataque por fuerza bruta.

Es el ataque más elemental, en él se hace una prueba exhaustiva con todas las claves posibles, para descifrar un criptograma. Aquí se requiere tiempo y paciencia.

El análisis de frecuencias.

Es la herramienta básica para romper los cifrados clásicos. Ya que en todas las lenguas conocidas, ciertas letras del alfabeto aparecen más frecuentemente que otras, pongamos por ejemplo el español en el cual las vocales son muy frecuentes, ya que como se puede ver en la figura 1.2 ocupan alrededor del 45% del texto, mientras que la frecuencia sumada de F, Z, J, X, W y K no alcanza el 2%. En el análisis de frecuencias se revelará el contenido original si el cifrado utilizado no es capaz de ocultar estas estadísticas. Por ejemplo en un cifrado por sustitución simple ya sea monoalfabético o polialfabético, la letra más frecuente en el texto cifrado sería un candidato probable para representar la letra A.

El análisis de frecuencias se basa tanto en el conocimiento de la lengua en que esta escrito el mensaje como en las estadísticas, pero al volverse cada vez más complicados los cifrados, las matemáticas se convirtieron gradualmente en el enfoque predominante en el criptoanálisis.

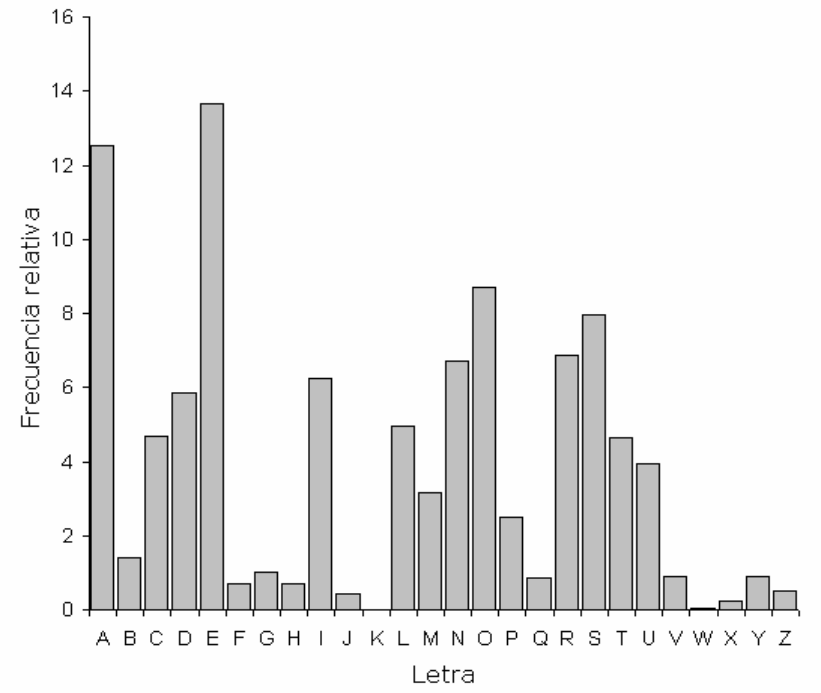


Figura 1.2 Frecuencia en que se utilizan las letras en un mensaje en español

El método Kasiski.

Fue ideado en 1863 con el motivo de romper el cifrado de Vigenere, después de ser considerado irrompible por 300 años. Se basa en la suposición de que la repetición de una cadena de dos o más caracteres a lo largo del criptograma se corresponderán con la misma cadena en el texto plano, por lo que la longitud entre las cadenas será un múltiplo de la longitud de la llave K.

Una vez encontrada la longitud n de la llave K, se divide el texto cifrado en bloques de tamaño n y se realiza un análisis de frecuencias por cada bloque, lo cual es más sencillo que realizarlo a todo el criptograma completo.

Criptoanálisis moderno.

Aunque la computación fue utilizada con gran éxito durante la Segunda Guerra Mundial, también se hizo posible la creación de nuevos métodos criptográficos que eran más complejos y que los seguimos utilizando hasta la fecha. Tomando esto en cuenta la criptografía moderna se ha vuelto mucho más sólida con respecto al criptoanálisis que en el pasado.

Los criptoanálisis exitosos han influido sin lugar a dudas en la historia ya que la capacidad de leer los mensajes que debieran ser secretos o los planes de otros puede ser una ventaja decisiva, y nunca con mayor razón que en tiempos como los actuales en los que la información se ha convertido en nuestro activo más valioso después de los recursos humanos.

1.2.2 Ataques a criptosistemas de clave privada

Los ataques criptoanalíticos varían en potencia y en su capacidad de amenaza para los sistemas criptográficos utilizados en el mundo real. El criptoanálisis puede realizarse bajo una serie de supuestos sobre cuánto puede observarse o descubrirse sobre el sistema en cuestión antes de realizar el ataque.

Existen diversos tipos de ataques y una clasificación posible es la siguiente:

- Ataque con texto de cifrado disponible, en este ataque el criptoanalista sólo tiene acceso a una colección de textos cifrados o codificados.
- Ataque con texto plano conocido, aquí el atacante tiene un conjunto de textos cifrados de los que conoce el correspondiente texto plano o descifrado.
- Ataque con texto cifrado elegido, en éste el atacante puede obtener los textos cifrados (planos) correspondientes a un conjunto arbitrario de textos planos (cifrados) de su propia elección.
- Ataque adaptativo de texto plano escogido, es como un ataque de texto plano escogido, pero el atacante puede elegir textos planos subsiguientes en base a la información obtenida de los descifrados anteriormente.
- Ataque de clave relacionada, es como un ataque de texto plano escogido, pero el atacante puede obtener texto cifrado utilizando dos claves diferentes. Las claves son desconocidas, pero la relación entre ambas es conocida tal vez dos claves que difieren en un bit.

En estos tipos de ataque varían en la plausibilidad de que ocurran en la práctica. Aunque algunos son más probables que otros, los criptógrafos suelen adoptar un enfoque conservador y asumir el peor caso imaginable cuando diseñan algoritmos, razonando que si un sistema es seguro incluso contra amenazas tan poco realistas, entonces debería resistir al criptoanálisis en el mundo real también. Los supuestos en los que se basan estos ataques son a menudo más realistas de lo que podría parecer a primera vista. Para obtener un ataque con texto plano conocido, el criptoanalista podría muy bien conocer o ser capaz de inferir una parte que probablemente forma

parte del texto plano, como por ejemplo el encabezamiento de una carta cifrada. Los ataques de clave relacionada son básicamente teóricos.

Algunas técnicas de criptoanálisis simétricos son las siguientes:

Criptoanálisis diferencial.

Se basa en la observación de pares de texto cifrado, los cuales tienen textos en claro que tienen ciertas diferencias entre sí. Si se estudia la evolución de estas diferencias nos podemos dar cuenta de que pueden atravesar las 16 rondas de criptograma DES, ya que éste se cifra con la misma clave.

Criptoanálisis lineal.

Este ataque utiliza aproximaciones lineales para describir la acción de un cifrado de bloque, es la técnica que intenta encontrar las aproximaciones lineales basadas en las transformaciones que un criptosistema ejecuta en un texto.

Las técnicas diferencial y lineal son conocidas como know plaintext exploit ya que utilizan un texto conocido para ejecutar los ataques.

Criptoanálisis imposible.

La idea básica es buscar diferencias entre pares de textos en claro que se mantienen con cierta probabilidad en las diferencias correspondientes de los textos cifrados que no pueden ocurrir. Resulta ser que estas diferenciales son más poderosas que las ordinarias y pueden usarse para atacar algoritmos[3].

Criptoanálisis Estadístico.

Usa información estadística para romper el texto cifrado, analizándolo y usando la probabilidad de ocurrencia de cada letra en un idioma. Un ejemplo de esto es el ya mencionado criptoanálisis por fuerza bruta en el cual se hace uso de herramientas de cómputo como podría ser un diccionario.

[3] Lo ideal es que el criptoanálisis sea imposible, pero lamentablemente siempre existirá la posibilidad de llevarlo a cabo exitosamente.

1.2.3 Ataques a criptosistemas de clave pública

La criptografía asimétrica o criptografía de clave pública como ya se ha mencionado utiliza dos claves: una privada y una pública. Estos sistemas invariablemente se utilizan en problemas matemáticos robustos como base para su seguridad, así que un punto obvio de ataque es desarrollar métodos para resolver el problema matemático con el cual fueron desarrollados. La seguridad de un sistema de dos claves depende de cuestiones matemáticas de una manera que no se aplica a la criptografía de clave secreta y a su vez se relaciona el criptoanálisis con la investigación matemática en general de nuevas maneras.

Los algoritmos asimétricos se diseñan en torno a la dificultad de resolver ciertos problemas matemáticos. Si se encuentra un algoritmo mejorado que puede resolver el problema matemático, el criptosistema se ve debilitado. Tomando por ejemplo la seguridad del protocolo Diffie-Hellman podemos ver que su seguridad depende de la dificultad de calcular un logaritmo discreto. En 1983, Don Coppersmith encontró una manera más rápida de calcular logaritmos discretos en ciertos grupos y por tanto obligando a los usuarios a utilizar claves más grandes. La seguridad del protocolo RSA depende parcialmente de la dificultad en la factorización de números enteros. Un avance en la factorización tendría un impacto claro en la seguridad de RSA.

Los avances en la tecnología de computación también han provocado que las operaciones se puedan realizar en un tiempo mucho menor. Se prevé con base en los avances hechos hasta ahora que las velocidades de computación continuarán aumentando, y las técnicas de factorización podrían mostrar un desarrollo parecido. Números de 150 cifras como los que son utilizados en RSA han sido ya factorizados. Al comienzo del siglo XXI, los números de 150 cifras ya no se consideran suficientemente seguros como clave para RSA de hecho se piensa que las claves para este criptosistema deben ser por lo menos de 200 dígitos. Números de varios cientos de dígitos se siguen considerando demasiado difíciles de factorizar en la actualidad, aunque los métodos probablemente continuarán mejorando con el tiempo, obligando a los tamaños de clave a mantener el ritmo de crecimiento o a desarrollar nuevos algoritmos tal como lo son los criptosistemas basados en curvas elípticas.

Otra característica importante de los algoritmos asimétricos es que cualquier criptoanálisis tiene la oportunidad de usar el conocimiento obtenido de la clave pública y ésta debe resistir los intentos por descifrarla.

En un futuro, dado que una computadora podría ser capaz de realizar búsquedas de claves mediante fuerza bruta extremadamente rápidas, los tamaños de

clave considerados hoy en día quedarían al alcance de cualquier persona que contara con estos equipos. Los tamaños de clave necesarios para quedar más allá de la capacidad de una computadora del futuro serían considerablemente más grandes que los actuales. Algunos aseguran que añadiendo bits a las longitudes de las claves se evitará los ataques por fuerza bruta. Pero es necesario utilizar algoritmos más eficientes y además que sean confiables no importando la capacidad de cómputo con la que se vean atacados.

Capítulo 2

Fundamentos matemáticos de la criptografía asimétrica

Como se ha mencionado, los algoritmos de hoy deben basar su seguridad en fundamentos matemáticos comprobables es por esto que vamos a introducirnos en la aritmética de campos finitos, la cual va incrementando su importancia dentro de la criptografía y sobre todo en la criptografía de clave pública.

También, en el presente capítulo se analiza el algoritmo de clave pública más utilizado hoy en día, el RSA, y los sistemas basados en Criptografía de Curvas Elípticas, porque éstos son de gran utilidad para establecer una comparación detallada de ambos sistemas de cifrado y permite establecer claramente bajo qué condiciones es más conveniente utilizar uno u otro tipo de algoritmo.

2.1 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.

2.1.1 Grupos, anillos y campos

Los grupos, los anillos y los 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

Un 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

Un 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

En algebra moderna, un 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

Un 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).

En la siguiente sección se detalla más acerca de la aritmética de campos primos.

2.1.2 Aritmética de campos primos

En la sección anterior se definió un campo como una colección de elementos que siguen ciertas reglas, además es importante mencionar que los campos finitos, mencionados en la parte final de la sección anterior, juegan un papel muy importante en muchos algoritmos de cifrado ya que contienen ciertas características especiales. Por ejemplo 2 enteros son primos relativos si su máximo común divisor es 1.

Los campos finitos de la forma p^m se escriben de la forma $GF(p^m)$[1], en esta parte se va a estudiar el caso cuando m=1, es decir GF(p), ya que este campo tiene una estructura diferente que los campos finitos con m>1.

Campos finitos de orden p

Para un primo p, el campo finito de orden p es definido como el conjunto de elementos GF(p) = {0, 1, … , p-1} junto con las operaciones aritméticas módulo p. Como p es primo, entonces todos los enteros diferentes de cero en GF(p) son primos relativos de p y existe un inverso multiplicativo para todos estos enteros en GF(p). Podemos entonces mencionar la siguiente propiedad para GF(p):

Inverso multiplicativo (w^{-1})	para cada w perteneciente a GF(p), $w \neq 0$, existe un z perteneciente a GF(p) tal que $w*z \equiv 1 \bmod p$.

Si w es primo relativo de p y si multiplicamos todos los elementos de GF(p) por w, el residuo resultante de todos los elementos de GF(p), tendremos que uno de los enteros del residuo tendrá el valor de 1. Esto quiere decir que cuando se multiplique un entero por w y el resultado sea congruente con módulo 1, entonces este entero será el inverso multiplicativo de w denotado por w^{-1}.

[1] GF significa Campo de Galois (Galois Field), en honor al primer matemático en estudiar los campos finitos.

$$w*w^{-1} \equiv 1 \text{modulo } p$$

De la misma forma como se ha definido el inverso multiplicativo dentro del campo de los números primos, también se puede definir el inverso aditivo en el cual se tiene que:

Inverso aditivo (-w)	Para cada w perteneciente a GF(p), existe un z perteneciente también a GF(p) tal que $w+z \equiv 0 \bmod p$.

De la misma forma se tiene que cuando se sume un entero con w y el resultado sea congruente con módulo 0, entonces este entero será el inverso aditivo de w denotado por -w.

$$w+(-w) \equiv 0 \text{modulo } p$$

Por ejemplo, para mostrar las propiedades anteriores se puede tomar en cuenta el campo finito de la forma GF(7), en las tablas siguientes se expresan los resultados aritméticos de las operaciones suma y multiplicación y en una tercer tabla 2.1(c) se indican el inverso aditivo de cada elemento, así como el inverso multiplicativo de los elementos que tengan esta característica.

Tabla 2.1 Aritmética en GF(7)

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

(a) Adición módulo 7

*	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

(b) Multiplicación módulo 7

W	-w	w^{-1}
0	0	-
1	6	1
2	5	4
3	4	5
4	3	2
5	2	3
6	1	6

(c) Inversos aditivo y multiplicativo módulo 7

Encontrando el inverso multiplicativo en GF(p)

Se puede ver que es fácil encontrar el inverso multiplicativo de un elemento en el campo finito de los primos para valores de p pequeños, pero para valores grandes sería inadecuado construir una tabla para obtener los valores directamente.

Es por esto que se vuelve importante utilizar las herramientas a nuestro alcance y de este modo utilizamos el algoritmo de Euclides para obtener el máximo común divisor y el inverso multiplicativo dentro de los campos finitos.

El código fuente en lenguaje Visual Basic para el algoritmo de Euclides se encuentre en el Anexo A.2 del presente trabajo.

2.1.3 Campos finitos de la forma $GF(2^m)$

El primero en trabajar con el concepto de campos finitos fue Evariste Galois, no obstante su muerte a los 20 años (1832), sus avances más notables fueron precisamente los relacionados con el desarrollo de la teoría de grupos, la cual tardo mucho en darse a conocer en su época, pero hoy es la parte esencial de todos los manuales de álgebra. Galois escribió pocos trabajos pero el significado de éstos es enorme y sus aplicaciones muy importantes sobre todo en criptografía.

Ya se había mencionado que el orden de un campo finito debe ser de la forma p^m, donde p es un primo y m es un entero positivo. En esta sección tal y como se hizo en la anterior se va a demostrar que los axiomas de campos finitos se satisfacen para los p^m elementos y específicamente en $GF(2^m)$.

Si todas las operaciones aritméticas van a ser utilizadas y se desea representar todo el rango de enteros con m bits, entonces la aritmética modular 2^m no se puede utilizar ya que el conjunto de enteros módulo 2^m no es un campo, además todos los algoritmos de cifrado sólo utilizan adición y multiplicación, pero no división. Es por esto que los campos finitos de la forma $GF(2^m)$ se vuelven atractivos para algoritmos de cifrado.

Aritmética polinomial modular

Considerando al conjunto S de todos los polinomios de grado n-1 o menor sobre el campo GF(p). Tenemos que los polinomios tienen la siguiente forma:

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

Donde cada a_i tiene un valor en el conjunto {0, 1, … , p-1}. Existe un total de p^n polinomios diferentes en el conjunto S.

Con la apropiada definición aritmética, cada elemento del conjunto de S es un campo finito. La definición consiste en los siguientes elementos:

1. La aritmética sigue las reglas ordinarias de la aritmética polinomial usada en las reglas básicas del algebra, con las siguientes dos redefiniciones.
2. La aritmética sobre los coeficientes sigue las reglas módulo p. Esto es que utilizamos las reglas de la aritmética para el campo finito Z_p.
3. Si el resultado de la multiplicación en un polinomio es de grado mayor que n-1, entonces al polinomio se le obtiene el módulo de otro polinomio m(x) de grado n. Esto significa que se procede a dividir entre m(x) y se mantiene el residuo. Para un polinomio f(x), el residuo se expresa como r(x)=f(x) mod m(x).

Como con la aritmética modular ordinaria, se tiene la noción de un conjunto de residuos en aritmética polinomial modular. El conjunto de residuos módulo m(x), un n-esimo grado polinomial, consiste de p^n polinomios de grado m<n.

Se puede ver que el conjunto de elementos de todos los polinomios satisfacen todos los axiomas vistos al principio de este capítulo, y por tanto son campos finitos.

2.2 Intercambio de claves de Diffie-Hellman

En 1976 Whitfield Diffie y Martin Hellman desarrollaron un protocolo por medio del cual dos personas pueden intercambiarse pequeñas informaciones secretas por un canal inseguro. A este protocolo se conoce como el intercambio de clave de Diffie-Hellman. El procedimiento está basado en un sistema de claves asimétrico en el que cada comunicante posee dos claves, una de ellas secreta y la otra pública. El

protocolo no sólo resolvió el problema del intercambio de claves, sino además fue el origen de la denominada criptografía de clave pública, cuya aplicación tiene importantes implicaciones en el ámbito de la seguridad dentro de las comunicaciones digitales.

El método propuesto está basado en la existencia de funciones de una sola dirección también llamadas funciones unidireccionales. Una función de una sola dirección es aquélla cuyo cálculo directo es viable, mientras que el cálculo de la función inversa tiene tal complejidad que resulta imposible. Una función de una sola dirección típica es la exponenciación modular dada por la ecuación:

$$y \equiv g^{x} (\mathrm{mod}\ p)$$

Con g, x valores enteros y siendo p un número primo grande de 200 dígitos o mayor. El cálculo de la función y es posible, ya que tiene una complejidad 0(log p), pero el cálculo de la función inversa:

$$x \equiv \log_{g} y (\mathrm{mod}\ p)$$

Tiene una complejidad tan elevada que es totalmente inviable para números de tamaño de 200 dígitos. Esta función se conoce como el problema del logaritmo discreto.

Con tales antecedentes podemos definir el intercambio de claves, en el cual existen 2 números: un número primo q y un entero α tal que sea un entero generador del grupo multiplicativo q, es decir, que α pertenezca al grupo q tal que sus potencias generan todos los elementos del grupo y donde también debe considerarse como públicos a los valores de q y α. Supóngase ahora que dos personas A y B desean intercambiar una clave secreta a través de un canal inseguro. Para ello, el usuario A selecciona un entero aleatorio secreto tal que $1<X_A<q$ y envía a B el valor público generado por $Y_A= \alpha^{XA} \pmod q$. Por su parte, B elige un entero aleatorio secreto X_B tal que $1<X_B<q$ y del mismo modo le envía a A su valor público generado por $Y_B= \alpha^{XB} \pmod q$, cabe mencionar que los valores X_A y X_B seleccionados por los usuarios deben ser números primos. El usuario A obtiene la clave con la siguiente fórmula $K= (Y_B)^{XA} \pmod q$ y el usuario B calcula la clave con la siguiente fórmula $K= (Y_A)^{XB} \pmod q$. Los dos cálculos anteriores producen resultados idénticos.

Ejemplo:

Sea el sistema de Diffie-Hellman para intercambio de claves en el que el número primo q = 71 y el generador α = 21. Supongamos que los usuarios A y B desean intercambiar un valor secreto para utilizarlo posteriormente. Para ello, el usuario A elige un valor entero aleatorio secreto de X_A = 47 y envía a B el valor público Y_A dado por:

$Y_A=21^{47}(\text{mod } 71) =47 \text{ (mod } 71)$

$21^1 \text{ mod}71=21$

$21^2 \text{ mod}71=15$

$21^4 \text{ mod}71\equiv15^2 \text{ mod}71=12$

$21^8 \text{ mod}71\equiv12^2 \text{ mod}71=2$

$21^{16} \text{ mod}71\equiv2^2 \text{ mod}71=4$

$21^{32} \text{ mod}71\equiv4^2 \text{ mod}71=16$

$C=21^{47} \text{ mod}71\equiv (16*2*12*15*21) \text{ mod}71=47.$

Por su parte, B elige un valor entero aleatorio secreto de X_B=61 y envía a A el valor público Y_B dado por:

$Y_B=21^{61}(\text{mod } 71) =22 \text{ (mod } 71)$

$21^1 \text{ mod}71=21$

$21^2 \text{ mod}71=15$

$21^4 \text{ mod}71\equiv15^2 \text{ mod}71=12$

$21^8 \text{ mod}71\equiv12^2 \text{ mod}71=2$

$21^{16} \text{ mod}71\equiv2^2 \text{ mod}71=4$

$21^{32} \text{ mod}71\equiv4^2 \text{ mod}71=16$

$C=21^{61} \text{ mod}71\equiv (16*4*2*12*21) \text{ mod}71=22.$

Entonces A calcula la clave secreta dado por:

$K= (Y_B)^{XA} (\text{mod } q) = 22^{47}(\text{mod } 71) =55 \text{ (mod } 71)$

22^1 mod71=22

22^2 mod71=58

22^4 mod71≡58^2 mod71=27

22^8 mod71≡27^2 mod71=19

22^{16} mod71≡19^2 mod71=6

22^{32} mod71≡6^2 mod71=36

C=22^{47} mod71≡ (36*19*27*58*22) mod71=55.

A continuación, B calcula el valor secreto dado por:

K= $(Y_A)^{XB}$ (mod q) = 47^{61}(mod 71) =55 (mod 71)

47^1 mod71=47

47^2 mod71=8

47^4 mod71≡8^2 mod71=64

47^8 mod71≡64^2 mod71=49

47^{16} mod71≡49^2 mod71=58

47^{32} mod71≡58^2 mod71=27

C=47^{61} mod71≡ (27*58*49*64*47) mod71=55.

En estas condiciones la clave secreta común es K=55.

2.3 Algoritmo RSA

El sistema de cifrado de clave pública RSA fue propuesto por Ron Rivest, Adi Shamir y Leonard Adleman, basados en el trabajo publicado por Diffie-Hellman, en el año de 1977.

El sistema de cifrado RSA es uno de los métodos más utilizados, debido a sus características especiales. Ya que cuando se envía un mensaje, el emisor pide la clave pública de cifrado al receptor y una vez que tal mensaje llega el receptor, éste descifra el mensaje utilizando su clave secreta. Los mensajes enviados utilizando RSA se representan mediante números y el funcionamiento se basa en el producto de dos números primos grandes elegidos al azar para conformar la clave de descifrado.

La seguridad de este algoritmo se basa en que no hay una forma rápida conocida de factorizar un número grande en sus factores primos utilizando computadoras tradicionales.

2.3.1 Generalidades del algoritmo RSA

El esquema desarrollado por sus creadores utiliza dos números primos, p y q, grandes (por lo menos 200 dígitos) y el producto de ellos, esto es, N= p*q. Se tiene φ(N) la función de Euler de N, indicadora del número de valores enteros menores que N que son relativamente primos con respecto a N. En este caso la función φ(N) viene dada por el producto:

$$\varphi(N) = (p-1)(q-1)$$

Así, se tiene un valor e entero y aleatorio que es primo relativo con φ(N) tal que 1<e<N y sea d otro entero tal que se verifica la congruencia:

$$ed \equiv 1(\mathrm{mod}\,\varphi(\mathrm{N}))$$

Es decir que d es el inverso multiplicativo de e:

$$ed = k\varphi(\mathrm{N}) + 1$$

Utilizando, desde luego, un valor k entero. Con estas condiciones, para cualquier valor entero M se verifica que si $C \equiv M^e (\text{mod} N)$ entonces $M \equiv C^d (\text{mod} N)$, es decir que:

$$M^{ed} \equiv M (\text{mod}\, N)$$

Tenemos que:

$$M^{ed} \equiv M (\text{mod}\, p) \equiv M (\text{mod}\, q)$$

El sistema RSA está constituido por las claves N, e y d, respectivamente módulo de trabajo y exponentes de cifrado y descifrado. Tenemos que el valor de N es público, así como el de uno de los dos exponentes (e o d), mientras que el otro exponente debe permanecer secreto, por ejemplo:

Claves públicas: N, e.

Clave privada: d.

Para realizar el cifrado C de un mensaje en claro M tal que $1 < M < N$, el mensaje se eleva a la potencia e y el resultado se reduce al módulo N ($C \equiv M^e (\text{mod } N)$).

La recuperación del mensaje en claro M se lleva a cabo elevando el mensaje cifrado C a la potencia d (clave pareja de e) y posteriormente reduciéndola al módulo N ($M \equiv C^d (\text{mod } N)$).

Ejemplo:

Sea el sistema RSA definido por los parámetros:

p=17

q=11

N=p*q=17*11=187.

$\varphi(N)=(p-1)(q-1)=16*10=160$.

$e=7$

$d \equiv 7^{-1} \equiv 23 \bmod 160$

Calculando el inverso de 7 en el GF(160) utilizando el algoritmo de Euclides (apéndice A.2), se tiene como clave pública {7, 187} y como clave privada {23,187}.

Ahora bien, utilizando el mensaje en claro M=88, para el cifrado se debe calcular $C=88^7 \bmod 187$. Utilizando las propiedades de la aritmética modular:

$88^1 \bmod 187=88$

$88^2 \bmod 187=7744 \bmod 187=77$

$88^4 \bmod 187 \equiv 77^2 \bmod 187=132$

$C=88^7 \bmod 187 \equiv (88*77*132) \bmod 187=11$

Para recuperar el mensaje original se calcula $M=11^{23} \bmod 187$

$11^1 \bmod 187=11$

$11^2 \bmod 187=121$

$11^4 \bmod 187 \equiv 121^2 \bmod 187=55$

$11^8 \bmod 187 \equiv 55^2 \bmod 187=33$

$11^{16} \bmod 187 \equiv 33^2 \bmod 187=154$

$M=11^{23} \bmod 187 \equiv (154*55*121*11) \bmod 187=88$

Los números primos p y q son secretos, constituyendo la trampa del sistema. El cálculo de la clave secreta d es sencillo conocida la trampa, en tanto que sin el conocimiento de los factores primos p y q, el cálculo de la clave secreta es equivalente al cálculo de la factorización de la clave pública N. Aquí es donde radica la seguridad del método RSA, puesto que el problema de la factorización de un número compuesto N, producto de dos factores primos, tiene una complejidad exponencial dada por $e^{\sqrt{[\ln(N)\ln\ln(N)]}}$. Por esta razón se requiere la utilización de números primos muy grandes. A su vez, aunque el sistema de cifrado RSA funciona para cualquier valor entero M, es conveniente que M sea relativamente primo con la clave pública N, puesto que de lo contrario la factorización de N es trivial.

Cuando se diseña un sistema RSA es necesario tener en cuenta ciertas condiciones para elegir los parámetros del mismo.

Elección de los números primos p y q

Sea N=p*q la clave pública de un sistema de cifrado RSA con p y q primos grandes. Una primera condición que deben satisfacer p y q es la de no estar demasiado próximos el uno del otro. En caso contrario, es decir, si p≈q y suponemos por ejemplo p>q entonces (p-q)/2 es un entero pequeño, mientras que (p+q)/2 es ligeramente superior a √N. Además, se verifica la relación dada por:

$$\frac{(p+q)^2}{4} - N = \frac{(p-q)^2}{4}$$

En estas condiciones, una forma de factorizar N es elegir aleatoriamente valores enteros x>√N hasta que se encuentre uno tal que $(x^2-N)= y^2$, es sencillo verificar que p=x+y y que q=x-y. Por esta razón, para que este procedimiento de factorización resulte impracticable es aconsejable que las longitudes de los números primos p y q difieran en unos pocos bits.

2.4 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. Básicamente criptografía de curvas elípticas es más difícil de explicar que hasta este punto lo han sido Diffie-Hellman o RSA.

Muchos criptosistemas requieren el uso de grupos algebraicos, los cuales se han analizado al principio del presente capítulo, las curvas elípticas pueden utilizar los que provienen de los grupos de curvas elípticas. Se ha mencionado ya que un grupo es un conjunto de elementos con operaciones aritméticas definidas en estos elementos. Para los grupos de curvas elípticas, estas operaciones específicas están definidas geométricamente.

Retomando un poco lo que se vio al principio del capítulo tenemos que muchos 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.

2.4.1 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, considerando el propósito que se persigue en cuanto a desarrollar un sistema de aprendizaje que nos sirva de herramienta para comprender como opera la criptografía de curvas elípticas, 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 y b=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 2.1.

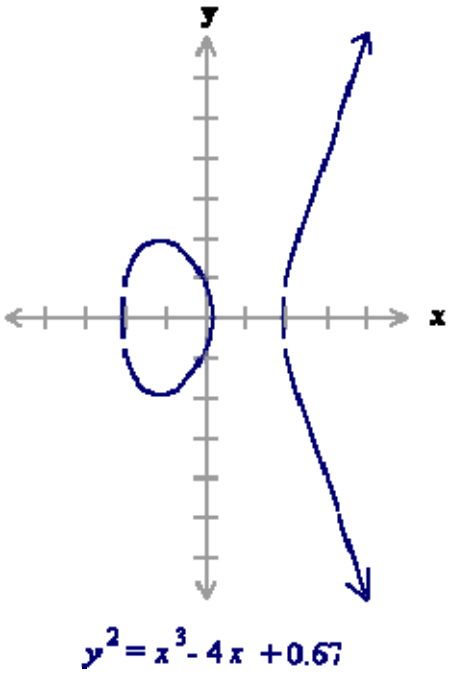


Figura 2.1 Curva elíptica con a=-4 y b=0.67

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 esta 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 2.2 se puede ver un ejemplo de esta regla.

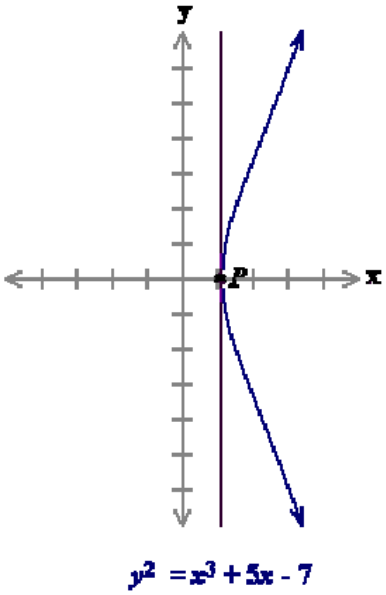


Figura 2.2 Se tiene el punto P = (1, 0), por lo tanto P + 0 = P, es decir la intersección con la curva es en un solo punto, la recta es tangencial a la curva en el punto P.

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 tenemos que P o Q son tangentes a dicha curva, entonces tenemos que R=P o R=Q respectivamente. Para formar una estructura de grupo, necesitamos 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 2.3 ilustra esta construcción.

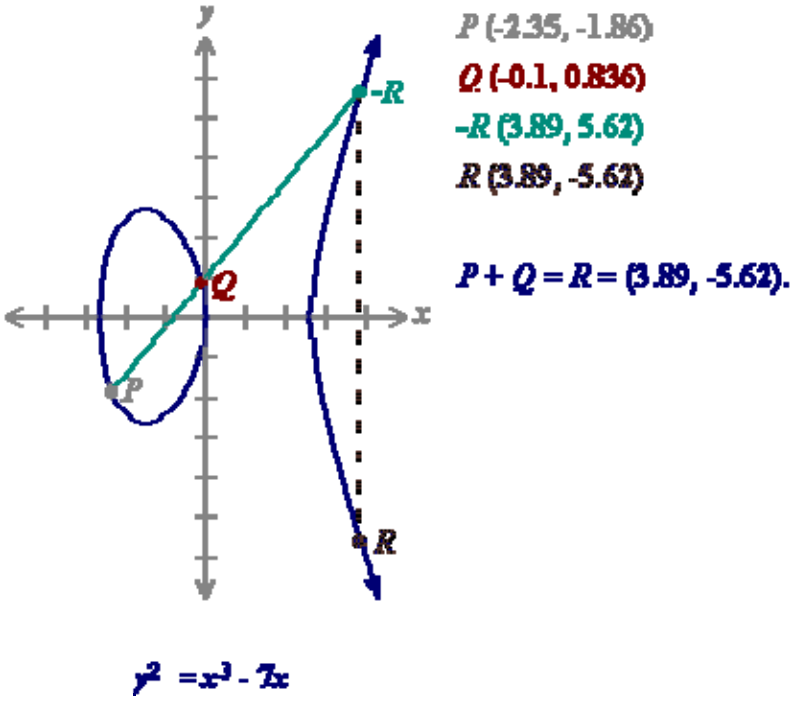


Figura 2.3 Para sumar los puntos P y Q, una línea se dibujara a través de los dos puntos. Esta línea intercepta a la curva elíptica en exactamente un punto –R. El punto –R se refleja en el eje de las x al punto R.

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. Tenemos que P+ (-P)=0. La figura 2.4 muestra lo referente a la presente regla.

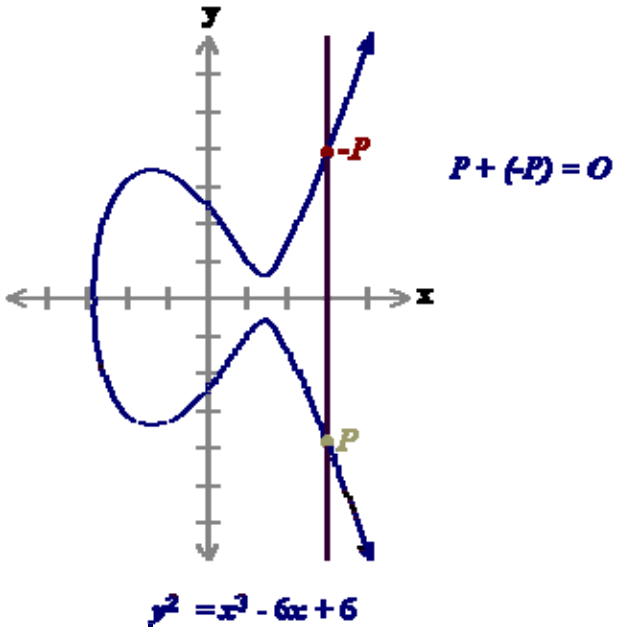


Figura 2.4 Suma de P+ (-P) = O

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 2.5 se ilustra esto.

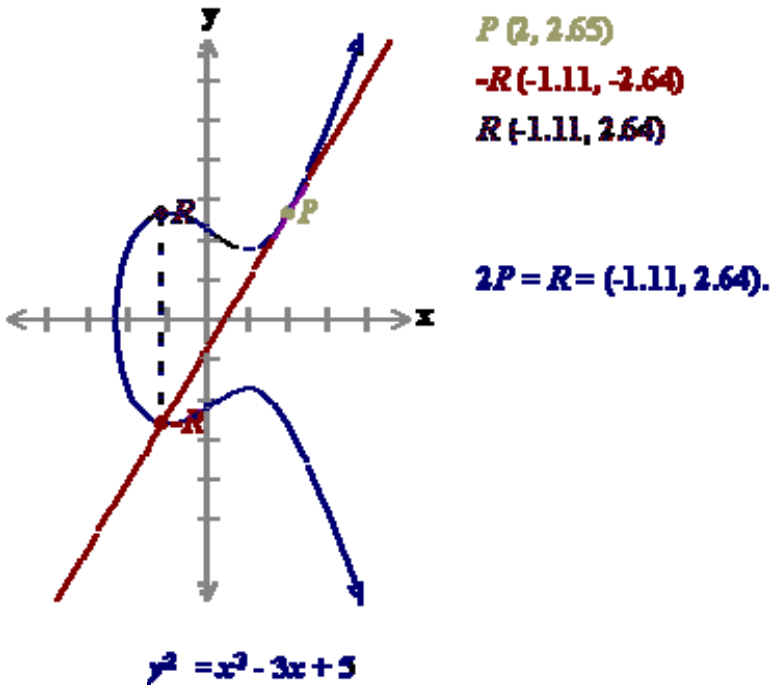


Figura 2.5 Suma de P + P

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).

Sin embargo se tiene que las reglas anteriores operan sobre los números reales los cuales en realidad no forman un grupo. Es por lo que en la siguiente sección se trabajará sobre el campo de los números primos.

2.4.2 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 de p. 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 están en 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, del cual ya se ha comentado que 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 2.6:

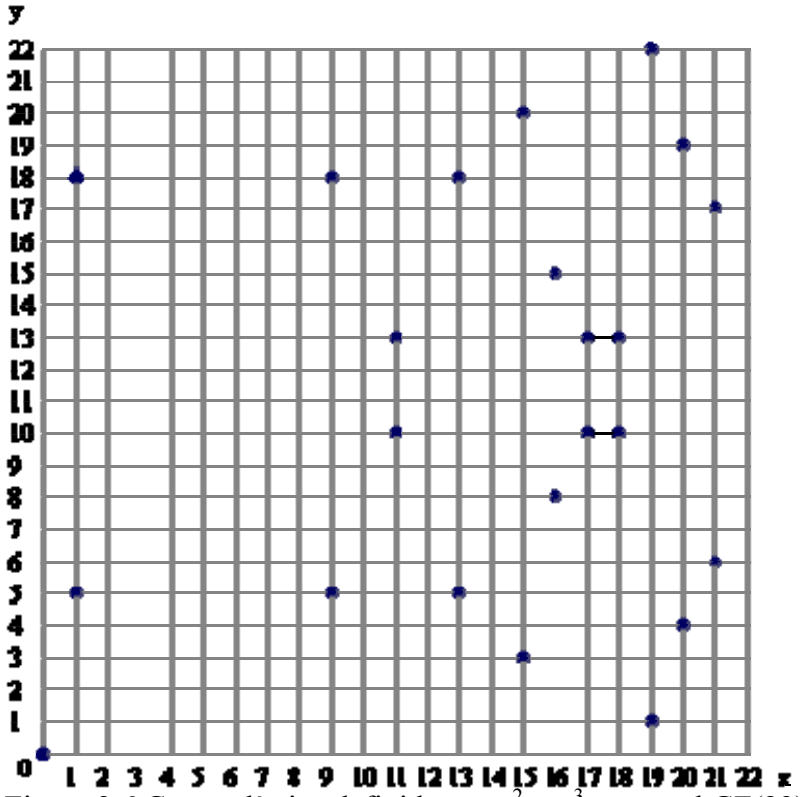


Figura 2.6 Curva elíptica definida por $y^2 = x^3 + x$ en el GF(23).

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 importantes 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), el cual como se observa en la figura 2.6 también se encuentra en $E_{23}(1,0)$.
3. Si P=(x_P, y_P) y Q=(x_Q, y_Q) con P≠-Q, entonces R=P+Q es determinada por las siguientes reglas:

Si P≠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:

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

Utilizando el algoritmo para obtener inversos que se encuentra en el anexo A.2 se tiene:

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

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

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

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:

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

Utilizando el algoritmo para obtener inversos que se encuentra en el anexo A.2 se tiene:

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

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

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

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

2.4.3 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 polinomial 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 cero 0.

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^4 y 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 2.7 de la siguiente forma:

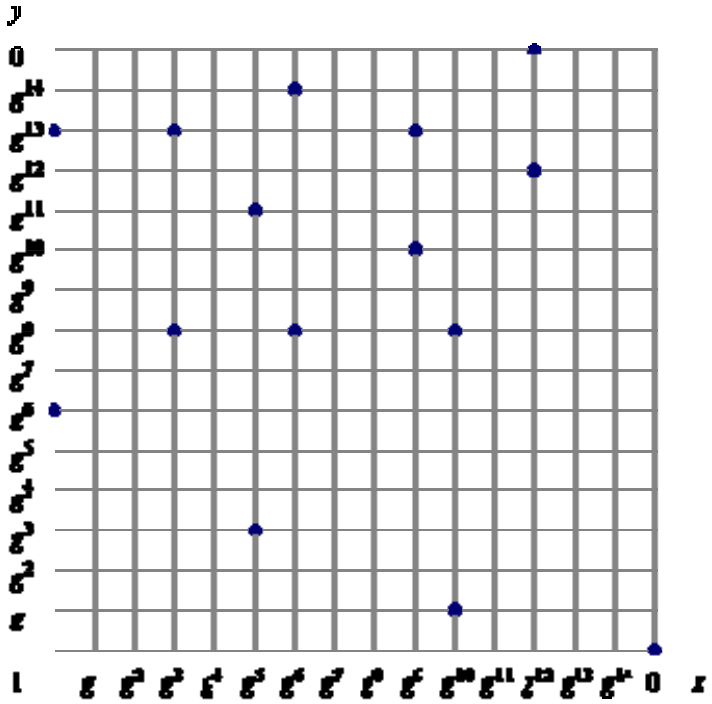


Figura 2.7 Puntos que satisfacen la ecuación $y^2 + xy = x^3 + g^4x^2 + 1$ sobre $GF(2^m)$

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:

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

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

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

2.4.4 Curvas elípticas y el problema del logaritmo discreto

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).

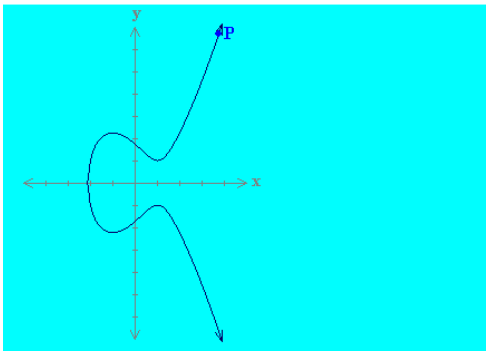
Como se ha podido observar existen dos operaciones básicas que son la adición de puntos y la suma de un punto consigo mismo. Si se selecciona un punto en un grupo elíptico de una curva, se puede sumar consigo mismo para obtener el punto 2P. Después de esto, se puede sumar el punto P al punto 2P para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

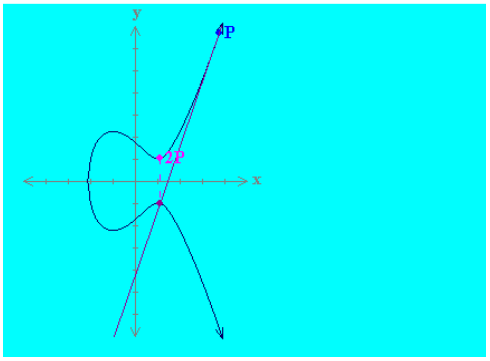
Multiplicación escalar.

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

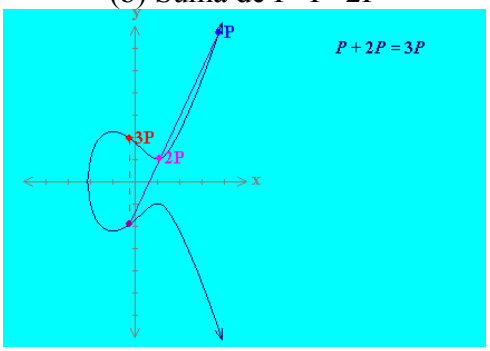
consiste en el multiplicar juntas las copias de k del punto P, obteniendo el punto P+P+P+P+…+P = kP. La figura 2.8 muestra una representación gráfica de sumar un mismo punto k veces.



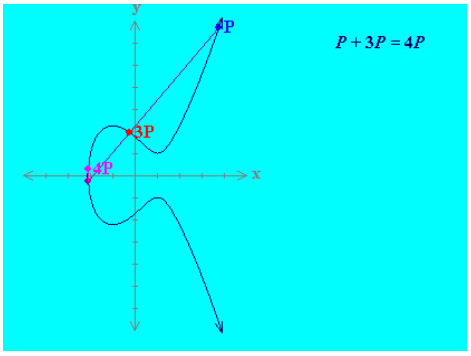
(a) Punto P



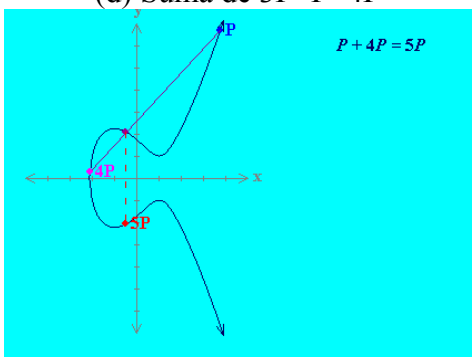
(b) Suma de P+P=2P



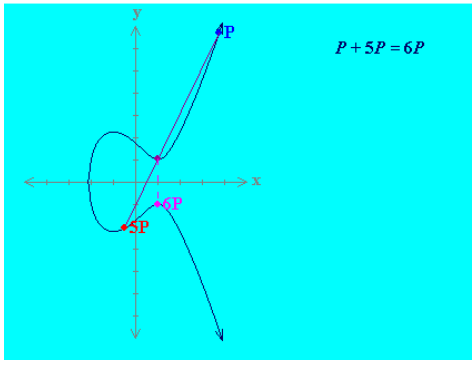
(c)Suma de 2P+P=3P



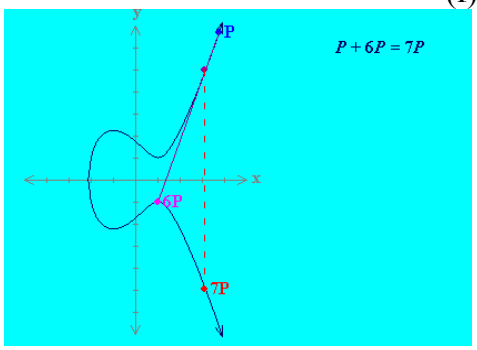
(d) Suma de 3P+P=4P



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



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



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

Figura 2.8 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.

2.5 Criptografía con curvas elípticas

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 tal que 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 $y^2 = x^3 + 9x + 17$ sobre GF(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 tenemos 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.

2.5.1 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, tenemos 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 utilizamos el punto cero P = (0,0) como inicio y para cada i = n-1, n-2, …, 1, 0 hacemos lo siguiente:

1. Hacemos P = 2P, es decir, se obtiene la multiplicación escalar por dos del punto.
2. Si k_i =1entonces hacemos P = P + G.
3. Si i es igual diferente de cero regresamos 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 2.2 se anotan los resultados.

Tabla 2.2 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.

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)

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.

2.5.2 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 estamos 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 nos permitirá efectuar la búsqueda de los posibles puntos generadores del grupo.

2.5.3 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.

2.5.4 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 $K = n_B * P_A$.

Obviamente, 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_A y 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[2].

- Entonces se tiene que la clave pública de A es P_A= 121(2, 2)= (115, 48). Como se puede ver en la tabla 2.3.

Tabla 2.3 El cálculo de la clave pública del usuario A, la cual es 121 veces el punto G = (2, 2) sobre la curva elíptica $y^2=x^3- 4$.

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)

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 2.4.

[2] En el anexo A.2 se tiene el código fuente para obtener el máximo común divisor.

Tabla 2.4 El cálculo de la clave pública del usuario B, la cual es 203 veces el punto G = (2, 2) sobre la curva elíptica $y^2=x^3- 4$.

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)

- 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 podemos usar simplemente la coordenada x o una función simple de la misma.

2.5.5 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,…, 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 2.5.

Tabla 2.5 Alfabeto para codificar sistemas de Curvas Elípticas

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

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= "OMAR". A continuación se codifica cada una de las unidades del mensaje:

O:16 → con j=1 x=16*20+1= 321 sustituyendo en $y^2=x^3$ -x +188 da como resultado en el segundo miembro de la igualdad un valor de 486 pero no existe un valor de "y" tal que $y^2=486$ para la curva elíptica elegida en el GF(751).

O:16 → con j=2 x=16*20+2= 322 sustituyendo en $y^2=x^3$ -x +188 da como resultado en el segundo miembro de la igualdad un valor de 409 pero no existe un valor de "y" tal que $y^2=409$ para la curva elíptica elegida en el GF(751).

O:16 → con j=3 x=16*20+3= 323 sustituyendo en $y^2=x^3$ -x +188 da como resultado en el segundo miembro de la igualdad un valor de 11 pero no existe un valor de "y" tal que $y^2=11$ para la curva elíptica elegida en el GF(751).

O:16 → con j=4 x=16*20+4= 324 sustituyendo en $y^2=x^3$ -x +188 da como resultado en el segundo miembro de la igualdad un valor de 49 entonces como $49 = y^2$ para un y=7 entonces el punto correspondiente a la curva para la parte del mensaje "O" será $P_O = (324, 7)$.

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:

(324, 7), (261, 288), (21, 133), (381, 69)

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

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

$$\frac{261-1}{20} = 13 \rightarrow M$$

$$\frac{21-1}{20} = 1 \rightarrow A$$

$$\frac{381-1}{20} = 19 \rightarrow R$$

Lo anterior se puede utilizar para manejar los métodos de cifrado de Massey-Omura y ElGamal para curvas elípticas, los que se explican a continuación.

2.5.6 Método de cifrado de Massey-Omura

El logaritmo elíptico puede ser igualmente utilizado como función de una sola dirección para el diseño de sistemas de cifrado de clave pública basados en curvas elípticas. El que se expone en esta parte es el propuesto por Massey y Omura. Los parámetros del procedimiento son una curva elíptica E de orden N definida sobre un campo GF(p) con p primo. La curva E y el orden de la curva N son parámetros públicos del procedimiento.

De tal manera que suponiendo a dos usuarios A y B que quieren intercambiar de forma secreta un mensaje confidencial el cual se representa por un punto P de la curva elíptica E definida sobre el campo GF(p) la cual fue elegida por los usuarios previamente y además se ponen de acuerdo en definir la codificación de las unidades del mensaje, a las cuales ya les ha sido asignado un elemento en la curva elíptica E. Para esto, cada usuario debe poseer un sistema de clave pública. En particular, el usuario A tiene una clave secreta igual a "a", tal que dicha clave también pertenecen al campo GF(p) y se sabe que "a" es un valor entero aleatorio. Por otro lado, el usuario B posee otra clave, de la misma forma que el usuario A, que puede ser "b", también con "b" secreto que pertenece a GF(p). Dado que ambos conocen la ecuación de la curva elíptica pueden calcular el número de puntos que tiene E y a este número le

llaman N. De tal forma que ahora A y B calculan sus respectivos valores secretos utilizando el apéndice A.2:

$a' \equiv a^{-1} \pmod N$

$b' \equiv b^{-1} \pmod N$

Con esto el usuario A puede transmitir al usuario B un mensaje confidencial a través del punto P_m con el siguiente proceso de cifrado:

1. El usuario A envía al usuario B el punto de la curva $Q=a* P_m$.
2. El usuario B recibe un punto (x, y) el cual sabe que pertenece a la curva elíptica, pero no podrá recuperar el mensaje ya que desconoce la clave secreta de A, es decir "a", ni tampoco conoce el inverso de ésta, a la que se le llama a'. Entonces, envía al usuario A el punto de la curva $Q'=b*Q=b*a* P_m$.
3. El usuario A recibe y procesa Q' y le envía al usuario B el punto de la curva $Q''= a'*Q'= a^{-1}*b*a* P_m =b* P_m$.
4. Finalmente el usuario B calcula $b'*Q''=b^{-1}*b* P_m = P_m$ de esta manera b recupera el mensaje que A le quería hacer llegar.

Para un posible espía que desea encontrar P_m lo único que conocerá será Q, Q' y Q''. Intentar encontrar P_m a partir de sólo estos tres datos conduce a resolver el problema del logaritmo discreto en curvas elípticas. Con este procedimiento de cifrado deben tomarse precauciones adicionales para garantizar la identidad de los usuarios, ya que es sencillo darse cuenta de que en este proceso de cifrado un eventual atacante podría interceptar el canal de comunicación suplantando cada vez a uno de los usuarios, lo cual permitiría descifrar todos los mensajes intercambiados por éstos e incluso modificarlos a su conveniencia.

2.5.7 Método de cifrado ElGamal con curvas elípticas

Otro posible algoritmo de cifrado de clave pública basado en curvas elípticas es el equivalente al de ElGamal, el cual es muy utilizado en nuestros días. En este caso, 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 el mensaje "O" utilizado el ejemplo sobre como codificar un mensaje en curvas elípticas es decir nuestro punto a cifrar será P_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 cual 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".

De ésta forma se puede tener un acercamiento a la criptografía basada en curvas elípticas, se puede observar que ésta es un poco más complicada con respecto al sistema de cifrado RSA, pero todavía no se ha mencionado las ventajas de utilizar éste tipo de criptografía asimétrica con respecto a RSA.

En el siguiente capítulo se hace una comparación más detallada de estos dos tipos de criptosistema.

Capítulo 3

Importancia de la CCE en el aprovechamiento de los recursos computacionales y de la seguridad en elementos reducidos.

A medida que las computadoras reducen su tamaño las velocidades de procesamiento aumentan, gracias a esto se ha visto en pocos años un avance en la distribución de sistemas de tipo portátil, los cuales se empiezan a volverse comunes en nuestros días.

Los equipos portátiles, en general, contienen información muy valiosa para los usuarios que los utilizan, por lo cual se tiene que entender que las formas de proteger estos sistemas deben evolucionar para no quedar rezagados con respecto a los individuos que intenten apoderarse de la información, independientemente del la utilidad que se les quiera dar.

No importa qué tan pequeño sea un sistema a proteger, se debe ofrecer una seguridad matemática comprobable, ya que los oponentes atacarán con los equipos más rápidos y eficientes que puedan conseguir y los productos que se tengan con limitantes de espacio y recursos deben ser capaces de soportar cualquier ataque que se pueda presentar.

Por ejemplo, en elementos de tipo inalámbrico tales como teléfonos celulares, computadoras móviles o organizadores personales se tiene la necesidad de mantener

la comunicación alejada de cualquier curioso que pudiese escuchar una conversación o ver intercambio de información. En Internet, del mismo modo, se vuelve muy necesaria la comunicación segura, confiable y que pueda operar con limitaciones en cuanto a ancho de banda, sobre todo en operaciones tales como de comercio electrónico donde la seguridad de los datos es fundamental.

Durante los últimos treinta años la criptografía asimétrica ha brindado seguridad en las comunicaciones a través de alguna red o alguna otra forma de comunicación digital. También ha proporciono la capacidad de administrar las claves, de ofrecer firmas de tipo digital a documentos que necesitan autenticarse y proteger la integridad de los mismos. El algoritmo RSA, el cual se basa en las investigaciones hechas por Diffie-Hellman, ha hecho seguras las comunicaciones en los últimos años, sin embargo la situación ha cambiado, ya que a pesar de que en el tiempo en que se desarrollo RSA los algoritmos de clave pública revolucionaron la criptografía, al día de hoy se han descubierto nuevas técnicas, que en términos generales son más adecuadas para los requerimientos actuales, también han mejorado el funcionamiento y han incrementado la seguridad en comparación con aquellos primeros criptosistemas, que sin embargo en nuestros días se siguen utilizando.

Los dos algoritmos de clave pública utilizados en la actualidad son RSA y Diffie-Hellman. La seguridad del primero, como se vio en el capítulo anterior, se basa en la dificultad de factorizar dos números primos largos y el segundo se relaciona con el problema del logaritmo discreto sobre los campos finitos. Los dos basan su uso en la teoría numérica elemental y ambos han sido objeto de estudio desde su invención con el fin de garantizar la seguridad de su uso.

Una de las técnicas que empiezan a tomar importancia, por su mejor funcionamiento con respecto a otros sistemas de cifrado de tipo asimétrico, son las basadas en la aritmética de curvas elípticas (vista en el capítulo anterior) ya que empiezan a ofrecer beneficios significativos con respecto a viejos sistemas criptográficos.

Desde que la criptografía de curvas elípticas fue descubierta en 1985 ha sido estudiada de la misma forma que los sistemas criptográficos asimétricos que fueron sus predecesores. Sin embargo, RSA y Diffie-Hellman sucumbieron lentamente a algunos ataques criptoanalíticos lo que a llevado a la necesidad de incrementar el tamaño de sus claves para ofrecer los mismos niveles de seguridad de antaño, la criptografía basada en curvas elípticas ha permanecido prácticamente intacta desde que fue presentada en 1985.

3.1 Comparación de CCE con RSA

Existen algunos criterios que se deben tomar en cuenta cuando seleccionamos un sistema de cifrado de tipo asimétrico para una aplicación en particular, en esta sección se analizan los que son considerados como los principales.

La mayoría de los sistemas de clave pública de hoy en día utiliza claves de 1024 bits para RSA. El Instituto Nacional para Estándares y Tecnología (NIST por sus siglas en inglés) dice que éste tamaño de clave es suficiente hasta el año 2010. A partir de este año se recomienda que la longitud de las claves se aumente para lograr mantener el nivel de seguridad. Esto implica que a medida que se tengan equipos con mayor capacidad de procesamiento será necesario aumentar la longitud de las claves para ofrecer un nivel de seguridad adecuado.

Por otro lado también se debe tomar en cuenta que algunos elementos tales como computadores de mano, memorias USB, teléfonos celulares, redes con ancho de banda limitado y las próximas computadoras ultra móviles (UMPC) necesitan proteger su información no importando que no tengan los recursos computacionales necesarios para manejar claves que cada día son de mayor tamaño para mantener una seguridad confiable en estos días.

Se ha mencionado que los requerimientos en cuanto a espacio son muy importantes, pero se tiene que tomar en cuenta la seguridad que pueden ofrecer el sistema de cifrado más utilizado actualmente, entiéndase RSA, y el cifrado con curvas elípticas para definir cual es más adecuado para los requerimientos actuales.

También es importante tomar en cuenta la eficiencia con la que operan cada uno de estos sistemas de cifrado para evaluar cuál podría ser el adecuado para darle solución a cada una de las necesidades específicas.

3.1.1 Seguridad

La seguridad en un algoritmo de cifrado de tipo asimétrico se refiere a las garantías que ofrece el sistema para que una información cifrada no pueda ser vista por una persona que no posea la clave secreta.

La seguridad en RSA y Criptografía de Curvas Elípticas se basa en el problema del logaritmo discreto, sin embargo para ambos sistemas se tienen características especiales.

RSA

La seguridad del criptosistema RSA está basada en el problema de factorizar números cuya longitud se considera grande (al menos 200 dígitos). Para un criptoanalista el descifrado completo de un texto utilizando RSA es actualmente intratable. Por esta razón los ataques contra RSA son basados en resolver el problema de factorizar dos enteros, es decir, en encontrar la clave privada de los usuarios.

Existe un algoritmo que se basa en la idea de utilizar como generador un número primo y obtener una asociación con un conjunto de ecuaciones lineales cuya solución finalmente consiste en factorizar. Este algoritmo puede ser utilizado para permitir la factorización en redes o estaciones de trabajo y no se necesitan computadoras con grandes recursos o supercomputadoras para factorizar números grandes.

En 1984 Carl Pomerance desarrolló un método que se utilizaba para factorizar números de aproximadamente 223 bits de largo. En 1994, fue utilizado por un grupo de investigación encabezado por Arjen Lenstra para factorizar números de 429 bits. Es aquí donde se empezó a incrementar la longitud de las claves en RSA ya que se pudo apreciar que se necesitaban claves de mayor tamaño para garantizar que el sistema fuese seguro a ataques de los criptoanálistas. La factorización que coordinó Lenstra llevó aproximadamente ocho meses, utilizando aproximadamente 1600 computadoras distribuidas alrededor del mundo. El tiempo total de cómputo para la factorización se estimó en 5000 millones de instrucciones por segundo (MIPS) durante un año.

El 22 agosto de 1999, un equipo de científicos provenientes de seis diferentes países, dirigidos por Herman Riele, encontraron la factorización de un número de 512 bits.

En otro estudio, para obtener la factorización de la clave RSA de 155 bits se gastaron aproximadamente 35 años de tiempo computacional, para esto se hizo que las computadoras trabajaran en paralelo, el trabajo se efectúo en 7 meses. Pero debemos considerar que teniendo la distribución adecuada en Internet con miles de participantes, sería posible reducir el tiempo de factorización de 7 meses a una semana. En la tabla 3.1 se puede ver que un número de 512 bits provee solamente seguridad marginal cuando se utiliza en criptosistemas RSA. Para mayores niveles

de seguridad, una clave de 1024 bits o mayor se vuelve necesario para garantizar el funcionamiento de RSA.

Tabla 3.1 Poder de cómputo necesario para factorizar números enteros

Tamaño del entero para ser factorizado(bits)	Millones de instrucciones por segundo en años (MIPS/años)
512	$3*10^4$
768	$2*10^8$
1024	$3*10^{11}$
1280	$1*10^{14}$
1536	$3*10^{16}$
2048	$3*10^{20}$

Para factorizar números enteros, existe un algoritmo rápido para n=p*q que da p-1 o q-1 pero sólo nos sirve para factores pequeños. Este algoritmo se conoce como método para factorizar con curvas elípticas (ECM por sus siglas en inglés), el cual fue creado por Hendrik Lenstra en 1985. El tiempo de cómputo de éste método depende del tamaño de los factores primos de n.

Criptografía de Curvas Elípticas (CCE)

Los ataques contra los criptosistemas de curvas elípticas se basan en resolver el problema del logaritmo discreto en curvas elípticas (ECDLP por sus siglas en inglés). Y la existencia de métodos usados para atacar el problema del logaritmo discreto depende de tener un campo finito GF(p). Es aquí donde se puede hacer una analogía con el problema del logaritmo discreto pero ahora en curvas elípticas. Sin embargo en el caso de curvas elípticas los métodos son mucho más lentos ya que las operaciones de suma (como se vio en el capítulo anterior) son mucho más complejas aunque esto nos proporciona a su vez que ofrece ciertas ventajas.

Teniendo una curva elíptica E(x, y) definida sobre un campo finito GF(p) y dos puntos P y Q pertenecientes a la curva, las principales formas de atacarlos se resumen como sigue:

El algoritmo de búsqueda exhaustiva tiene un tiempo estimado de cómputo de $\sqrt{\pi r/2}$ pasos, donde se le llama paso a una operación de sumar dos puntos (el multiplicar el mismo punto por 2, 2P = P + P, tal como se explico en el capítulo

anterior) y r es el orden del punto P. La tabla 3.2 muestra el poder de cómputo necesario para procesar el logaritmo de curvas elípticas con un método llamado Pollard o método Monte Carlo. En 1993, Van Oorshot y Wiener utilizaron en paralelo éste método para obtener con $\frac{\sqrt{(\pi r/2)}}{m}$ pasos en m procesos paralelos. Wiener y Zuccherato además redujeron esto en $\sqrt{2}$ en el año de 1998.

Algoritmo de Pohlig-Hellman. Este algoritmo explota la factorización de r (el orden del punto P). El algoritmo reduce el problema de recuperar la clave secreta a y recobrar un módulo cada vez que se tiene un factor de r, a pueda entonces ser recuperado utilizando el teorema chino del residuo.

Tabla 3.2 Poder de cómputo requerido para procesar logaritmo de curvas elípticas con el método de Pollard.

Tamaño del campo [bits]	Tamaño de r [bits]	$\sqrt{\pi r/2}$	Millones de instrucciones por segundo en años
163	160	2^{80}	$9.6*10^{11}$
191	186	2^{93}	$7.9*10^{15}$
239	234	2^{117}	$1.6*10^{23}$
359	354	2^{177}	$1.5*10^{41}$
431	426	2^{213}	$1.0*10^{52}$

Además, algunas clases especiales de curvas elípticas son susceptibles a ataques peculiares, por ejemplo Smart, Satoh y Araki mostraron independientemente que el problema del logaritmo discreto para clases especiales de curvas elípticas anómalas es fácil de resolver. Una curva elíptica anómala sobre GF(p) es una curva elíptica que tiene exactamente q puntos, es decir que el orden de la curva es igual al número primo sobre el cual se define. Este ataque no se extiende a ninguna otra clase de curvas elípticas. Consecuentemente, para verificar que el número de puntos en una curva elíptica no sea igual al número de elementos en el campo marcado se tiene que asegurar que el ataque de Smart-Satoh-Araki no se aplique a una curva en particular.

Si existe o no un tiempo subexponencial para resolver el problema del logaritmo elíptico es una cuestión importante y de gran relevancia para la seguridad de los sistemas criptográficos basados en curvas elípticas. A pesar del trabajo requerido, para resolver el problema del logaritmo discreto desarrollado en los últimos años y específicamente en el problema sobre curvas elípticas, no se ha

descubierto un algoritmo para encontrar en un tiempo subexponencial la forma de resolver el problema del logaritmo discreto en curvas elípticas.

Se ha comparado el tiempo requerido para romper los sistemas de Criptografía de Curvas Elípticas (CCE) con el tiempo que se requiere para romper RSA para varios tamaños de claves utilizando los mejores algoritmos conocidos para factorizar los números. Los valores fueron procesados en millones de instrucciones por segundo en año (MIPS/año), lo cual representa un tiempo de cómputo de un año en una máquina que logra un millón de instrucciones por segundo. Los resultados se muestran en la tabla 3.3. Esta comparación ilustra el atractivo de la criptografía de curvas elípticas, en especial cuando se necesitan altos niveles de seguridad.

Tabla 3.3 Comparación del tamaño de clave entre RSA y CCE y el tiempo que se requiere para romperla

Tiempo de ruptura [MIPS/años]	Tamaño de clave RSA [bits]	Tamaño de clave CCE [bits]	Proporción de tamaño de clave entre RSA y CCE
10^4	512	106	5:1
10^8	768	132	6:1
10^{11}	1024	160	7:1
10^{20}	2048	210	10:1
10^{78}	21000	600	35:1

En general en nuestros días se acepta que 10^{12} [MIPS/años] representa una seguridad razonable, ya que con esto el factorizar un número con tales características requiere más poder de procesamiento en el planeta del que se es capaz de utilizar al mismo tiempo.

Se puede decir que para un nivel razonable de seguridad el sistema de cifrado RSA requiere una clave con tamaño de 1024 bits, mientras que para ofrecer el mismo nivel de seguridad el cifrado con curvas elípticas requiere de una clave de tan sólo 160 bits. Por lo tanto se observa que el sistema de curvas elípticas requiere claves menores que RSA y que la diferencia en el nivel de seguridad entre ambos se hace mayor conforme el tamaño de la clave aumenta. Además una clave de 300 bits en CCE es significativamente más segura que una clave de 2000 bits en RSA ya que el problema del logaritmo discreto elíptico (en el cual se base la CCE) es considerado como un problema más complicado, por las características vistas en el capítulo anterior.

En la figura 3.1 se muestra una comparación de los niveles de seguridad que ofrecen tanto RSA (junto con su firma digital) como CCE. Se puede ver que para tener el mismo tiempo de ruptura RSA requiere claves de mucho mayor tamaño que con CCE.

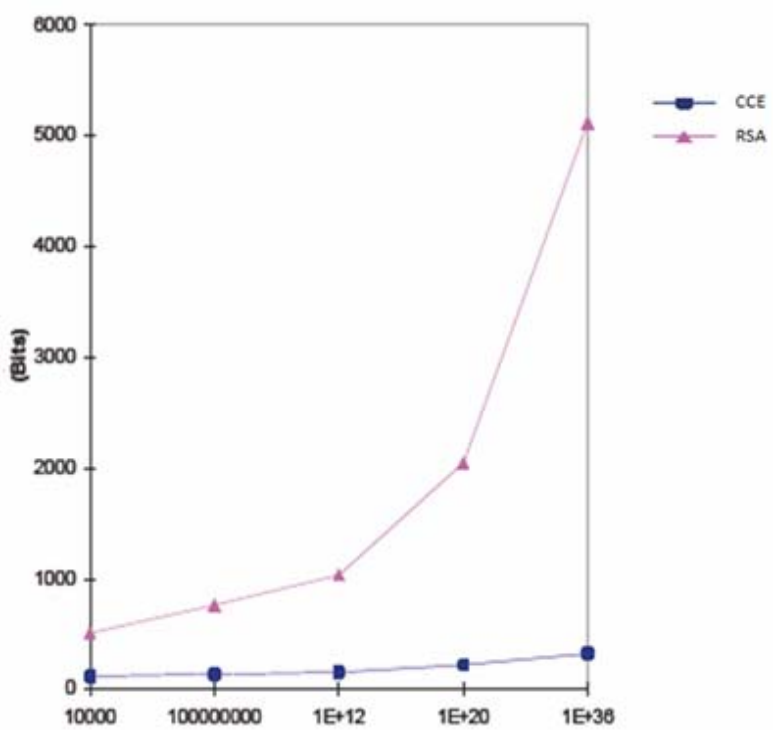


Figura 3.1 Comparación de niveles de seguridad entre RSA y CCE, en las ordenadas tenemos el tamaño de la clave y el las accisas el tiempo para romper la clave en MIPS/años.

Por otro lado, al ver ambos criptosistemas de seguridad y de las equivalencias en sus claves para ofrecer los mismos niveles de seguridad, se debe tomar en cuenta que para elementos reducidos como tarjetas electrónicas, teléfonos celulares, memorias USB se necesitan niveles de seguridad mayores, ya que estos elementos trabajan en ambientes más rudos, y se tiene que CCE puede ofrecer estos niveles mayores de seguridad utilizando claves de menor tamaño que RSA, además de no necesitar grandes adaptaciones a los recursos del sistema.

En septiembre de 1999, alrededor de 200 personas utilizaron 740 computadoras para descifrar una clave de 97 bits cifrado con Criptografía de Curvas Elípticas. El proceso tomo 16000 [MIPS/años] de procesamiento, aproximadamente dos veces el utilizado por otro equipo para encontrar una clave de 512 bits cifrado con RSA.

El 4 de abril de 2000, un equipo internacional de investigadores franceses anuncio la solución del reto lanzado por la compañía Certicom ECC2k-108.

La solución tomó cuatro meses e involucró aproximadamente 9500 computadoras y 1300 trabajadores voluntarios de 40 países. Para darle solución a tal reto se utilizo el método Pollard. El trabajo requerido para resolver ECC2k-108 fue aproximadamente 50 veces mayor que el requerido para resolver el sistema de 512 bits utilizando RSA. Lo que da una idea de la ventaja que puede tener CCE con una clave casi 5 veces menor ofreciendo el mismo nivel de seguridad que RSA.

El esfuerzo realizado por los investigadores en Francia para darle solución a éste desafío es de vital importancia porque nos provee información práctica acerca de la teoría estimada en la dificultad del problema del logaritmo discreto en curvas elípticas y la seguridad que brinda CCE.

3.1.2 Eficiencia

La seguridad no es el único atractivo de la criptografía de curvas elípticas, ya que los sistemas basados en CCE además son computacionalmente más eficientes que RSA. Además, como ya se puede intuir, la aritmética de curvas elípticas es más complicada también a nivel de bits que RSA.

La eficiencia de un sistema de cifrado de tipo asimétrico se refiere a las capacidades que provee un algoritmo en cuanto a su funcionalidad en un espacio requerido.

La eficiencia de un algoritmo es medido por los recursos que éste consume. Normalmente la forma de medir la eficiencia es el tiempo, aunque algunas veces otras medidas son importantes tales como espacio y número de procesos utilizados. Es razonable esperar que un algoritmo consuma muchos recursos cuando se procesan grandes entradas y la eficiencia de un algoritmo puede ser descrita como una función del tamaño de entrada.

Expresiones como tiempo de procesamiento en un sistema son muy usuales, sobre todo cuando hay independencia de plataforma utilizada para implementar un algoritmo. Éste se calcula por estimación del número de operaciones elementales que se ejecutan.

En la figura 3.2 podemos observar el impacto en el tiempo de reacción del servidor con claves de RSA y CCE que ofrecen el mismo nivel de seguridad. Como se puede notar las diferencias son importantes muy a favor de CCE, así mismo cabe aclarar que en la figura CCE aparece como ECC (por sus siglas en inglés) en la figura.

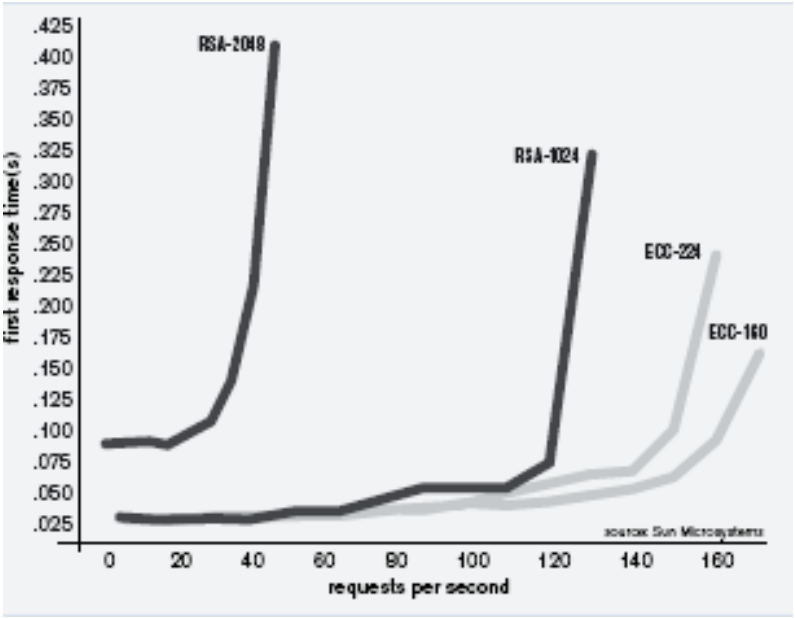


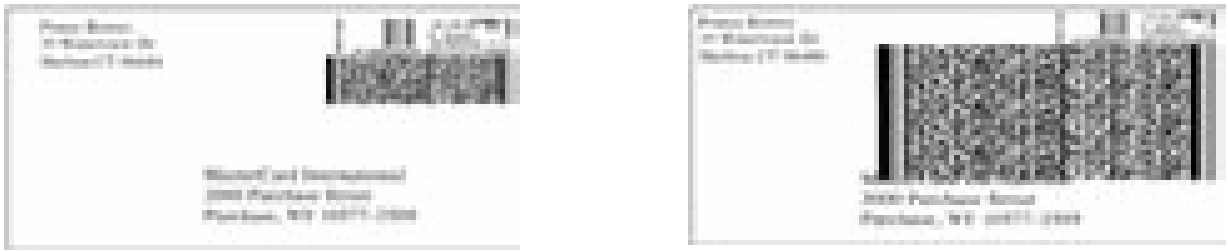
Figura 3.2 Respuesta del servidor con longitudes de claves en RSA y CCE que ofrecen el mismo nivel de seguridad.

En la tabla 3.4 se muestra los resultados obtenidos en 1998 por la empresa Certicom, la cual hizo una lista de los tiempos requeridos para operación de una clave de 163 bits utilizando CCE y otra de 1024 bits utilizando RSA.

Tabla 3.4 Comparación hecha por la empresa Certicom comparando los tiempos de operación requeridos para una clave de CCE y RSA con igual nivel de seguridad

Función	Seguridad CCE con clave de 163 bits[ms]	Seguridad RSA con clave de 1024 bits[ms]
Generación de claves	3.8	40708.3
Intercambio de claves con Diffie-Hellman	7.3	1654.7

También se tiene que las firmas digitales, las cuales son aplicaciones exclusivas de la criptografía asimétrica y que se detallarán un poco más en el siguiente subtema de este mismo capítulo, también tienen una diferencia considerable en cuanto a su tamaño y se aprecia en la figura 3.3 que es mucho menor la firma generada por la CCE que por RSA. Se aprecia que mucho del espacio de un pequeño correo electrónico es ocupado por la información de la marca postal digital, por lo cual el tener una firma digital pequeña se vuelve muy importante.



(a) (b)

Figura 3.3 Una comparación física el la marca postal de tipo digital basada en la firma digital CCE (a) y en la firma digital RSA (b).

La eficiencia es un parámetro importante a considerar cuando comparamos RSA con CCE, ya que la mayor eficiencia en el uso de los recursos computacionales ayuda a tener velocidades más altas, un consumo de energía más bajo y reducciones en el tamaño del código.

3.1.3 Espacio y requerimientos

Los criptosistemas de curvas elípticas tienen el potencial para proveer una seguridad equivalente a otros esquemas de seguridad existentes, pero con claves de menor tamaño. Tener claves pequeñas es un factor que puede ser importante en algunas aplicaciones, por ejemplo en aquellas donde los recursos computacionales son reducidos. Otra ventaja que puede dar la criptografía de curvas elípticas es que cada usuario puede seleccionar libremente una curva E(x, y) diferente, aún cuando se trabaje sobre el mismo campo GF(p).

También las CCE pueden hacer compactas las aplicaciones en hardware utilizando una clave que no requiere mucho espacio para ser procesada.

En la tabla 3.5 se compara la selección de pares de claves para los dos sistemas. Se pone en evidencia que los parámetros y las claves utilizadas son de menor tamaño en CCE que en RSA.

Tabla 3.5 Requerimientos de las claves de RSA y CCE

	Clave pública [bits]	Clave privada [bits]
RSA con 1024 bits	1088	2048
CCE con 160 bits	161	160

Ambos sistemas tienen similar requerimiento en el ancho de banda cuando se utilizan para cifrar mensajes largos, pero la situación cambia cuando se desea cifrar un mensaje menor. Pero se tiene que los sistemas de cifrado asimétrico son usualmente empleados para transmitir mensajes cortos como por ejemplo para transmitir la clave de un sistema de cifrado de tipo simétrico. Es por esto que la CCE presenta ventajas también en el ancho de banda utilizado.

En la figura 3.4 se puede ver una caricatura de cómo las claves (o las llaves) demasiado grandes pueden causar problemas a dispositivos que no tienen tantos recursos computacionales.

No obstante, RSA es mundialmente aceptado y domina el campo de la criptografía de clave pública. Esto cambiará hasta que existan mayores investigaciones acerca como trabaja la criptografía de curvas elípticas y se entienda como funcionan.

Existen muchos factores que pueden influenciar para hacer una elección. Todas deben ser consideradas simultáneamente para garantizar la mejor solución para una aplicación en particular. Los factores más relevantes deben incluir consideraciones de seguridad, plataforma de implementación, ambiente particular de cómputo y de

las características especiales de los servicios de comunicaciones con los que se cuenten.

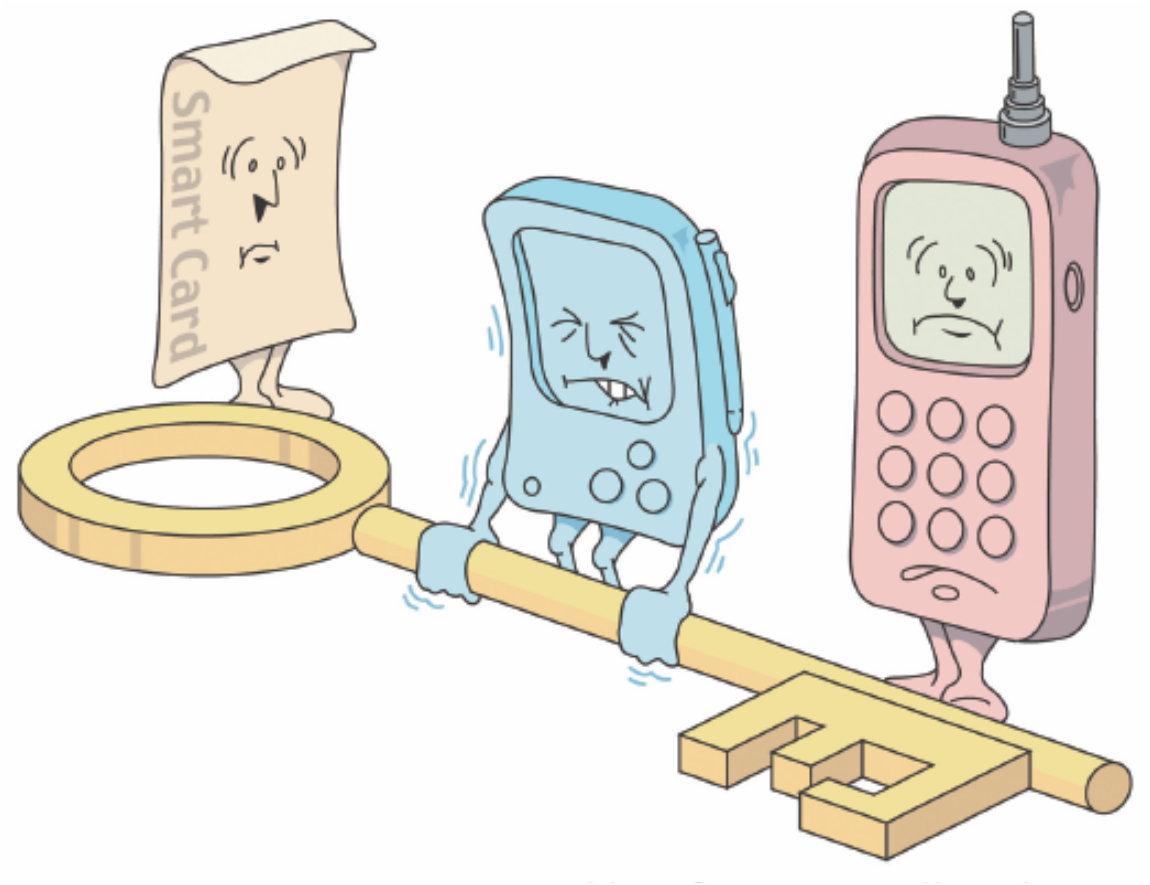


Figura 3.4 Llaves de cifrado demasiado grandes pueden ser un problema para elementos pequeños.

En resumen, la criptografía de curvas elípticas provee mayor eficiencia que el sistema de cifrado RSA, en términos computacionales, tamaño de clave y ancho de banda. En implementaciones, éstas diferencias se traducen en mayores velocidades, menor consumo de recursos y reducción en el tamaño de código.

Se puede considerar que por ejemplo para proteger información clasificada y no clasificada la Agencia de Seguridad Nacional en Estados Unidos (NSA, por sus siglas en inglés), ha decidido utilizar curvas elípticas para hacerse cargo de las comunicaciones basadas en criptografía de clave pública.

Los Estados Unidos, Gran Bretaña y otras naciones tienen todo para adaptar de alguna forma la Criptografía de Curvas Elípticas en futuros sistemas de protección

clasificada entre sus gobiernos. Muchas de las necesidades pueden ser satisfechas con sistemas basados en el uso de la Criptografía de Curvas Elípticas. Sería ideal que México se uniera a este grupo para aprovechar las ventajas que ofrecen las CCE.

3.2 Aplicaciones de la CCE y sus posibles implementaciones

El nombre de Criptografía de Curvas Elípticas se debe a las estructuras matemáticas que utilizan, las cuales se empleaban para calcular las áreas de elipses. Como se ha visto hacen uso de una aritmética más complicada de la que se utilizó cuando se inventó el algoritmo RSA, pero también se puede reducir la longitud de las claves y tener aún la misma seguridad.

Es relevante resumir que las ventajas ofrecidas por la Criptografía de Curvas Elípticas son muy importantes en ambientes donde el procesamiento, el almacenamiento, ancho de banda y el poder de consumo es limitado. Es por esto que la Criptografía de Curvas Elípticas se vuelve especialmente atractiva en sistemas de bajos recursos como teléfonos celulares y smart cards.

Los factores que promueven la adopción de la criptografía de curvas elípticas son claves más pequeñas lo cual implica cómputos más rápidos, menor consumo de energía, menor consumo de memoria, ya que aunque ésta se considera en nuestros días barata, la diferencia entre cientos de bits y miles de bits es de tomarse en cuenta, ya que existen billones de claves en el mundo.

En la actualidad la tecnología de la información hace uso de equipos de cómputo y comunicaciones de tamaño reducido que requieren de la seguridad informática. El problema como ya se mencionó es que, sin importar qué tan pequeño sea el dispositivo que se quiera proteger, los adversarios van a atacar con las computadoras más grandes que puedan conseguir. De esta forma independientemente de lo pequeño que sea el sistema de cómputo, se tendrá que proteger con los criptosistemas más sólidos de los que se disponga. Es aquí donde interviene la criptografía de curvas elípticas, ya que además cada vez se conectan a Internet dispositivos más pequeños y en la medida de que el comercio electrónico y otras comunicaciones de la red continúen creciendo la criptografía de curvas elípticas puede volverse cada vez más necesaria.

3.2.1 Firmas digitales utilizando Curvas Elípticas

El propósito de utilizar una firma manuscrita es la de asociar la identidad de una persona con la información que esta registrada en algún documento, las firmas manuscritas permiten realizar esta función en tanto que las firmas digitales por el contrario permiten asociar la identidad del firmante con el documento firmado y detectar modificaciones que pudiera sufrir el mismo.

Las firmas digitales se construyen utilizando únicamente criptografía de clave pública. Para firmar se utiliza la clave privada y para verificar la firma se utiliza la clave pública. Las firmas digitales permiten garantizar los servicios de seguridad de integridad y autenticidad al mismo tiempo.

Existen diferentes tecnologías para realizar firmas digitales, la más utilizada en la actualidad es RSA. Existen otros esquemas de firma digital aunque también se basan para su funcionamiento en el problema del logaritmo discreto. Sin embargo utilizar CCE para firmas digitales tiene las ventajas antes mencionadas sobre seguridad y eficiencia con respecto a otros algoritmos de cifrado como RSA.

Se vuelve importante el mencionar que las funciones Hash permiten realizar un resumen en cuanto a la longitud fija de cualquier mensaje $h= H(m)$. Son funciones que se comportan como funciones de un solo sentido por lo cual deben cumplir las siguientes propiedades:

1. Con un mensaje m se puede calcular $h= H(m)$.
2. Dado h, es computacionalmente difícil encontrar M , tal que $H(M) = h$;
3. Con M, es computacionalmente difícil encontrar M´, tal que $H(M)=H(M´)$.

También se requiere de una condición adicional, la cual se denomina resistencia a las colisiones y consiste en que es difícil encontrar dos mensajes aleatorios M y M´ tales que $H(M) = H(M´)$.

Ahora se tiene que una firma digital en un documento es un segmento de información con base en el documento que se quiere firmar, en la clave privada del usuario que pone la firma en el documento y en una función o esquema de firma. El objetivo de una firma digital, al igual que las firmas manuscritas, es el de ligar en forma indiscutible la identidad del usuario al documento firmado.

En la figura 3.5 se muestra un esquema general de la firma digital, el emisor toma el mensaje m y le aplica una función Hash, esta función genera un resumen h(m) del archivo, el cual el emisor cifra con su clave privada. Este proceso produce un conjunto de bits que el emisor concatena con el mensaje y se lo envía al receptor. El receptor recibe el paquete y procede a separar el mensaje de su firma, luego aplica una función hash al mensaje y calcula un resumen h'(m). El siguiente paso que debe ejecutar el receptor es descifrar o realizar una transformación inversa sobre los bits de la firma utilizando la clave pública del emisor**.** En este momento el emisor puede comparar h(m) y h'(m). Si los dos son iguales el receptor entonces puede concluir que el mensaje efectivamente fue enviado por A y que éste no sufrió alteración alguna.

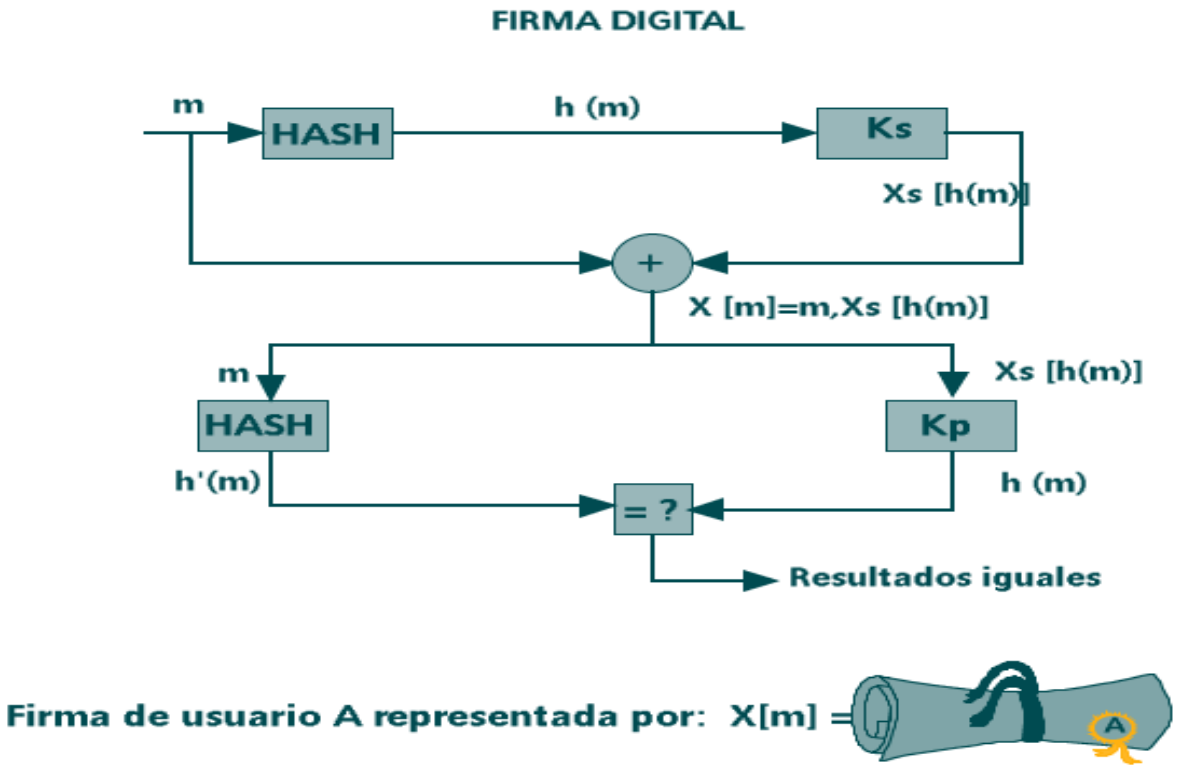


Figura 3.5 Esquema general de una firma digital.

Para realizar la firma de un documento utilizando CCE se tiene que:

Con h(m) el valor hash del documento m, utilizando E(x, y) como la curva elíptica y teniendo un punto P con (x_P, y_P) sobre la curva elíptica mencionada definida sobre el GF(p) con p un número primo grande (al menos 20 dígitos), con k como la clave privada del usuario, entonces se tiene que la clave pública Q= kP.

Para realizar la firma se debe escoger una secuencia aleatoria de bits n y calcular $R = nP = (x_R, y_R)$ y se toma su componente x_R para calcular

$c = x_R + h(m) \bmod p$

$d = n - kc \bmod p$

La pareja (c, d) es la firma digital del documento m representado por su valor hash h(m).

La firma digital se concatena con el mensaje, se comprime y se envía como un mensaje de correo electrónico. El destinatario recibe el mensaje, descomprime el archivo y separa la firma del mensaje original. Con los bits de la firma realiza los siguientes cálculos:

$R' = dP + cQ$

Usando la componente $x_{R'}$ de $R'=(x_{R'}, y_{R'})$ se calcula:

$h'(m) = c - x_{R'} \bmod p$

Como paso final el destinatario calcula el valor hash h(m) del mensaje recibido y lo compara con h'(m) . Si los valores son iguales la firma se valida. Si los valores son diferentes, entonces significa que el mensaje fue alterado y la firma es rechazada.

Se puede ver que las firmas digitales son una aplicación exclusiva de la criptografía asimétrica y por tanto son relativamente nuevas, sin embargo al necesitar claves de longitudes de al menos 1024 bits la firma digital aumenta el tamaño del archivo (no del mensaje) lo cual puede volverse un problema cuando se desea mandarlo por un canal con ancho de banda limitado, pero al utilizar CCE se tiene que las firmas digitales no deben ser tan grandes para garantizar la autenticidad del emisor y la integridad del mensaje.

3.2.2 Criptografía de Curvas Elípticas en marcas postales de tipo digital

La tecnología digital ha tenido un cambio radical sobretodo en compañías de negocios y el sistema postal no es la excepción. Las marcas postales de tipo digital

son la prueba de que se puede hacer el pago tradicional a través de enviar un correo electrónico como si se tratara del sistema postal y a vuelta de correo se envía el equivalente del timbre postal. Esto ofrece un ahorro significativo con respecto al uso tradicional en la utilización de timbres postales por el servicio postal ordinario, ya que es sencillo y económico de producir.

Mientras se vuelva mucho más sencillo de reproducir una marca postal de tipo digital será mucho más difícil continuar utilizando los timbres postales tradicionales. Existe un esquema llamado Firma Vanstone Pintsov de Curvas Elípticas (ECPVS, por sus siglas en inglés). Este método se basa en la CCE y ofrece una firma seis veces menor que una de RSA conservando los altos niveles de seguridad.

Para entender la importancia de la seguridad en las marcas postales de tipo digital, es necesario ver cómo se hace el proceso postal.

La compañía A genera una marca postal digital vía correo electrónico y lo envía a un determinado número de clientes. El sistema postal verifica que la marca postal sea valida y envía cada correo electrónico a el destino previsto. Entonces un fraude puede ocurrir cuando:

1. Un adversario intercepta un correo electrónico, copiando la marca postal digital originalmente generada por la compañía A, y la utiliza con fines maliciosos.

2. La compañía A toma la marca postal digital que generó legalmente para un correo electrónico y la utiliza con fines fraudulentos en otros correos electrónicos, reduciendo sus costos totales.

En cada caso, cada ejemplo ilustra la necesidad para mejorar la autenticidad de la información contenida con la generación de la marca postal de tipo digital. Un método podría ser la firma digital para verificar la autenticidad de las marcas postales.

El uso de la CCE basado en el esquema de firmas digitales para la industria postal no es nuevo, porque el tamaño de la firma digital es afectado por el total de la marca postal digital, firmas basadas en CCE provee una opción de seguridad con el uso de una firma digital extremadamente pequeña.

La firma Pintsov Vanstone basada en curvas elípticas se puede hacer tan pequeña como de 20 bytes los cuales se agregan a la longitud del mensaje original, esto es como seis veces menor al tamaño de la firma que utiliza RSA lo que lo hace más eficiente.

Ciertos datos como la fecha y el monto de envío son fáciles de leer por las personas, mientras que otros elementos como la dirección del emisor o la confirmación de la dirección del destinatario están estrictamente destinados a ser leídas por la computadora. Además, si la verificación de la firma es secreta, exceptuando la clave pública, entonces la parte oculta del mensaje es difícil de obtener, aún con una computadora.

Otro rasgo específico de la firma Pintsov Vanstone basada en curvas elípticas es su habilidad para ajustarse determinado nivel de seguridad dependiendo de los requerimientos que se tengan.

En la firma Pintsov Vanstone basada en curvas elípticas se tiene un mensaje en claro m el cual es dividido en dos partes: parte C y parte V. La parte C representa datos elementales que requieren protección confidencial, tales como información del emisor o algún otro valor. Esta puede ser recuperada durante el proceso de verificación de la firma y utilizada para probar el importe del proceso que se genero vía Internet. La parte V contiene datos elementales presentados en el mensaje en claro con la marca postal digital, tales como la fecha o el código postal del emisario y del receptor.

$m= C+V$

La firma Pintsov Vanstone basada en curvas elípticas utiliza una curva elíptica con un generador G de orden n. La terminal A tiene Q_A como clave pública y A como clave privada.

Para generar la firma la terminal A empieza por escoger un número entero positivo aleatorio k que sea menor que n.

La terminal obtiene la información vía correo electrónico y escoge un número para cifrar el mensaje. Primero calcula un punto R sobre la curva, para ser utilizado como clave para la transformación de C. Este punto sobre la curva elíptica es después utilizado en una transformación biyectiva (T_R) para destruir cualquier estructura algebraica que C que se pueda tener, con el resultado se obtiene e.

$R=kG$

$e=T_RC$

Después se calcula la variable d utilizando una función hash, la parte V del mensaje cifrado, la identidad del cliente denominada I_A y la transformación e.

$d=h(e, I_A, V)$

Finalmente se calcula s (la segunda parte de la firma) usando d, k y a, que es la clave privada de la terminal A.

$s= a\ d+k \bmod n$

El par de la firma (s, e) es entonces puesta dentro de la marca postal digital junto con la porción V del mensaje en claro.

Para verificar la firma de la marca postal digital de un determinado correo electrónico, un verificador en la otra terminal del proceso postal pasa la marca postal digital dentro de I_A, la firma (s, e) y la verificación de datos V. Utilizando esto y la clave pública de la terminal A, Q_A, el verificador postal recobra C y verifica que la marca postal digital sea efectiva.

$d=h(e, I_A, V)$

$U=sG – dQ_A$

$C=T_U^{-1}(e)$

Los beneficios relativos de la firma Pintsov Vanstone basada en curvas elípticas (menor tamaño, flexibilidad y eficiencia) también lo hacen ideal para aplicaciones más allá del servicio postal, tal vez como verificador de cheques emitidos vía Internet.

3.2.3 Comprobación de compra con cheques electrónicos utilizando Curvas Elípticas

La adopción del pago electrónico y los mecanismos de envío se están volviendo más extensos en la industria financiera que ve los beneficios de automatizarla, ya que no sólo reduce los costos, además mejora el servicio a los clientes. Una aplicación novedosa es el pago a través de Internet con imágenes en forma de cheque.

Obviamente, la seguridad alrededor de las imágenes digitales es muy importante. Sin alguna forma de autenticar las imágenes digitales, alguien podría interceptar la transmisión de una imagen digital, si se quisiera enviar un cheque en forma de imagen a través de la red sin tomar en cuenta algunos aspectos

importantes, esto traería pérdidas adicionales a los bancos. La autenticación además asegura que la información del cheque no contenga ninguna alteración.

El proceso de compra con cheques electrónicos representa un reto mayor, ya que las máquinas verifican miles de cheques por minuto y cada imagen debe ser firmada para asegurar que la información que contenga el cheque no sea falsa.

Estos niveles de automatización en la industria financiera pueden ser alcanzados utilizando nuevamente CCE, pero los bancos involucrados deben ser capaces de crear certificaciones para verificar la imagen de los cheques.

La imagen de los cheques se verifica al recibirse en el banco. Para verificar que el cheque es válido, la imagen se firma utilizando firma digital con curvas elípticas. El cheque en papel puede llegar después para servir como respaldo de que la operación fue realizada exitosamente. En la figura 3.6 se puede ver cómo se hace el proceso.

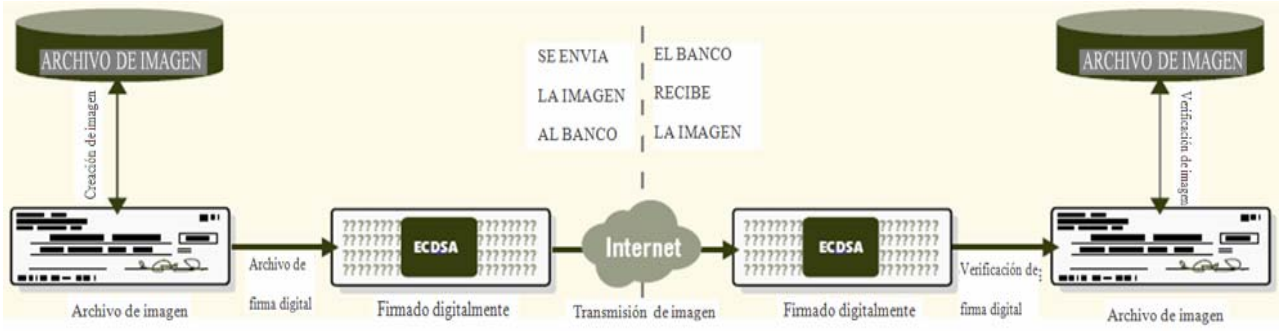


Figura 3.6 Esquema de creación y verificación de un cheque de forma electrónica.

Teniendo un alto volumen de cheques que deben ser firmados y verificados, las claves menores de CCE proveen una mejor eficiencia y desempeño, mientras se cuente con los requerimientos de hardware para verificar los cheques a un nivel razonable.

Desde que los datos se tuvieron que mantener en archivos por algunos años, se requieren firmas eficientes y fuertes. Por lo que utilizando las firmas digitales con curvas elípticas o la firma Pintsov Vanstone basada en curvas elípticas, la información permanecerá segura por un largo periodo, hasta que el poder de cómputo demande un nivel superior de seguridad.

3.2.4 Mejoramiento de las comunicaciones en Internet

Como se ha visto la CCE es la mejor opción de criptografía de tipo asimétrica cuando preocupa el rendimiento de los sistemas. Una de las áreas donde el desempeño es muy importante es en las comunicaciones vía Internet y la CCE provee los beneficios de desempeño que requieren las comunicaciones vía Internet en la actualidad.

El organismo que ve que se cumplan los estándares para la comunicación en Internet es la Internet Engineering Task Force (IETF).

La IETF es un conjunto de software abierto internacional junto con una comunidad de diseñadores de redes, operadores, vendedores e investigadores preocupados por el desarrollo de la arquitectura de Internet y de su adecuado funcionamiento. La IETF sostiene tres reuniones al año y cualquier persona puede asistir. El resto del tiempo los participantes se comunican por correo electrónico.

La forma actual de trabajo del IETF se hace en grupos, los cuales son organizados en ocho áreas diferentes y la seguridad es uno de estos.

Los estándares de la IETF comienzan como un borrador en Internet (I-D, por sus siglas en inglés). Para comenzar un estándar, un borrador del documento debe ser publicado como un I-D para que los participantes interesados puedan hacer comentarios y se retroalimenten. Después de un tiempo el borrador es presentado a un grupo para su revisión y publicación y se le asigna un número de RFC (Request For Comments).

IPSec es uno de los protocolos donde CCE se usa actualmente. Típicamente IPSec utiliza el algoritmo Diffie-Hellman para generación de claves, el cual es adecuado para computadoras de escritorio, pero es muy lento para dispositivos pequeños. Diffie-Hellman con Curvas Elípticas (ECDH) o intercambio de claves Diffie-Hellman con curvas elípticas provee un desempeño más rápido. Ya que muchas compañías desean garantizar la seguridad de las conexiones en elementos pequeños como teléfonos celulares considerando al tiempo como un factor importante para que esto se lleve a cabo.

Utilizar CCE para transacciones seguras se vuelve indispensable ya que las transacciones necesitan ser procesadas de una manera eficiente, porque cada día más

elementos diminutos están comenectándose a Internet y éstos requieren seguridad desde smart cards hasta sensores de control de procesos.

El poder de procesamiento se ha incrementado y los crackers tienen el día de hoy más recursos a su disposición como nunca antes. Aunque una clave de 1024 bits para una clave de RSA sea el tamaño más utilizado en nuestros días, el uso de una clave de 2048 bits para el mismo algoritmo de cifrado se vuelve cada día más común. El inconveniente de aumentar el tamaño de las claves, además de consumir recursos en el sistema, es el impacto que esto trae a un servidor. De acuerdo con esto el tráfico que se genera por alguien que utiliza RSA comparado con alguien que utiliza CCE sería de una proporción de 3.5 veces mayor.

Se ha hecho un esfuerzo considerable para que la CCE se incorpore a los estándares de la IETF, las compañías están utilizando algoritmos basados en CCE para sus comunicaciones vía Internet. La criptografía de curvas elípticas está emergiendo como la mejor opción, pero es necesario que se sigan escribiendo borradores que sean adaptados como RFCs.

3.2.5 Implementación de seguridad en smart card con Curvas Elípticas

Una smart card típica tiene un procesador con frecuencia de 3 a 5 MHz y aproximadamente de 4 a 32 Kbytes de memoria para leer y escribir. Esto nos impedía utilizar la criptografía de clave pública de manera eficiente, ya que las claves de la criptografía asimétrica necesitan ser de mayor tamaño en comparación con las claves utilizadas por las claves de la criptografía simétrica, sin embargo la criptografía de curvas elípticas puede solucionar este problema ya que nos provee del mismo nivel de seguridad que se tendría con RSA pero con claves más pequeñas.

Un factor importante es el crecimiento en la capacidad de procesar información que hoy en día tienen todos los usuarios de computadoras. Esto se debe a la amplia variedad de elementos que cada día son más económicos y que pueden estar al alcance de más personas casi siempre con una conexión a Internet. De esta forma el acceder a la información de una smart card, sin conocer su clave se vuelve muy atractivo para un criptoanalisista y por tanto un problema latente para los usuarios de este tipo de dispositivos. La criptografía de curvas elípticas puede ser la solución a un problema como éste ya que la diferencia en el tamaño entre las claves aumentará

constantemente, pues la capacidad de cómputo también aumentará y se requieren niveles de seguridad en las smart card que sólo la CCE puede llegar a ofrecer.

Las smart card son dispositivos con la forma y el tamaño de una tarjeta de crédito. Sin embargo, estos dispositivos rinden mucho más. Se llaman 'smart' debido al microchip que se integra sobre ellos. Dentro de este chip las tarjetas tienen un CPU, una memoria permanente y periféricos de entrada y salida, de hecho se les considera microprocesadores portátiles.

La información contenida en las smart card se almacena en una memoria EEPROM y se puede ver solamente usando la interfaz bien definida para las tarjetas con los permisos del sistema operativo, la tarjeta y el software. Aunque son inferiores a los ordenadores personales en términos de velocidad, memoria y los dispositivos de entrada y salida, son de hecho superiores por sus características sofisticadas de la seguridad. Es por esto que se vuelven candidatos excelentes a cifrar información con CCE basados en implementaciones de hardware ya que además trabajan en ambientes más rudos que las computadoras de escritorio.

3.3 El futuro de la CCE

La criptografía de curvas elípticas como ya se mencionó fue descubierta en 1985. Antes de esto se pensaba que las curvas elípticas pertenecían a la matemática pura, hasta que hubo quienes consideraron que podría tener aplicaciones prácticas.

A lo largo de los últimos años, se han hecho muchas investigaciones, trayendo consigo numerosos estándares que se han incorporado a la CCE.

El 24 de octubre de 2003 el gobierno de Estados Unidos a través de su agencia NSA (National Security Agency) en un movimiento sin precedentes decidió aplicar la CCE para misiones de carácter crítico, declarando a ésta como una "una tecnología crucial".

La adopción hecha por el gobierno de los Estados Unidos puede ayudar a empujar a la CCE al ancho uso comercial. Ya que hoy en día los gobiernos, las agencias y los departamentos buscan productos con aplicaciones comerciales que provean de seguridad y que tengan ventajas significativas con respecto a la tecnología del pasado. Y las ventajas que presenta CCE provee aplicaciones igualmente exitosas tanto en la industria como en las finanzas y, como se vio, en el servicio postal.

Además de los Estados Unidos, otros gobiernos están volteando a la CCE, por ejemplo, el gobierno de China está considerando el utilizar a la CCE para que resuelva el problema de la seguridad en las redes inalámbricas de ese país. Los niveles de seguridad que presenta la CCE puede ser el ideal para solucionar problemas como éste.

3.3.1 Seguridad en Internet con CCE

El comercio electrónico juega un papel importante en la economía global y su importancia va aumentando cada año. En la actualidad en México no se tiene la cultura de hacer compras en Internet por considerarlas inseguras y lentas, sin embargo esto puede cambiar con un sistema que ofrezca un nivel de seguridad que garantice a los usuarios que su dinero no corre peligro y que los proveedores cumplirán con lo pactado en un tiempo razonable. Para esto los protocolos se vuelven necesarios para asegurar el aislamiento y la seguridad de las transacciones hechas en línea.

De acuerdo con cifras del Banco de México, en 2006 a través de Internet se realizaron transacciones por casi 11 billones 879 mil millones de pesos, 9.7 por ciento más que el año anterior, y éstas operaciones fueron hechas principalmente por grandes empresas e instituciones de Gobierno. Mientras que en la parte bancaría en línea el crecimiento fue de 50 por ciento, y aunque el universo de usuarios aun es pequeño, que son alrededor de 29 por ciento de los 20.2 millones de cibernautas en nuestro país que había en 2006.

El principal reto es promover entre las personas el uso de la tecnología, sus ventajas, riesgos y sobretodo las medidas de seguridad que se necesiten hacer para realizar operaciones en línea. Y en el caso de la Ingeniería aplicar un método de cifrado sólido que pueda ofrecer un nivel de seguridad adecuado para dichas operaciones.

Se tiene que varios estudios han divulgado que los servidores seguros funcionan de tres a nueve veces más lentos que los servidores regulares en la misma plataforma de hardware. Por esto el tiempo de reacción es lento y esto es una causa importante de que los compradores desistan de hacer una compra o de llevar a cabo una operación en línea.

Hasta ahora, los abastecedores del servicio le han hecho frente al problema con equipos y conexiones más y más poderosos lo cual repercute en los altos costos de procesamiento.

Cada vez más dispositivos de menor tamaño están utilizando la tecnología de conectarse a Internet para facilitar algunas funciones, tal como la tecnología U3 la cual corre aplicaciones en computadoras que no necesitan tener instalada dichas aplicaciones. Muchos de estos dispositivos requieren asegurar la conexión de manera confidencial, es decir que ningún curioso tenga acceso a su información. Generalmente estos dispositivos son mucho más limitados en memoria y energía que una computadora tradicional, por lo que para proporcionar los niveles adecuados de seguridad es necesario hacer uso de la criptografía.

El volumen de las transacciones comerciales hechas por Internet se proyecta para duplicarse en pocos años, trayendo con esto aumento en el costo de la conexión por la carga en el servidor de la empresa.

Los consumidores empiezan a expresar su deseo por tener más privacidad en todas las actividades que realicen en línea, de manera que mucha gente siente que la protección que se le brinda a la información de su tarjeta de crédito no es suficiente y por lo tanto desean incrementar la seguridad en sus consumos.

Como se ha explicado, la CCE puede ser la solución a problemas de este tipo ya que al utilizar claves pequeñas también se reduce el tiempo de respuesta en el servidor manteniendo el mismo nivel de seguridad que se tendría con un sistema basado en el tradicional algoritmo RSA, por lo que se hace cada vez más necesario que los diseñadores de este tipo de dispositivos conozcan más de cerca las ventajas que ofrece el utilizar CCE.

3.3.2 El futuro de los certificados digitales

Los méritos técnicos de los certificados digítales basados en CCE hacen que sea una excelente elección para diversas aplicaciones. La seguridad, la fuerza y el menor tamaño proveen numerosos beneficios de seguridad y un mejor funcionamiento, también asegura una larga vida al sistema de seguridad.

Además de las aplicaciones de firmas digitales y certificados mencionados en este capítulo, también se tiene un amplio rango de usos en otros mercados como el de los consumidores, ya que se puede utilizar la certificación basada en CCE para

garantizar que un usuario desea adquirir un determinado producto y no se tenga problemas con el no repudio o con la autenticación.

Se puede aplicar también en el tratamiento médico, para garantizar que a pesar de realizarse un chequeo médico a través de lo que se conoce como telemedicina, se tenga la confianza de ser atendido por un especialista a pesar de que éste se encuentre en un lugar lejano.

Otra aplicación se puede dar en la transmisión de imágenes en dos dimensiones que necesiten garantizar su autenticidad, esta aplicación se vuelve muy atractiva para la publicación de imágenes en los periódicos, las noticias impresas y en la ya mencionada telemedicina para diagnóstico de radiografías, ultrasonidos y otros.

La CCE tiene todo a su alcance para ser considera en el futuro como la forma de autenticarse más común, y más segura, a través de las operaciones realizadas por Internet.

3.3.3 Seguridad en dispositivos diminutos utilizando CCE

Los investigadores de los laboratorios SUN han credo el servidor más pequeño del mundo. Este servidor, del tamaño de una moneda, se puede incrustar en una amplia variedad de dispositivos ligeros, incluyendo aparatos electrodomésticos, los dispositivos médicos de tipo personal y sensores industriales para la supervisión a través de Internet. En la figura 3.7 se tiene una fotografía de este servidor.

A pesar de su tamaño el servidor no tiene ningún dispositivo de seguridad, es por esto que utilizar CCE resulta la mejor opción ya que comparada con la tecnología de clave pública mas utilizada en Internet, RSA, ofrece el mismo nivel de seguridad utilizando menos recursos. La ventaja del funcionamiento de CCE se puede garantizar, ya que el tamaño de sus claves es mucho menor.

Figura 3.7 El servidor más pequeño del mundo, del tamaño de una moneda.

Así es que en forma general CCE nos ofrece aplicaciones para diversas áreas y muchas de ellas son las mejores opciones cuando los recursos de cómputo son limitados y se requiere tener el mismo nivel de seguridad que se tendría en un dispositivo o un equipo de mayores dimensiones. Esto hace a la CCE muy atractiva para aplicaciones futuras en donde cada vez tendremos más usuarios conectados a Internet, dispositivos más pequeños y muchas otras.

3.3.4 Seguridad en el Registro Público Vehicular utilizando CCE

En términos generales hemos visto que la CCE tiene muchas aplicaciones, las cuales son las más adecuadas cuando se tienen recursos limitados en los equipos. Es por esto que la posibilidad de implementar la CCE para encargarse de la seguridad en un proyecto de nivel nacional como lo es el Registro Público Vehicular (REPUVE) es una opción bastante interesante.

La Secretaría de Seguridad Pública, junto con tres universidades del país, ha decidido evaluar la posibilidad de utilizar un sistema de identificación de radiofrecuencia (RFID) para llevar a cabo el Registro Público Vehicular en todo el país. Además de los datos del vehículo y del dueño del mismo se tendría la posibilidad de saber si el automóvil es reportado como robado o si tiene alguna multa pendiente. Con esto se podría ubicar rápidamente su localización y su posible captura en caso de infligir en algún delito.

Un sistema de RFID (Radio Frequency IDentification) es la tecnología inalámbrica que permite básicamente la comunicación entre un lector y una etiqueta. Estos sistemas dan la oportunidad de almacenar información en sus etiquetas mediante comunicaciones de radiofrecuencia, esta información que puede ir desde un bit hasta kilobytes, dependiendo principalmente del sistema de almacenamiento que posea la etiqueta electrónica.

Un tag, transponder o etiqueta electrónica contiene un microchip y una antena que puede adherirse a cualquier producto. Incluso se están desarrollando tags que son de un tamaño tan pequeño que pasarían inadvertidas en algunos objetos.

El funcionamiento del sistema es bastante sencillo, como se observa en la figura 3.8, el lector envía una serie de ondas de radiofrecuencia al tag, que son captadas por la microantena de éste, dichas ondas activan el microchip, el cual a través de la microantena y mediante ondas de radiofrecuencia, transmite al lector la información que se tenga en su memoria.

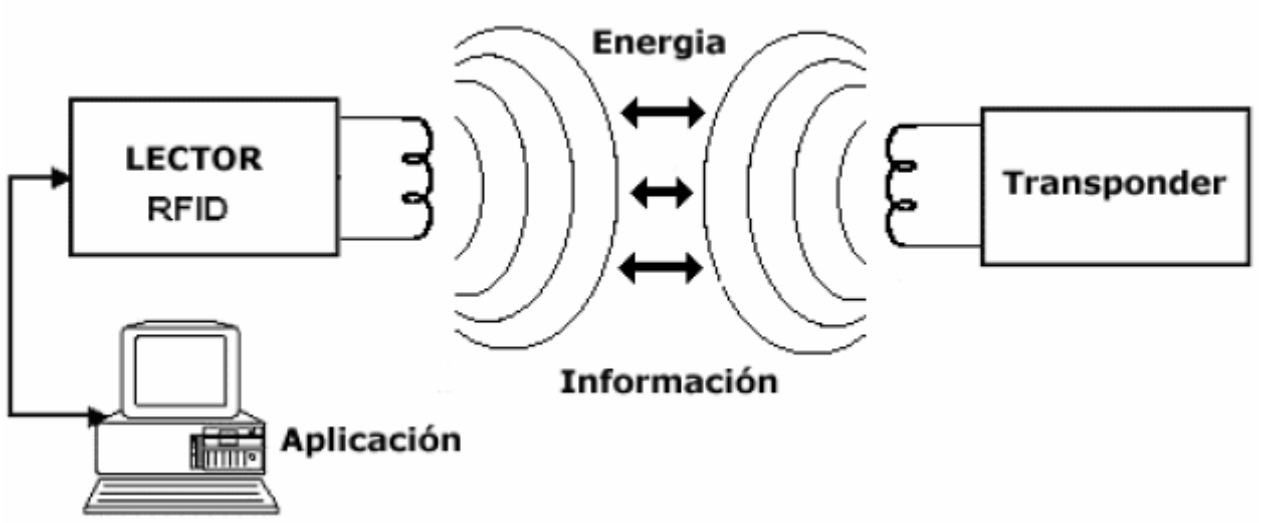


Figura 3.8 Esquema de un sistema RFID.

Finalmente, el lector recibe la información que tiene el tag y lo envía a una base de datos en la que previamente se han registrado las características del producto o puede procesarlo según convenga a cada aplicación.

Uno de los mecanismos más importantes en este esquema es el tag o transporder, el cual tiene ciertas características. La palabra transponder deriva de TRANSmitter/resPONDER. Los componentes básicos de un transponder son:

• Una memoria no volátil donde se almacenan datos.
•Una memoria ROM donde se almacenan instrucciones básicas para el funcionamiento, como son temporizadores, controladores de flujo de datos, etcétera.
•Una memoria RAM para almacenar datos durante la comunicación con el lector(es opcional).
• La antena por la cual detecta el campo creado por el interrogador, y del que extrae energía para su comunicación con él.
• Componentes electrónicos que procesan la señal de la antena y para el proceso de datos, como buffers, filtros.

Se pueden distinguir dos tipos de etiquetas dependiendo de la energía que se utiliza para la comunicación:

• Etiquetas activas: son transponders que necesitan el apoyo de baterías adicionales, ya que no tienen suficiente energía con la que proporciona el lector. Este tipo de etiqueta tiene la ventaja de poseer un alcance mayor de comunicación e incluso no necesitan que el lector sea quién inicie la comunicación. Además permiten habitualmente procesos de lectura y reescritura enviando previamente instrucciones al lector y la utilización de memorias más grandes (existen etiquetas con 1Mb de memoria). Por el contrario ofrecen una vida útil limitada (menos de diez años), dependiendo del tipo de batería y de las temperaturas a las que opera.
• Etiquetas pasivas: son transponders que no necesitan baterías adicionales, ya que únicamente se alimentan de la energía del campo generado por el lector. Para las etiquetas pasivas, la energía que necesitan para transmitir la información que contienen, proviene en su totalidad de la señal generada por el lector. Estas etiquetas aprovechan la energía subministrada por un lector para generar su propia señal que recibe nuevamente el lector.

Los transponders tienen diversas formas y tamaños, se puede apreciar una de ellas en la figura 3.9, todo dependiendo de la aplicación a la cual están destinados. Los transponders que se utilizan para el control y localización tienen un tamaño inferior a 10 mm.

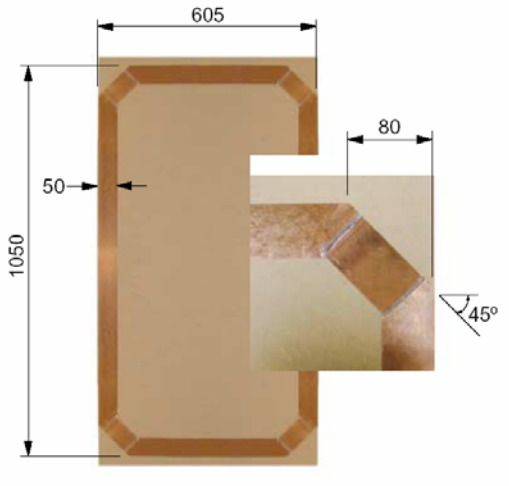


Figura 3.9 Detalle de un tag típico de aplicaciones logísticas, con las unidades expresadas en milímetros.

La capacidad de almacenar información en un tag es muy limitada dado su tamaño, por lo que se vuelve importante utilizar un algoritmo para cifrar información de los datos del vehículo y del dueño con un algoritmo de cifrado que no consuma grandes recursos. Ya que como se mencionó en el caso de los tags activos se tiene limitaciones en la duración de la pila y los tags pasivos no tienen grandes dimensiones para guardar mucha información. Es por esto que utilizar CCE se vuelve una alternativa particularmente aceptable para la seguridad de este sistema, ya sea porque se requiere ahorro de energía o se tenga muy poca memoria para almacenar una clave.

En general, si se necesita que los dispositivos pequeños sean seguros, se requiere CCE, si se intenta que funcionen por mucho más tiempo con la misma batería produciendo mucho menos calor, es necesaria CCE, y si se hace indispensable un criptosistema asimétrico que funcione en aplicaciones para el futuro, la mejor opción es CCE.

Capítulo 4

Sistema de aprendizaje de criptografía de curvas elípticas

En los capítulos anteriores se hablo acerca de las 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 (ofrecen el mismo nivel de seguridad que otros algoritmos asimétricos como RSA pero con claves de menor tamaño).

En el presente capítulo se presenta el desarrollo del sistema de aprendizaje de criptografía de curvas elípticas basado en la idea de que los algoritmos criptográficos que la sostienen tienen ventajas en cuanto a la seguridad, la eficiencia en el uso de los recursos, en el espacio y en los requerimientos computacionales que necesitan para operar, sin embargo son poco difundidos en comparación de algunos otros algoritmos de cifrado asimétrico como RSA, tal vez por que requiere una comprensión y uso de elementos matemáticos mas profundos que en ocasiones dificultan la concepción de las bases de la criptografía basada en curvas elípticas.

Por tal motivo el uso del software desarrollado ayudará a un mejor entendimiento de algunos elementos abstractos explicados en el capítulo 2, de tal forma que puede servir de apoyo para introducir a las personas interesadas en criptografía de curvas elípticas al estudio de esta parte de la criptografía asimétrica que empieza a ser utilizada en la criptología moderna.

4.1 Selección de la herramienta de software

Para cubrir el último objetivo de la presente tesis se desarrollo el sistema que permite a todo aquel interesado en criptografía comprender y manejar los conceptos que se utilizan en la Criptografía de Curvas Elípticas, en este sistema se implementaron ejemplos y un sistema que permite a los usuarios manipular algunos parámetros de las curvas elípticas, esto para observar como se ve afectado el usar diversos parámetros. Lo anterior hace que el software no sólo funcione para los ejemplos propuestos, también para aquellos que a consideración del usuario puedan servir para el estudio y la mejor compresión de cómo es que trabaja la criptografía con las curvas elípticas.

Para el desarrollo del sistema se utilizó Microsoft Visual Basic 2005, ya que presenta algunas ventajas con respecto a las versiones anteriores como lo son Visual Basic 6.0 y Visual Basic. NET, la principal es que en la versión 2005 se pueden correr las aplicaciones que fueron creadas en sus dos versiones anteriores, cosa que no ocurre con ninguna de las versiones precedentes, esto con la finalidad de integrar aplicaciones y funciones que sirvieron para el desarrollo del software.

El tutorial puede ser modificado en un futuro con versiones en Visual Basic y Visual Basic.NET gracias a la integración de sistemas antiguos con los nuevos esquemas, en la versión 2005 se proveen herramientas específicas de integración de sistemas antiguos con las versiones recientes.

En la versión .NET, el código fuente se traduce en un lenguaje intermedio, llamado MSIL (Microsoft Intermediate Language). Esto proporciona servicios como verificación de código, manejo de memoria, colección de basura y seguridad.

Visual Basic está diseñado para aplicaciones en los sistemas operativos Windows, esto trae ciertas ventajas como el manejo de ventanas con lo cual hace que el sistema de aprendizaje de Criptografía de Curvas Elípticas sea amigable con los usuarios y éstos sólo se enfoquen la comprensión del contenido del tutorial más que en el manejo del software.

Una de las razones principales por las que se eligió la plataforma Visual Basic 2005 es por la independencia del lenguaje y la transparencia a través de las redes, ya que se puede combinar lenguajes tan representativos de una plataforma .NET como C#, C++, Visual J y el mismo Visual Basic, utilizando la ventaja que permite que se utilice un lenguaje amigable, pero aprovechando los servicios y beneficios de la plataforma. Así se podría utilizar diversos lenguajes en el sistema sin que se tuviera que hacer grandes modificaciones, esto con la finalidad de poder manejar combinaciones con los gráficos desarrollados y las ecuaciones matemáticas que se

manejan en Curvas Elípticas, haciendo que en las láminas se puedan ejecutar ambas aplicaciones.

Por eso la diversidad de los lenguajes de programación que ofrece la plataforma .NET (además de Visual Basic), tiene soporte para un extenso número de lenguajes, cosa que no sucede con otras plataformas que sólo soportan un solo tipo de lenguaje como JEE que sólo soporta el lenguaje de programación Java. Así en caso de realizar mejoras en el futuro al sistema un programador que domine un lenguaje distinto a Visual Basic podrá hacer las modificaciones que considere necesarias sin necesidad de tener que aprender a manejar el lenguaje de programación Visual Basic.

En términos generales se puede apreciar que se eligió el lenguaje Visual Basic 2005 por que es factible de utilizar ya que soporta aplicaciones realizadas en el mismo lenguaje pero con versiones anteriores, la plataforma que lo soporta (.NET) es multilenguaje lo que permite combinar distintos lenguajes de programación sin que afecte a las aplicaciones y finalmente por lo amigable que resultan sus aplicaciones al estar diseñado para correr en sistemas operativos tipo Windows, haciendo que elementos dentro de las láminas del software tales como gráficos o ecuaciones sean el principal atractivo para los usuarios que utilicen el tutorial de curvas elípticas que se desarrollo.

4.2 Diseño y desarrollo

El desarrollo del sistema se hizo con base en la creación de prototipos de software, ya que de este modo fue posible efectuar refinamientos al producto final. Además, se utilizó una gama de herramientas de construcción para generar pantallas que permitieron definir la disposición de diversos elementos en las mismas para aplicaciones interactivas.

Tal y como ocurre en los sistemas de construcción de prototipos se desarrollo el sistema de aprendizaje de Criptografía de Curvas Elípticas, ya que se comenzó con recolectar y analizar toda la información que se tenía referente al tema en cuestión. Se determinaron cuales eran los objetivos de la tesis y particularmente el de programar los algoritmos que se utilizan para que estos estuvieran al alcance de las personas interesadas en criptografía de tal forma que funcionara como herramienta de apoyo para la comprensión del fascinante mundo de las Curvas Elípticas y principalmente de su uso a la criptografía. Después se diseño un modelo con las características que hasta ese momento se consideraban importantes, después se centro en las características visibles para el usuario del sistema, es decir la interfaz

y a partir de ahí se hizo el primer prototipo que sería el inicio del sistema de aprendizaje de Criptografía de Curvas Elípticas. A partir de ahí se comenzó a revisar y actualizar el contenido que se exponía en el tutorial y se definieron nuevas características que debía contener el sistema.

En la primera lámina se da una bienvenida al tutorial, explicando cual fue el objetivo del mismo, figura 4.1.

Esta primera lámina es la base de todas las pantallas, ya que con base en ésta se diseñaron todas las siguientes, utilizando la programación orientada a objetos, en donde el objeto se definió con características especiales que se fueron utilizando en todas las láminas sólo con algunos detalles que las hacen diferentes. Por ejemplo, en una primera versión se tenía un menú en la parte superior el cual se encontraba en todas las láminas del tutorial y desde éste se podía acceder a cualquier parte del mismo, con lo cual se podía regresar a consultar nuevamente una sección ya estudiada, tal vez para repasar o comprobar resultados que se exponen en láminas posteriores o para estudiar el tutorial por secciones, éste menú se cambio por el de la parte izquierda para tener una mejor visión de en que parte del tutorial se encuentra el usuario.

Como característica común en todas las pantallas se encuentra el escudo de la Universidad Nacional Autónoma de México; los botones para continuar secuencialmente el tutorial y finalmente el cuadro de texto que se ocupa en todas las diapositivas para identificar a las diferentes láminas que componen el programa y explicar las diversas características de las Curvas Elípticas.

En el anexo A.2 de la presente tesis se muestra el código fuente de cada una de las láminas y el módulo sobre el que se programaron las funciones básicas de la Criptografía de Curvas Elípticas.

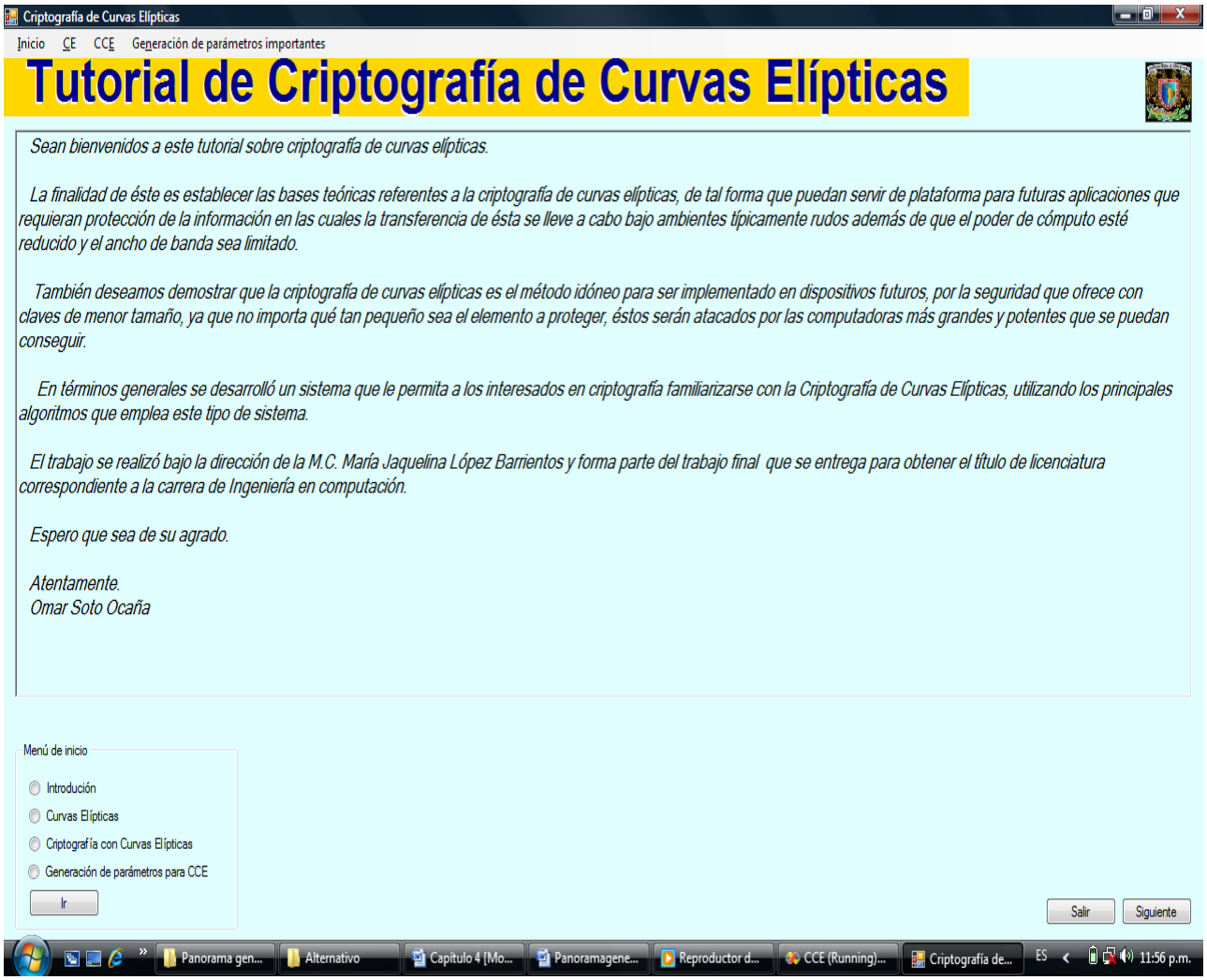


Figura 4.1 Presentación del tutorial de Criptografía de Curvas Elípticas.

En la segunda lámina se hace una introducción a las curvas elípticas figura 4.2, en ella también se menciona a las dos ramas que componen a la criptología, la primera la criptografía (encargada de ocultar información con el objetivo de protegerla cundo es transferida a través de un canal inseguro) y la segunda criptoanálisis (encargada de descubrir la información oculta sin ser el usuario autorizado para acceder a ella). También se hace referencia a Neal Koblitz y Victor Miller que propusieron de forma independiente el uso de Curvas Elípticas en Criptografía, por último se mencionan la ventajas de utilizar algoritmos basados en Curvas Elípticas para Criptografía[1].

[1] Para mayor información acerca de la Criptología revisar el capítulo 1(Estado del arte de la Criptología) del presente trabajo.

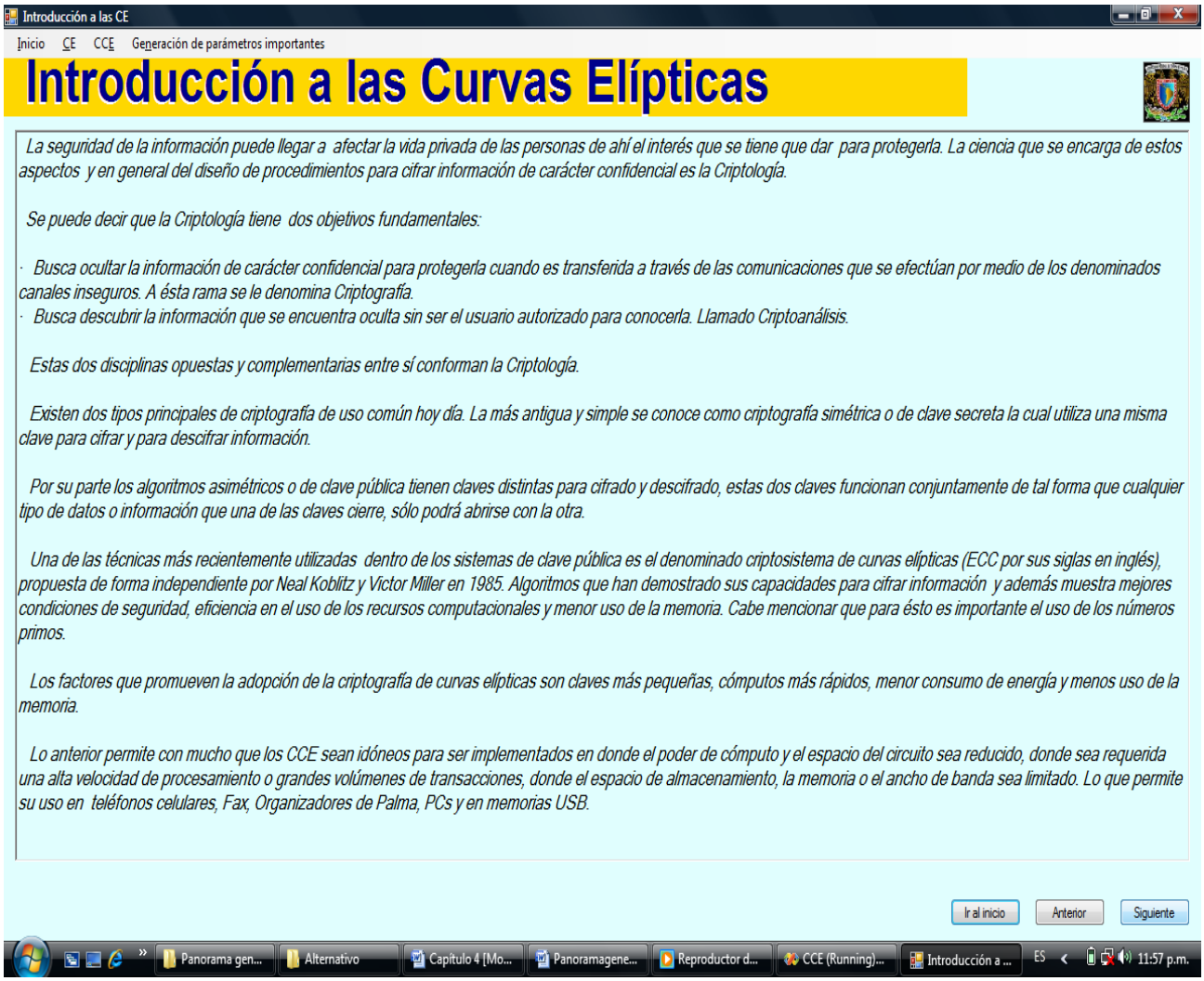


Figura 4.2 Introducción a las Curvas Elípticas

En la tercera lámina se puede realizar un bosquejo de diversas Curvas Elípticas modificando los parámetros “a” y “b” de la ecuación que define a la misma, figura 4.3 y se menciona si dichas curva son útiles para ser emplearse en Criptografía de Curvas Elípticas.

Para hacer el bosquejo de las Curvas Elípticas se introdujo una caja de dibujo con coordenadas en el eje de las abscisas “x” y el eje de las ordenadas, “y” con un fondo verde y el trazo de la curva con color rojo para hacer notar la zona de la Curva Elíptica. Los botones numéricos son importantes ya que con éstos se modifican los parámetros “a” y “b” de la curva a graficar; los botones pueden tomar valores que van de -999 hasta +999, esto con el fin de mostrar como se van modificando las curvas al usar diferentes parámetros para “a” y “b”, además se limito a estos ya que valores mayores o menores serán imprácticos para la visualización por parte del usuario.

El botón para limpiar refresca la caja de dibujo, lo cual sirve para poder graficar una nueva curva borrando la curva graficada anteriormente, esto sólo en caso de que

el usuario lo crea conveniente ya que se puede presentar el caso de que se desee comparar dos curvas elípticas de diferentes parámetros.

En el proceso se ocupan librerías para gráficos en dos dimensiones y librerías para efectuar operaciones matemáticas.

En esta lámina también se menciona si la curva es supersingular, es decir si la ecuación $4a^3 + 27b^2 = 0$, ya que si es así la curva graficada se descarta para utilizarse en criptografía ya que se efectuaría operaciones sobre el punto en el infinito, lo cual como se menciono en el capítulo 2 no es deseable.

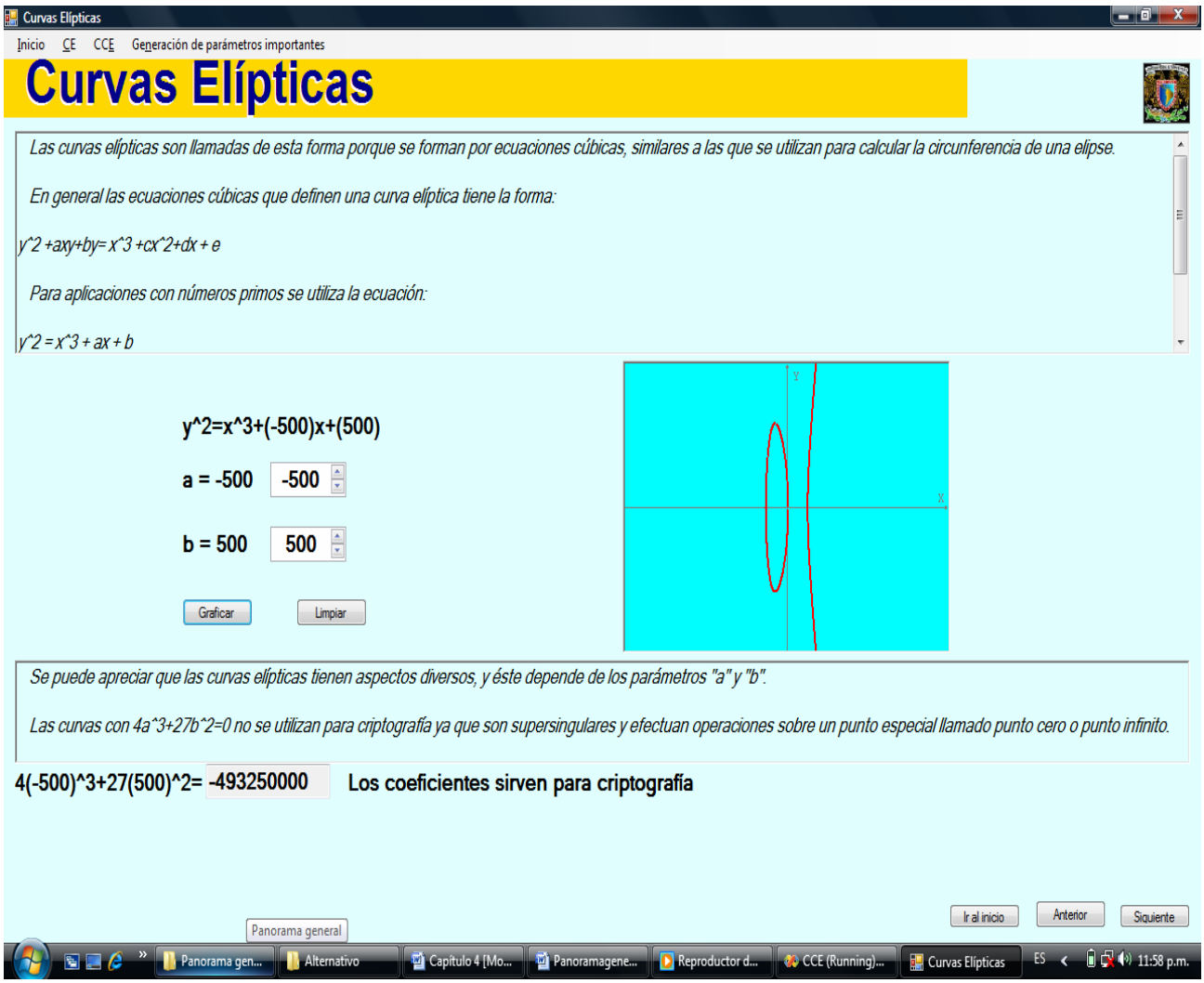


Figura 4.3 Gráfica de una Curva Elíptica cambiando sus parámetros “a” y “b”.

En la lámina de adición Geométrica en Curvas Elípticas se hace referencia a las 5 reglas necesarias para sumar dos puntos pertenecientes a una Curva Elíptica figura

4.4, además se hace uso de imágenes para ver un ejemplo de cada una de estas reglas[2].

En esta lámina se utilizó una caja de dibujo y un botón numérico para mostrar las imágenes que ejemplifican cada una de las cinco reglas de la adición geométrica para curvas elípticas. Aquí fueron indispensables las librerías para importar imágenes y poder manipular la caja de dibujo.

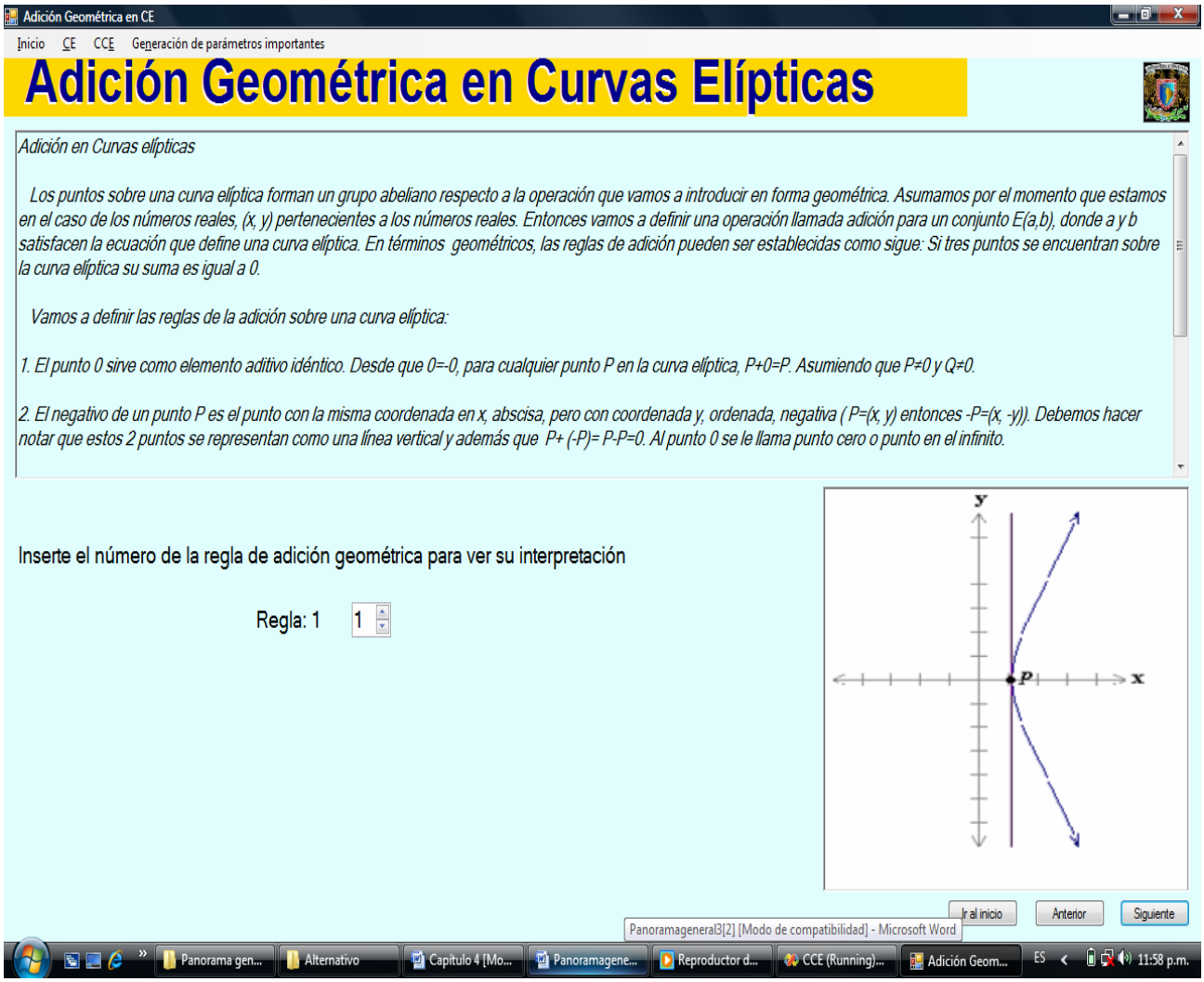


Figura 4.4 Adición Geométrica en Curvas Elípticas

Resultó trascendental el mencionar la importancia de los parámetros P, Q y R en las Curvas Elípticas, figura 4.5, ya que éstos funcionan como claves privadas (puntos P y Q de la Curva Elíptica) y clave pública (punto R de la Curva, suma de los puntos P+Q). Es una lámina que al principio no se había contemplado sin embargo al revisar el primer prototipo se descubrió que era importante el mencionar

[2] Las reglas para la suma geométrica en curvas elípticas se describen con mayor detalle en el capítulo 2 (Fundamentos matemáticos) del presente trabajo.

que gracias a estos puntos se le puede considera a la Criptografía de Curvas Elípticas dentro de la criptografía de tipo asimétrica.

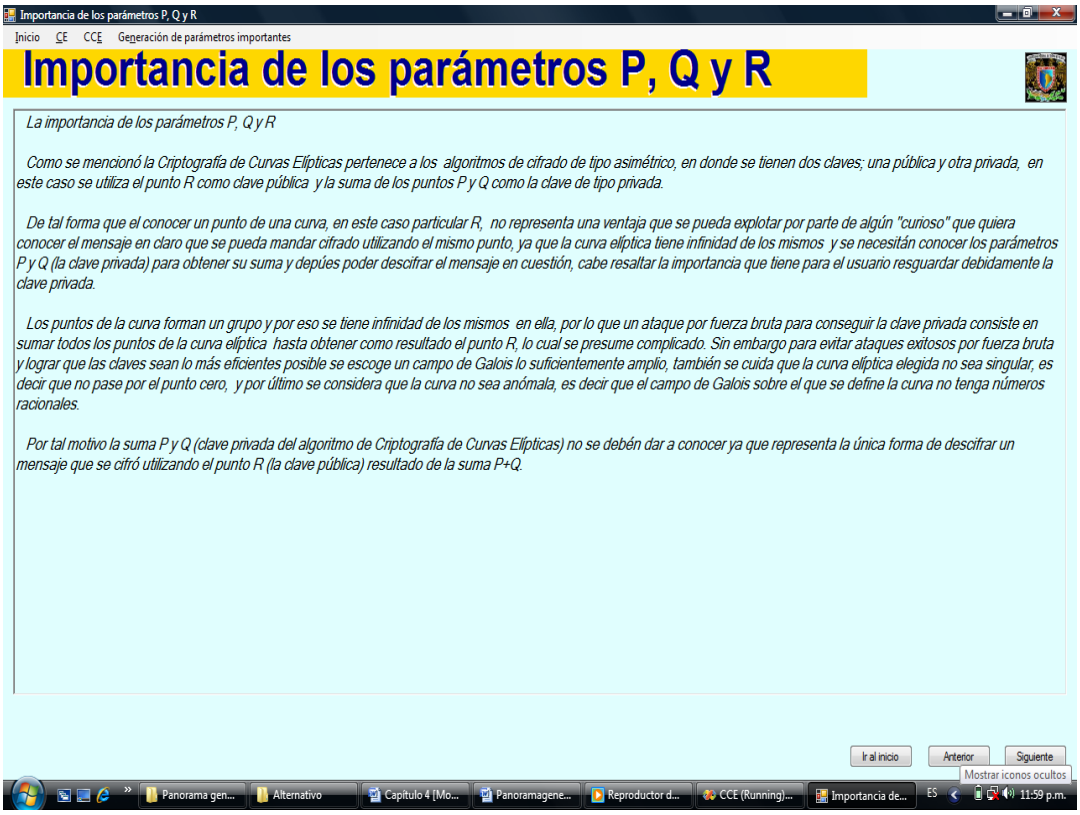


Figura 4.5 Importancia de los parámetros P, Q y R

Se mencionan las ecuaciones para obtener la suma de dos puntos de una forma algebraica, figura 4.6, lo cual es útil para ser implementado en los sistemas de cómputo ya que de esta forma se procesa los datos de forma eficiente, cosa que no sucede con la suma geométrica, por lo que la lámina utiliza librerías para realizar operaciones matemáticas muy sencillas, porque se trabaja en ésta parte con los números reales los cuales, por diversas características entre las que se incluye que no son un campo de Galois, no sirven para criptografía; sin embargo se expone en esta lámina para facilitar la comprensión de material más complejo que se presenta más adelante dentro del tutorial.

En la lámina se presenta un ejercicio de cómo obtener las coordenadas de un punto que pertenezca a una curva elíptica dentro del campo de los números reales y el usuario puede verificar si el resultado que obtuvo es correcto o necesita revisar nuevamente la lámina.

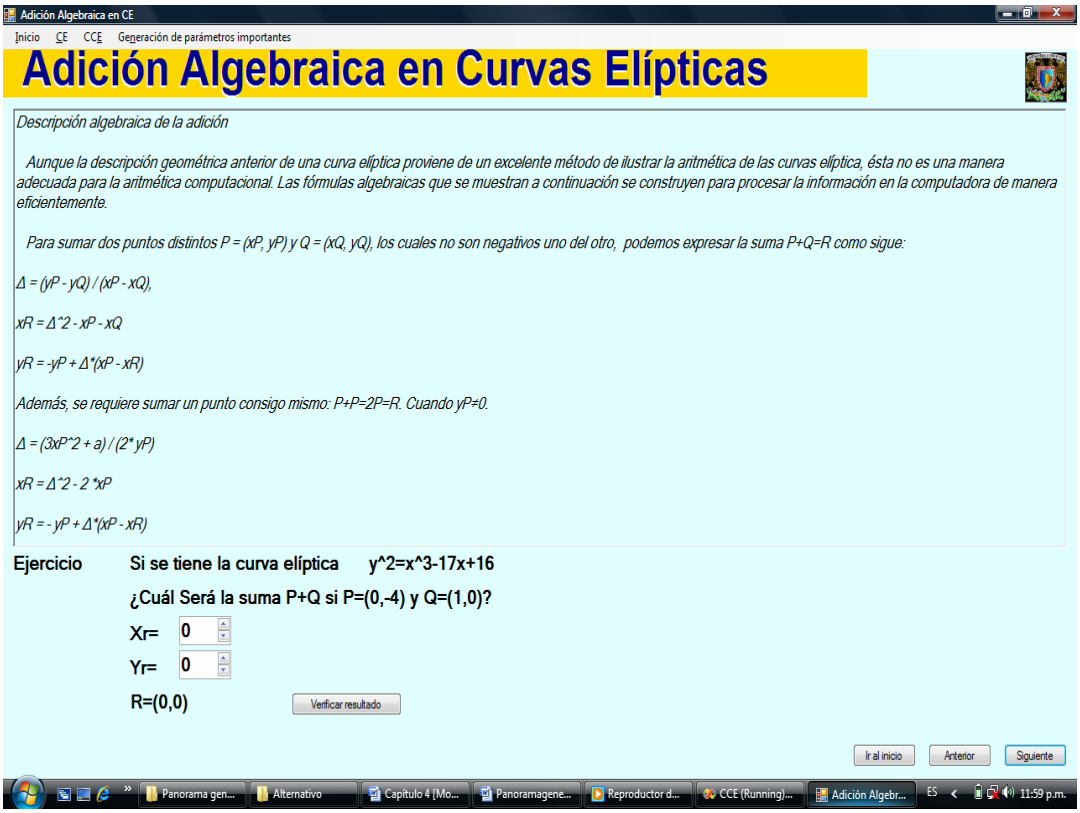


Figura 4.6 Adición Algebraica en Curvas Elípticas

En la lámina de Curvas Elípticas sobre los números primos figura 4.7 se trabaja con los campos que se utilizan en criptografía ya que se sabe de la importancia de los números primos en criptografía, también se hace referencia a la suma extendida de Euclides para encontrar el inverso de un número en un campo de Galois o campo finito determinado.

En la lámina hace un llamado a la función del algoritmo de Euclides para mostrar esta función de manera particular, en caso de que el usuario deseará desarrollar paso por paso los ejercicios propuestos más adelante se tendrá que recurrir a esta lámina para encontrar las coordenadas de algún punto deseado y se necesitaría el número inverso del campo en el que se trabaja.

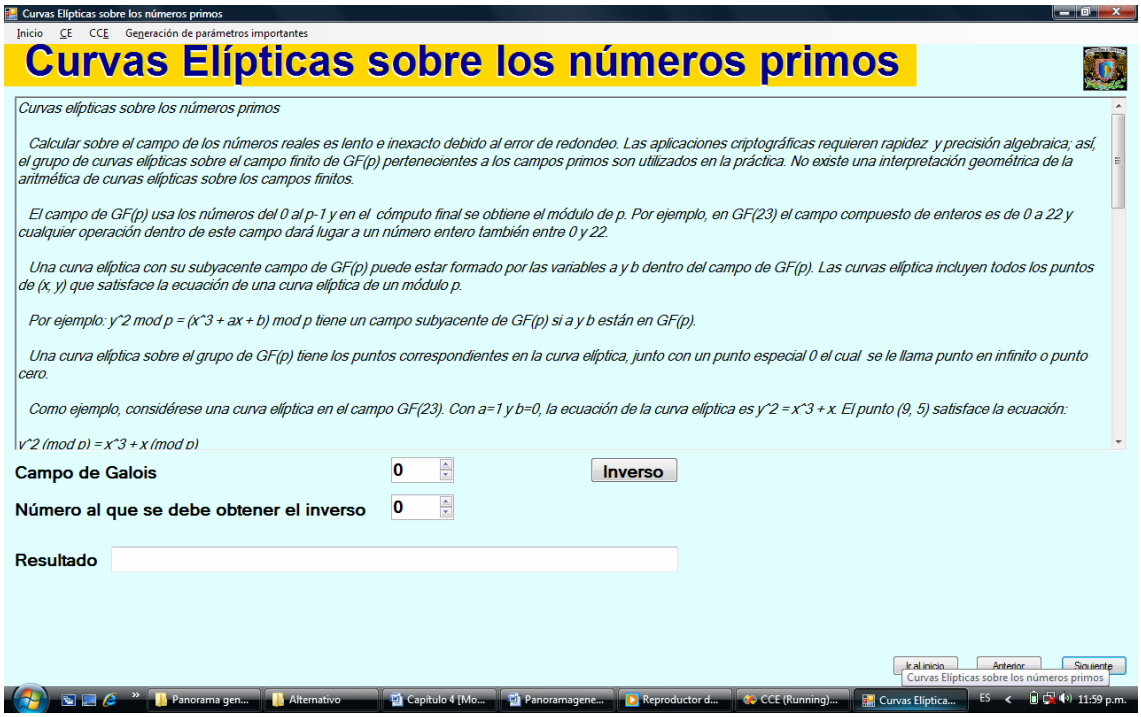


Figura 4.7 Curvas Elípticas sobre los números primos

En la lámina de “Curvas Elípticas y el Problema del Logaritmo Discreto” figura 4.8 se empieza a trabajar con las Curvas Elípticas sobre un campo de Galois y los puntos que la forman que también se encuentran definidos en el mismo, se plantea un ejemplo en el que se puede multiplicar un punto de una curva por un número entero.

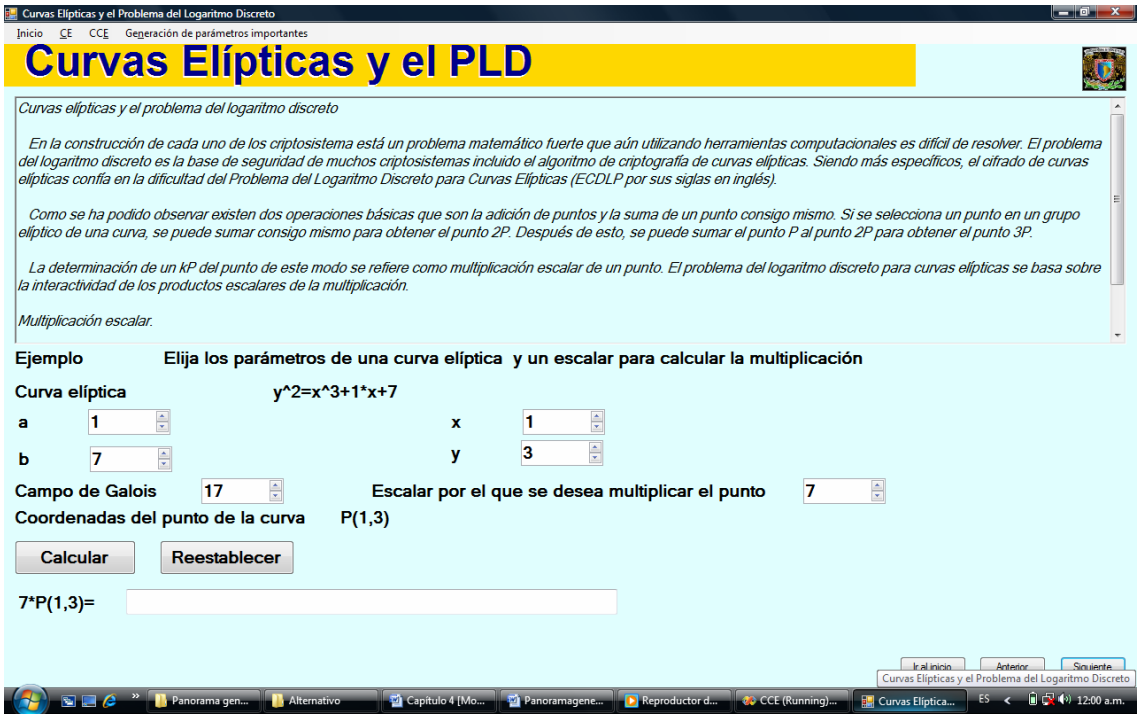


Figura 4.8 Curvas Elípticas y el Problema del Logaritmo Discreto

Posteriormente en la lámina de “Importancia de la multiplicación en curvas elípticas”, figura 4.9, de la misma forma que en la lámina “Importancia de los parámetros P, Q y R” se añadió al primer prototipo ya que se considera importante el mencionar como en algunos sistemas de Criptografía de Curvas Elípticas, como ElGamal para curvas elípticas, el número entero aleatorio por el que se multiplica el punto generador hace la función de clave privada y el punto de la curva obtenido de esta multiplicación es la clave pública.

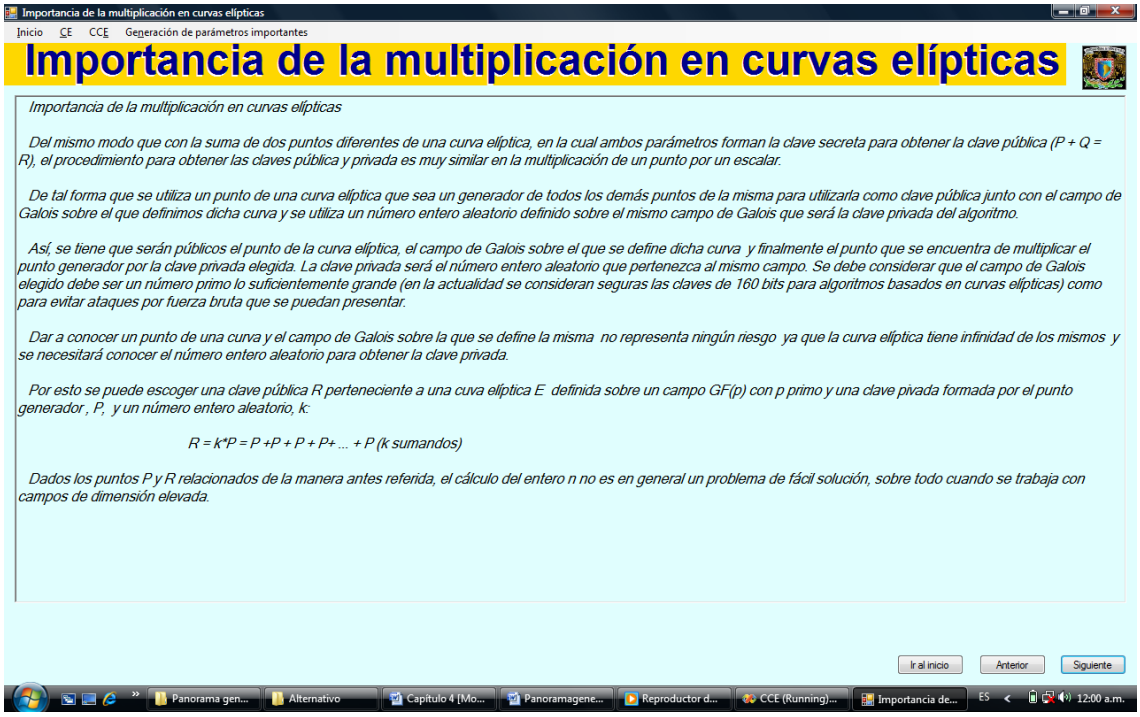


Figura 4.9 Importancia de la multiplicación en curvas elípticas

En la lámina de “Criptografía de Curvas Elípticas” figura 4.10 se hace referencia al Problema del Logaritmo Discreto para Curvas Elípticas en el cual basa su seguridad la Criptografía con Curvas Elípticas, y se sabe como se menciona en el capítulo 3 que el problema del logaritmo discreto para curvas elípticas es un problema que es más difícil de resolver que el problema del logaritmo discreto, con lo cual se tiene que con claves de menor tamaño se puede ofrecer el mismo nivel de seguridad que con otros algoritmos asimétricos que tengan claves mayores a las utilizadas por CCE, también se da un ejemplo de cómo se podría llevar a cabo un ataque por fuerza bruta y porque para evitar éstos se debe escoger números de 160 bits los cuales se consideran seguros en la actualidad.

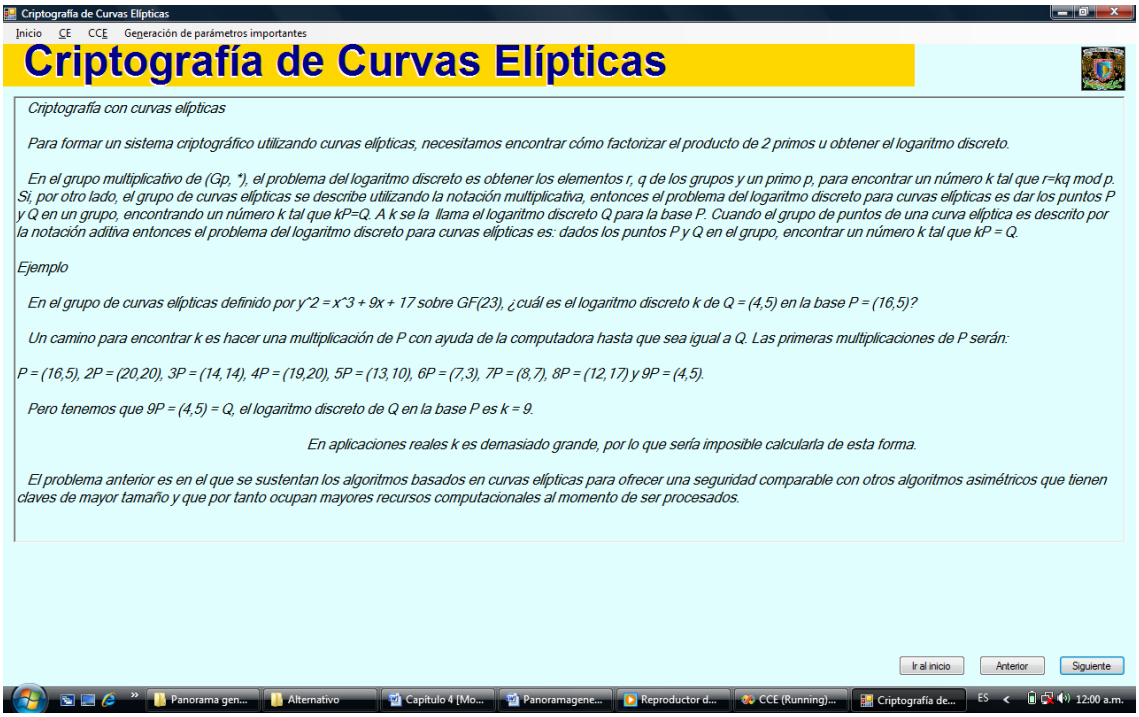


Figura 4.10 Criptografía de Curvas Elípticas

En la lámina de “Obtención de puntos en Curvas Elípticas”, figura 4.11, se maneja la programación utilizada en la figura 4.8 para obtener un punto multiplicado por un escalar y se explica como poder hacer más eficiente el cómputo de la multiplicación para números enteros mayores.

En el ejercicio de la lámina se pueden modificar los valores de los parámetros “a” y “b” de la curva elíptica, el campo sobre el que se definen, GF(p) con p primo o la potencia de un primo, los parámetros “x” y “y” del punto generador que pertenece a la curva elíptica y el escalar por el que se desea multiplicar el punto generador de la curva elíptica.

En ocasiones no es sencillo conocer si un punto pertenece a la curva propuesta o no, por tal motivo en caso de modificar los valores propuestos y no encontrar un punto que pertenezca a la curva propuesta se tiene la opción de restablecer los valores propuestos.

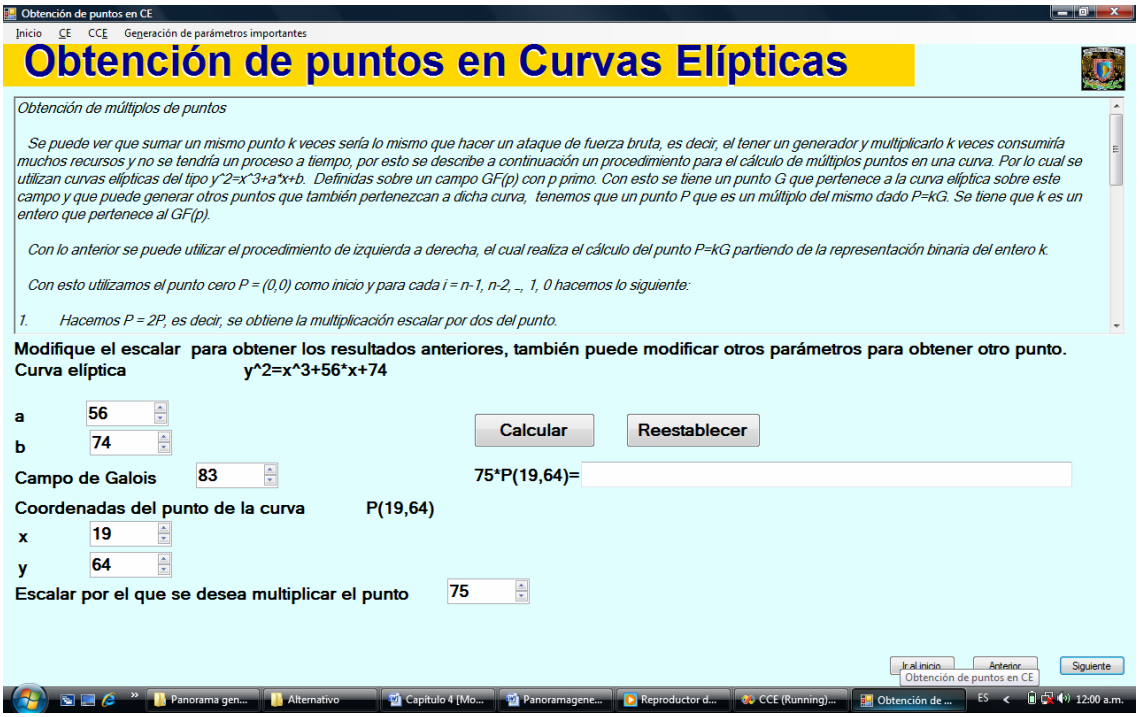


Figura 4.11 Obtención de puntos en Curvas Elípticas

En la lámina de “Intercambio de claves en Curvas Elípticas” figura 4.12 se hace una analogía con el intercambio de claves de Diffie-Hellman esta vez para Curvas Elípticas, aquí se los usuarios hacen uso de un punto de una curva para efectuar las operaciones de multiplicación por un escalar visto en la figura 4.11.

En la lámina se presenta un ejemplo en el cual dos usuarios virtuales hacen uso del intercambio de claves secretas para curvas elípticas, para esto necesitan ponerse de acuerdo en la curva elíptica a utilizar y el punto generador que se ocupa para el proceso, en el ejercicio se propone una curva y un punto generador, pero se puede ocupar cualquier curva elíptica y cualquier punto que pertenezca a dicha curva. De igual forma se proponen los números que serán la clave secreta de cada uno de los usuarios, pero se pueden utilizar los que se crean convenientes, cabe señalar que se recomienda que los números que se utilizan para claves secretas de los usuarios deben ser primos o primos relativos con respecto al campo finito sobre el que están definidos la curva elíptica y el punto generador, esto es, que su máximo común divisor[3] entre el campo finito y el número propuesto debe ser igual a uno.

En la lámina también se presenta un botón para restablecer los valores predeterminados.

[3] El algoritmo para encontrar el máximo común divisor se encuentra en el modulo1 del anexo 2 del presente trabajo.

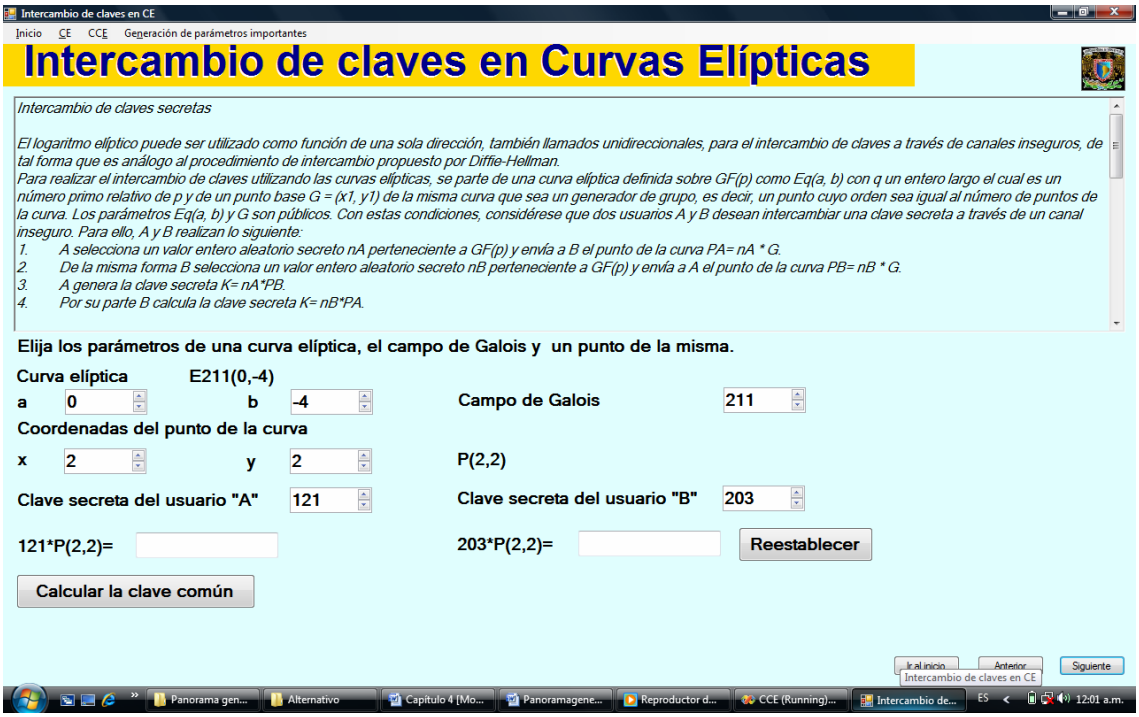


Figura 4.12 Intercambio de clave en Curvas Elípticas

La lámina de "Codificación en Curvas Elípticas", figura 4.13, es de gran relevancia ya que es aquí donde se estudia como asignar cada una de las letras del alfabeto a un punto de una Curva Elíptica, los parámetros que son necesarios para este proceso, y además se da un ejemplo interactivo para asignar a un punto de una curva elegida por el usuario a un carácter y de esta forma trabajar posteriormente con todos los algoritmos utilizados en Curvas Elípticas.

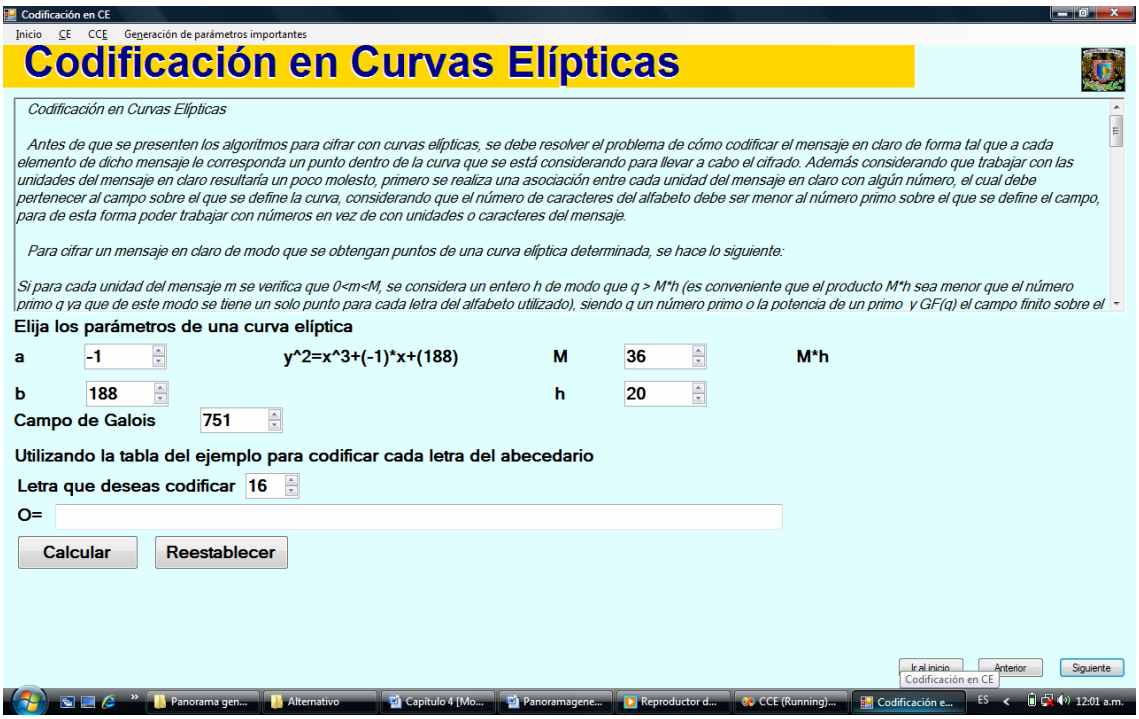


Figura 4.13 Codificación en Curvas Elípticas

En las figuras 4.14 y 4.15 se presentan las láminas del cifrado de tipo ElGamal pero para su versión en Curvas Elípticas, aquí se plantea un ejercicio de cómo cifrar información con un punto que previamente fue codificado para poder utilizar este tipo de algoritmo para cifrar la información codificada.

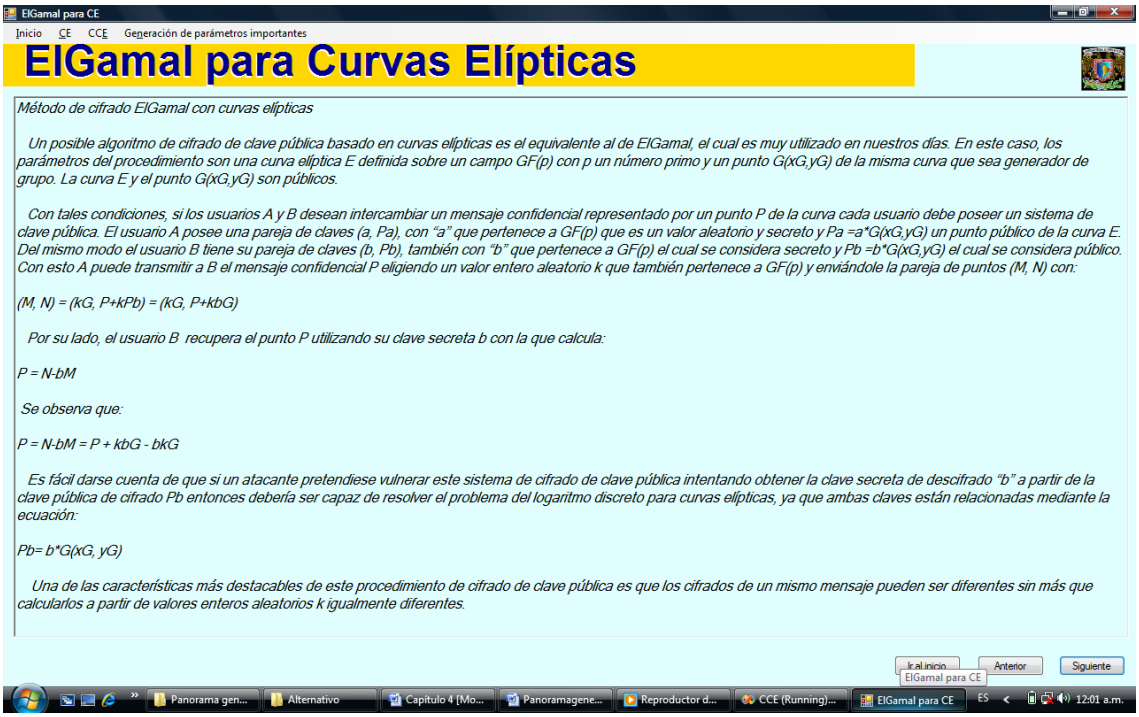


Figura 4.14 ElGamal para Curvas Elípticas

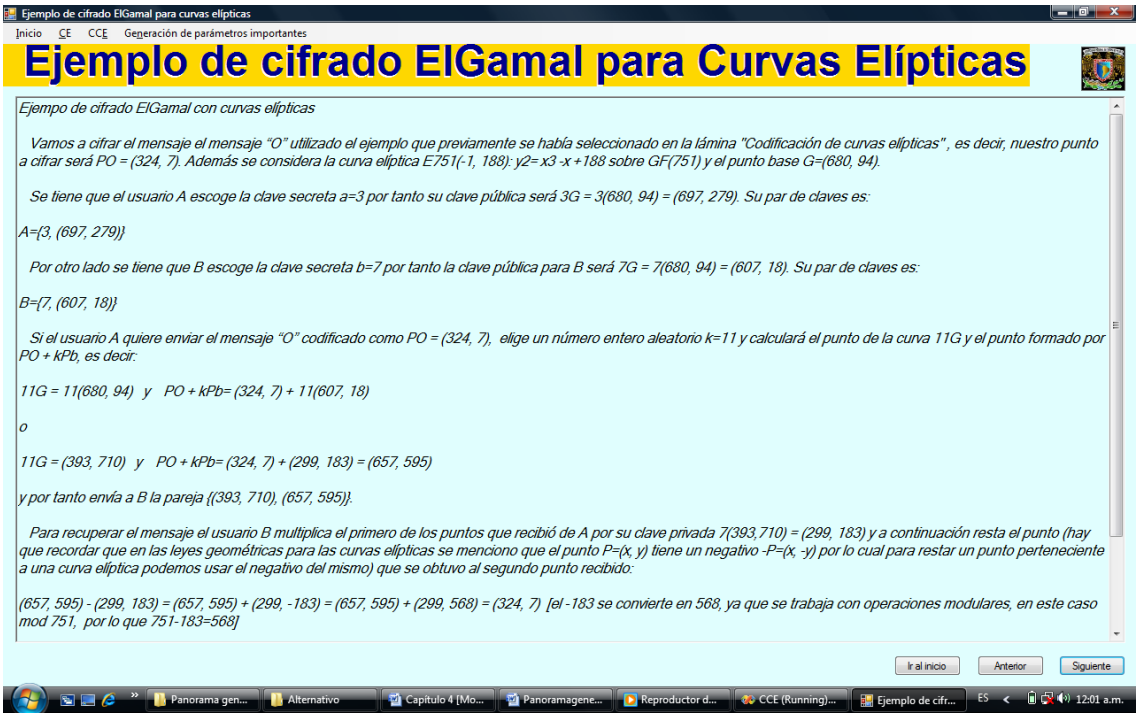


Figura 4.15 Ejemplo de cifrado ElGamal para Curvas Elípticas

En la lámina “Generación de una curva elíptica aleatoria”, figura 4.16, se trabaja con número primos por lo cual es necesario verificar que el número utilizado pertenezca al campo de dichos números. En la lámina se tiene la posibilidad de verificar lo anterior, ya que es importante trabajar sobre este campo en criptografía, además en el ejercicio se puede generar una curva elíptica aleatoria que pertenezca a dicho campo.

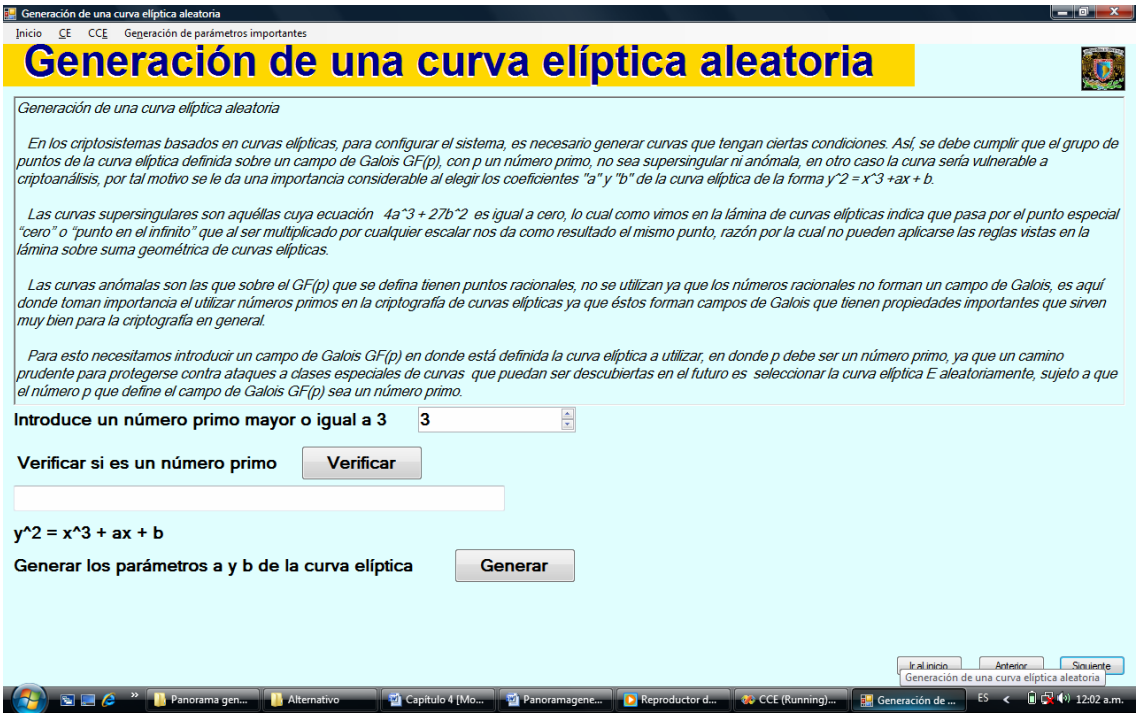


Figura 4.16 Generación de una curva elíptica aleatoria

En la lámina de “Generación de parámetros para Curvas Elípticas” figura 4.17 se utilizan todos los algoritmos de Criptografía de Curvas Elípticas y se presentan los parámetros que son considerados importantes, que incluye el número primo sobre el que se define las curvas elípticas y los puntos de la misma, la representación de un campo finito, los parámetros “a” y “b” de la curva elíptica, las coordenadas del punto generador y el orden de la curva, también se genera una curva elíptica aleatoria a partir de un número primo.

Esta es una lámina importante que como ya se menciono utiliza todos los algoritmos de curvas elípticas y representa un resumen de todo lo visto a lo largo del tutorial.

En el ejercicio solamente se pide introducir un número primo para generar los parámetros ya mencionados y en caso de que el número que se introduzca no pertenezca al campo de los números primos se manda un mensaje avisando que el número no es primo.

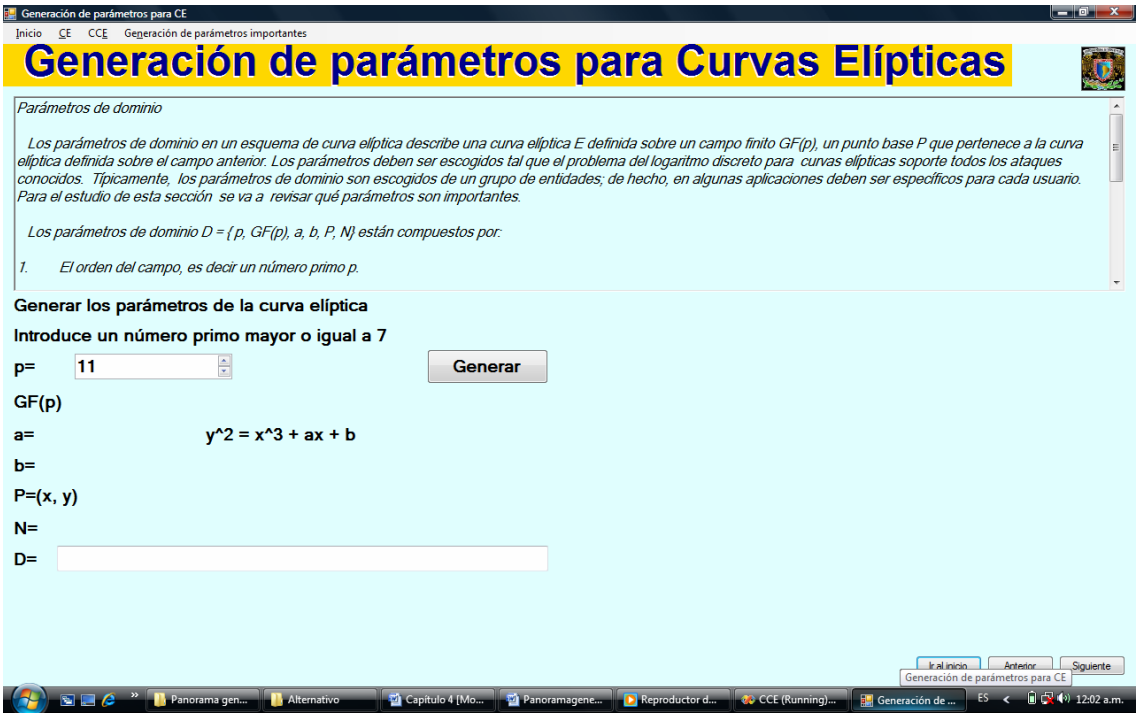


Figura 4.17 Generación de parámetros para Curvas Elípticas

En la lámina “Graficación de puntos en una Curvas Elípticas”, figura 4.18, presentan los parámetros importantes que se mencionaron en la lámina de “Generación de parámetros importantes” ya que se vuelven a ocupar para mostrar una característica importante de las Curvas Elípticas útiles en criptografía. Esta característica es la simetría que se presentan en los puntos que forman una curva elíptica paralela al eje de las abscisas y que toma la ordenada al origen igual a la mitad del número primo elegido y el índice de la curva es decir el número de los puntos que pertenecen a la misma.

Del mismo modo como se generaron los parámetros importantes en la lámina sobre “Generación de parámetros importantes” ya mencionada en la figura 4.17, aquí solamente se requiere de un número primo para obtener la representación del campo de Galois utilizado, los coeficientes “a” y “b” de la curva elíptica, el punto generador y el índice de la misma.

En la lámina se tiene la opción de regresar al inicio del tutorial o salir del mismo con botones para “Ir al inicio” y “Salir”, con lo que se puede o repasar algún tema que hubiese quedado claro o salir para aplicar los principios básicos sobre la Criptografía de Curvas Elípticas.

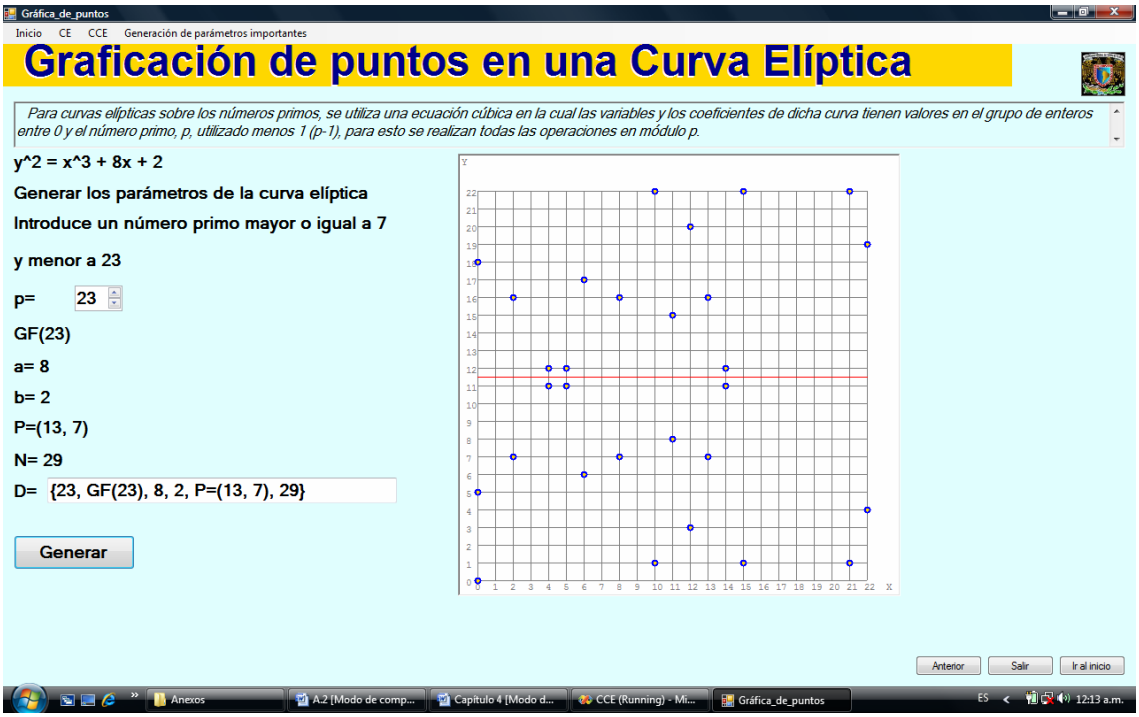


Figura 4.18 Graficación de puntos en una Curva Elíptica

4.3 Pruebas y liberación

Una vez que se genero el código fuente y las láminas que interactúan con los usuarios del tutorial de Criptografía con Curvas Elípticas se deben probar para descubrir y corregir el máximo número de errores posibles antes de su liberación final. Para esto es necesario hacer uso de las técnicas de prueba de software para que se evalué la lógica interna del programa (a este tipo de pruebas se les conoce como pruebas de caja blanca) y se verifique los componentes de entrada y salida del software para descubrir errores en la funcionalidad, el comportamiento y rendimiento (conocidas como pruebas de caja negra). El objetivo de las pruebas es descubrir algún error que se encuentre en el tutorial que hasta este momento no se había detectado.

4.3.1 Prueba de caja blanca

A la prueba de caja blanca, también conocida como prueba de caja de cristal, es un método de diseño de casos de prueba que comprueba los caminos lógicos del software proponiendo casos de prueba que ejercitan conjuntos específicos de condiciones o ciclos.

La prueba de condición ejercita las condiciones lógicas contenidas en el módulo principal del tutorial. Los tipos posibles de componentes en una condición pueden ser: un operador lógico, una variable lógica, un par de paréntesis lógicos, un operador relacional o una expresión aritmética.

Si una condición es incorrecta, entonces es incorrecto al menos un componente de la condición. Así, los tipos de errores de una condición pueden ser los siguientes:

- Error en operador lógico (existencia de operadores lógicos incorrectos, que sobren o que hayan desaparecido).
- Error en variable lógica.
- Error en paréntesis lógico
- Error en operador relacional.
- Error en expresión aritmética.

En términos generales el propósito de aplicar la prueba de condiciones al tutorial de Curvas Elípticas es detectar, no sólo los errores en las condiciones que se presentan en la sintaxis del mismo, sino también detectar otros errores que se pueden arrastrar a lo largo del tutorial.

En la prueba de flujo de datos se seleccionan caminos de prueba en el tutorial de acuerdo con la ubicación de las diferentes secciones que lo componen y los usos de las variables que son utilizadas dentro del mismo tutorial. Esta es una prueba importante ya que como se menciono en el desarrollo del sistema se puede acceder a diversas partes del mismo por diversos caminos gracias al menú que se tiene en todas las láminas.

Los ciclos son muy importantes dentro del tutorial ya que muchos de los algoritmos utilizados requieren de éstos para poder funcionar y en general son indispensables en la inmensa mayoría de los algoritmos que se implementan en cualquier programa. Por esto se les debe prestar la debida atención al momento de realizar las pruebas.

La prueba de los ciclos es una técnica que se centra exclusivamente en la validez de las construcciones de diferentes ciclos. En el caso del tutorial se utilizaron ciclos simples a los que se les aplico el siguiente conjunto de pruebas:

- Se dejo pasar por alto el ciclo.
- Se paso una vez por cada ciclo.
- Se paso dos veces por el ciclo.
- Se paso un número menor de veces por el ciclo que el número máximo permitido por el ciclo.
- Se paso una vez más por el ciclo que el número máximo permitido por el ciclo.

Dentro del tutorial muchas veces se hace uso de ciclos anidados en el que se probó primero el ciclo más interno, estableciendo los demás a sus valores mínimos y posteriormente se progreso hacia afuera con los demás ciclos. Muchos de los ciclos anidados hacen uso de ciclos simples por lo que esto facilito su revisión.

Se debe tomar en cuenta uno de los principios de las pruebas, el cual establece que no son posibles las pruebas exhaustivas, ya que el número de permutaciones de caminos para incluso un programa de tamaño moderado como el tutorial de Curvas Elípticas es excepcionalmente grande.

Por tal motivo resulta imposible recorrer todas las combinaciones de caminos durante las pruebas. Pero fue posible cubrir adecuadamente la lógica del programa y asegurarse de que se satisfacen todas las condiciones en el diseño de todos los componentes.

4.3.2 Prueba de caja negra

Las pruebas de caja negra, también llamados prueba de comportamiento, se encargan de los requisitos funcionales de los programas, es decir permite obtener conjuntos de condiciones de entrada que ejerciten completamente los requisitos funcionales del sistema.

La prueba de caja negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.

- Errores de interfaz.
- Errores en la estructura de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de iniciación y de terminación.

A diferencia de la prueba de caja blanca, la cual se lleva a cabo previamente en el proceso de prueba, la prueba de caja negra se aplica durante fases posteriores, ya que la prueba de caja negra ignora intencionalmente la prueba de control, centra su atención en el campo de la información.

Para hacer más eficaces las pruebas de caja negra, es decir que las pruebas tengan una mayor probabilidad de encontrar errores, éstas deben ser realizadas por personas independientes a las que desarrollaron el sistema a probar. Por tal motivo las personas que participamos en el desarrollo del sistema de aprendizaje de Curvas Elípticas no somos las más adecuadas para llevar a cabo éste tipo de prueba.

Por lo anterior se realizó una prueba con los alumnos que cursan la materia de Criptografía en el grupo 1 del semestre 2008-1 con las siguientes observaciones:

Algunos usuarios necesitaron instalar framework 2.0 en sus computadoras para poder correr la aplicación sin problemas, lo cual resulta muy útil para una aplicación que se utiliza como apoyo en el tema de Criptografía de Curvas Elípticas visto en clase.

En términos generales la aplicación resulto atractiva para los usuarios, ya que cumplió con el objetivo de ser una herramienta que les permitiera repasar el tema visto en clase y resolver problemas relacionados con criptografía de curvas elípticas.

Con base en los comentarios realizados por los usuarios se modifico el tipo de letra utilizada en el tutorial, la organización de gráficos, esquemas, tablas y llaves con el fin de hacerlo más atractivo para los alumnos.

Se corrigieron algunas etiquetas que no se actualizaban y algunos colores para resaltar los resultados.

Los títulos y el menú del lado izquierdo se modificaron para tener una organización y presentación más agradable a los alumnos, el escudo de la universidad también se modifico para hacerlo más atractivo a la vista.

El icono de la aplicación y el de la interfaz se modifico para tener una imagen representativa de lo que contenía el programa, utilizando para esto la curva elíptica en color rojo que se aprecia en la aplicación.

En general el rendimiento del tutorial al momento de su aplicación fue adecuado y no se necesito ninguna modificación adicional.

Las pruebas se realizaron el día lunes 3 de diciembre de 2007 y sirvió de apoyo para realizar un examen referente a la criptografía de curvas elípticas por lo que resulto de gran ayuda para los usuarios.

Después de las pruebas el diseño final del "Tutorial de Criptografía de Curvas Elípticas" quedo como muestra la figura 4.19. Lo anterior para hacer más atractivo el diseño para los usuarios.

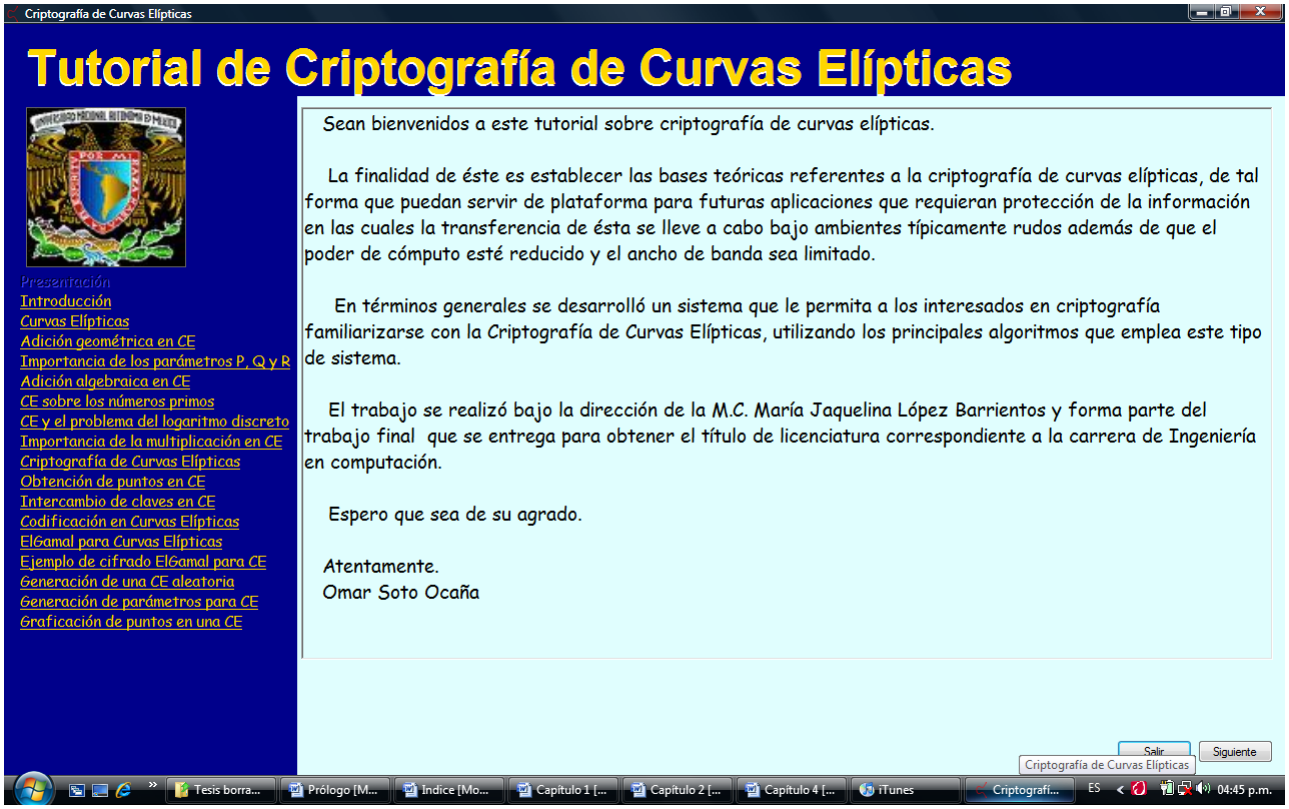


Figura 4.19 Presentación final del Tutorial de Criptografía de Curvas Elípticas

También el icono para acceder al tutorial quedo como muestra la figura 4.20.

Figura 4.20 Icono para acceder al Tutorial de Criptografía de Curvas Elípticas

Conclusiones

Con base en el desarrollo del presente trabajo de tesis se llegó a las siguientes conclusiones:

Se plantearon las bases teóricas utilizadas en la criptografía de curvas elípticas, las cuales abarcan desde el estudio de los campos finitos, pasando por la interpretación geométrica y algebraica de las curvas elípticas hasta el análisis y estudio de la criptografía de curvas elípticas y de otros sistemas de cifrado de tipo asimétrico. Logrando con esto tener una visión detallada de la forma de trabajar de los algoritmos utilizados y de las bases matemáticas que las fundamentan.

Se realizó una comparación entre los sistemas de cifrado RSA y la Criptografía de Curvas Elípticas a nivel de seguridad, eficiencia, espacio y requerimientos dando como resultado diferencias significativas muy favorables para la criptografía de curvas elípticas, esto se debe a que la longitud de las claves de CCE ofrecen el mismo nivel de seguridad que RSA pero con claves de menor tamaño. Lo anterior se debe al problema del logaritmo discreto para curvas elípticas es una función más segura (necesita mayor procesamiento) que el logaritmo discreto, en el que se basa el algoritmo RSA. En términos generales se encontró que las claves de CCE son hasta 6 veces menores que las utilizadas por RSA. Se tiene que ambos sistemas tienen similares requerimientos en el ancho de banda cuando se utilizan para cifrar mensajes largos, pero la situación cambia cuando se desea cifrar un mensaje menor. Pero se tiene que los sistemas de cifrado asimétrico son usualmente empleados para transmitir mensajes cortos como por ejemplo para transmitir la clave de un sistema de cifrado de tipo simétrico. Es por esto que la CCE presenta ventajas también en el ancho de banda utilizado.

Por todo lo anterior se puede decir que los algoritmos basados en Criptografía de Curvas Elípticas resultan ideales para utilizarse en dispositivos diminutos y en donde al ancho de banda es limitado.

Se revisaron algunas aplicaciones que actualmente utilizan la Criptografía de Curvas Elípticas como su sistema de cifrado, tales como firmas digitales, marcas postales de tipo digital, comprobación de compra con cheques, seguridad en smart cards y en general el mejoramiento en la seguridad de las comunicaciones. Y se proponen aplicaciones para encargarse de la seguridad en dispositivos diminutos tales como los chips de radiofrecuencia para el Registro Público Vehicular (REPUVE).

No obstante las ventajas que presenta la Criptografía basada en Curvas Elípticas, el algoritmo de cifrado RSA es mejor aceptado mundialmente y domina el campo de la criptografía asimétrica. Es por esto que se desarrollo un tutorial de Criptografía de Curvas Elípticas, con la finalidad de difundir la CCE y de comprender como operan.

El tutorial permitió a los alumnos inscritos en la materia de Criptografía de la Facultad de Ingeniería en el semestre 2008-1, familiarizarse con la Criptografía de Curvas Elípticas y encontrar material de apoyo que sirvió como complemento a lo visto en su clase.

Se apoyó en la exposición de las clases de criptografía con respecto al tema de curvas elípticas, con resultados favorables en el aprendizaje de los alumnos. Por tener contacto con el temario de la clase de Criptografía se propone una reestructuración del temario de tal forma que pueda abarcar los temas de Criptografía de Curvas Elípticas más representativos. Esta propuesta se encuentra en el anexo 6 del presente trabajo.

Además se espera que en un futuro el tutorial se pueda correr en otras plataformas como Linux, por lo que tal vez sea necesario migrar al lenguaje de programación Java.

El tutorial de Criptografía de Curvas Elípticas es funcional y resultará de gran ayuda para las generaciones que cursen la materia de Criptografía, ya que se considera un tema actual dentro del mundo de la seguridad por ser una de las técnicas más recientes en ser utilizadas para cifrar información, se vuelve importante la existencia de trabajos de este tipo y la actualización constante de la Facultad de Ingeniería en sus planes de estudio en las asignaturas correspondientes al módulo de redes y seguridad para que de este modo se pueda contar con los ingenieros en computación mejor preparados.

Anexo 1

Glosario

Activo. Un recurso necesario para que una organización funcione correctamente y alcance los objetivos propuestos por su dirección.

AES (Advanced Encrytion Standard). Es un esquema de cifrado por bloques y uno de los algoritmos más populares usados en criptografía simétrica.

Anillo. Un anillo es una colección de elementos con dos operaciones binarias, llamadas adición y multiplicación, denotada por {R, +, *}.

Amenaza. Es todo aquello que intenta o pretende destruir un sistema.

Ataque. Es la realización de una amenaza. Para su realización se requiere que exista una motivación, la capacidad de llevarlo a cabo y que exista una oportunidad para realizarlo.

Autenticación. Proceso mediante el cual se verifica la identidad de las personas que tienen acceso a un sistema, es decir, si la persona es quien dice ser. También se determina si una persona está autorizada para llevar a cabo una acción determinada o si los datos han sido modificados.

Campo. En general un cuerpo o campo son abstracciones de sistemas numéricos y de sus propiedades esenciales. Un campo F es un anillo que además debe cumplir con la propiedad conmutativa, se denotado por {F,+,*}.

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

CCE. Criptografía de curvas elípticas, es el algoritmo asimétrico para cifrar información que ha sido recientemente utilizado dentro de sistemas con recursos limitados.

Cifrado. Es la conversión de un mensaje en claro en una forma ininteligible usando una transformación basada en una tabla o en un algoritmo.

Cifrado en bloque. Es un algoritmo de encriptación simétrico en el cual se opera en grupos de bits de longitud fija aplicándole una transformación invariante.

Cifrado simétrico. También llamado Criptografía de clave privada, es el proceso de ocultar los datos o la información en la cual el emisario y el receptor poseen la misma clave para cifrar y descifrar.

Cifrado asimétrico. También llamado Criptografía de clave pública, es el proceso de ocultar los datos o la información en la cual el emisario y el receptor posen claves distintas para cifrar y descifrar.

Clave privada. Es una de las dos llaves utilizadas en los sistemas asimétricos y la única en los sistemas simétricos. Para comunicaciones seguras el único que la debe conocer es el dueño de esa clave. También se le conoce como llave privada.

Clave pública. Es una de las dos llaves utilizadas en los sistemas asimétricos. La clave pública debe ser manejada de manera pública, ya que se utiliza en conjunto con una correspondiente clave privada. También se le conoce como llave pública.

Confidencialidad. Servicio de seguridad en el cual se puede asegurar la capacidad de que sólo las personas, sistemas o procesos autorizados puedan tener acceso a la información.

Control de acceso. Es uno de los servicios de seguridad. Es la habilidad para limitar y controlar los diferentes tipos de acceso que puedan tener los usuarios, sistemas, procesos y aplicaciones mediante los diferentes puentes de comunicación.

Criptoanálisis. Es una de las dos ramas de la Criptología, en la cual se intenta obtener el sentido de una información cifrada de manera ilícita, esto es, sin ser el usuario autorizado para obtenerla.

Criptografía. Es la ciencia que estudia los métodos y procedimientos, mediante algoritmos matemáticos, para modificar los datos de tal manera que solamente las

personas que tengan la llave adecuada puedan tener acceso a los mismos y asegurar que estos datos no fueron modificados entre el remitente y el destinatario.

Criptología. Es la ciencia encargada de las comunicaciones en forma segura y usualmente en forma secreta. Engloba dos disciplinas opuestas y a la vez complementarias estas son la Criptografía y el Criptoanálisis.

Curva elíptica anómala. Es una curva elíptica que tiene exactamente p puntos, es decir que el orden de la curva es igual al número primo sobre el cual se define. Smart, Satoh y Araki mostraron independientemente que el problema del logaritmo discreto para estas clases especiales de curvas es fácil de resolver, por lo que se descartan para ser utilizadas en criptografía.

Descriptado. Es el mensaje que es interceptado por el criptoanalista con el fin de analizar información en la cual no se está autorizado para acceder a ella y se obtiene el mensaje en claro de manera ilícita utilizando técnicas de criptoanálisis.

DES. Data Encription Standard, es un algoritmo de cifrado de clave privada en el cual se oculta la información en bloques y durante 16 rondas.

Descifrado. Es la transformación de un texto cifrado en el mensaje en claro original utilizando una tabla o un algoritmo y la clave correspondiente.

Disponibilidad. Es uno de los servicios de seguridad y se encarga de que las personas autorizadas puedan acceder a la información deseada cuando lo requieran y tantas veces como lo deseen.

Eficiencia. Se refiere a las capacidades que provee un algoritmo de cifrado en cuanto a su funcionalidad en un espacio requerido. La eficiencia de un algoritmo es medido por los recursos que éste consume. Normalmente la forma de medir la eficiencia es el tiempo, aunque algunas veces otras medidas son importantes tales como espacio y número de procesos utilizados.

Emisario. Entidad que envía un mensaje o información generalmente en forma oculta para la seguridad de la misma.

Firmas digitales. Permiten asociar la identidad en forma indiscutible del firmante con el documento digital firmado y detectar modificaciones que pudiera sufrir el mismo, se construyen utilizando únicamente criptografía de clave pública. Para firmar se utiliza la clave privada y para verificar la firma se utiliza la clave pública. Las firmas digitales permiten garantizar los servicios de seguridad de integridad y autenticidad al mismo tiempo.

Generador de una curva elíptica. 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.

Grupo. Es una dupla ordenada {G, *}, donde G es un conjunto y el * es una operación binaria, que cumple con las propiedades asociativa, de cerradura, elemento idéntico y elemento inverso.

Grupo abeliano. Debe cumplir con las características y las propiedades de un grupo y además cumple con la propiedad conmutativa.

Integridad. Es uno de los servicios de seguridad y se refiere al control sobre los datos a fin de asegurar que el contenido de la información no se modifique sin la debida autorización y que durante la transmisión la secuencia de los datos se mantenga.

Intercambio de claves de Diffie-Hellman. Protocolo por medio del cual dos personas pueden intercambiarse pequeñas informaciones secretas por un canal inseguro. El procedimiento está basado en un sistema de claves asimétrico en el que cada comunicante posee dos claves, una de ellas secreta y la otra pública. El protocolo no sólo resolvió el problema del intercambio de claves, sino además fue el origen de la denominada criptografía de clave pública, cuya aplicación tiene importantes implicaciones en el ámbito de la seguridad dentro de las comunicaciones digitales.

No repudio. Es uno de los servicio de seguridad y previene a emisores y/o receptores de negar la emisión y/o recepción de un mensaje transmitido.

Orden de la curva elíptica. Es el número total de puntos pertenecen a la 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.

Prueba de caja blanca. También conocida como prueba de caja de cristal, es un método de diseño de casos de prueba que comprueba los caminos lógicos del software.

Pruebas de caja negra: También llamados prueba de comportamiento, se encargan de los requisitos funcionales de los programas, es decir permite obtener conjuntos de condiciones de entrada que ejerciten completamente los requisitos funcionales del sistema.

Punto cero. También llamado punto en el infinito, es un punto que contienen todas las curvas elípticas y tiene características particulares que no permite utilizar este punto para generar los otros puntos que forman dicha curva ya que es el elemento neutro aditivo para las curvas elípticas.

Receptor. Entidad que recibe la información o mensaje enviado por el emisario y que generalmente debe venir en forma oculta para no ser vista ni modificada por usuarios no autorizados.

RSA. Rivest, Shamir, Adleman, llamado así por las iniciales de sus creadores, es un algoritmo de cifrado de clave pública basado en aritmética modular. Es el único algoritmo aceptado en la práctica como algoritmo asimétrico.

Seguridad en un algoritmo de cifrado. Se refiere a las garantías que ofrece el sistema para que una información cifrada no pueda ser vista por una persona que no posea la clave secreta.

Seguridad informática. Es la colección de herramientas diseñadas para la protección de los sistemas de cómputo a fin de evitar amenazas de confidencialidad, integridad y/o disponibilidad.

Sustitución. Es una de las dos operaciones básicas en la de criptografía en la que se cambian los caracteres del mensaje original por otros según una regla determinada de posición del alfabeto.

TDES (Triple DES). Se le llama de ésta forma al algoritmo que hace tres veces el cifrado del DES. En términos generales utiliza el mismo algoritmo de DES, sólo que se efectúa tres veces con tres diferentes claves.

Transposición. Es una de las dos operaciones básicas en la criptografía en la cual se cambian de posición los caracteres que componen el mensaje en claro según una regla determinada de posición del orden del mensaje.

Vulnerabilidad. Es una debilidad que puede ser explotada para violar la seguridad.

Anexo 2

Código fuente del tutorial del Criptografía de Curvas Elípticas

Módulo donde se encuentran los algoritmos de cifrado principales.

```
Module Module1


    'Algoritmo para encontrar el maximo común divisor
    Public Function mcd(ByVal primernumero As Long, ByVal
segundonumero As Long)
        'La función pide dos números y regresa el máximo común divisor
        Dim A As Long
        Dim B As Long
        Dim residuo As Long
        A = primernumero
        B = segundonumero
        Do While (1)
            If (B = 0) Then
                Exit Do
            Else
                residuo = A Mod B
                A = B
                B = residuo
            End If
        Loop
        Return A
    End Function


    'Algoritmo para verificar si se tiene un número primo
    Public Function primo(ByVal numero As Long)
        'El algoritmo pide un número y regresa un "uno" si es un
'número primo y un "cero" si no es un número primo
        Dim contador As Long
        Dim verificador As Long
        Dim regreso As Byte
```

```
        contador = numero
        Do Until (contador = 1)
            contador = contador - 1
            'Utiliza el algoritmo de MCD para verificar si es primo
            verificador = mcd(numero, contador)
            If (verificador <> 1) Then
                regreso = 0
                Exit Do
            End If
        Loop
        If (contador = 1) Then
            regreso = 1
        End If
        Return regreso
    End Function


    'Algoritmo de Euclides
    Public Function inverso(ByVal Campo As Long, ByVal
numeroparainverso As Long)
        'El algoritmo pide dos números, el campo de Galois y el número
'al que hay que obtener el inverso
 'Regresa el inverso de un número en un campo de Galois definido
        Dim A1, A2, A3, B1, B2, B3, T1, T2, T3, cociente As Long
        A1 = 1
        A2 = 0
        A3 = Campo
        B1 = 0
        B2 = 1
        B3 = numeroparainverso
        Do While (1)
            If (B3 = 0) Then
                B2 = 0
                Exit Do
            ElseIf (B3 = 1) Then
                If (B2 < 0) Then
                    B2 = Campo + B2
                End If
                Exit Do
            Else
                cociente = A3 \ B3
                T1 = A1 - (cociente * B1)
                T2 = A2 - (cociente * B2)
                T3 = A3 - (cociente * B3)
                A1 = B1
                A2 = B2
                A3 = B3
                B1 = T1
                B2 = T2
                B3 = T3
            End If
        Loop
        Return B2
    End Function



'Algoritmo para calcular el indice de una curva
    Public Function indice(ByVal a As Long, ByVal b As Long, ByVal p
As Long, ByVal x As Long, ByVal y As Long)
```

```
'El indice de una curva es el número de puntos no repetidos que tiene
'una curva elíptica
'El algoritmo pide los coeficientes de la curva elíptica "a" y "b", el
'campo de Galois sobre el que se define
'También pide las coordenadas de un punto, no el punto cero, que
'pertenece a la curva elóptica sobre el campo
 'definido. Retorna un número entero
        Dim z As Single
        Dim s As Single
        Dim t As Single
        Dim Q As Integer
        Dim R As Integer
        Dim i As Integer
        Dim j As Integer
        Dim k As Integer
        Dim l As Integer
        Dim primero(2) As Long
        Dim segundo(2) As Long
        Dim n As Long
        s = p ^ (1 / 4)
        z = s
        s = (s \ 1)
        If (z > s) Then
            s = s + 1
        End If
        t = (2 * (p ^ (1 / 2))) \ (2 * s + 1)
        Q = 2 * s + 1
        R = p + 1
        For i = -t To t
            k = R + (i * Q)
 'k*P
            primero = Suma(a, b, k, p, x, y)
            For j = -s To s
                l = j
                If (j < 0) Then
                    l = -j
                End If
 'l*P
                segundo = Suma(a, b, l, p, x, y)
                If (j < 0) Then
                    segundo(1) = p - segundo(1)
                End If
                If ((primero(0) = segundo(0)) And (primero(1) =
segundo(1))) Then
                    n = R + (i * Q) - j
                End If
            Next
        Next
        Return n
    End Function



'Algoritmo para encontrar la multiplicación de un escalar por un
'punto de una curva elíptica
    Public Function Suma(ByVal a As Long, ByVal b As Long, ByVal k As
Long, ByVal p As Long, ByVal x As Long, ByVal y As Long)
'El algoritmo requiere los parametros de la curva elíptica, el campo
'de Galoi sobre el que se define
'las coordenadas de un punto perteneciente a la curva y el entero por
'el que se quiere multipolicar a algún punto de la curva
```

```
'Regresa las coordenadas del punto multiplicada por el escalar
'Definimos las variables a utilizar
'Dim a, b, k, p, x, y As Long
'Definimos algunas variables de control
        Dim contador, i, j, divisor, resul, parcial, count, landa,
Qrx, Qry As Long
        Dim Qx, Qy As Long
        Qx = 0
        Qy = 0
        Dim binario_k(3000) As Integer
'Se declara un arreglo para los resultados
        Dim resultado(2) As Long
        'convertimos el número k a número binario
        i = k
        contador = 0
        Do While (i <> 0)
            j = i Mod 2
            binario_k(contador) = j
            i = i \ 2
            contador = contador + 1
        Loop
'Iniciamos el proceso de calcular la multiplicación
        Do
            If (contador = 0) Then
                Exit Do
            End If
            contador = contador - 1
'Calculamos el doble del punto de acuerdo a las reglas de adición
'para curvas elípticas
            divisor = 2 * Qy
            If (divisor < 0) Then
                divisor = divisor + p
            End If
            resul = inverso(p, divisor)
            parcial = (((3 * Qx * Qx) + a)) Mod p
            count = ((parcial) * resul)
            landa = count Mod p
            If (landa < 0) Then
                landa = landa + p
            End If

            Qrx = ((landa * landa) - (2 * Qx)) Mod p
            If (Qrx < 0) Then
                Qrx = Qrx + p
            End If
            Qry = (landa * (Qx - Qrx) - Qy) Mod p
            If (Qry < 0) Then
                Qry = Qry + p
            End If
            Qx = Qrx
            Qy = Qry
'calculamos la suma de dos puntos diferentes de acuerdo a las reglas
'de adición para curvas elípticas
            If (binario_k(contador) = 1) Then
                If (Qx = 0 And Qy = 0) Then
                    Qrx = x
                    Qry = y
                Else
                    divisor = x - Qx
                    If (divisor < 0) Then
                        divisor = divisor + p
```

```
                        End If
                        resul = inverso(p, divisor)
                        parcial = (y - Qy)
                        If (parcial < 0) Then
                            parcial = parcial + p
                        End If
                        count = ((parcial) * resul)
                        landa = count Mod p

                        If (landa < 0) Then
                            landa = landa + p
                        End If
                        Qrx = ((landa * landa) - x - Qx) Mod p
                        If (Qrx < 0) Then
                            Qrx = Qrx + p
                        End If
                        Qry = (landa * (x - Qrx) - y) Mod p
                        If (Qry < 0) Then
                            Qry = Qry + p
                        End If
                    End If
                End If
                Qx = Qrx
                resultado(0) = Qrx
                Qy = Qry
                resultado(1) = Qry
            Loop While (contador)
    'Se regresa el apuntador al arreglo con los resultados de la
    'múltiplicación
            Return resultado
        End Function
    End Module

End Class
```

Primera lámina: Presentación de tutorial de Criptografía de Curvas Elípticas.

```
Public Class frmInicio

'boton para ir a la siguiente diapositiva

Private Sub Siguiente_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Siguiente.Click

            Dim oform As frmIntroduccion
            oform = New frmIntroduccion
            oform.Show()
            oform = Nothing
            Me.Hide()

        End Sub
    'Boton para salir de la aplicación
        Private Sub btnSalir_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnSalir.Click
```

```
            Dim Reply As DialogResult

    'Mensaje para verificar que se quiere salir del programa
     Reply = MsgBox("¿Realmente desea salir del tutorial?",
    MsgBoxStyle.Question Or MsgBoxStyle.YesNo)
            If Reply = Windows.Forms.DialogResult.Yes Then
                Me.Close()
            End If

        End Sub
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
    'Se crea un cuadro con el nombre de la lámina
            Dim Grafico As Graphics = Me.CreateGraphics
    'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1030, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("Tutorial de Criptografía de Curvas
    Elípticas", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.White), 14, 17)
            Grafico.DrawString("Tutorial de Criptografía de Curvas
    Elípticas", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.DarkBlue), 15, 15)

        End Sub
```

Segunda lámina: Introducción a las Curvas Elípticas.

```
    Public Class frmIntroduccion
    'Se dibuja un cuadro en la lámina para poner el nombre de ésta
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)

            Dim Grafico As Graphics = Me.CreateGraphics
    'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1030, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("Introducción a las Curvas Elípticas", New
    Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.White), 14,
    17)
            Grafico.DrawString("Introducción a las Curvas Elípticas", New
    Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.DarkBlue), 15,
    15)
        End Sub

End Class
```

Tercera lámina: Bosquejo de Curvas Elípticas.

```
    'Bibliotecas para utilizar funciones matemáticas y graficos en 2-D
    Imports System.Math
```

```
Imports System.Drawing
Imports System.Drawing.Drawing2D

Public Class frmCE
'Botón para el coeficiente "a" de la curva elíptica
    Private Sub NumericUpDown1_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown1.ValueChanged
        Label1.Text = "a = " & NumericUpDown1.Value
        Label3.Text = "y^2=x^3+(" & NumericUpDown1.Value & ")x+(" &
NumericUpDown3.Value & ")"
        Label4.Text = "4(" & NumericUpDown1.Value & ")^3+27(" &
NumericUpDown3.Value & ")^2="
        TextBox1.Text = 4 * (NumericUpDown1.Value) ^ 3 + 27 *
(NumericUpDown3.Value) ^ 2
        If (4 * (NumericUpDown1.Value) ^ 3 + 27 *
(NumericUpDown3.Value) ^ 2 = 0) Then
            Label5.ForeColor = Color.Red
            Label5.Text = "Los coeficientes no sirven para
criptografía"
        Else
            Label5.ForeColor = Color.Black
            Label5.Text = "Los coeficientes sirven para criptografía"
        End If
    End Sub
 'Botón para el coeficiente "b" de la curva elíptica
    Private Sub NumericUpDown3_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown3.ValueChanged
        Label2.Text = "b = " & NumericUpDown3.Value
        Label3.Text = "y^2=x^3+(" & NumericUpDown1.Value & ")x+(" &
NumericUpDown3.Value & ")"
        Label4.Text = "4(" & NumericUpDown1.Value & ")^3+27(" &
NumericUpDown3.Value & ")^2="
        TextBox1.Text = 4 * (NumericUpDown1.Value) ^ 3 + 27 *
(NumericUpDown3.Value) ^ 2
        If (4 * (NumericUpDown1.Value) ^ 3 + 27 *
(NumericUpDown3.Value) ^ 2 = 0) Then
            Label5.ForeColor = Color.Red
            Label5.Text = "Los coeficientes no sirven para
criptografía"
        Else
            Label5.ForeColor = Color.Black
            Label5.Text = "Los coeficientes sirven para criptografía"
        End If
    End Sub
 'Cuadro para poner el nombre de la lámina
    Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
        Dim Grafico As Graphics = Me.CreateGraphics
 'dibujamos un rectangulo
        Grafico.FillRectangle(New SolidBrush(Color.Gold), New
Rectangle(0, 0, 1030, 72))
 'dibujamos un texto dándole efecto 3D
        Grafico.DrawString("Curvas Elípticas", New Font("Arial", 36,
FontStyle.Bold), New SolidBrush(Color.White), 14, 17)
        Grafico.DrawString("Curvas Elípticas", New Font("Arial", 36,
FontStyle.Bold), New SolidBrush(Color.DarkBlue), 15, 15)
    End Sub
 'Cuadro para gráficar una curva elíptica
    Private Sub picturebox1_paint(ByVal sender As System.Object, ByVal
e As System.Windows.Forms.PaintEventArgs) Handles PictureBox1.Paint
```

```
 'Movemos las coordenadas para gráficar

e.Graphics.TranslateTransform(CSng(PictureBox1.ClientSize.Width / 2),
CSng(PictureBox1.ClientSize.Height / 2))
 'y dibujamos un rectangulo a partir de ese centro
 'y otro en la parte negativa de los ejes
        e.Graphics.FillRectangle(Brushes.Aqua, -175, -120, 350, 240)
        Dim grafico As Graphics = e.Graphics
 'ejes de las coordenadas
        Dim lapiz As New Pen(Color.Gray, 1)
        lapiz.EndCap = Drawing.Drawing2D.LineCap.ArrowAnchor
        grafico.DrawLine(lapiz, -175, 0, 172, 0)
        lapiz.EndCap = Drawing.Drawing2D.LineCap.Round
        lapiz.StartCap = Drawing.Drawing2D.LineCap.ArrowAnchor
        grafico.DrawLine(lapiz, 0, -118, 0, 120)
 'letras de las coordenadas
        Dim tipoletra As Font = New Font("courier new", 8,
FontStyle.Regular)
        e.Graphics.DrawString("X", tipoletra, Brushes.Gray, 160, -15)
        e.Graphics.DrawString("Y", tipoletra, Brushes.Gray, 5, -115)
    End Sub
 'Se gráfica la función que se introdujo en las coeficientes "a" y "b"
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
 'Se crea el lienzo para dibujar el bosquejo de la gráfica
        Dim lienzo As Graphics = PictureBox1.CreateGraphics
 'Se divide en cuadrantes el lienzo de dibujo
        lienzo.TranslateTransform(CSng(PictureBox1.ClientSize.Width /
2), CSng(PictureBox1.ClientSize.Height / 2))
        Dim x As Integer
        Dim y As Integer
        Dim a As Integer
        Dim b As Integer
        Dim discriminante As Integer
'Se convierten a valores enteros los valores tomados de las cajas de
'número
        a = CInt(NumericUpDown1.Value)
        b = CInt(NumericUpDown3.Value)
        Dim puntos(350) As Point
        Dim puntos2(350) As Point

        For x = -34 To 175
'Se evaluala el discriminante
            discriminante = (x * x * x) + (a * x) + b
            If discriminante < 0 Then
                y = 0
                puntos(x + 175) = New Point(x, y)
                puntos2(x + 175) = New Point(x, -y)
            ElseIf discriminante > 0 Then
'Se evaluala la ecuación
                y = Sqrt((x * x * x) + (a * x) + b)

                puntos(x + 175) = New Point(x, y)
                puntos2(x + 175) = New Point(x, -y)
'Se obtiene un punto anterior
                Dim PuntoInicial As PointF
                PuntoInicial = puntos(x + 174)
                Dim PuntoInicial2 As PointF
                PuntoInicial2 = puntos2(x + 174)
'Se obtiene el punto actual
                Dim PuntoFinal As PointF
```

```
                    PuntoFinal = puntos(x + 175)
                    Dim PuntoFinal2 As PointF
                    PuntoFinal2 = puntos2(x + 175)

                    Try
    'Se gráfica la ecuación con el punto anterior y el punto actual
                        lienzo.DrawLine(New Pen(Color.Red, 2),
    PuntoInicial, PuntoFinal)
                        lienzo.DrawLine(New Pen(Color.Red, 2),
    PuntoInicial2, PuntoFinal2)

                    Catch ex As Exception
                      lienzo.DrawLine(New Pen(Color.Green, 3), 0, 0, 0, 0)
                    End Try

                End If

            Next

        End Sub
    'Limpia la caja de dibujo
        Private Sub BtnLimpiar_Click(ByVal sender As System.Object, ByVal
    e As System.EventArgs) _
            Handles BtnLimpiar.Click
            Me.PictureBox1.Invalidate()
        End Sub


End Class
```

Cuarta lámina: Adición geométrica para Curvas Elípticas.

```
    'Librerias para dibujo e importación de imagenes
    Imports System.Drawing
    Imports System.Drawing.Imaging

    Public Class frmAdicionGeometrica
    'Selecciona una imagen de acuerdo a las reglas para adición en CE
        Private Sub NumericUpDown1_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown1.ValueChanged
            Label2.Text = "Regla: " & NumericUpDown1.Value
    'Case para seleccionar la imagen que se mostrara con ayuda del botón
    'numérico
            Select Case True
                Case NumericUpDown1.Value = 1
    'Selecionamos regla 1
                    PictureBox1.Image = My.Resources.Imagen1
                Case NumericUpDown1.Value = 2
    'Selecionamos regla 2
                    PictureBox1.Image = My.Resources.Imagen2
                Case NumericUpDown1.Value = 3
    'Selecionamos regla 3
                    PictureBox1.Image = My.Resources.Imagen3
                Case NumericUpDown1.Value = 4
    'Selecionamos regla 4
                    PictureBox1.Image = My.Resources.Imagen4
```

```
                Case NumericUpDown1.Value = 5
    'Selecionamos regla 5
                    PictureBox1.Image = My.Resources.Imagen5
            End Select
        End Sub
    'Dibujamos un cuadro para anotar el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
    'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1030, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("Adición Geométrica en Curvas Elípticas",
    New Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.White),
    14, 17)
            Grafico.DrawString("Adición Geométrica en Curvas Elípticas",
    New Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.DarkBlue),
    15, 15)
        End Sub

End Class
```

Importancia de los parámetros P, Q y R.

```
    Public Class ImportanciapuntoRQ
    'Menu para ir a la Introdución del programa
        Private Sub mnuInicioIntroduccion_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    mnuInicioIntroduccion.Click
            Dim oform As frmIntroduccion
            oform = New frmIntroduccion
            oform.Show()
            oform = Nothing
            Me.Close()
        End Sub
     ' Ir al inicio del programa
        Private Sub ToolStripMenuItem1_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    ToolStripMenuItem1.Click
            frmInicio.Show()
            Me.Close()
        End Sub
    'Cuadrto para anotar el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
     'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1030, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("Importancia de los parámetros P, Q y R",
    New Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.White),
    14, 17)
            Grafico.DrawString("Importancia de los parámetros P, Q y R",
    New Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.DarkBlue),
    15, 15)
```

```
        End Sub

End Class
```

Quinta lámina: Descripción algebraica de la adición para Curvas Elípticas.

```
    Public Class frmAdicionAlgebraica
    'Menu para ir a la Introdución del programa
        Private Sub mnuInicioIntroduccion_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    mnuInicioIntroduccion.Click
            Dim oform As frmIntroduccion
            oform = New frmIntroduccion
            oform.Show()
            oform = Nothing
            Me.Close()
        End Sub
    'Elige el elemento Xr
        Private Sub NumericUpDown2_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown2.ValueChanged
            Label7.Text = "R=(" & NumericUpDown2.Value & "," &
    NumericUpDown3.Value & ")"
            Label8.Text = " "
        End Sub
    'Elige el elemneto Yr
        Private Sub NumericUpDown3_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown3.ValueChanged
            Label7.Text = "R=(" & NumericUpDown2.Value & "," &
    NumericUpDown3.Value & ")"
            Label8.Text = " "
        End Sub
    'Verifica si el resultado es correcto o no
        Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles Button1.Click
            If (NumericUpDown2.Value = 15 And NumericUpDown3.Value = -56)
    Then
                Label8.ForeColor = Color.Black
                Label8.Text = "Bien, resultado correcto"
            Else
                Label8.ForeColor = Color.Red
                Label8.Text = "Resultado incorrecto, pruebe otros valores"
            End If
        End Sub
    'Dibuja un cuadro para escribir el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
            'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1030, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("Adición Algebraica en Curvas Elípticas",
    New Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.White),
    14, 17)
```

```
            Grafico.DrawString("Adición Algebraica en Curvas Elípticas",
    New Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.DarkBlue),
    15, 15)
        End Sub

End Class
```

Sexta lámina: Curvas Elípticas sobre los números primos.

```
    Public Class frmCEPrimos
    'Menu para ir a la Introdución del programa
        Private Sub mnuInicioIntroduccion_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    mnuInicioIntroduccion.Click
            Dim oform As frmIntroduccion
            oform = New frmIntroduccion
            oform.Show()
            oform = Nothing
            Me.Close()
        End Sub
    'Se obtiene el inverso según la división extendida de Euclides
        Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles Button1.Click
    ' boton para obtener el inverso multiplicativo de un número en un
    'campo de Galois definido
            If (NumericUpDown1.Value < NumericUpDown2.Value) Then
                TextBox1.ForeColor = Color.Red
                TextBox1.Text = "No se puede obtener un inverso
    multiplicativo mayor que el campo"
            Else
                Dim resultado As Long
                resultado = inverso(CLng(NumericUpDown1.Value),
    CLng(NumericUpDown2.Value))
                TextBox1.ForeColor = Color.Black
                TextBox1.Text = resultado
            End If
        End Sub
     'Se gráfica un cuadro para poner el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
    'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1030, 72))
    'dibujamos un texto dándole efecto 3D
           Grafico.DrawString("Curvas Elípticas sobre los números primos",
    New Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.White),
    14, 17)
            Grafico.DrawString("Curvas Elípticas sobre los números
    primos", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.DarkBlue), 15, 15)
        End Sub
    'Campo de Galois
        Private Sub NumericUpDown1_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown1.ValueChanged
```

```
            TextBox1.Text = " "
        End Sub
    'Número al que se le quiere obtener el inverso
        Private Sub NumericUpDown2_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown2.ValueChanged
            TextBox1.Text = " "
        End Sub


End Class
```

Séptima lámina: Curvas Elípticas y el problema del logaritmo discreto.

```
    Public Class frmCEPLD
    'Cambia el valor del parámetro "a" de la Curva
        Private Sub NumericUpDown1_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown1.ValueChanged
            Label4.Text = "y^2=x^3+" & NumericUpDown1.Value & "*x+" &
    NumericUpDown2.Value
            TextBox1.Text = " "
        End Sub
    'Cambia el valor del parámetro "b" de la Curva
        Private Sub NumericUpDown2_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown2.ValueChanged
            Label4.Text = "y^2=x^3+" & NumericUpDown1.Value & "*x+" &
    NumericUpDown2.Value
            TextBox1.Text = " "
        End Sub
    'Cambia el valor de la coordenada "x" del punto
        Private Sub NumericUpDown4_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown4.ValueChanged
            Label10.Text = "P(" & NumericUpDown4.Value & "," &
    NumericUpDown5.Value & ")"
            TextBox1.Text = " "
        End Sub
    'Cambia el valor de la coordenada "y" del punto
        Private Sub NumericUpDown5_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown5.ValueChanged
            Label10.Text = "P(" & NumericUpDown4.Value & "," &
    NumericUpDown5.Value & ")"
            TextBox1.Text = " "
        End Sub
    'Cambia el valor del escalar que se quiere multiplicar
        Private Sub NumericUpDown6_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown6.ValueChanged
            Label12.Text = NumericUpDown6.Value & "*P(" &
    NumericUpDown4.Value & "," & NumericUpDown5.Value & ")="
            TextBox1.Text = " "
        End Sub
```

```
    ' boton para obtener la suma multiplicación de un punto de una curva
    'elíptica por un escalar
        Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles Button1.Click
            If (NumericUpDown1.Value > NumericUpDown3.Value Or
    NumericUpDown2.Value > NumericUpDown3.Value) Then
                TextBox1.ForeColor = Color.Red
                TextBox1.Text = "Los parametros a ó b son mayores que el
    campo de Galois"
            ElseIf ((((NumericUpDown5.Value) ^ 2 Mod
    NumericUpDown3.Value)) <> (((NumericUpDown4.Value) ^ 3 +
    ((NumericUpDown1.Value) * (NumericUpDown4.Value)) +
    (NumericUpDown2.Value)) Mod NumericUpDown3.Value)) Then
                TextBox1.ForeColor = Color.Red
                TextBox1.Text = "El punto dado no perternece a la curva"
            Else
                Dim resultados(2) As Long
                resultados = Suma(CLng(NumericUpDown1.Value),
    CLng(NumericUpDown2.Value), CLng(NumericUpDown6.Value),
    CLng(NumericUpDown3.Value), CLng(NumericUpDown4.Value),
    CLng(NumericUpDown5.Value))
                TextBox1.ForeColor = Color.Black
                TextBox1.Text = "(" & resultados(0) & "," & resultados(1)
    & ")"
            End If
        End Sub
    ' boton para reestablecer los valores predeterminados
        Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles Button2.Click
            NumericUpDown1.Value = 1
            NumericUpDown2.Value = 7
            NumericUpDown3.Value = 17
            NumericUpDown4.Value = 1
            NumericUpDown5.Value = 3
            NumericUpDown6.Value = 7
            TextBox1.Text = ""
        End Sub
    'Dibuja un cuadro para poner el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
    'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1030, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("Curvas Elípticas y el PLD", New
    Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.White), 14,
    17)
            Grafico.DrawString("Curvas Elípticas y el PLD", New
    Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.DarkBlue), 15,
    15)
        End Sub
    'Cambia el valor del Campo de Galois de la Curva
        Private Sub NumericUpDown3_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown3.ValueChanged
            TextBox1.Text = " "
        End Sub

End Class
```

Importancia de la multiplicación en Curvas Elípticas.

```
    Public Class Importanciamultiplicacion
    'Dibuja un cuadro para poner el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
            'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1200, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("Importancia de la multiplicación en curvas
    elípticas", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.White), 14, 17)
            Grafico.DrawString("Importancia de la multiplicación en curvas
    elípticas", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.DarkBlue), 15, 15)
        End Sub

End Class
```

Octava lámina: Curvas Elípticas.

```
    Public Class frmCCE
    'Dibuja un cuadro para poner el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
    'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1030, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("Criptografía de Curvas Elípticas", New
    Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.White), 14,
    17)
            Grafico.DrawString("Criptografía de Curvas Elípticas", New
    Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.DarkBlue), 15,
    15)
        End Sub

End Class
```

Novena lámina: Obtención de múltiplos de puntos.

```
    Public Class frmObtencionPuntos
    'Valor del parámetro "b" de la curva
        Private Sub NumericUpDown2_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown2.ValueChanged
            Label4.Text = "y^2=x^3+" & NumericUpDown1.Value & "*x+" &
    NumericUpDown2.Value
            TextBox1.Text = " "
        End Sub
```

```
'Valor del parámetro "x" del punto
    Private Sub NumericUpDown4_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown4.ValueChanged
        Label12.Text = NumericUpDown6.Value & "*P(" &
NumericUpDown4.Value & "," & NumericUpDown5.Value & ")="
        Label10.Text = "P(" & NumericUpDown4.Value & "," &
NumericUpDown5.Value & ")"
        TextBox1.Text = " "
    End Sub
'Valor del parámetro "x" del punto
    Private Sub NumericUpDown5_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown5.ValueChanged
        Label12.Text = NumericUpDown6.Value & "*P(" &
NumericUpDown4.Value & "," & NumericUpDown5.Value & ")="
        Label10.Text = "P(" & NumericUpDown4.Value & "," &
NumericUpDown5.Value & ")"
        TextBox1.Text = " "
    End Sub
'Valor del escalar por el que se quiere multiplicar el punto
    Private Sub NumericUpDown6_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown6.ValueChanged
        Label12.Text = NumericUpDown6.Value & "*P(" &
NumericUpDown4.Value & "," & NumericUpDown5.Value & ")="
        TextBox1.Text = " "
    End Sub
' boton para obtener la suma multiplicación de un punto de una curva
'elíptica por un escalar
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        If (NumericUpDown1.Value > NumericUpDown3.Value Or
NumericUpDown2.Value > NumericUpDown3.Value) Then
            TextBox1.ForeColor = Color.Red
            TextBox1.Text = "Los parametros a ó b son mayores que el
campo de Galois"
        ElseIf ((((NumericUpDown5.Value) ^ 2 Mod
NumericUpDown3.Value)) <> (((NumericUpDown4.Value) ^ 3 +
((NumericUpDown1.Value) * (NumericUpDown4.Value)) +
(NumericUpDown2.Value)) Mod NumericUpDown3.Value)) Then
            TextBox1.ForeColor = Color.Red
            TextBox1.Text = "El punto dado no perternece a la curva"
        Else
            Dim resultados(2) As Long
            resultados = Suma(CLng(NumericUpDown1.Value),
CLng(NumericUpDown2.Value), CLng(NumericUpDown6.Value),
CLng(NumericUpDown3.Value), CLng(NumericUpDown4.Value),
CLng(NumericUpDown5.Value))
            TextBox1.ForeColor = Color.Black
            TextBox1.Text = "(" & resultados(0) & "," & resultados(1)
& ")"
        End If
     End Sub
 ' boton para reestablecer los valores predeterminados
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        NumericUpDown1.Value = 56
        NumericUpDown2.Value = 74
        NumericUpDown3.Value = 83
        NumericUpDown4.Value = 19
```

```
        NumericUpDown5.Value = 64
        NumericUpDown6.Value = 75
        TextBox1.Text = ""
    End Sub
 'Dibuja un cuadro para anotar el nombre de la lámina
    Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)

        Dim Grafico As Graphics = Me.CreateGraphics
        'dibujamos un rectangulo
        Grafico.FillRectangle(New SolidBrush(Color.Gold), New
Rectangle(0, 0, 1030, 72))
 'dibujamos un texto dándole efecto 3D
        Grafico.DrawString("Obtención de puntos en Curvas Elípticas",
New Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.White),
14, 17)
        Grafico.DrawString("Obtención de puntos en Curvas Elípticas",
New Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.DarkBlue),
15, 15)
    End Sub
 'Valor del parámetro "a" de la curva
    Private Sub NumericUpDown1_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown1.ValueChanged
        Label4.Text = "y^2=x^3+" & NumericUpDown1.Value & "*x+" &
NumericUpDown2.Value
        TextBox1.Text = " "
    End Sub
 'Valor del campo de Galois de la curva
    Private Sub NumericUpDown3_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown3.ValueChanged
        TextBox1.Text = " "
    End Sub

End Class
```

Décima lámina: Intercambio de claves secretas.

```
Public Class frmIntercambioClaves
'Parámetro "a" de la curva elíptica
    Private Sub NumericUpDown1_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown1.ValueChanged
        Label4.Text = "E" & NumericUpDown3.Value & "(" &
NumericUpDown1.Value & "," & NumericUpDown2.Value & ")"
        TextBox1.Text = " "
        TextBox2.Text = " "
        Label15.Text = ""
    End Sub
'Parámetro "a" de la curva elíptica
    Private Sub NumericUpDown2_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown2.ValueChanged
        Label4.Text = "E" & NumericUpDown3.Value & "(" &
NumericUpDown1.Value & "," & NumericUpDown2.Value & ")"
        TextBox1.Text = " "
```

```
        TextBox2.Text = " "
        Label15.Text = ""
    End Sub
'Coordenada "x" de la curva elíptica
    Private Sub NumericUpDown4_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown4.ValueChanged
        Label10.Text = "P(" & NumericUpDown4.Value & "," &
NumericUpDown5.Value & ")"
        TextBox1.Text = " "
        TextBox2.Text = " "
        Label12.Text = NumericUpDown6.Value & "*P(" &
NumericUpDown4.Value & "," & NumericUpDown5.Value & ")="
        Label13.Text = NumericUpDown7.Value & "*P(" &
NumericUpDown4.Value & "," & NumericUpDown5.Value & ")="
        Label15.Text = ""
    End Sub
'Coordenada "x" de la curva elíptica
    Private Sub NumericUpDown5_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown5.ValueChanged
        Label10.Text = "P(" & NumericUpDown4.Value & "," &
NumericUpDown5.Value & ")"
        TextBox1.Text = " "
        TextBox2.Text = " "
        Label15.Text = ""
        Label12.Text = NumericUpDown6.Value & "*P(" &
NumericUpDown4.Value & "," & NumericUpDown5.Value & ")="
        Label13.Text = NumericUpDown7.Value & "*P(" &
NumericUpDown4.Value & "," & NumericUpDown5.Value & ")="
    End Sub
'Clave secreta del usuario A
    Private Sub NumericUpDown6_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown6.ValueChanged
        Label12.Text = NumericUpDown6.Value & "*P(" &
NumericUpDown4.Value & "," & NumericUpDown5.Value & ")="
        TextBox1.Text = " "
        TextBox2.Text = " "
        Label15.Text = ""
    End Sub
'Clave secreta del usuario B
    Private Sub NumericUpDown7_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown7.ValueChanged
        Label13.Text = NumericUpDown7.Value & "*P(" &
NumericUpDown4.Value & "," & NumericUpDown5.Value & ")="
        TextBox1.Text = " "
        TextBox2.Text = " "
        Label15.Text = ""
    End Sub
' boton para obtener la suma multiplicación de un punto de una curva
'elíptica por un escalar
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        If (NumericUpDown1.Value > NumericUpDown3.Value Or
NumericUpDown2.Value > NumericUpDown3.Value) Then
            Label15.ForeColor = Color.Red
            Label15.Text = "Los parametros a ó b son mayores que el
campo de Galois"
```

```
        ElseIf ((((NumericUpDown5.Value) ^ 2 Mod
NumericUpDown3.Value)) <> (((NumericUpDown4.Value) ^ 3 +
((NumericUpDown1.Value) * (NumericUpDown4.Value)) +
(NumericUpDown2.Value)) Mod NumericUpDown3.Value)) Then
            Label15.ForeColor = Color.Red
            Label15.Text = "El punto dado no perternece a la curva"
        Else
            Dim resultados(2) As Long
            TextBox1.ForeColor = Color.Black
            Label15.ForeColor = Color.Black
            resultados = Suma(CLng(NumericUpDown1.Value),
CLng(NumericUpDown2.Value), CLng(NumericUpDown6.Value),
CLng(NumericUpDown3.Value), CLng(NumericUpDown4.Value),
CLng(NumericUpDown5.Value))
            TextBox1.Text = "(" & resultados(0) & "," & resultados(1)
& ")"
            resultados = Suma(CLng(NumericUpDown1.Value),
CLng(NumericUpDown2.Value), CLng(NumericUpDown7.Value),
CLng(NumericUpDown3.Value), CLng(NumericUpDown4.Value),
CLng(NumericUpDown5.Value))
            TextBox2.Text = "(" & resultados(0) & "," & resultados(1)
& ")"
            resultados = Suma(CLng(NumericUpDown1.Value),
CLng(NumericUpDown2.Value), CLng(NumericUpDown6.Value),
CLng(NumericUpDown3.Value), CLng(resultados(0)), CLng(resultados(1)))
            Label15.Text = "La clave secreta común será: K = " &
NumericUpDown6.Value & TextBox2.Text & " = " & NumericUpDown7.Value &
TextBox1.Text & " =" & "(" & resultados(0) & "," & resultados(1) & ")"
        End If
    End Sub
' boton para reestablecer los valores predeterminados
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        NumericUpDown1.Value = 0
        NumericUpDown2.Value = -4
        NumericUpDown3.Value = 211
        NumericUpDown4.Value = 2
        NumericUpDown5.Value = 2
        NumericUpDown6.Value = 121
        NumericUpDown7.Value = 203
        TextBox1.Text = ""
        TextBox2.Text = ""
        Label15.Text = ""
    End Sub
'Campo de Galois
    Private Sub NumericUpDown3_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown3.ValueChanged
        TextBox1.Text = " "
        TextBox2.Text = " "
        Label15.Text = ""
    End Sub
'Dibuja un cuadro para anotar el nombre de la lámina
    Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)

        Dim Grafico As Graphics = Me.CreateGraphics
'dibujamos un rectangulo
        Grafico.FillRectangle(New SolidBrush(Color.Gold), New
Rectangle(0, 0, 1030, 72))
'dibujamos un texto dándole efecto 3D
```

```
            Grafico.DrawString("Intercambio de claves en Curvas
    Elípticas", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.White), 14, 17)
            Grafico.DrawString("Intercambio de claves en Curvas
    Elípticas", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.DarkBlue), 15, 15)
        End Sub

End Class
```

Onceava lámina: Codificación en Curvas Elípticas.

```
    Public Class frmCodificacion
    'Dibuja un cuadro para poner el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
            'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1030, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("Codificación en Curvas Elípticas", New
    Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.White), 14,
    17)
            Grafico.DrawString("Codificación en Curvas Elípticas", New
    Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.DarkBlue), 15,
    15)
        End Sub
    'Parámetro "a" de la curva elíptica
        Private Sub NumericUpDown1_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown1.ValueChanged
            Label4.Text = "y^2=x^3+(" & NumericUpDown1.Value & ")*x+(" &
    NumericUpDown2.Value & ")"
            TextBox1.Text = " "
        End Sub
    'Parámetro "b" de la curva elípica
        Private Sub NumericUpDown2_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown2.ValueChanged
            Label4.Text = "y^2=x^3+(" & NumericUpDown1.Value & ")*x+(" &
    NumericUpDown2.Value & ")"
            TextBox1.Text = " "
        End Sub
    'Parámetro "M" para codificar
        Private Sub NumericUpDown4_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown4.ValueChanged
            Dim Mh As Long
            Dim M As Long
            M = 27
            Mh = CLng(NumericUpDown4.Value) * CLng(NumericUpDown5.Value)
            Label10.Text = NumericUpDown4.Value & "*" &
    NumericUpDown5.Value & "=" & Mh
            If (Mh > CLng(NumericUpDown3.Value)) Then
                TextBox1.ForeColor = Color.Red
```

```
            TextBox1.Text = "La multiplicación de los parámetros M y h
deben ser menores que el campo Galois"
        ElseIf (M > CLng(NumericUpDown4.Value)) Then
            TextBox1.ForeColor = Color.Red
            TextBox1.Text = "El parámetro M debe ser mayor para poder
tener puntos suficientes para todo el alfabeto"
        Else
            TextBox1.ForeColor = Color.Black
            TextBox1.Text = ""
        End If
    End Sub
'Parámetro "h" para codificar
    Private Sub NumericUpDown5_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown5.ValueChanged
        Dim Mh As Long
        Mh = CLng(NumericUpDown4.Value) * CLng(NumericUpDown5.Value)
        Label10.Text = NumericUpDown4.Value & "*" &
NumericUpDown5.Value & "=" & Mh
        If (Mh > CLng(NumericUpDown3.Value)) Then
            TextBox1.ForeColor = Color.Red
            TextBox1.Text = "La multiplicación de los parámetros M y h
deben ser menores que el campo Galois"
        Else
            TextBox1.ForeColor = Color.Black
            TextBox1.Text = ""
        End If
    End Sub
' boton para obtener la suma multiplicación de un punto de una curva
'elíptica por un escalar
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        Dim Mh As Long
        Dim M As Long
        M = 27
        Mh = CLng(NumericUpDown4.Value) * CLng(NumericUpDown5.Value)

        If (NumericUpDown1.Value > NumericUpDown3.Value Or
NumericUpDown2.Value > NumericUpDown3.Value) Then
            TextBox1.ForeColor = Color.Red
            TextBox1.Text = "Los parámetros a ó b son mayores que el
campo de Galois"

        ElseIf (Mh > CLng(NumericUpDown3.Value)) Then
            TextBox1.ForeColor = Color.Red
            TextBox1.Text = "Los parámetros M ó h deben ser
modificados"
        ElseIf (M > CLng(NumericUpDown4.Value)) Then
            TextBox1.ForeColor = Color.Red
            TextBox1.Text = "El valor de M debe ser mayor para cubrir
todo el alfabeto"
        Else
            Dim x As Long
            Dim funcion As Long
            Dim y As Long
            Dim yy As Long
            Dim contador As Integer
            contador = 1
            While contador < NumericUpDown3.Value
                x = NumericUpDown6.Value * NumericUpDown5.Value +
contador
```

```
                funcion = x * x * x + NumericUpDown1.Value * x +
NumericUpDown2.Value
                funcion = funcion Mod NumericUpDown3.Value
                y = 0
                Do Until funcion = yy
                    y = y + 1
                    If y = NumericUpDown3.Value Then
                        Exit Do
                    End If
                    yy = (y * y) Mod NumericUpDown3.Value
                Loop
                contador = contador + 1
                If funcion = yy Then
                    Exit While
                End If
            End While
             TextBox1.ForeColor = Color.Black
            TextBox1.Text = "(" & x & "," & y & ")"
        End If
    End Sub
 ' boton para reestablecer los valores predeterminados
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        NumericUpDown1.Value = -1
        NumericUpDown2.Value = 188
        NumericUpDown3.Value = 751
        NumericUpDown4.Value = 36
        NumericUpDown5.Value = 20
        NumericUpDown6.Value = 16
        TextBox1.Text = ""
    End Sub
 'Elección de la letra a codificar
    Private Sub NumericUpDown6_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown6.ValueChanged
        Dim xbyte As Byte
        Dim strcar As String
        xbyte = CByte(NumericUpDown6.Value)
        strcar = "O"
 'Se hace declaración de letras de acuerdo a la tabla de la lámina
        If xbyte = 1 Then strcar = "A"
        If xbyte = 2 Then strcar = "B"
        If xbyte = 3 Then strcar = "C"
        If xbyte = 4 Then strcar = "D"
        If xbyte = 5 Then strcar = "E"
        If xbyte = 6 Then strcar = "F"
        If xbyte = 7 Then strcar = "G"
        If xbyte = 8 Then strcar = "H"
        If xbyte = 9 Then strcar = "I"
        If xbyte = 10 Then strcar = "J"
        If xbyte = 11 Then strcar = "K"
        If xbyte = 12 Then strcar = "L"
        If xbyte = 13 Then strcar = "M"
        If xbyte = 14 Then strcar = "N"
        If xbyte = 15 Then strcar = "Ñ"
        If xbyte = 16 Then strcar = "O"
        If xbyte = 17 Then strcar = "P"
        If xbyte = 18 Then strcar = "Q"
        If xbyte = 19 Then strcar = "R"
        If xbyte = 20 Then strcar = "S"
        If xbyte = 21 Then strcar = "T"
```

```
            If xbyte = 22 Then strcar = "U"
            If xbyte = 23 Then strcar = "V"
            If xbyte = 24 Then strcar = "W"
            If xbyte = 25 Then strcar = "X"
            If xbyte = 26 Then strcar = "Y"
            If xbyte = 27 Then strcar = "Z"
            Label12.Text = strcar & "="
            TextBox1.Text = " "
        End Sub
     'Campo de Galois
        Private Sub NumericUpDown3_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown3.ValueChanged
            TextBox1.Text = " "
        End Sub

End Class
```

Doceava lámina: ElGamal para Curvas Elípticas.

```
    Public Class frmElgamal
    'Se dibuja un cuadro para escribir el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
    'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1030, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("ElGamal para Curvas Elípticas", New
    Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.White), 14,
    17)
            Grafico.DrawString("ElGamal para Curvas Elípticas", New
    Font("Arial", 36, FontStyle.Bold), New SolidBrush(Color.DarkBlue), 15,
    15)
        End Sub

End Class
```

Ejemplo de ElGamal para Curvas Elípticas.

```
    Public Class Ejemplo_de_el_ElGamal
    'Se gráfica un cuadro para poner el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
    'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1160, 72))
    'dibujamos un texto dándole efecto 3D
```

```
            Grafico.DrawString("Ejemplo de cifrado ElGamal para Curvas
    Elípticas", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.White), 14, 17)
            Grafico.DrawString("Ejemplo de cifrado ElGamal para Curvas
    Elípticas", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.DarkBlue), 15, 15)
        End Sub

End Class
```

Treceava lámina: Generación de Curvas Elípticas aleatorias.

```
    Public Class frmGeneracionCE
    'Se dibuja un cuadro para escribir el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
    'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1030, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("Generación de una curva elíptica
    aleatoria", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.White), 14, 17)
            Grafico.DrawString("Generación de una curva elíptica
    aleatoria", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.DarkBlue), 15, 15)
        End Sub
    'Número primo
        Private Sub NumericUpDown1_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown1.ValueChanged
            TextBox1.Text = " "
            Label4.Text = "y^2 = x^3 + ax + b"
        End Sub
    ' Verifica que se trate de un número primo
        Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles Button1.Click
            Dim verificar As Byte
            verificar = primo(CLng(NumericUpDown1.Value))
            If verificar Then
                TextBox1.ForeColor = Color.Black
                TextBox1.Text = "Número primo"
            Else
                TextBox1.ForeColor = Color.Red
                TextBox1.Text = "No es un número primo"
            End If
        End Sub
    'Genera los parámetros "a" y "b" de la curva elíptica
        Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles Button2.Click
            Dim verificar As Byte
            Dim a As Long
            Dim b As Long
            Dim campo As Long
    'verifica que se trata de un número primo
```

```
            verificar = primo(CLng(NumericUpDown1.Value))
            If verificar Then
    'campo de Galois
                campo = CLng(NumericUpDown1.Value)
                Do Until (4 * a * a * a + 27 * b * b <> 0)
    'Genera los números enteros aleatorios
                    a = Int(((campo - 1) * Rnd()) + 1)
                    b = Int(((campo - 1) * Rnd()) + 1)
                Loop
                Label4.Text = "y^2 = x^3 + " & a & "x + " & b
                TextBox1.ForeColor = Color.Black
                TextBox1.Text = "a=" & a & "    b=" & b
            Else
                TextBox1.ForeColor = Color.Red
                TextBox1.Text = "El número elegido no es primo"
            End If
        End Sub

End Class
```

Catorceava lámina: Generación de parámetros para Curvas Elípticas.

```
    Public Class frmGeneracionClave
    'Dibuja un cuadro para escribir el nombre de la lámina
        Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
            Dim Grafico As Graphics = Me.CreateGraphics
    'dibujamos un rectangulo
            Grafico.FillRectangle(New SolidBrush(Color.Gold), New
    Rectangle(0, 0, 1140, 72))
    'dibujamos un texto dándole efecto 3D
            Grafico.DrawString("Generación de parámetros para Curvas
    Elípticas", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.White), 14, 17)
            Grafico.DrawString("Generación de parámetros para Curvas
    Elípticas", New Font("Arial", 36, FontStyle.Bold), New
    SolidBrush(Color.DarkBlue), 15, 15)
        End Sub
    'Número primo utilizado
        Private Sub NumericUpDown1_ValueChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    NumericUpDown1.ValueChanged
            TextBox1.Text = " "
            Label5.Text = "GF(p)"
            Label6.Text = "a="
            Label7.Text = "b="
            Label8.Text = "P=(x, y)"
            Label1.Text = "y^2 = x^3 + ax + b"
        End Sub
    'Generara los parámetros de las curvas elípticas con ayuda de otras
    'funciones
        Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles Button2.Click
            Dim verificar As Byte
            Dim a As Long
            Dim b As Long
```

```
            Dim campo As Long
            Dim x As Long
            Dim y As Long
            Dim n As Long
            Dim contando As Single
            contando = 0
    'verifica que el número sea primo
            verificar = primo(CLng(NumericUpDown1.Value))
            If verificar Then
    'Verifica que el orden sea impar
                Do Until ((n Mod 2) <> 0)
                    contando = contando + 1
    'Campo de Galois
                    campo = CLng(NumericUpDown1.Value)
                    If (contando <> 0) Then
    'Genera los números enteros aleatorios para los parámetros "a" y "b"
    'de la curva elíptica
                        a = Int(((campo - 1) * Rnd()) + 1)
                        b = Int(((campo - 1) * Rnd(contando * Rnd())) + 1)
                    End If
    'Verifica que la curva no sea supersingular
                    Do Until (4 * a * a * a + 27 * b * b <> 0)
    'Genera los números enteros aleatorios para los parámetros "a" y "b"
    'de la curva elíptica
                        a = Int(((campo - 1) * Rnd()) + 1)
                        b = Int(((campo - 1) * Rnd(contando * Rnd())) + 1)
                    Loop
    'Verifica que el punto generado pertenezca a la curva dada
                    Do Until ((y * y) Mod campo = (x * x * x + a * x + b)
    Mod campo)
    'Genera los números enteros que servirán como coordenadas de un punto
    'de la curva
                        x = Int(((campo - 1) * Rnd(contando * Rnd())) + 1)
                        y = Int(((campo - 1) * Rnd(contando * Rnd())) + 1)
                    Loop
    'Calcula el orden de la curva
                    n = indice(a, b, campo, x, y)
    'Regresa lod resultados a las etiquetas y al cuadro de texto
                    Label1.Text = "y^2 = x^3 + " & a & "x + " & b
                    Label5.Text = "GF(" & CLng(NumericUpDown1.Value) & ")"
                    Label6.Text = "a= " & a
                    Label7.Text = "b= " & b
                    Label8.Text = "P=(" & x & ", " & y & ")"
                    Label10.Text = "N= " & n
                    TextBox1.ForeColor = Color.Black
                    TextBox1.Text = "{" & CLng(NumericUpDown1.Value) & ",
    GF(" & CLng(NumericUpDown1.Value) & "), " & a & ", " & b & ", P=(" & x
    & ", " & y & ")" & ", " & n & "}"
                Loop

            Else
                TextBox1.ForeColor = Color.Red
                TextBox1.Text = "El número elegido no es primo"
            End If
        End Sub

End Class
```

Graficación de puntos en una Curvas Elípticas.

```
Imports System.Math
Imports System.Drawing
Imports System.Drawing.Drawing2D
Public Class Gráfica_de_puntos
'Mensaje para verificar que se quiere salir del programa
 Reply = MsgBox("¿Realmente desea salir del tutorial?",
MsgBoxStyle.Question Or MsgBoxStyle.YesNo)
        If Reply = Windows.Forms.DialogResult.Yes Then
            Me.Close()
        End If

'Dibuja un rectángulo para escribir el nombre de la lámina
    Protected Overrides Sub onpaint(ByVal e As PaintEventArgs)
        Dim Grafico As Graphics = Me.CreateGraphics
'dibujamos un rectangulo
        Grafico.FillRectangle(New SolidBrush(Color.Gold), New
Rectangle(0, 0, 1140, 72))
'dibujamos un texto dándole efecto 3D
        Grafico.DrawString("Graficación de puntos en una Curva
Elíptica", New Font("Arial", 36, FontStyle.Bold), New
SolidBrush(Color.White), 14, 17)
        Grafico.DrawString("Graficación de puntos en una Curva
Elíptica", New Font("Arial", 36, FontStyle.Bold), New
SolidBrush(Color.DarkBlue), 15, 15)
    End Sub
'Dibuja las coordenadas donde se gráfica
    Private Sub picturebox2_paint(ByVal sender As System.Object, ByVal
e As System.Windows.Forms.PaintEventArgs) Handles PictureBox2.Paint
'La base de puntos con coordenadas hasta de (22,22)
        Dim grafico As Graphics = e.Graphics
        Dim tipoletra As Font = New Font("courier new", 8,
FontStyle.Regular)
'ejes de las coordenadas
        Dim lapiz As New Pen(Color.Gray, 0.01)
        Dim i As Integer
        i = 20
        Dim j As Integer
        j = 20
        Dim k As Byte
        k = 0
        lapiz.EndCap = Drawing.Drawing2D.LineCap.NoAnchor
'Verifica todas las coordenadas posibles
        Do Until (j = 480)
'Paralelas eje y
            grafico.DrawLine(lapiz, j, 40, j, 480)
            j = j + 20
            e.Graphics.DrawString(k, tipoletra, Brushes.Gray, (j -
25), 480)
            k = k + 1
        Loop
        Do Until (i = 480)
'Paralelas eje x
            k = k - 1
            i = i + 20
            lapiz.EndCap = Drawing.Drawing2D.LineCap.NoAnchor
            grafico.DrawLine(lapiz, 20, i, 460, i)
```

```
            e.Graphics.DrawString(k, tipoletra, Brushes.Gray, 5, (i - 5))
        Loop
        e.Graphics.DrawString("X", tipoletra, Brushes.Gray, 480, 480)
        e.Graphics.DrawString("Y", tipoletra, Brushes.Gray, 0, 0)
' puntos para verificar circulo
    End Sub
'Campo de Galois que borra los puntos graficados anteriormente
    Private Sub NumericUpDown1_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown1.ValueChanged
        Me.PictureBox2.Invalidate()
        TextBox1.Text = " "
        Label5.Text = "GF(p)"
        Label6.Text = "a="
        Label7.Text = "b="
        Label8.Text = "P=(x, y)"
        Label1.Text = "y^2 = x^3 + ax + b"

    End Sub
'Genera los parámetros de una curva elíptica y gráfica los puntos
'pertenecientes a la curva
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
'Declaración de variables
        Dim verificar As Byte
        Dim a As Long
        Dim b As Long
        Dim campo As Long
        Dim x As Long
        Dim y As Long
        Dim n As Integer
        Dim contando As Single
        Dim i As Integer
        Dim j As Integer
        Dim discriminante As Integer
        Dim x1 As Integer
        Dim y1 As Integer
'Prepara el cuadro para graficar
        Dim grafico As Graphics = PictureBox2.CreateGraphics
        Dim camino As New Drawing2D.GraphicsPath()
        Dim lapiz As New Pen(Color.Red, 1)
        contando = 0
'Verifica que el número sea primo
        verificar = primo(CLng(NumericUpDown1.Value))
        If verificar Then
            contando = contando + 1
            campo = CLng(NumericUpDown1.Value)
            If (contando <> 0) Then
'Genera los números enteros aleatorios para los parámetros de la curva
                a = Int(((campo - 1) * Rnd()) + 1)
                b = Int(((campo - 1) * Rnd(contando * Rnd())) + 1)
            End If
'Verifica que la curva no sea supersingular
            Do Until (4 * a * a * a + 27 * b * b <> 0)
'Si es supersingular vuelve a calcular los parámetros
                a = Int(((campo - 1) * Rnd()) + 1)
                b = Int(((campo - 1) * Rnd(contando * Rnd())) + 1)
            Loop
'Verifica que los puntos pertenezcan a la curva
            Do Until ((y * y) Mod campo = (x * x * x + a * x + b) Mod
campo)
```

```
  'Genera los números enteros aleatorios para el punto de la curva
                x = Int(((campo - 1) * Rnd(contando * Rnd())) + 1)
                y = Int(((campo - 1) * Rnd(contando * Rnd())) + 1)
            Loop
'Obtiene el orden de la curva
            n = 1
'Graficación de puntos
            Dim veri1 As Integer
            Dim veri2 As Integer
            Dim eje As Byte
            eje = campo
'Selecciona el eje en el cual son simétricos los puntos de la curva
            Select Case eje
                Case 7
                    grafico.DrawLine(lapiz, 20, 410, 460, 410)
                Case 11
                    grafico.DrawLine(lapiz, 20, 370, 460, 370)
                Case 13
                    grafico.DrawLine(lapiz, 20, 350, 460, 350)
                Case 17
                    grafico.DrawLine(lapiz, 20, 310, 460, 310)
                Case 19
                    grafico.DrawLine(lapiz, 20, 290, 460, 290)
                Case 23
                    grafico.DrawLine(lapiz, 20, 250, 460, 250)
            End Select
'Verifica todos los puntos posibles
            For i = 0 To 22
                For j = 0 To 22
'verifica que sean reales los puntos
                    discriminante = (i * i * i) + (a * i) + b
                    If discriminante >= 0 Then
'verifica que el valor de i que será el valor de "x" sea menor que el
'campo de Galois
                        If (i < campo) Then
'verifica que el valor de j que será el valor de "y" sea menor que el
'campo de Galois
                            If (j < campo) Then
                                veri1 = (j * j) Mod campo
                            veri2 = ((i * i * i) + (a * i) + b) Mod campo
                        If (((j * j) Mod campo) = (((i * i * i) + (a *
i) + b) Mod campo)) Then
                                    If (veri1 = veri2) Then
                                        x1 = 17 + (i * 20)
                                        y1 = 477 - (j * 20)
                                        camino.AddArc(x1, y1, 6, 6, 0,
360)
'circulo dorado
                grafico.FillPath(New SolidBrush(Color.Gold), camino)
'perímetro del círculo
                grafico.DrawPath(New Pen(Color.Blue, 2), camino)
                                        n = n + 1
                                    End If
                                End If
                            End If
                        End If
                    End If
                Next
            Next
'Regresa los valores a las etiquetas y al cuadro de texto
            Label1.Text = "y^2 = x^3 + " & a & "x + " & b
```

```
                Label5.Text = "GF(" & CLng(NumericUpDown1.Value) & ")"
                Label6.Text = "a= " & a
                Label7.Text = "b= " & b
                Label8.Text = "P=(" & x & ", " & y & ")"
                Label10.Text = "N= " & n
                TextBox1.ForeColor = Color.Black
                TextBox1.Text = "{" & CLng(NumericUpDown1.Value) & ", GF("
    & CLng(NumericUpDown1.Value) & "), " & a & ", " & b & ", P=(" & x & ",
    " & y & ")" & ", " & n & "}"
    'Grafica el punto infinito
                camino.AddArc(17, 477, 6, 6, 0, 360)
    'circulo dorado
                grafico.FillPath(New SolidBrush(Color.Gold), camino)
    'perímetro del círculo
                grafico.DrawPath(New Pen(Color.Blue, 2), camino)
            Else
                TextBox1.ForeColor = Color.Red
                TextBox1.Text = "El número elegido no es primo"
            End If
        End Sub

End Class
```

Anexo 3

Tablas de Curvas Elípticas

Dentro del tutorial de curvas elípticas se hacen los cálculos y la verificación de parámetros de manera digital y automática. Sin embargo en la tesis "Criptografía y Curvas Elípticas" del licenciado Christopher Silva se hacen algunos ejercicios de manera manual, verificando cada uno de los parámetros para desarrollar solamente un ejemplo.

Aquí se muestran las tablas que utilizó en su trabajo para realizar las operaciones de cifrado con el algoritmo de el ELGamal para curvas elípticas, codificado del mensaje, obtención de índice de la curva y todas las tablas mostradas sólo sirven para la curva elíptica $y^2 = x^3 - x + 188$, en caso de cambiar la curva o el campo de Galois que se utilizó (la curva se define sobre el GF(751)) se tendría que realizar nuevamente todas las tablas mostradas. También se tendría que realizar todas las operaciones para multiplicar un punto, cifrar la información, etcétera.

Se presentan 5 tablas todas sobre el campo de Galois de los números primos con GF(751), en la Tabla 1 de la página 153 hasta la página 157 de la tesis "Criptografía y Curvas Elípticas" se presenta los valores que toma el parámetro "x" que van desde 0 hasta 750 y se evalúa como una función $f(x) = x^3 - x + 188$ y se le aplica la operación módulo 751.

En la Tabla 2 que va de la página 158 hasta la página 162 de la tesis mencionada se presentan los valores que toma el parámetro "y" que van desde 0 hasta 750 y se evalúa como una función f(y) = y2 después se la aplica la operación módulo 751.

La curva elíptica que se utilizó tiene un índice de 727, esto quiere decir que la curva esta definida por 727 puntos en el GF(751), en la Tabla 3, que se muestra de la página 163 a la página 169, se aprecian las coordenadas de todos estos puntos.

En la Tabla 4 que se puede ver en la página170 se aprecia la codificación que se utilizó para el alfabeto de 27 letras, la codificación se refiere a asignar a cada carácter del alfabeto un punto de la curva elíptica sobre la que se trabaja.

Finalmente en la Tabla 5 que se encuentran desde la página 171 hasta la página 175, se presentan los resultados de las operaciones que se efectuaron para encontrar el índice de la ecuación, en términos generales es un proceso muy laborioso ya que se debe realizar la multiplicación de cada uno de los puntos por un escalar y comparar los resultados con otra multiplicación. En la tabla se aprecia que todos los valores son diferentes excepto para el valor de 726(1, 376) = -1(1,376).

Lo anterior es equivalente a despejar todo al primer lado de la igualdad 726 (1, 376) + 1(1,376) = 0. Por lo que 727(1, 376) = 0, es decir que se necesitan 727 puntos para cubrir todo el espacio de puntos que pertenecen a la curva donde se encuentra el punto en el infinito.

Todas las tablas resultan muy laboriosas, por lo que el uso de herramientas de software, como el tutorial del presente trabajo, resulta muy útil para realizar las todas las operaciones sin la necesidad de realizar todas las tablas mostradas.

Tabla 1

x	$x^3 - x$ +188	x	$x^3 - x$ +188	x	$x^3 - x$ +188	x	$x^3 - x$ +188
1	188	2	194	3	212	4	248
5	308	6	398	7	524	8	692
9	157	10	427	11	6	12	402
13	119	14	665	15	544	16	513
17	578	18	745	19	269	20	658
21	416	22	300	23	316	24	470
25	17	26	465	27	318	28	333
29	516	30	122	31	659	32	631
33	44	34	406	35	221	36	246
37	487	38	199	39	139	40	313
41	727	42	636	43	46	44	465
45	397	46	599	47	326	48	335
49	632	50	472	51	612	52	307
53	314	54	639	55	537	56	14
57	578	58	733	59	485	60	591
61	306	62	387	63	89	64	169
65	633	66	736	67	484	68	634
69	441	70	662	71	552	72	117
73	114	74	549	75	677	76	504
77	36	78	30	79	492	80	677
81	591	82	240	83	381	84	269
85	661	86	61	87	728	88	415
89	630	90	628	91	415	92	748
93	131	94	72	95	577	96	150
97	299	98	279	99	96	100	507
101	16	102	131	103	107	104	701
105	417	106	12	107	243	108	365
109	384	110	306	111	137	112	634
113	301	114	646	115	173	116	390
117	552	118	665	119	735	120	17
121	19	122	747	123	705	124	650
125	588	126	525	127	467	128	420
129	390	130	383	131	405	132	462

x	$x^3 - x$ +188
133	560
137	731
141	536
145	359
149	584
153	93
157	21
161	1
165	417
169	151
173	338
177	611
181	603
185	698
189	529
193	480
197	184
201	25
205	387
209	152
213	455
217	178
221	456
225	171
229	458
233	199
237	529
241	330
245	737
249	632
253	399
257	422
261	334
265	519
269	610
273	240
277	544
281	404
285	204

x	$x^3 - x$ +188
134	705
138	373
142	522
146	34
150	44
154	185
158	90
162	143
166	728
170	727
174	524
178	503
182	297
186	290
190	115
194	156
198	46
202	169
206	158
210	397
214	519
218	157
222	446
226	268
230	7
234	47
238	21
242	313
246	556
250	383
254	178
258	325
262	457
266	207
270	710
274	97
278	254
282	63
286	659

x	$x^3 - x$ +188
135	152
139	92
143	609
147	585
151	404
155	450
159	356
163	506
167	533
171	70
175	252
179	712
183	332
187	247
191	90
195	245
199	345
203	23
207	414
211	400
215	365
219	693
223	266
227	219
231	185
235	548
239	190
243	246
247	349
251	132
255	730
259	274
263	650
267	740
271	177
275	96
279	130
283	663
287	577

x	$x^3 - x$ +188
136	409
140	645
144	52
148	516
152	168
156	143
160	74
164	345
168	589
172	439
176	279
180	493
184	714
188	575
192	460
196	2
200	336
204	344
208	410
212	167
216	750
220	290
224	673
228	30
232	247
236	206
240	291
244	135
248	122
252	636
256	559
260	275
264	168
268	622
272	519
276	243
280	178
284	708
288	715

x	$x^3 - x$ +188
289	328
293	409
297	80
301	476
305	479
309	473
313	91
317	468
321	486
325	529
329	230
333	724
337	142
341	370
345	290
349	286
353	742
357	540
361	64
365	449
369	577
373	81
377	96
381	255
385	191
389	288
393	179
397	248
401	128
405	203
409	106
413	221
417	181
421	370
425	421
429	718
433	143
437	582
441	166

x	$x^3 - x$ +188
290	173
294	491
298	495
302	569
306	346
310	210
314	545
318	233
322	409
326	706
330	6
334	195
338	155
342	270
346	173
350	248
354	128
358	197
362	88
366	185
370	121
374	280
378	295
382	550
386	678
390	312
394	587
398	385
402	90
406	86
410	6
414	234
418	403
422	146
426	598
430	641
434	659
438	285
442	654

x	$x^3 - x$ +188
291	256
295	84
299	445
303	221
307	547
311	305
315	630
319	404
323	11
327	586
331	260
335	168
339	694
343	720
347	630
351	57
355	136
359	500
363	31
367	615
371	383
375	470
379	509
383	133
387	477
391	423
395	355
399	657
403	211
407	152
411	113
415	478
419	129
423	201
427	327
431	140
435	24
439	363
443	39

x	$x^3 - x$ +188
292	583
296	696
300	687
304	189
308	337
312	13
316	352
320	236
324	49
328	175
332	247
336	649
340	263
344	224
348	165
352	470
356	21
360	704
364	650
368	243
372	618
376	657
380	744
384	512
388	345
392	627
396	240
400	319
404	497
408	407
412	433
416	208
420	116
424	541
428	365
432	723
436	497
440	71
444	580

x	$x^3 - x$ +188
445	30
449	558
453	632
457	636
461	203
465	468
469	313
473	122
477	279
481	417
485	169
489	670
493	51
497	198
501	744
505	571
509	63
513	355
517	329
521	369
525	108
529	681
533	219
537	608
541	730
545	218
549	207
553	330
557	220
561	261
565	86
569	79
573	624
577	603
581	400
585	399
589	233
593	286
597	191

x	$x^3 - x$ +188
446	648
450	651
454	296
458	718
462	48
466	172
470	723
474	583
478	136
482	517
486	608
490	42
494	705
498	728
502	495
506	390
510	46
514	598
518	177
522	669
526	205
530	671
534	198
538	672
542	224
546	740
550	351
554	192
558	647
562	598
566	429
570	524
574	516
578	38
582	225
586	710
590	375
594	355
598	283

x	$x^3 - x$ +188
447	187
451	440
455	431
459	544
463	412
467	419
471	198
475	133
479	608
483	505
487	208
491	101
495	568
499	491
503	254
507	241
511	85
515	170
519	129
523	346
527	454
531	86
535	377
539	209
543	717
547	32
551	40
555	374
559	667
563	552
567	413
571	634
575	97
579	688
583	538
587	31
591	302
595	233
599	208

x	$x^3 - x$ +188
448	155
452	682
456	292
460	120
464	550
468	464
472	246
476	280
480	199
484	387
488	477
492	102
496	397
500	244
504	27
508	130
512	186
516	579
520	191
524	157
528	110
532	434
536	11
540	727
544	713
548	353
552	31
556	131
560	286
564	129
568	44
572	415
576	124
580	306
584	594
588	621
592	20
596	677
600	723

x	$x^3 - x$ +188	x	$x^3 - x$ +188	x	$x^3 - x$ +188	x	$x^3 - x$ +188
601	332	602	543	603	611	604	542
605	342	606	17	607	324	608	518
609	605	610	591	611	482	612	284
613	3	614	396	615	718	616	224
617	422	618	567	619	665	620	722
621	744	622	737	623	707	624	660
625	602	626	539	627	477	628	422
629	380	630	357	631	359	632	392
633	462	634	575	635	737	636	203
637	481	638	75	639	493	640	239
641	70	642	743	643	11	644	133
645	364	646	710	647	426	648	269
649	245	650	360	651	620	652	280
653	97	654	77	655	226	656	550
657	304	658	245	659	379	660	712
661	499	662	497	663	712	664	399
665	315	666	466	667	107	668	746
669	136	670	536	671	450	672	635
673	346	674	340	675	623	676	450
677	578	678	262	679	259	680	575
681	465	682	686	683	493	684	643
685	391	686	494	687	207	688	287
689	740	690	70	691	536	692	642
693	394	694	549	695	362	696	590
697	488	698	62	699	69	700	515
701	655	702	495	703	41	704	50
705	528	706	730	707	662	708	330
709	491	710	400	711	63	712	237
713	177	714	640	715	130	716	155
717	721	718	332	719	496	720	468
721	254	722	611	723	43	724	58
725	662	726	359	727	657	728	60
729	76	730	711	731	469	732	107
733	382	734	549	735	614	736	583
737	462	738	257	739	725	740	370
741	700	742	219	743	435	744	603
745	729	746	68	747	128	748	164
749	182	750	188	0	188		

Tabla 2

y	y^2	y	y^2	y	y^2	y	y^2
1	1	2	4	3	9	4	16
5	25	6	36	7	49	8	64
9	81	10	100	11	121	12	144
13	169	14	196	15	225	16	256
17	289	18	324	19	361	20	400
21	441	22	484	23	529	24	576
25	625	26	676	27	729	28	33
29	90	30	149	31	210	32	273
33	338	34	405	35	474	36	545
37	618	38	693	39	19	40	98
41	179	42	262	43	347	44	434
45	523	46	614	47	707	48	51
49	148	50	247	51	348	52	451
53	556	54	663	55	21	56	132
57	245	58	360	59	477	60	596
61	717	62	89	63	214	64	341
65	470	66	601	67	734	68	118
69	255	70	394	71	535	72	678
73	72	74	219	75	368	76	519
77	672	78	76	79	233	80	392
81	553	82	716	83	130	84	297
85	466	86	637	87	59	88	234
89	411	90	590	91	20	92	203
93	388	94	575	95	13	96	204
97	397	98	592	99	38	100	237
101	438	102	641	103	95	104	302
105	511	106	722	107	184	108	399
109	616	110	84	111	305	112	528
113	2	114	229	115	458	116	689
117	171	118	406	119	643	120	131
121	372	122	615	123	109	124	356
125	605	126	105	127	358	128	613
129	119	130	378	131	639	132	151
133	416	134	683	135	201	136	472

y	y^2
137	745
141	355
145	748
149	422
153	128
157	617
161	387
165	189
169	23
173	640
177	538
181	468
185	430
189	424
193	450
197	508
201	598
205	720
209	123
213	309
217	527
221	26
225	308
229	622
233	217
237	595
241	254
245	696
249	419
253	174
257	712
261	531
265	382
269	265
273	180
277	127
281	106
285	117

y	y^2
138	269
142	638
146	288
150	721
154	435
158	181
162	710
166	520
170	362
174	236
178	142
182	80
186	50
190	52
194	86
198	152
202	250
206	380
210	542
214	736
218	211
222	469
226	8
230	330
234	684
238	319
242	737
246	436
250	167
254	681
258	476
262	303
266	162
270	53
274	727
278	682
282	669
286	688

y	y^2
139	546
143	172
147	581
151	271
155	744
159	498
163	284
167	102
171	703
175	585
179	499
183	445
187	423
191	433
195	475
199	549
203	655
207	42
211	212
215	414
219	648
223	163
227	461
231	40
235	402
239	45
243	471
247	178
251	668
255	439
259	242
263	77
267	695
271	594
275	525
279	488
283	483
287	510

y	y^2
140	74
144	459
148	125
152	574
156	304
160	66
164	611
168	437
172	295
176	185
180	107
184	61
188	47
192	65
196	115
200	197
204	311
208	457
212	635
216	94
220	336
224	610
228	165
232	503
236	122
240	524
244	207
248	673
252	420
256	199
260	10
264	604
268	479
272	386
276	325
280	296
284	299
288	334

y	y^2
289	160
293	235
297	342
301	481
305	652
309	104
313	339
317	606
321	154
325	485
329	97
333	492
337	168
341	627
345	367
349	139
353	694
357	530
361	398
365	298
369	230
373	194
377	190
381	218
385	278
389	370
393	494
397	650
401	87
405	307
409	559
413	92
417	408
421	5
425	385
429	46
433	490
437	215
441	723

y	y^2
290	739
294	71
298	186
302	333
306	512
310	723
314	215
318	490
322	46
326	385
330	5
334	408
338	92
342	559
346	307
350	87
354	650
358	494
362	370
366	278
370	218
374	190
378	194
382	230
386	298
390	398
394	530
398	694
402	139
406	367
410	627
414	168
418	492
422	97
426	485
430	154
434	606
438	339
442	104

y	y^2
291	569
295	660
299	32
303	187
307	374
311	593
315	93
319	376
323	691
327	287
331	666
335	326
339	18
343	493
347	249
351	37
355	608
359	460
363	344
367	260
371	208
375	188
379	200
383	244
387	320
391	428
395	568
399	740
403	193
407	429
411	697
415	246
419	578
423	191
427	587
431	264
435	724
439	465
443	238

y	y^2
292	401
296	500
300	631
304	43
308	238
312	465
316	724
320	264
324	587
328	191
332	578
336	246
340	697
344	429
348	193
352	740
356	568
360	428
364	320
368	244
372	200
376	188
380	208
384	260
388	344
392	460
396	608
400	37
404	249
408	493
412	18
416	326
420	666
424	287
428	691
432	376
436	93
440	593
444	374

$\mathbb{F}_{751}$	
y	y^2
445	512
449	333
453	186
457	71
461	739
465	688
469	669
473	682
477	727
481	53
485	162
489	303
493	476
497	681
501	167
505	436
09	737
513	319
517	684
521	330
525	8
529	469
533	211
537	736
541	542
545	380
549	250
553	152
557	86
561	52
565	50
569	80
573	142
577	236
581	362
585	520
589	710
593	181

$\mathbb{F}_{751}$	
y	y^2
446	652
450	481
454	342
458	235
462	160
466	117
470	106
474	127
478	180
482	265
486	382
490	531
494	712
498	174
502	419
506	696
510	254
514	595
518	217
522	622
526	308
530	26
534	527
538	309
542	123
546	720
550	598
554	508
558	450
562	424
566	430
570	468
574	538
578	640
582	23
586	189
590	387
594	617

$\mathbb{F}_{751}$	
y	y^2
447	43
451	631
455	500
459	401
463	334
467	299
471	296
475	325
479	386
483	479
487	604
491	10
495	199
499	420
503	673
507	207
511	524
515	122
519	503
523	165
527	610
531	336
535	94
539	635
543	457
547	311
551	197
555	115
559	65
563	47
567	61
571	107
575	185
579	295
583	437
587	611
591	66
595	304

$\mathbb{F}_{751}$	
y	y^2
448	187
452	32
456	660
460	569
464	510
468	483
472	488
476	525
480	594
484	695
488	77
492	242
496	439
500	668
504	178
508	471
512	45
516	402
520	40
524	461
528	163
532	648
536	414
540	212
544	42
548	655
552	549
556	475
560	433
564	423
568	445
572	499
576	585
580	703
584	102
588	284
592	498
596	744

y	y^2
597	435
601	721
605	288
609	638
613	269
617	683
621	378
625	105
629	615
633	406
637	229
641	84
645	722
649	641
653	592
657	575
661	590
665	637
669	716
673	76
677	219
681	394
685	601
689	89
693	360
697	663
701	247
705	614
709	262
713	693
717	405
721	149
725	676
729	484
733	324
737	196
741	100
745	36
749	4

y	y^2
598	128
602	422
606	748
610	355
614	745
618	416
622	119
626	605
630	372
634	171
638	2
642	616
646	511
650	438
654	397
658	388
662	411
666	466
670	553
674	672
678	72
682	255
686	470
690	717
694	245
698	556
702	148
706	523
710	179
714	618
718	338
722	90
726	625
730	441
734	289
738	169
742	81
746	25
750	1

y	y^2
599	574
603	125
607	459
611	74
615	472
619	151
623	613
627	356
631	131
635	689
639	528
643	399
647	302
651	237
655	204
659	203
663	234
667	297
671	392
675	519
679	678
683	118
687	341
691	596
695	132
699	451
703	51
707	434
711	98
715	545
719	273
723	33
727	576
731	400
735	256
739	144
743	64
747	16
0	0

y	y^2
600	271
604	581
608	172
612	546
616	201
620	639
624	358
628	109
632	643
636	458
640	305
644	184
648	95
652	38
656	13
660	20
664	59
668	130
672	233
676	368
680	535
684	734
688	214
692	477
696	21
700	348
704	707
708	347
712	19
716	474
720	210
724	729
728	529
732	361
736	225
740	121
744	49
748	9

Tabla 3

#	x	y
1	1	375
4	2	378
7	5	225
10	6	390
13	12	235
16	13	622
19	18	137
22	19	613
25	24	65
28	26	439
31	30	236
34	32	451
37	36	336
40	38	495
43	41	274
46	43	429
49	45	97
52	47	416
55	52	346
58	54	620
61	59	325
64	62	590
67	64	13
70	66	537
73	69	21
76	72	466
79	77	6
82	79	418
85	86	184
88	92	606
91	94	73
94	97	467
97	102	120
100	103	571

#	x	y
2	1	376
5	3	211
8	5	526
11	7	240
14	12	516
17	17	332
20	18	614
23	21	133
26	24	686
29	28	302
32	30	515
35	34	118
38	36	415
41	39	349
44	41	477
47	44	312
50	45	654
53	50	136
56	52	405
59	57	332
62	59	426
65	63	62
68	64	738
71	67	22
74	69	730
77	74	199
80	77	745
83	84	138
86	86	567
89	93	120
92	94	678
95	101	4
98	102	631
101	121	39

#	x	y
3	2	373
6	3	540
9	6	361
12	7	511
15	13	129
18	17	419
21	19	138
24	21	618
27	26	312
30	28	449
33	32	300
36	34	633
39	38	256
42	39	402
45	43	322
48	44	439
51	47	335
54	50	615
57	54	131
60	57	419
63	62	161
66	63	689
69	66	214
72	67	729
75	72	285
78	74	552
81	79	333
84	84	613
87	92	145
90	93	631
93	97	284
96	101	747
99	103	180
102	121	712

#	x	y
103	124	354
106	126	476
109	131	34
112	135	553
115	144	190
118	147	576
121	153	315
124	154	575
127	157	55
130	158	722
133	160	140
136	161	750
139	170	274
142	172	496
145	174	240
148	177	587
151	179	257
154	180	408
157	187	50
160	188	657
163	190	196
166	191	722
169	195	57
172	196	638
175	198	322
178	200	531
181	202	13
184	203	582
187	205	161
190	207	536
193	210	97
196	211	731
199	214	76
202	217	504
205	224	248
208	225	634
211	229	115
214	231	575
217	233	256

#	x	y
104	124	397
107	128	252
110	131	717
113	139	338
116	144	561
119	152	337
122	153	436
125	155	193
128	157	696
131	159	124
134	160	611
137	169	132
140	170	477
143	173	33
146	174	511
149	178	232
152	179	494
155	182	84
158	187	701
161	189	23
164	190	555
167	192	359
170	195	694
173	197	107
176	198	429
179	201	5
182	202	738
185	204	363
188	205	590
191	209	198
194	210	654
197	212	250
200	214	675
203	219	38
206	224	503
209	227	74
212	229	636
215	232	50
218	233	495

#	x	y
105	126	275
108	128	499
111	135	198
114	139	413
117	147	175
120	152	414
123	154	176
126	155	558
129	158	29
132	159	627
135	161	1
138	169	619
141	172	255
144	173	718
147	177	164
150	178	519
153	180	343
156	182	667
159	188	94
162	189	728
165	191	29
168	192	392
171	196	113
174	197	644
177	200	220
180	201	746
183	203	169
186	204	388
189	207	215
192	209	553
195	211	20
198	212	501
201	217	247
204	219	713
207	225	117
210	227	677
213	231	176
216	232	701
219	234	188

#	x	y
220	234	563
223	238	55
226	239	377
229	243	336
232	245	509
235	248	236
238	251	695
241	254	247
244	256	409
247	258	276
250	261	463
253	263	354
256	264	414
259	266	244
262	267	399
265	269	224
268	270	589
271	274	329
274	278	510
277	280	247
280	283	697
283	291	16
286	295	641
289	297	182
292	299	568
295	302	291
298	304	586
301	310	31
304	311	640
307	314	36
310	317	570
313	320	174
316	324	744
319	329	369
322	331	384
325	333	316
328	335	414
331	339	353
334	341	389

#	x	y
221	237	23
224	238	696
227	241	230
230	243	415
233	246	53
236	248	515
239	253	108
242	254	504
245	257	149
248	258	475
251	262	208
254	263	397
257	265	76
260	266	507
263	268	229
266	269	527
269	272	76
272	274	422
275	279	83
278	280	504
281	285	96
284	291	735
287	296	245
290	297	569
293	301	258
296	302	460
299	305	268
302	310	720
305	312	95
308	314	715
311	318	79
314	320	577
317	325	23
320	329	382
323	332	50
326	333	435
329	337	178
332	339	398
335	343	205

#	x	y
222	237	728
225	239	374
228	241	521
231	245	242
234	246	698
237	251	56
240	253	643
243	256	342
246	257	602
249	261	288
252	262	543
255	264	337
258	265	675
261	267	352
264	268	522
267	270	162
270	272	675
273	278	241
276	279	668
279	283	54
282	285	655
285	295	110
288	296	506
291	299	183
294	301	493
297	304	165
300	305	483
303	311	111
306	312	656
309	317	181
312	318	672
315	324	7
318	325	728
321	331	367
324	332	701
327	335	337
330	337	573
333	341	362
336	343	546

#	x	y
337	348	228
340	352	686
343	356	55
346	358	551
349	361	8
352	364	397
355	367	122
358	370	740
361	373	9
364	375	686
367	380	155
370	381	682
373	385	328
376	386	679
379	389	146
382	391	564
385	393	41
388	394	427
391	398	326
394	400	513
397	402	29
400	403	533
403	406	194
406	407	553
409	412	191
412	414	663
415	417	158
418	421	389
421	426	201
424	430	649
427	440	294
430	446	532
433	452	278
436	454	471
439	465	181
442	466	608
445	470	310
448	472	415
451	479	355

#	x	y
338	348	523
341	354	153
344	356	696
347	359	296
350	361	743
353	366	176
356	367	629
359	372	37
362	373	742
365	378	172
368	380	596
371	384	306
374	385	423
377	387	59
380	389	605
383	392	341
386	393	710
389	395	141
392	398	425
395	401	153
398	402	722
401	405	92
404	406	557
407	409	281
410	412	560
413	416	371
416	417	593
419	423	135
422	426	550
425	432	310
428	440	457
431	447	303
434	452	473
437	461	92
440	465	570
443	467	249
446	470	441
449	473	236
452	479	396

#	x	y
339	352	65
342	354	598
345	358	200
348	359	455
351	364	354
354	366	575
357	370	11
360	372	714
363	375	65
366	378	579
369	381	69
372	384	445
375	386	72
378	387	692
381	391	187
384	392	410
387	394	324
390	395	610
393	400	238
396	401	598
399	403	218
402	405	659
405	407	198
408	409	470
411	414	88
414	416	380
417	421	362
420	423	616
423	430	102
426	432	441
429	446	219
432	447	448
435	454	280
438	461	659
441	466	143
444	467	502
447	472	336
450	473	515
453	480	256

#	x	y
454	480	495
457	485	13
460	486	396
463	488	59
466	490	544
469	493	48
472	495	395
475	500	368
478	501	596
481	508	83
484	510	429
487	513	141
490	514	550
493	522	282
496	529	497
499	532	44
502	533	677
505	538	77
508	540	477
511	545	370
514	546	399
517	549	244
520	551	520
523	555	307
526	556	631
529	565	194
532	566	407
535	575	329
538	578	652
541	581	20
544	582	736
547	584	271
550	585	643
553	589	79
556	591	647
559	594	141
562	595	672
565	599	371
568	600	441

#	x	y
455	484	161
458	485	738
461	487	371
464	488	692
467	492	167
470	493	703
473	496	97
476	500	383
479	503	241
482	508	668
485	512	298
488	513	610
491	520	328
494	522	469
497	531	194
500	532	707
503	537	355
506	538	674
509	543	61
512	545	381
515	547	299
518	549	507
521	553	230
524	555	444
527	562	201
530	565	557
533	570	240
536	575	422
539	579	286
542	581	731
545	583	177
548	584	480
551	586	162
554	589	672
557	592	91
560	594	610
563	597	328
566	599	380
569	603	164

#	x	y
456	484	590
459	486	355
462	487	380
465	490	207
468	492	584
471	495	356
474	496	654
477	501	155
480	503	510
483	510	322
486	512	453
489	514	201
492	520	423
495	529	254
498	531	557
501	533	74
504	537	396
507	540	274
510	543	690
513	546	352
516	547	452
519	551	231
522	553	521
525	556	120
528	562	550
531	566	344
534	570	511
537	578	99
540	579	465
543	582	15
546	583	574
549	585	108
552	586	589
555	591	104
558	592	660
561	595	79
564	597	423
567	600	310
570	603	587

#	x	y
571	604	210
574	605	454
577	609	125
580	612	588
583	620	106
586	621	596
589	623	47
592	624	456
595	628	149
598	629	545
601	634	94
604	635	509
607	637	301
610	639	408
613	648	138
616	649	694
619	653	329
622	654	488
625	658	57
628	660	494
631	663	257
634	664	643
637	667	180
640	671	558
643	676	193
646	677	419
649	680	94
652	681	439
655	684	119
658	686	393
661	688	327
664	689	399
667	694	199
670	695	581
673	697	279
676	701	548
679	705	112
682	708	521

#	x	y
572	604	541
575	607	18
578	609	626
581	617	149
584	620	645
587	622	242
590	623	704
593	627	59
596	628	602
599	632	80
602	634	657
605	636	92
608	637	450
611	646	162
614	648	613
617	650	58
620	653	422
623	657	156
626	658	694
629	661	179
632	663	494
635	666	85
638	667	571
641	672	212
644	676	558
647	678	42
650	680	657
653	683	343
656	684	632
659	687	244
662	688	424
665	693	70
668	694	552
671	696	90
674	697	472
677	704	186
680	705	639
683	710	20

#	x	y
573	605	297
576	607	733
579	612	163
582	617	602
585	621	155
588	622	509
591	624	295
594	627	692
597	629	206
600	632	671
603	635	242
606	636	659
609	639	343
612	646	589
615	649	57
618	650	693
621	654	263
624	657	595
627	660	257
630	661	572
633	664	108
636	666	666
639	671	193
642	672	539
645	677	332
648	678	709
651	681	312
654	683	408
657	686	358
660	687	507
663	689	352
666	693	681
669	695	170
672	696	661
675	701	203
678	704	565
681	708	230
684	710	731

#	x	y
685	712	100
688	714	578
691	717	150
694	720	570
697	722	164
700	723	447
703	731	222
706	732	571
709	734	199
712	735	705
715	742	74
718	743	597
721	747	153
724	750	376
727	∞	∞

#	x	y
686	712	651
689	715	83
692	717	601
695	721	241
698	722	587
701	729	78
704	731	529
707	733	265
710	734	552
713	740	362
716	742	677
719	745	27
722	747	598
725	751	375

#	x	y
687	714	173
690	715	668
693	720	181
696	721	510
699	723	304
702	729	673
705	732	180
708	733	486
711	735	46
714	740	389
717	743	154
720	745	724
723	750	375
726	751	376

Tabla 4

Unidad de mensaje	Punto en E
A	(0,375)
C	(41,274)
E	(84,138)
G	(121,39)
I	(160,140)
K	(200,220)
M	(241,230)
Ñ	(280,247)
P	(320,174)
R	(361,8)
T	(400,238)
V	(440,294)
X	(480,256)
Z	(520,328)

Unidad de mensaje	Punto en E
B	(21,133)
D	(62,161)
F	(101,4)
H	(144,190)
J	(180,343)
L	(224,248)
N	(261,288)
O	(301,258)
Q	(341,362)
S	(380,155)
U	(421,362)
W	(461,92)
Y	(500,368)

Tabla 5

$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(\ 97\ ,\ 215)$	$-6P$
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(\ 57\ ,\ 332)$	$-5P$
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(121\ ,\ 39)$	$-4P$
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(\ 6\ \ ,\ 390)$	$-3P$
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(\ 2\ \ ,\ 373)$	$-2P$
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(\ 1\ \ ,\ 376)$	$-P$
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(\ \infty\ \ ,\ \infty)$	0
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(\ 1\ \ ,\ 375)$	P
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(\ 2\ \ ,\ 378)$	$2P$
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(\ 6\ \ ,\ 361)$	$3P$
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(121\ ,\ 712)$	$4P$
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(\ 57\ ,\ 419)$	$5P$
$R-4Q$	$700P=(205\ ,\ 161)$	$\neq$	$(\ 97\ ,\ 129)$	$6P$

$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(\ 97\ ,\ 215)$	$-6P$
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(\ 57\ ,\ 332)$	$-5P$
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(121\ ,\ 39)$	$-4P$
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(\ 6\ \ ,\ 390)$	$-3P$
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(\ 2\ \ ,\ 373)$	$-2P$
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(\ 1\ \ ,\ 376)$	$-P$
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(\ \infty\ \ ,\ \infty)$	0
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(\ 1\ \ ,\ 375)$	P
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(\ 2\ \ ,\ 378)$	$2P$
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(\ 6\ \ ,\ 361)$	$3P$
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(121\ ,\ 712)$	$4P$
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(\ 57\ ,\ 419)$	$5P$
$R-3Q$	$713P=(734\ ,\ 552)$	$\neq$	$(\ 97\ ,\ 129)$	$6P$

$R-2Q$	$726P = (1, 376)$	$\neq$	$(97, 215)$	$-6P$
$R-2Q$	$726P = (1, 376)$	$\neq$	$(57, 332)$	$-5P$
$R-2Q$	$726P = (1, 376)$	$\neq$	$(121, 39)$	$-4P$
$R-2Q$	$726P = (1, 376)$	$\neq$	$(6, 390)$	$-3P$
$R-2Q$	$726P = (1, 376)$	$\neq$	$(2, 373)$	$-2P$
$R-2Q$	$726P = (1, 376)$	$=$	$(1, 376)$	$-P$
$R-2Q$	$726P = (1, 376)$	$\neq$	(∞, ∞)	0
$R-2Q$	$726P = (1, 376)$	$\neq$	$(1, 375)$	P
$R-2Q$	$726P = (1, 376)$	$\neq$	$(2, 378)$	$2P$
$R-2Q$	$726P = (1, 376)$	$\neq$	$(6, 361)$	$3P$
$R-2Q$	$726P = (1, 376)$	$\neq$	$(121, 712)$	$4P$
$R-2Q$	$726P = (1, 376)$	$\neq$	$(57, 419)$	$5P$
$R-2Q$	$726P = (1, 376)$	$\neq$	$(97, 129)$	$6P$

$R-1Q$	$739P = (180, 408)$	$\neq$	$(57, 332)$	$-5P$
$R-1Q$	$739P = (180, 408)$	$\neq$	$(121, 39)$	$-4P$
$R-1Q$	$739P = (180, 408)$	$\neq$	$(6, 390)$	$-3P$
$R-1Q$	$739P = (180, 408)$	$\neq$	$(2, 373)$	$-2P$
$R-1Q$	$739P = (180, 408)$	$\neq$	$(1, 376)$	$-P$
$R-1Q$	$739P = (180, 408)$	$\neq$	(∞, ∞)	0
$R-1Q$	$739P = (180, 408)$	$\neq$	$(1, 375)$	P
$R-1Q$	$739P = (180, 408)$	$\neq$	$(2, 378)$	$2P$
$R-1Q$	$739P = (180, 408)$	$\neq$	$(6, 361)$	$3P$
$R-1Q$	$739P = (180, 408)$	$\neq$	$(121, 712)$	$4P$
$R-1Q$	$739P = (180, 408)$	$\neq$	$(57, 419)$	$5P$
$R-1Q$	$739P = (180, 408)$	$\neq$	$(97, 129)$	$6P$

$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(\ 97\ ,\ 215)$	$-6P$
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(\ 57\ ,\ 332)$	$-5P$
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(121\ ,\ 39)$	$-4P$
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(\ 6\ \ ,\ 390)$	$-3P$
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(\ 2\ \ ,\ 373)$	$-2P$
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(\ 1\ \ ,\ 376)$	$-P$
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(\ \infty\ \ ,\ \infty)$	0
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(\ 1\ \ ,\ 375)$	P
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(\ 2\ \ ,\ 378)$	$2P$
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(\ 6\ \ ,\ 361)$	$3P$
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(121\ ,\ 712)$	$4P$
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(\ 57\ ,\ 419)$	$5P$
$R-0Q$	$752P=(296\ ,\ 245)$	$\neq$	$(\ 97\ ,\ 129)$	$6P$

$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(\ 57\ ,\ 332)$	$-5P$
$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(121\ ,\ 39)$	$-4P$
$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(\ 6\ \ ,\ 390)$	$-3P$
$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(\ 2\ \ ,\ 373)$	$-2P$
$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(\ 1\ \ ,\ 376)$	$-P$
$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(\ \infty\ \ ,\ \infty)$	0
$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(\ 1\ \ ,\ 375)$	P
$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(\ 2\ \ ,\ 378)$	$2P$
$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(\ 6\ \ ,\ 361)$	$3P$
$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(121\ ,\ 712)$	$4P$
$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(\ 57\ ,\ 419)$	$5P$
$R+1Q$	$765P=(484\ ,\ 590)$	$\neq$	$(\ 97\ ,\ 129)$	$6P$

$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(\ 97\ ,\ 215)$	$-6P$
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(\ 57\ ,\ 332)$	$-5P$
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(121\ ,\ 39)$	$-4P$
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(\ 6\ \ ,\ 390)$	$-3P$
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(\ 2\ \ ,\ 373)$	$-2P$
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(\ 1\ \ ,\ 376)$	$-P$
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(\ \infty\ \ ,\ \infty)$	0
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(\ 1\ \ ,\ 375)$	P
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(\ 2\ \ ,\ 378)$	$2P$
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(\ 6\ \ ,\ 361)$	$3P$
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(121\ ,\ 712)$	$4P$
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(\ 57\ ,\ 419)$	$5P$
$R+2Q$	$778P = (508\ ,\ 668)$	$\neq$	$(\ 97\ ,\ 129)$	$6P$

$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(\ 57\ ,\ 332)$	$-5P$
$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(121\ ,\ 39)$	$-4P$
$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(\ 6\ \ ,\ 390)$	$-3P$
$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(\ 2\ \ ,\ 373)$	$-2P$
$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(\ 1\ \ ,\ 376)$	$-P$
$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(\ \infty\ \ ,\ \infty)$	0
$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(\ 1\ \ ,\ 375)$	P
$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(\ 2\ \ ,\ 378)$	$2P$
$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(\ 6\ \ ,\ 361)$	$3P$
$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(121\ ,\ 712)$	$4P$
$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(\ 57\ ,\ 419)$	$5P$
$R+3Q$	$791P = (720\ ,\ 570)$	$\neq$	$(\ 97\ ,\ 129)$	$6P$

$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(\ 97\ ,\ 215)$	$-6P$
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(\ 57\ ,\ 332)$	$-5P$
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(121\ ,\ 39)$	$-4P$
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(\ 6\ \ ,\ 390)$	$-3P$
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(\ 2\ \ ,\ 373)$	$-2P$
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(\ 1\ \ ,\ 376)$	$-P$
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(\ \infty\ \ ,\ \infty)$	0
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(\ 1\ \ ,\ 375)$	P
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(\ 2\ \ ,\ 378)$	$2P$
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(\ 6\ \ ,\ 361)$	$3P$
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(121\ ,\ 712)$	$4P$
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(\ 57\ ,\ 419)$	$5P$
$R+4Q$	$804P = (47\ ,\ 335)$	$\neq$	$(\ 97\ ,\ 129)$	$6P$

Anexo 4

Resumen de Criptografía de Curvas Elípticas

Introducción

Las curvas elípticas son llamadas de esta forma porque se forman por ecuaciones cúbicas, similares a las que se utilizan para calcular la circunferencia de una elipse. En general las ecuaciones cúbicas que definen una curva elíptica tiene la forma:

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

Para aplicaciones con números primos se utiliza la ecuación:

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

Las curvas con $4a^3+27b^2=0$ no se utilizan para criptografía ya que son supersingulares y efectúan operaciones sobre un punto especial llamado punto cero o punto infinito.

Descripción algebraica de la adición en los números reales

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:

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

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

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

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

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

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

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

Descripción algebraica de la adición en los números primos

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 $R=2P=P+P$.

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

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

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

Obtención de múltiplos de puntos

Se utiliza 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 utilizamos el punto cero $P = (0,0)$ como inicio y para cada $i = n-1, n-2, ..., 1, 0$ hacemos lo siguiente:

1. *Hacemos $P = 2P$, es decir, se obtiene la multiplicación escalar por dos del punto.*
2. *Si $k_i = 1$ entonces hacemos $P = P + G$.*
3. *Si k_i es igual a cero regresamos al paso 1, pero si i es igual a cero se tiene la suma del punto k veces.*

Intercambio de claves secretas

Para realizar el intercambio de claves utilizando las curvas elípticas, se parte de una curva elíptica $E_p(a, b)$ definida sobre GF(p) con p un número primo y de un punto base $G = (x_1, y_1)$ de la misma curva. Los parámetros $E_p(a, b)$ 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 $K = n_B * P_A$.*

Codificación y decodificación en curvas elípticas

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 p > M*h, siendo p un número primo y GF(p) 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,..., 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(p). Para cada x que se obtenga se calcula el valor de la ecuación de la curva elíptica que se escoja que pertenece a GF(p). 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".*

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

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

Método de cifrado ElGamal con curvas elípticas

En este caso, 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.

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) = (P_a, P+aP_b) = (P_a, P+abG)

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:

$$P = N–bM = P + abG – baG$$

***Tablas** para la curva elíptica $y^2=x^3+4x+6$, definida sobre el campo GF(7) y el punto base G=(1, 2).*

w	w^{-1}
0	-
1	1
2	4
3	5
4	2
5	3
6	6

x	x^3+4x+6
0	6
1	4
2	1
3	3
4	2
5	4
6	1

y	y^2
0	0
1	1
2	4
3	2
4	2
5	4
6	1

Anexo 5

Ejemplos de Criptografía de Curvas Elípticas con números de un dígito

Descripción algebraica de la adición en los números reales

En 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)?

$$\Delta = \frac{y_P - y_Q}{x_P - x_Q} = \frac{-4 - 0}{0 - 1} = 4$$

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

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

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

En 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)?

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

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

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

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

Descripción algebraica de la adición en los números primos

Con la curva elíptica $y^2 = x^3 + x + 7$ sobre GF(7) ¿Cuál sería la suma P+Q si P=(3,4) y Q=(1,3)?

$$\lambda = \left[\frac{y_P - y_Q}{x_P - x_Q}\right] mod\ p = \left[\frac{4-3}{3-1}\right] mod\ 7 = \left[\frac{1}{2}\right] mod\ 7 = [1 * 2^{-1}] mod\ 7$$
$$= [1 * 4] mod\ 7 = 4$$

$$x_R = [\lambda^2 - x_P - x_Q] mod\ p = [4^2 - 3 - 1] mod\ 7 = 12\ mod\ 7 = 5$$

$$y_R = [-y_P + \lambda(x_P - x_R)] mod\ p = [-4 + 4(3-5)] mod\ 7$$
$$= -12\ mod\ 7 = 2$$

Por lo tanto P+Q = R = (5, 2).

Con la curva elíptica $y^2 = x^3 + 4x + 6$ sobre GF(7) ¿Cuál sería la multiplicación 2P, si P=(1,2)?

$$\lambda - \left[\frac{3x_P^2 + a}{2 * y_P}\right] mod\ p - \left[\frac{3 * 1^2 + (4)}{2 * 2}\right] mod\ 7 - \left[\frac{7}{4}\right] mod\ 7$$
$$= [0 * 4^{-1}] mod\ 7 = [0 * 2] mod\ 7 = 0$$

$$x_R = [\lambda^2 - 2 * x_P] mod\ p = [0^2 - 2(1)] mod\ 7 = -2\ mod\ 7 = 5$$

$$y_R = [-y_P + \lambda(x_P - x_R)] mod\ 7 = [-2 + 0 * (1-5)] mod\ 7$$
$$= -2\ mod\ 7 = 5$$

Con la curva elíptica $y^2 = x^3 + 4x + 6$ sobre GF(7), encontrar la suma de los puntos (1,2) y (5,5).

$$\lambda = \left[\frac{y_P - y_Q}{x_P - x_Q}\right] mod\ p = \left[\frac{2-5}{1-5}\right] mod\ 7 = \left[\frac{-3}{-4}\right] mod\ 7 = \left[\frac{4}{3}\right] mod\ 7$$
$$= [4 * 3^{-1}] mod\ 7 = [4 * 5] mod\ 7 = 20\ mod\ 7 = 6$$

$$x_R = [\lambda^2 - x_P - x_Q] mod\ p = [6^2 - 1 - 5] mod\ 7 = 30\ mod\ 7 = 2$$

$$y_R = [-y_P + \lambda(x_P - x_R)] mod\ 7 = [-2 + 6 * (1 - 2)] mod\ 7$$
$$= -8\ mod\ 7 = 6$$

(1,2) + (5,5) = R = (2, 6).

Obtención de múltiplos de puntos

Con la curva elíptica $y^2 = x^3 + 4x + 6$ sobre GF(7) y el punto generador G=(1,2) encontrar la multiplicación del punto k*G, con k=6.
Primero se procede a representar al escalar en su forma binaria:

$k=6_{10}= 110_2$

$k=k_{n-1},k_{n-2},\ldots,k_1,k_0=1,1,0$

Con base en la representación binaria del entero escalar se procede a llenar la tabla siguiente:

i	k_i	P	G
3	-	(0, 0)	0G
2	1	2(0, 0)+(1, 2)	G=(1, 2)
1	1	2(1, 2)+(1, 2)	3G=(2, 6)
0	0	2(2, 6)	6G=(4, 3)

Con las operaciones siguientes:

Para 2P, si P=(1,2)

$$\lambda = \left[\frac{3x_P^2 + a}{2 * y_P}\right] mod\ p = \left[\frac{3 * 1^2 + (4)}{2 * 2}\right] mod\ 7 = \left[\frac{7}{4}\right] mod\ 7$$
$$= [0 * 4^{-1}] mod\ 7 = [0 * 2] mod\ 7 = 0$$

$$x_R = [\lambda^2 - 2 * x_P] mod\ p = [0^2 - 2(1)] mod\ 7 = -2\ mod\ 7 = 5$$

$$y_R = [-y_P + \lambda(x_P - x_R)] mod\ 7 = [-2 + 0 * (1 - 5)] mod\ 7$$
$$= -2\ mod\ 7 = 5$$

Después se suma 2(1, 2) + (1, 2), es decir, (5,5)+(1, 2).

$$\lambda = \left[\frac{y_P - y_Q}{x_P - x_Q}\right] mod\ p = \left[\frac{2-5}{1-5}\right] mod\ 7 = \left[\frac{-3}{-4}\right] mod\ 7 = \left[\frac{4}{3}\right] mod\ 7$$
$$= [4 * 3^{-1}] mod\ 7 = [4 * 5] mod\ 7 = 20\ mod\ 7 = 6$$

$$x_R = [\lambda^2 - x_P - x_Q] mod\ p = [6^2 - 1 - 5] mod\ 7 = 30\ mod\ 7 = 2$$

$$y_R = [-y_P + \lambda(x_P - x_R)] mod\ 7 = [-2 + 6 * (1 - 2)] mod\ 7$$
$$= -8\ mod\ 7 = 6$$

3G = (5,5) + (1,2) = (2, 6).

Finalmente se obtiene 6G=2(2, 6)

$$\lambda = \left[\frac{3x_P^2 + a}{2 * y_P}\right] mod\ p = \left[\frac{3 * 2^2 + (4)}{2 * 6}\right] mod\ 7 = \left[\frac{16}{12}\right] mod\ 7$$
$$= \left[\frac{2}{5}\right] mod\ 7 = [2 * 5^{-1}] mod\ 7 = [2 * 3] mod\ 7 = 6$$

$$x_R = [\lambda^2 - 2 * x_P] mod\ p = [6^2 - 2(2)] mod\ 7 = 32\ mod\ 7 = 4$$

$$y_R = [-y_P + \lambda(x_P - x_R)] mod\ 7 = [-6 + 6 * (2 - 4)] mod\ 7$$
$$= -18\ mod\ 7 = 3$$

Entonces 6G=6(1, 2)=2(2, 6) = (4, 3).

Intercambio de claves secretas

Para un usuario se tiene un par de claves:
{n, n*G}
n es privada
n*G es pública

Con los parámetros públicos $y^2 = x^3 + 4x + 6$ sobre GF(7) y el punto generador G=(1,2) encontrar la clave secreta común a los usuarios A y B si:
A tiene una clave secreta n_A=3.
B tiene una clave secreta n_B=2.
Para el usuario A se tiene:

$A=\{n_A,\ n_A*G\}= \{n_A,\ P_A\}=\{\ 3,\ 3(1,\ 2)\}= \{\ 3,\ (2,6)\}$

Por su parte el usuario B tiene su par de claves:

$B=\{n_B,\ n_B*G\}= \{n_B,\ P_B\}=\{\ 2,\ 2(1,\ 2)\}= \{\ 2,\ (5,5)\}$

Entonces para obtener la clave común el usuario A multiplica su clave privada por la clave pública del usuario B:

$k=n_A*P_B=3(5,\ 5)=2(5,\ 5)+(5,\ 5)$

$$\lambda = \left[\frac{3x_P^2 + a}{2 * y_P}\right] mod\ p = \left[\frac{3 * 5^2 + (4)}{2 * 5}\right] mod\ 7 = \left[\frac{79}{10}\right] mod\ 7$$
$$= \left[\frac{2}{3}\right] mod\ 7 = [2 * 3^{-1}] mod\ 7 = [2 * 5] mod\ 7 = 10 mod\ 7$$
$$= 3$$

$$x_R = [\lambda^2 - 2 * x_P] mod\ p = [3^2 - 2(5)] mod\ 7 = -1\ mod\ 7 = 6$$

$$y_R = [-y_P + \lambda(x_P - x_R)] mod\ 7 = [-5 + 3 * (5 - 6)] mod\ 7$$
$$= -8\ mod\ 7 = 6$$

$k=n_A*P_B=3(5,\ 5)= 2(5,\ 5)+(5,\ 5)=(6,\ 6)+(5,\ 5)$

$$\lambda = \left[\frac{y_P - y_Q}{x_P - x_Q}\right] mod\ p = \left[\frac{6 - 5}{6 - 5}\right] mod\ 7 = \left[\frac{1}{1}\right] mod\ 7 = [1 * 1^{-1}] mod\ 7$$
$$= [1 * 1] mod\ 7 = 1\ mod\ 7 = 1$$

$$x_R = [\lambda^2 - x_P - x_Q] mod\ p = [1^2 - 6 - 5] mod\ 7 = -10\ mod\ 7 = 4$$

$$y_R = [-y_P + \lambda(x_P - x_R)] mod\ 7 = [-6 + 1 * (6 - 4)] mod\ 7$$
$$= -4\ mod\ 7 = 3$$

$k=n_A*P_B=3(5,\ 5)= 2(5,\ 5)+(5,\ 5)=(6,\ 6)+(5,\ 5)=(4,\ 3)$.

Por su parte el usuario B multiplica su clave privada por la clave pública del usuario A:

$k=n_B*P_A=2(2, 6)$

$$\lambda - \left[\frac{3x_P^2 + a}{2 * y_P}\right] mod\ p - \left[\frac{3 * 2^2 + (4)}{2 * 6}\right] mod\ 7 - \left[\frac{16}{12}\right] mod\ 7$$
$$= \left[\frac{2}{5}\right] mod\ 7 = [2 * 5^{-1}] mod\ 7 = [2 * 3] mod\ 7 = 6 mod\ 7$$
$$= 6$$

$$x_R = [\lambda^2 - 2 * x_P] mod\ p = [6^2 - 2(2)] mod\ 7 = 32\ mod\ 7 = 4$$

$$y_R = [-y_P + \lambda(x_P - x_R)] mod\ 7 = [-6 + 6 * (2 - 4)] mod\ 7$$
$$= -18\ mod\ 7 = 3$$

$k=n_B*P_A= 2(2, 6) =(4, 3)$.

Se puede demostrar entonces que ambas claves son iguales y los usuarios A y B solamente tomarán en cuenta la coordenada "x" o la coordenada "y" del punto encontrado para utilizarla como clave común.

Codificación en curvas elípticas

Para codificar un mensaje de modo que se obtenga puntos de una curva elíptica se hace lo siguiente:

Utilizando una curva elíptica $y^2 = x^3 + 4x + 6$ sobre GF(7) y tomando en cuenta la regla p>M*h donde:

M=6 el cual representa el tamaño del alfabeto codificado
h=1 el número de iteraciones

Con el siguiente alfabeto:

A	E	I	O	U
1	2	3	4	5

j=1, 2,…, h-1

A:1 para j=1

x=A+j=1+1=2 evaluando en el segundo miembro de la igualdad de la curva elíptica se tiene:
$x^3 + 4x + 6 = 2^3 + 4*2 + 6 = 8+8+6 \equiv 22 \bmod 7 \equiv 1 \equiv y^2$

En tablas

y	y^2
1	1

$P_A = (2, 1)$.

E:2 para j=1

x=E+j=2+1=3 evaluando en el segundo miembro de la igualdad de la curva elíptica se tiene:
$x^3 + 4x + 6 = 3^3 + 4*3 + 6 = 27+12+6 \equiv 45 \bmod 7 \equiv 3 \neq y^2$

para j=2
x=E+j=2+2=4 evaluando en el segundo miembro de la igualdad de la curva elíptica se tiene:
$x^3 + 4x + 6 = 4^3 + 4*4 + 6 = 64+16+6 \equiv 86 \bmod 7 \equiv 2 \equiv y^2$

En tablas

y	y^2
3	2

$P_E = (4, 3)$.

I:3 para j=1
x=I+j=3+1=4 evaluando en el segundo miembro de la igualdad de la curva elíptica se tiene:
$x^3 + 4x + 6 = 4^3 + 4*4 + 6 = 64+16+6 \equiv 86 \bmod 7 \equiv 2 \equiv y^2$

En tablas

y	y^2
4	2

$P_I = (4, 4)$.
O:4 para j=1

x=O+j=4+1=5 evaluando en el segundo miembro de la igualdad de la curva elíptica se tiene:
$x^3 + 4x + 6 = 5^3 + 4*5 + 6 = 125+20+6 \equiv 151 \bmod 7 \equiv 4 \equiv y^2$

En tablas

y	y^2
2	4

$P_O = (5, 2)$.
U:5 para j=1
x=U+j=5+1=6 evaluando en el segundo miembro de la igualdad de la curva elíptica se tiene:
$x^3 + 4x + 6 = 6^3 + 4*6 + 6 = 216+24+6 \equiv 246 \bmod 7 \equiv 1 \equiv y^2$

En tablas

y	y^2
1	1

$P_U = (6, 1)$.

Método de cifrado ElGamal con curvas elípticas

Con la curva elíptica $y^2 = x^3 + 4x + 6$ sobre GF(7) y el punto generador G=(1,2) si el usuario A con su par de claves:
$A = \{n_A,\ n_A*G\} = \{3,\ 3(1, 2)\} = \{3,\ (2,6)\}$
Donde A utiliza una k=3.

Si el usuario A quiere mandar el mensaje $P_O = (5, 2)$ efectuará lo siguiente:
$(M, N) = (P_A, P+n_A P_B) = \{(2,6), (5, 2) + 3(5, 5)\} = \{(2,6), (5, 2) + (4, 3)\}$

$$\lambda = \left[\frac{y_P - y_Q}{x_P - x_Q}\right] mod\ p = \left[\frac{3-2}{4-5}\right] mod\ 7 = \left[\frac{1}{-1}\right] mod\ 7 = \left[\frac{1}{6}\right] mod\ 7$$
$$= [1 * 6^{-1}] mod\ 7 = [1 * 6] mod\ 7 = 6$$

$$x_R = [\lambda^2 - x_P - x_Q] mod\ p = [6^2 - 4 - 5] mod\ 7 = 27\ mod\ 7 = 6$$

$$y_R = [-y_P + \lambda(x_P - x_R)] mod\ p = [-3 + 6(4-6)] mod\ 7$$
$$= -15\ mod\ 7 = 6$$

Entonces el usuario A le envía a B el par de puntos:

(M, N)=(P_A, P+$n_A P_B$)= {(2,6), (5, 2) + (4, 3)}={(2,6), (6, 6)}

Por su parte el usuario B descifra el mensaje de la siguiente forma:

P_x=N-n_BM=(6,6) - 2(2,6)=(6,6) - (4,3)=(6,6)+(4,-3) =(6,6)+(4,4)

$$\lambda = \left[\frac{y_P - y_Q}{x_P - x_Q}\right] mod\ p = \left[\frac{6-4}{6-4}\right] mod\ 7 = \left[\frac{2}{2}\right] mod\ 7 = [2 * 2^{-1}] mod\ 7$$
$$= [2 * 4] mod\ 7 = 1$$

$$x_R = [\lambda^2 - x_P - x_Q] mod\ p = [1^2 - 6 - 4] mod\ 7 = -9\ mod\ 7 = 5$$

$$y_R = [-y_P + \lambda(x_P - x_R)] mod\ p = [-6 + 1(6-5)] mod\ 7 = -5\ mod\ 7$$
$$= 2$$

P_x=(5,2)

Para decodificar se toma en cuenta sólo el parámetro “x”

$$m = \frac{x-1}{h} = \frac{5-1}{1} = 4$$

Por lo tanto del alfabeto empleado para codificar las vocales:

4→”O”

Bibliografía

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.

Charte, Francisco, *"Programación con Visual Basic 2005"*, Ediciones Anaya Multimedia, España, 2005.

Charte, Francisco, *"Programación con Visual Basic .NET"*, Ediciones Anaya Multimedia, España, 2002.

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

Evjen, Bill, *"El libro de Visual Basic .NET"*, Ediciones Anaya Multimedia, España, 2002.

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

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

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.

Rodríguez, Luis Ángel, *"Seguridad de la Información en sistemas de cómputo"*, Ventura Ediciones, México, 1995.

Serrano, Jorge, *"Manual Avanzado de Visual Basic .NET"*, Ediciones Anaya Multimedia, España, 2002.

Som, Guillermo, *"Manual Imprescindible de Visual Basic .NET"*, Ediciones Anaya Multimedia, España, 2002.

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

Tesis y trabajos de investigación

Pietiläinen, Henna, "Elliptic curve cryptography on smart cards", Helsinki University of Technology, Finland, 2000.

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

Zoltán, Adam, "Implementing elliptic curve cryptography on pc and smart card", Budapest University of Technology and Economics, Hungary, 2002.

Revistas y boletines

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

Boletín “Code and Cipher”, volumen 1, número 1, Certicom, Cánada, 2003.

Boletín “Code and Cipher”, volumen 1, número 2, Certicom, Cánada, 2003.

Boletín “Code and Cipher”, volumen 1, número 3, Certicom, Cánada, 2003.

Boletín “Code and Cipher”, volumen 1, número 4, Certicom, Cánada, 2003.

Boletín “Code and Cipher”, volumen 2, número 1, Certicom, Cánada, 2003.

Boletín “Code and Cipher”, volumen 2, número 2, Certicom, Cánada, 2003.

Boletín “Code and Cipher”, volumen 3, número 1, Certicom, Cánada, 2003.

Internet

http://www.htmlweb.net/seguridad/cripto_p/cripto_princ_4.html
http://www.dei.uc.edu.py/tai2000/criptografia/casimet2.htm
http://www.detectum.com/news/noticiascriptografia.htm
http://paraisomat.ii.uned.es/paraiso/cripto.php?id=elip_1
http://www.certicom.com/
http://www.microsoft.com/spanish/msdn/articulos/archivo/100105/voices/LanguageEnhancements.mspx
http://www.embedded.com/showArticle.jhtml?articleID=177101463