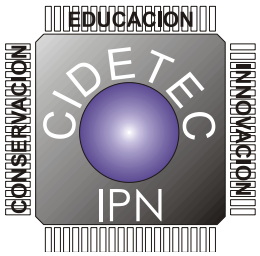
INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INNOVACIÓN Y DESARROLLO TECNOLÓGICO EN CÓMPUTO



Criptografía Basada en Identidad

Tesis que para obtener el grado de

Maestría en Tecnología de Cómputo

Presenta:

Lic. Ricardo Felipe Díaz Santiago

Directores:

Dr. Víctor Manuel Silva García

Dr. Rolando Flores Carapia

México D.F. **Enero 2012**

SIP-14-BIS

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 17:45 horas del día 13 del mes de diciembre del 2011 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del CIDETEC para examinar la tesis titulada:

"CRIPTOGRAFÍA BASADA EN IDENTIDAD"

Presentada por el alumno:

DÍAZ	SANTIAGO	RICARDO FELIPE
Apellido paterno	Apellido materno	Nombre(s)

Con registro:

B	0	9	2	2	8	3

aspirante de:

Maestría en Tecnología de Cómputo

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA
Directores de tesis

DR. VÍCTOR MANUEL SILVA GARCÍA
Primer Vocal

DR. ROLANDO FLORES CARAPIA
Segundo Vocal

DR. MAURICIO OLGUÍN CARBAJAL
Presidente

M. EN C. EDUARDO RODRÍGUEZ ESCOBAR
Secretario

DR. CARLOS RENTERÍA MÁRQUEZ
Tercer Vocal

PRESIDENTE DEL COLEGIO DE PROFESORES

DR. VÍCTOR MANUEL SILVA GARCÍA

S. E. P.
INSTITUTO POLITECNICO NACIONAL
CENTRO DE INNOVACION Y DESARROLLO TECNOLOGICO EN COMPUTO

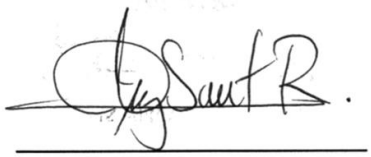
INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F., el día 06 del mes de enero del año 2012, el que suscribe Ricardo Felipe Díaz Santiago alumno del Programa de Maestría en Tecnología de Cómputo con número de registro B092283, adscrito al Centro de Innovación y Desarrollo Tecnológico en Cómputo, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del Dr. Víctor Manuel Silva García y el Dr. Rolando Flores Carapia y cede los derechos del trabajo intitulado Criptografía Basada en Identidad, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección rdiazs@ipn.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



Ricardo Felipe Díaz Santiago

Resumen

En la presente tesis, se muestra cómo el uso de los apareamientos bilineales, da lugar a criptosistemas que proporcionan nuevas funcionalidades.

Se lleva a cabo la implementación del protocolo tripartita Diffie-Hellman, que es un protocolo de establecimiento de claves entre partes que no han tenido contacto previo, utilizando un canal inseguro y de manera anónima (no autenticada). Debe su nombre a sus creadores que lo propusieron en 1976 [11].

Este protocolo generalmente es usado para intercambiar claves simétricas que serán empleadas por ejemplo para cifrar una sesión (es decir, establecer la clave de sesión). Es un protocolo no autenticado, sin embargo, provee los fundamentos para otros protocolos autenticados.

La seguridad del protocolo radica en la extrema dificultad (conjeturada, no demostrada) de calcular logaritmos discretos sobre campos finitos.

La versión del protocolo que se implementó, fue desarrollada por Joux [20], utiliza apareamientos bilineales y curvas elípticas, permite que tres participantes intercambien claves en un solo paso de comunicación.

Para la implementación desarrollada, se utilizaron las bibliotecas GMP para el manejo de números grandes y PBC para el manejo de los apareamientos.

Abstract

In this thesis, it is shown how the use of bilinear pairings leads to cryptosystems that provide new functionality.

The Tripartite Diffie Hellman Protocol is implemented; this is a protocol for the establishment ok keys between parties who have not had a previous contact, using an insecure channel in an anonymous (unauthenticated) way. This protocol was named after its creators, who proposed it in 1976 [11].

Generally, the Diffie-Hellman protocol It is used to exchange symmetric keys that will be used as an example for encrypting a session (i.e. to establish the session key). Although this is an unauthenticated protocol, it provides the foundation for several authenticated protocols .

The security of this protocol resides in the extreme difficulty (conjectured, not proven) to compute discrete logarithms over finite fields.

The version of the Diffie-Hellman protocol implemented in this thesis was developed by Joux [20] and it uses bilinear pairings and elliptic curves, allowing three participants to share keys in a single step comunication.

For the developed implementation, GMP libraries were used for handling large numbers, while PBC libray was used to manage bilinear pairings.

Agradecimientos

A Bety, por todo su amor y su apoyo.

A mi hermana Sandrita, por su invaluable ayuda.

A mis padres y toda mi familia, por todo lo que soy.

Al doctor Víctor Manuel Silva, por toda su paciencia y guía.

Al doctor Rolando Flores, por su asesoría.

A mis sinodales, por sus valiosas observaciones.

A los profesores del CIDETEC, por todas sus enseñanzas.

Al personal de apoyo del CIDETEC, por toda la atención proporcionada.

Índice general

Resumen II

Abstract III

Agradecimientos IV

Índice general V

Índice de figuras VII

1. Introducción **1**

1.1. Objetivo General 2
1.2. Objetivos Particulares 2
1.3. Justificación 2
1.4. Criptografía 3
1.5. Criptografía de Clave Secreta 4
1.6. Estado del arte 6
1.7. El estándar Rivest-Shamir-Adleman (RSA) 8
1.8. Criptografía Basada en Identidad 9

2. Preliminares Matemáticos **11**

2.1. Conjuntos y relaciones 12
2.2. Grupos 16
2.3. Anillos 17
2.4. Ideales 18
2.5. Campos Finitos 19
2.6. Curvas Elípticas 20
2.6.1. Aritmética sobre curvas elípticas 24

3. Apareamientos Bilineales **27**
3.1. Introducción 27
3.2. Exponenciación en Grupos Generales Cíclicos 30
3.3. El apareamiento Tate 32
3.4. Cálculo del apareamiento Tate y el algoritmo de Miller 34

4. Desarrollo e Implementación **37**
4.1. Introducción 37
4.2. Hardware utilizado 38
4.3. Software adicional utilizado 38
4.4. La biblioteca GMP 39
4.4.1. Procedimiento de instalación 39
4.4.2. Archivos de encabezado y compilación 40
4.4.3. Ejemplo de uso 40
4.4.4. Funciones Importantes 42
4.5. La biblioteca PBC 43
4.5.1. Instalación 43
4.5.2. Ejemplo de uso 44
4.6. El Protocolo tripartita de Diffie Hellman 45
4.6.1. Planteamiento general 45
4.6.2. Una solución a partir del apareamiento Tate 46
4.7. Implementación en C 48
4.8. Resultados 48

5. Conclusiones y Trabajo a Futuro **54**
5.1. Conclusiones 54
5.2. Trabajo a Futuro 55

Bibliografía **56**

Índice de figuras

1.1. Criptografía simétrica 5
1.2. Criptografía de Clave Pública 6
1.3. Uso de una Autoridad Certificadora 7

2.1. Ejemplos de dos curvas elípticas $y^2 = x^3 - x$ y $y^2 = x^3 + x$ 22
2.2. Suma de puntos en una curva elíptica 23

4.1. Ejemplo de uso de la biblioteca GMP 41
4.2. Ejemplo de uso de la calculadora de apareamientos 44
4.3. Intercambio de claves tripartita 46
4.4. Codificación del protocolo tripartita de Diffie-Hellman en C 49
4.5. Codificación del protocolo tripartita de Diffie-Hellman en C 50
4.6. Primera ejecución del protocolo de secreto tripartita 51
4.7. Segunda ejecución del protocolo de secreto tripartita 52

Capítulo 1

Introducción

Hay una antigua historia sobre una persona que quería que su computadora fuera tan fácil de utilizar como su teléfono. Sus deseos se han hecho realidad, ya no sé cómo usar mi teléfono.

Bjarne Stroustrup

La Criptografía Simétrica o de Clave Secreta, ha sido utilizada a través de los tiempos para fines militares y diplomáticos. Es hasta la década de 1970 que se desarrolla la Criptografía de Clave Pública, que permite aplicaciones en el comercio y las finanzas entre otros ámbitos.

La importancia de la Criptografía de Clave Pública radica en que resuelve el problema del intercambio de claves de forma práctica, al funcionar con un par de claves para cada usuario, una pública y otra privada.

Las soluciones que se han propuesto, no tienen un acceso tan sencillo, incluso para los sectores especializados. En México, de acuerdo al estudio AMIPCI de Comercio Electrónico 2009 [2], las ventas por internet representaron el 16 % de las ventas totales para comercios que contaban con este esquema de venta, lo cual tiene que ver entre

otras cosas, con el nivel de confianza que los usuarios tienen para comprar en Internet, que es de acuerdo al mismo estudio, del 55 %.

La falta de popularidad de este tipo de tecnología se debe a esquemas complicados de administración, lo que ha llevado al desarrollo de nuevas soluciones, una de ellas, la Criptografía Basada en Identidad, de la que trataremos en la presente tesis.

1.1. Objetivo General

Desarrollar un sistema de Criptografía Basada en Identidad, mediante apareamientos bilineales.

1.2. Objetivos Particulares

- Aplicar apareamientos bilineales en Criptografía.
- Utilizar lenguajes de programación de alto nivel en el desarrollo del sistema.

1.3. Justificación

Conforme crece el número de usuarios y áreas que necesitan proteger y validar información, se plantean problemas que requieren hacer uso de servicios Criptográficos, de forma novedosa y en muchas ocasiones inédita, pues los sistemas tradicionales no siempre cumplen con los nuevos requisitos planteados. En nuestro caso, se plantea explorar el uso de los apareamientos bilineales como mecanismo de cifrado y descifrado. Este esquema podría ser usado, por ejemplo, por el círculo de los más altos ejecutivos de una empresa, donde hay un número reducido de usuarios y se requiere de una agilidad muy dinámica en el manejo de las claves y sería deseable tener un

esquema que no implique el uso de firmas electrónicas.

Cabe señalar que hoy en día, la mayoría de los sistemas criptográficos de clave pública están basados en el estándar RSA y solo algunos hacen uso de las Curvas elípticas, aún cuando comparativamente el uso de estas proporciona igual o mayor seguridad con la misma longitud de clave.

1.4. Criptografía

De acuerdo con el Glosario de Seguridad en Internet [32], la Criptografía se define como: *La Ciencia Matemática que trata con la transformación de datos para que su significado sea ilegible, es decir, para esconder su contenido semántico, prevenir alteraciones no detectadas, o prevenir su uso no autorizado. Si la transformación es reversible, la Criptografía tiene que ver también con la restauración de los datos encriptados a su forma comprensible.*

De acuerdo con [9], esta es una definición que sólo considera uno de los cuatro objetivos fundamentales y que se listan a continuación.

Privacidad. Consiste en ocultar la información de un mensaje a través de un medio de comunicación, que podría incluso ser observado por un tercero.

Integridad. Es la propiedad de asegurar que un mensaje no ha sido alterado de su forma original por un tercero.

Autenticación. Consiste en determinar de manera inequívoca la identidad de alguno de los participantes, sea una persona o una computadora. Por ejemplo, durante los siglos XVI y XVII se utilizaba un sello lacre a base de goma laca y trementina para garantizar la autenticidad de cartas importantes. Otro ejemplo, es el uso de una contraseña alfanumérica para el acceso a una computadora.

No repudio. Es el acuerdo de adherirse a cierta obligación, ó más específicamente la incapacidad para refutar responsabilidad [9]. Por ejemplo, una vez que una persona firma un contrato legal, este se convierte en un medio de no repudio.

Un Sistema Criptográfico, se define como una tupla de cinco elementos a saber (P, C, K, E, D), donde se cumplen las siguientes condiciones: [33]

- P es un conjunto finito de posibles *textos en claro*.
- C es un conjunto finito de posibles *textos cifrados*.
- K es el *espacio de claves*, un conjunto finito de claves válidas.
- Para cada $k \in K$, existe un algoritmo de cifrado $e_k \in E$ y un correspondiente algoritmo de descifrado $d_k \in D$. Tanto $e_k : P \to C$ como $d_k : C \to P$ son funciones tales que $d_k(e_k(x)) = x$ para cada texto en claro $x \in P$.

En este punto cabe señalar que todos los Sistemas Criptográficos deberían seguir religiosamente el principio de Kerckhoffs (Auguste KerckHoffs, 1883), que dice: *"Un Criptosistema debe ser seguro, incluso si un atacante conoce todos los detalles del sistema, excepto por la Clave. En particular, el sistema debería ser seguro cuando tal atacante conozca los algoritmos de Cifrado y Descifrado."*
Esta observación no debe entenderse como mantener en secreto los algoritmos, pues la historia ha mostrado en múltiples ocasiones que tales sistemas terminan siendo violentados fácilmente por medio de ingeniería inversa ó por filtraciones del funcionamiento.

1.5. Criptografía de Clave Secreta

También Conocida como Criptografía simétrica, consiste en utilizar una misma clave, que es conocida solamente por el emisor y por el receptor del mensaje, para cifrar la

información mediante un algoritmo (Algoritmo de Cifrado Simétrico), que consiste en aplicar una o mas operaciones que transforman la información original.

En la figura 1.1, se ejemplifica el funcionamiento del método antes descrito. En este caso, Alicia le envía a Beto el texto claro "Nos vemos a las tres", utilizando una clave única (que es de conocimiento solamente de Alicia y de Beto).

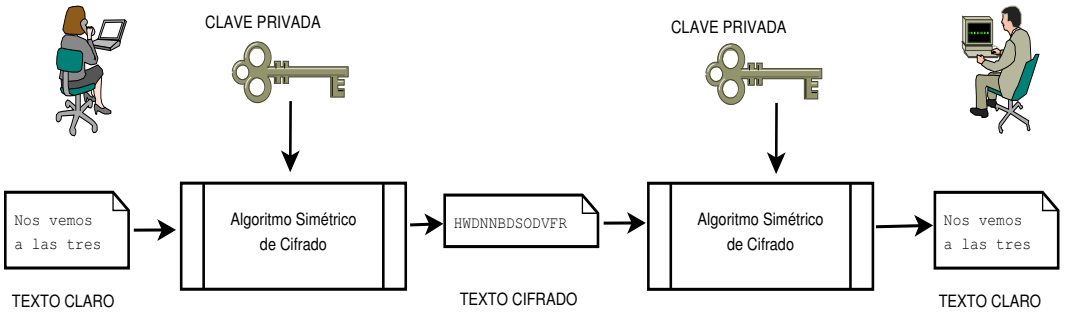


Figura 1.1: Criptografía simétrica

Las operaciones fundamentales de la Criptografía Simétrica son las siguientes:

Sustitución. Como su nombre lo indica, es reemplazar todos los caracteres del mensaje original por otros de acuerdo a cierta regla establecida, por ejemplo,por el quinto caracter en la secuencia del alfabeto. Así, por ejemplo, una A se sustituiría por una F.

Transposición. En este caso, cada caracter se sustituye por otro del mismo texto de acuerdo a la posición en que se encuentre, indicado por la regla de transposición. Por ejemplo, puede indicar que cambiemos el primer caracter por el octavo, el segundo por el octavo, etc.

Hoy en día, podemos clasificar a los Criptosistemas Simétricos, en dos grandes familias, los cifradores de flujo y los cifradores de bloque. La diferencia entre ambos es la

cantidad de bits sobre los que operan: uno o un bloque de ellos.

1.6. Estado del arte

La Criptografía de clave pública es un método de cifrado de información, donde cada usuario debe generar dos claves, una de ellas es pública, es decir, es conocida y utilizada por todos los usuarios del sistema y se utiliza para cifrar información. La otra clave, se conoce como clave privada y es usada para que el usuario poseedor de la misma pueda descifrar la información enviada.

En la figura 1.2 se ejemplifica el funcionamiento del método antes descrito. En este caso, Alicia le envía a Beto el texto claro "Nos vemos a las tres", utilizando la clave pública de Beto, con ello se obtiene un texto cifrado que únicamente Beto puede descifrar a través de su clave privada.

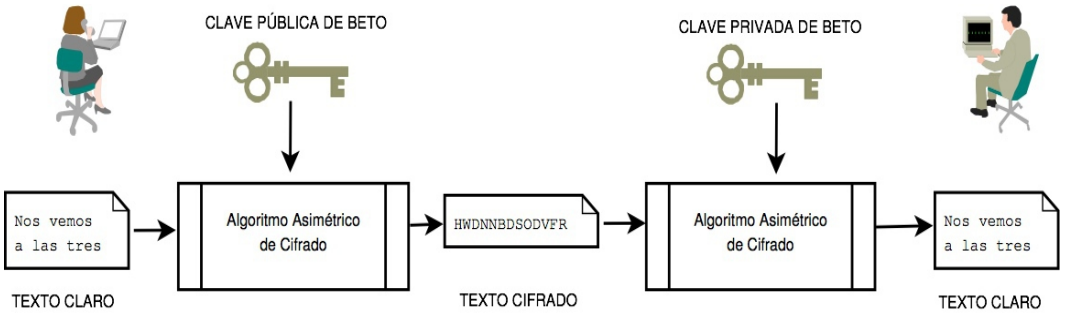


Figura 1.2: Criptografía de Clave Pública

Este esquema se puede utilizar siempre que Alicia esté segura que la Clave Pública que utiliza para cifrar mensajes realmente sea del destinatario correspondiente.

En el caso que acabamos de ejemplificar, suponemos que Beto le proporcionó de alguna manera segura, su Clave Pública a Alicia, tal vez en persona.

La Criptografía Asimétrica tiene muchas aplicaciones, una de ellas es el Comercio Electrónico. En este caso, no siempre es posible ni práctico proporcionar personalmente las Claves Públicas de cada participante, por ejemplo, si vendedor y comprador están físicamente en zonas geográficas diferentes, al otro lado del planeta.

Con esta complicación adicional se requiere de un esquema general que permita distribuir claves públicas que garantice que una clave corresponde a quien dice corresponder y no ha sido falsificada por un tercero, lo que le permitiría descifrar mensajes.

La solución más ampliamente aceptada es el uso de certificados que son emitidos por un tercero, una entidad en la cual confían todos los participantes, denominada Autoridad Certificadora. Su función es comprobar la identidad de los solicitantes de tales certificados, crearlos o publicar listas de revocación cuando estos ya no son válidos.

En la figura 1.3, se muestra gráficamente la operación de una de tales Autoridades Certificadoras.

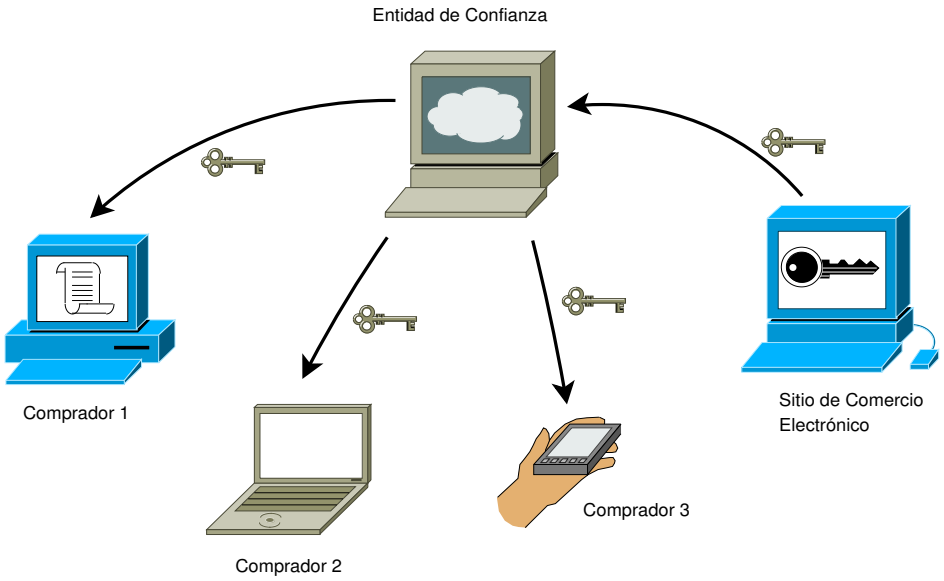


Figura 1.3: Uso de una Autoridad Certificadora

La administración de tales Autoridades Certificadoras representa un problema adicio-

nal, pues una sola Autoridad no es suficiente, lo que plantea el uso de una estructura jerárquica distribuida conocida como Infraestructura de Clave Pública (PKI por sus siglas en Inglés). El uso de una PKI representa un inconveniente para el uso de la Criptografía de Clave Pública.

La Criptografía Basada en Identidad ofrece una solución a gran parte de los problemas planteados anteriormente, al basar su funcionamiento en alguna característica pública del destinatario, por ejemplo su dirección de correo electrónico. Aunque la Criptografía Basada en Identidad no está exenta de costos adicionales, en esta tesis se considera como una alternativa para ciertas aplicaciones.

1.7. El estándar Rivest-Shamir-Adleman (RSA)

En 1976, Whitfield Diffie y Martin Hellman [11] desarrollaron la Criptografía de Clave Pública, de forma inmediata criptologistas empezaron a trabajar en métodos para implementar este novedoso esquema de cifrado.

En 1977, Ronald Rivest, Adi Shamir y y Leonard Adleman propusieron el esquema de Criptografía de Clave Pública más popular hasta nuestros días: *RSA*.

Tanto el proceso de cifrado como el de descifrado, se lleva a cabo en RSA en el anillo de los enteros $\mathbb{Z}_n$, donde las operaciones modulares tienen gran relevancia. *RSA* cifra un texto plano x, considerando que es una cadena de bits que representa un elemento en $\mathbb{Z}_n = \{1, 2, \ldots, n\}$. Por lo tanto, el valor binario de x debe ser menor a n y esto mismo se aplica para el texto cifrado y. Estos dos procesos se describen a continuación.

Cifrado RSA. Dada la clave pública $(n, e) = k_{pub}$ y el texto plano x, la función de cifrado es:

$$y = e\ k_{pub}\ (x) \equiv x^e\ mod\ n, \tag{1.1}$$

donde $x, y \in \mathbb{Z}_n$.

Descifrado RSA. Dada la clave privada $d = k_{priv}$ y el texto cifrado y, la función de descifrado es:

$$x = d_{k_{priv}}(y) \equiv y^d \ mod \ n, \tag{1.2}$$

donde $x, y \in \mathbb{Z}_n$.

En la práctica, x, y, n y d son números muy grandes, de al menos 1024 bits.

1.8. Criptografía Basada en Identidad

De acuerdo con [8], la Criptografía Basada en Identidad, *"es un esquema de Criptografía simétrica, donde la clave pública de un usuario puede ser cualquier cadena arbitraria, por ejemplo la dirección de correo electrónico"*.

Continuando con nuestro ejemplo de Comunicación encriptada entre Alicia y Beto, bajo este nuevo esquema, si Alicia deseara enviarle un mensaje a Beto le bastaría con conocer su direccción de correo electrónico, no requiere la comprobación de la clave por parte de la Autoridad Certificadora. En lugar de una Autoridad Certificadora, el papel del tercero consiste en ser un *Generador de Claves Privadas* (o PKG, por sus siglas en inglés).

Este esquema fue desarrollado por Shamir [31] en 1984, quien lo describió así:

"Hace los aspectos criptográficos de la comunicación casi transparentes al usuario y puede ser usado de forma efectiva incluso por abogados que no saben nada acerca de Claves o Protocolos".

De acuerdo con [8], La Criptografía Basada en Identidad consiste de los siguientes algoritmos a saber:

Establecimiento. Generar un conjunto de parámetros públicos, junto con una Clave

maestra *Key-Gen*, que generará la clave privada de cualquier entidad.

Cifrar. A partir de una identidad, este algoritmo cifrará un texto.

Descifrar. Dados un mensaje cifrado y una clave privada, este algoritmo se encarga de descifrarlo.

Aún cuando Shamir planteó un desafío a la comunidad criptográfica de desarrollar un esquema práctico de la Criptografía Basada en Identidad, tal reto permaneció sin solución hasta después del año 2000, cuando surgieron tres propuestas, dos de ellas basadas en apareamientos bilineales sobre curvas elípticas y una tercera basada en factorización, menos práctica. Antes de abordar con detalle uno de estos esquemas, describiremos en el siguiente capítulo los conceptos matemáticos necesarios.

En resumen, la Criptografía ha tenido un gran desarrollo en las últimas décadas. El espectro de los usuarios y las aplicaciones se han ampliado del ámbito diplomático y militar al comercial y personal. Esto no ha sido gratuito, pues siempre hay nuevos retos que resolver.

Capítulo 2

Preliminares Matemáticos

El Álgebra es generosa: a menudo da más de lo que se le pide.

D´ Alembert

De acuerdo con [29], en los últimos cuarenta o cincuenta años, los sistemas algebraicos han jugado un rol crítico en el desarrollo de la tecnología de las computadoras y las comunicaciones que nos rodea en nuestras vidas diarias. El Álgebra ya no es sólo un área bonita de las Matemáticas con una historia interesante, sino una disciplina que juega un papel primordial en el mundo moderno, lo que afecta incluso, áreas como el comercio moderno, la comunicación y el entretenimiento. Las técnicas de Criptografía de Clave Pública Moderna tales como RSA y la Criptografía de Curvas Elípticas, están basadas en la teoría de Números, la Teoría de Grupos, y la Geometría Algebraica.
En este capítulo se enumeran con ejemplos, los conceptos matemáticos que utilizaremos posteriormente en el desarrollo del trabajo de tesis.

2.1. Conjuntos y relaciones

Definición 2.1 *Un* conjunto *es una colección de objetos a los que llamaremos elementos o miembros del conjunto.*

Los conjuntos se pueden describir de forma exhaustiva, al listar cada uno de sus elementos, por ejemplo $A = \{1, 3, 5, 8\}$.

También se puede decribir un conjunto a través de una lista parcial, por ejemplo $B = \{1, 2, 3, 4, 5, \ldots\}$.

Finalmente, es posible describir un conjunto a través de una regla o condición. Por ejemplo. $C = \{x | x$ es el nombre de un estado de la república mexicana$\}$.

Ejemplo 2.1 *En nuestro caso, hay varios conjuntos importantes de números:*

- *Los números naturales o enteros positivos,* $\mathbb{N} = \{1, 2, 3, \ldots\}$.
- *El conjunto de los enteros,* $\mathbb{Z}$.
- $\mathbb{Z}_n = \{1, 2, 3, \ldots, n\}$.
- *El conjunto de los números racionales,* $\mathbb{Q}$.

Definición 2.2 *Una* Relación R *en un conjunto* A*, es un subconjunto de* $A \times A$*. Si* $(a, b) \in R$*, entonces también escribimos* aRb *y decimos que* a *está relacionada con* b.

Ejemplo 2.2 *Sea* $A = \mathbb{N}_6 = \{1, 2, 3, 4, 5\}$ *y*

$R = \{(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 2), (2, 4), (3, 3), (4, 4), (5, 4)\}$

Esta es la relación que se conoce comúnmente como "divide a".

Así pues, podemos definir esta relación como:

$R = \{(a, b) | a, b \in \mathbb{N}_6 \text{ y } a | b\}$

Definición 2.3 *Una* Relación de equivalencia*, es aquella que cumple con las siguientes propiedades:*

Reflexividad. *si* $a\ R\ a,\ \forall\ a\ \in A$,

Simetría. *si* $a\ R\ b\implies b\ R\ a, \forall\ a, b\ \in A$,

Transitividad. *si* $a\ R\ b,\ b\ R\ c\implies a\ R\ c\ \ \forall\ a, b, c\ \in A$,

Definición 2.4 *Sean* $n \in \mathbb{N}$ *y* $a, b \in \mathbb{Z}$. *Decimos que* **a es congruente con b módulo n**, *o bien que* **a es congruente con b mod n**, *si n divide a* $a - b$.

La siguiente notación, es de uso común para indicar que **a es congruente con b módulo n**.

$$a \equiv b \mod n \ o\ a \equiv_n b\ o\ a \equiv b \tag{2.1}$$

Ejemplo 2.3 *Si* $n = 2$, *entonces* $a \equiv b \mod 2$, *si y sólo sí 2 divide a* $a - b$. *Para que esto suceda, se requiere que tanto a como a, sean ambos pares o ambos impares.*

Definición 2.5 *Sea I un conjunto no vacío. Para cada* $i \in I$, *sea* X_i, *un subconjunto no vacío. Entonces* $\chi = \{X_i | i \in I\}$ *es una* familia *de conjuntos* indizados *por I, o* con índice I.

Una partición *de un conjunto no vacío* X *es una* familia $\{X_i | i \in I\}$ *de subconjuntos de X, tales que:*

1. $X = \cup_{i\in I}\ X_i$.
2. $i \neq j\ X_i \cap X_j = \emptyset$.

Ejemplo 2.4 *Sea* $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},\ X_1 = \{1, 5, 7\},\ X_2 = \{2, 4, 6, 8\},\ X_3 = \{3, 9\}$. *Entonces* $\{X_i | i = 1, 2, 3\}$ *es una partición de X.*

Lema 2.1 *Las relaciones de equivalencia y las particiones son conceptos muy relacionados. Sea R una* relación de equivalencia *de un conjunto X, para cada* $x \in X$, *sea*

$C_x\ =\ \{z \in X | x\ R\ z\}$.

$X = \cup_{x \in X} C_x$,

$\mathcal{P}_R = \{C_x | x \in X\}$ *es una partición de* X. *Entonces,*

1. *para todo* $x \in X,\ x \in C_x$,
2. *para todo* $x, y \in X$,

$$C_x \cap C_y \neq \emptyset \implies C_x = C_y.$$

Definición 2.6 *Si* R *es una relación de equivalencia en un conjunto no vacío* X *y* $x \in X$, *entonces a*

$$C_x = \{z \in X | xRz\}$$

y se le conoce como una clase de equivalencia de x *con respecto a* R. *Si* R *es* $\equiv_n$, *entonces se acostumbra adoptar la notación*

$$C_x = [x]_n \text{ ó } C_x = [x].$$

Ejemplo 2.5 *La relación de equivalencia* $\equiv_2$ *tiene exactamente dos clases*

$$[0] = \{0, \pm 2, \pm 4, \ldots\} \text{ y } [1] = \{\pm 1, \pm 3, \ldots\}$$

Es de notar, que en general, para $n \in \mathbb{N}$ *y* $a \in \mathbb{Z}$, *la clase de* a *es*

$$[a] = \{a + kn | k \in \mathbb{Z}\}.$$

Así, por ejemplo,

$$[0]_2 = [2]_2 = [-2]_2 = [4]_2 = \ldots$$
$$[2]_3 = [-1]_3 = [5]_3 = [8]_3 = \ldots$$
$$[a]_n = [a + n]_n = [a - n]_n = \ldots$$

Definición 2.7 (Ecuaciones en $\mathbb{Z}$) *Sea* $a \in \mathbb{Z},\ n \in \mathbb{N}$ *y*

$$f(x) = a_k x^k + a_{k-1}x^{k-1} + \ldots + a_1 x + a_0$$

es un polinomio con coeficientes enteros. Supongamos que $f(a) \equiv 0 \mod n$. *Entonces, para todo* $b \equiv a \ mod \ n$, *tenemos que* $f(b) \equiv f(a) \equiv 0 \ mod \ n$. *Esto nos lleva a considerar polinomios sobre* $\mathbb{Z}_n$. *De hecho, si calculamos módulo n, deberíamos tratar a los coeficientes como elementos de* $\mathbb{Z}_n$, *de forma que podamos escribir*

$$f(x) = [a_k]x^k + [a_{k-1}]x^{k-1} + \ldots + [a_1]x + [a_0]$$

como un polinomo con coeficientes en $\mathbb{Z}_n$. *Entonces, para todo* $[a] \in \mathbb{Z}_n$,

$$\begin{aligned} f([a]) &= [a_k][a]^k + [a_{k-1}][a]^{k-1} + \ldots + [a_1][a] + [a_0] \\ &= [a_k a^k + a_{k-1}a]^{k-1} + \ldots + a_1 a + a_0] \\ &= [f(a)]. \end{aligned}$$

Por lo tanto,

$$f(a) \equiv 0 \mod n \iff f([a]) \ = \ 0 \ en \ \mathbb{Z}_n.$$

de tal manera que podemos hablar de la clase $[a]$, *como la* solución de la congruencia *o el* zero de la congruencia

$$f(a) \equiv 0 \mod n.$$

El número de soluciones (módulo n) de esta congruencia por lo tanto significa el número de distintas clases (o elementos diferentes de $\mathbb{Z}_n$*) que son soluciones.*

Esta última definición está íntimamente relacionada con el:

Teorema 2.1 (Teorema Fundamental de la Aritmética.) *Todo entero n no cero, se puede expresar de la siguiente manera*

$$n = \mp p_1^{e_1} \ldots p_r k^{e_r} \tag{2.2}$$

Donde cada uno de los p_i son primos diferentes y los e_i son enteros positivos. Esta expresión es única, excepto por el orden de los primos.

2.2. Grupos

Definición 2.8 *Un* **Grupo** $(\mathbb{G}, \bullet)$ *es un conjunto* $\mathbb{G}$*, con una operación binaria* $\bullet$ *que satisface los siguientes axiomas:*

Cerradura. $A \bullet B \in \mathbb{G}, \forall A, B, \in \mathbb{G}$.

Asociatividad $(A \bullet B) \bullet C = A(B \bullet C), \forall A, B, C \in \mathbb{G}$.

Existencia de Identidad. *Hay un elemento único* $E \in \mathbb{G}$ *tal que* $A \bullet E = E \bullet A = A, \forall A \in \mathbb{G}$.

Existencia de inverso. *Para cada* $A \in \mathbb{G}$*, existe un elemento único,* $B \in \mathbb{G}$*, tal que* $A \bullet B = B \bullet A$.

Ejemplo 2.6 *Un ejemplo de grupo, es el conjunto* $\mathbb{Z}$ *de todos los enteros con respecto a la suma. En este caso el elemento identidad es el 0 y el inverso de* $a, \in \mathbb{Z}$ *es* $-a$*. De aquí en adelante podemos hacer referencia a este, como el grupo aditivo* $\mathbb{Z}$*. Por otro lado,* $\mathbb{Z}$ *no es un grupo multiplicativo, puesto que por ejemplo, ni 0, ni 5 tienen inverso multiplicativo.*

Definición 2.9 *El grupo* $(\mathbb{G}, \bullet)$*, se dice que es un* grupo conmutativo *o* grupo abeliano*, si satisface un axioma adicional, el de la conmutatividad, definida de la siguiente manera:*

Conmutatividad. $A \bullet B = B \bullet A \; \forall A, B \in \mathbb{G}$.

Ejemplo 2.7 *El conjunto $\mathbb{Z}$ bajo la suma es un conjunto abeliano, en este caso el elemento identidad es el 0 y el inverso de a es $-a$, $\forall a \in \mathbb{Z}$.*

Ejemplo 2.8 *A manera de contra-ejemplo, el conjunto de los enteros positivos bajo la suma no es un conjunto abeliano, pues no existen los inversos aditivos para los enteros positivos.*

Definición 2.10 *Un Grupo $\mathbb{G}$ se dice que es cíclico, si para alguna $a \in \mathbb{G}$, cualquier $x \in \mathbb{G}$ es de la forma: a^m, con $m \in \mathbb{Z}$. Al elemento a, se le conoce como* el generador *de $\mathbb{G}$.*

Ejemplo 2.9 *El grupo aditivo $\mathbb{Z}$ es cíclico con generador $a = 1$, para cada $m \in \mathbb{Z}$, $a^m = ma = m$.*

Definición 2.11 *Sea $\mathbb{G}$ un grupo con respecto a $\bullet$. Cualquier subconjunto no vacío $\mathbb{G}$'de $\mathbb{G}$ es un* subgrupo *de $\mathbb{G}$ si $\mathbb{G}$' es un grupo respecto a $\bullet$.*

Ejemplo 2.10 *Sea $\mathbb{Z}_8$ con los elementos $\{0, 1, 2, 3, 4, 5, 6, 7\}$, como operación de grupo la suma módulo 8. Este grupo tiene dos subgrupos $\mathbb{H}$ y $\mathbb{J}$ no triviales cuyas tablas de Cayley se muestran a continuación.*

$+$	0	4
0	0	4
4	4	0

y

$+$	0	2	4	6
0	0	2	4	6
2	2	4	6	0
4	4	6	0	2
6	6	0	2	4

2.3. Anillos

Definición 2.12 *Un* **Anillo** *$(\mathcal{A}, +, \times)$ es un conjunto $\mathcal{A}$, con dos operaciones binarias $+$ y $\times$ que satisface los siguientes axiomas:*

Existencia de Identidad. $(\mathcal{A}, +)$ *es un grupo Abeliano con elemento identidad, denotado por* 0.

Asociatividad. *La Operación* $\times$ *es asociativa:* $(A \times B) \times C = A \times (B \times C), \forall A, B, C \in \mathcal{A}$.

Identidad multiplicativa. *La identidad multiplicativa se denota por* 1, *(con* $1 \neq 0$*):* $1 \times A = A \times 1 = A, \forall A \in \mathcal{A}$.

Distributividad. *La operación* $\times$ *distribuye sobre* $+$*:* $A \times (B + C) = (A \times B) + (A \times C)$ *y* $(B + C) \times A = (B \times A) + (C \times A), \forall A, B, C \in \mathcal{A}$.

El anillo $(\mathcal{A}, +)$, se dice que es un *anillo conmutativo* si $A \times B = (B \times A), \forall A, B \in \mathcal{A}$.

Ejemplo 2.11 *El siguiente ejemplo, tomado de [3], forma un anillo sobre el conjunto* $\{a, b, c, d\}$*, con la suma y la multiplicación definidas por:*

$+$	a	b	c	d
a	a	b	c	d
b	b	a	d	c
b	c	d	a	b
d	d	c	b	a

y

$\times$	a	a	a	a
a	a	b	a	b
b	a	c	a	c
b	c	d	a	b
d	a	d	a	d

2.4. Ideales

Definición 2.13 *Sea* R *un anillo, un* **Ideal** *de* R *es un subgrupo* I *del grupo aditivo de* R *que es cerrado bajo la multiplicación por los elementos de* R*, esto es, para todas las* $a \in I$ *y* $r \in R$*, tenemos que* $ar \in I$.

Ejemplo 2.12 *Para algún* $m \in \mathbb{Z}$*, el conjunto* $m\mathbb{Z}$*, no sólo es un subgrupo del grupo aditivo* $\mathbb{Z}$*, también es un ideal del anillo* $\mathbb{Z}$.

2.5. Campos Finitos

Definición 2.14 *Un campo es un anillo conmutativo en el que cada elemento no cero tiene un único inverso multiplicativo. Un campo finito es un campo con un número finito de elementos. Se denota por $\mathbb{F}_q$ a un campo con q elementos.*

$\mathbb{F}_q$ tiene

- *Un grupo aditivo $\mathbb{F}_q^+$, con q elementos.*
- *Un grupo multiplicativo $\mathbb{F}_q^*$ con $q-1$ elementos.*

Ambos grupos son abelianos.

$\mathbb{F}_q^*$ es cíclico:

$\exists A \in \mathbb{F}_q^*$ tal que $\mathbb{F}_q = \{A^i | 0 \leq i \leq q-2\}$

Un campo finito con q elementos existe si y solo si $q = p^m$, donde p es un número primo y $m \geq 1$. A p se le conoce como la característica *del campo finito.*

Ejemplo 2.13 *Ejemplo tomado de [17]. Los enteros mod p, $\mathbb{Z}_p$, con p primo, forman un campo. En $\mathbb{Z}_p$, tenemos la relación*

$$p_1 = \underbrace{1+1+\ldots+1}_{n \text{ veces}} = 0$$

Definición 2.15 *Si K y L son campos, con $K \subseteq L$, K es un* subcampo *de L y L es una* extensión *de K*

Ejemplo 2.14 *$\mathbb{R}$ es una extensión del campo $\mathbb{Q}$.*

Definición 2.16 *Se le conoce como* extensión de grado m *de un campo finito $\mathbb{F}_q$ a toda extensión finita de $\mathbb{F}_q$ y que es de la forma $\mathbb{F}_{q^m}$.*

$\mathbb{F}_q \cong \mathbb{F}_q[x]/(f(x))$, donde $f(x)$ es un polinomio irreductible de grado m sobre $\mathbb{F}_q$

Si $q = p^m$, entonces $\mathbb{F}_{q^k} = \mathbb{F}_{p^{km}}$, que se puede construir con:

- *Un polinomio irreductible de grado km sobre* $\mathbb{F}_p$
- *Un polinomio irreductible de grado k sobre* $\mathbb{F}_q$

Finalmente, $\mathbb{F}_{q^r} \subseteq \mathbb{F}_{q^s}$*, si y solo si* $r|s$*.*

Definición 2.17 *La* cerradura algebraica *se define de la siguiente manera: Sean* K *y* L *dos campos con* $K \subseteq L$ *y* $\alpha \in L$*.* α *es* algebraica *sobre* K *si existe un polinomio no constante:*

$f(X) = a_0 + a_1X + \ldots + a_{m-1}X^{m-1} + X^m,\ a_i \in K$

tal que $f(\alpha) = 0$*.* L *es* algebraico *sobre* K *si cada elemento de* L *es* algebraico *sobre* K*.*

Una cerradura algebraica *de un campo* K*, es un campo* $\overline{K}$ *que:*

- $\overline{K}$ *es algebraico sobre* K*.*
- *cada polinomio no constante* $g(x)$ *con coeficientes en* $\overline{K}$ *tiene una raíz en* $\overline{K}$*.*

Ejemplo 2.15 *Los siguientes son ejemplos de cerradura algebraica.*

- $\mathbb{C}$ *es la cerradura algebraica de* $\mathbb{R}$*.*
- *La cerradura algebraica de* $\mathbb{F}_p$*:*

  $\overline{\mathbb{F}_p} = \bigcup_{m\geq 1} \mathbb{F}_{p^m}$

  $\mathbb{F}_p$ *es infinito.*

2.6. Curvas Elípticas

De acuerdo con [12], las *Curvas elípticas* se pueden equipar con una ley de grupo que se puede calcular de forma muy eficiente, de tal forma que son altamente susceptibles para la implementación de sistemas criptográficos de clave pública. Son particularmente importantes porque pueden lograr el mismo nivel de seguridad que un criptosistema

de clave pública basado en campos finitos, pero con longitudes de clave mucho más cortas, lo que resulta en procesos de cifrado y descifrado más eficientes.

Definición 2.18 *Tomaremos la definición de [36] para una curva elíptica. Una* curva elíptica *E es la gráfica de una ecuación de la forma $y^2 = x^3 + Ax + B$, Donde A y B son constantes. A esto se le conoce comúnmente como la* Ecuación de Weierstrass *de una curva elíptica.*

Es necesario especificar a qué conjuntos pertenecen A, B, x e y. Usualmente son los elementos de un campo, por ejemplo los números reales $\mathbb{R}$, los números complejos $\mathbb{C}$ o uno de los campos finitos $\mathbb{F}_p (= \mathbb{Z}_p)$ para algún primo p, o uno de los campos $\mathbb{F}_q$ donde $q = p^k$, con $k \geq 1$

Si K es un campo, con $A, B \in K$, entonces decimos que E está definida sobre K.

Si queremos considerar puntos con coordenadas en algún campo $L \supseteq K$, entonces escribimos $E(L)$. Por definición este conjunto siempre contiene el punto al infinito ∞, que definiremos posteriormente: $E(L) = \{\infty\} \cup \{(x, y) \in L \times L | y^2 = x^3 + Ax + B\}$.

En general, no es posible obtener gráficas significativas de curvas elípticas en la mayoría de los campos, sin embargo y por intuición es últil pensar en términos de las gráficas de curvas elípticas sobre los números reales, las cuales tienen dos formas que se muestran en la figura 2.1:

De manera más general, una curva elíptica se define por la siguiente forma más general, que se conoce como *Ecuación Generalizada de Weierstrass*:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{2.3}$$

Esta forma es útil cuando se trabaja con campos de característica 2 y 3 e inclusive se puede transformar para obtener formas simplificadas, por ejemplo para campos con

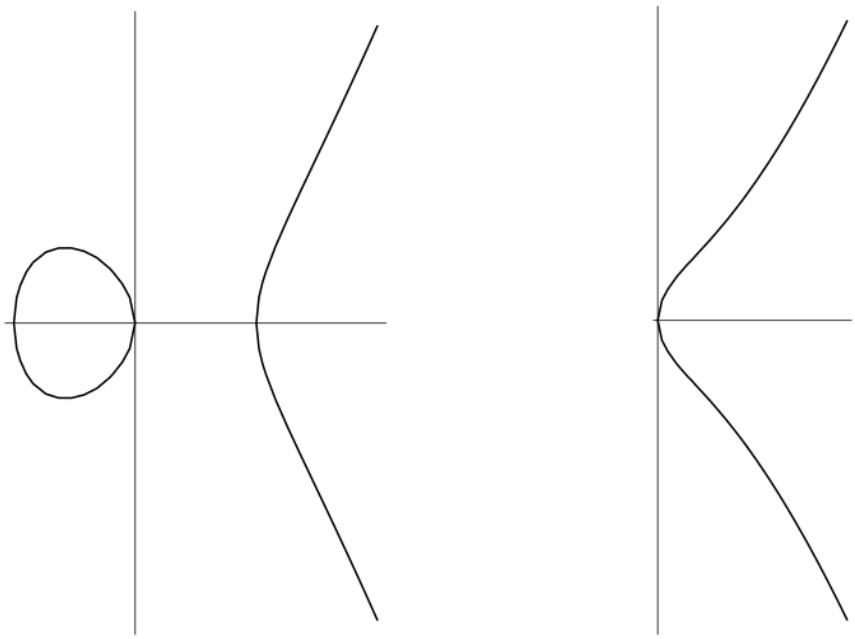


Figura 2.1: Ejemplos de dos curvas elípticas $y^2 = x^3 - x$ y $y^2 = x^3 + x$

características diferentes a 2 y 3 y con característica 3.

Definición 2.19 *Sea K un campo con característica $\neq 2, 3$ y sea $x^3 + ax + b$, con $a, b \in K$ un polinomio cúbico con la condición $4a^3 + 27b^2 \neq 0$ para asegurarnos que el polinomio no tiene múltiples raíces. En ese caso, una curva elíptica E sobre K es el conjunto de puntos (x, y) con $x, y \in K$ que satisfacen la ecuacíon*

$$y^2 = x^3 + ax + b \tag{2.4}$$

y también al elemento al infinito $\mathcal{O}$.

Definición 2.20 *Si K es un campo con característica 2, entonces hay dos tipos de curvas elípticas:*

- *Aquellas que satisfacen el conjunto de puntos*

$$y^2 + a_3 y = x^3 + a_4 x + a_6 \tag{2.5}$$

Donde $a_3, a_4, a_6 \in \mathbb{F}_q$, $a_3 \neq 0$, y $\mathcal{O}$, el punto al infinito. En este caso no importa si la cúbica del lado derecho de la ecuación tiene múltiples raíces o no.

- *Aquellas que satisfacen el conjunto de puntos*

$$y^2 + xy = x^3 + a_2x^2 + a_6 \tag{2.6}$$

Donde $a_2, a_6 \in \mathbb{F}_q$, $a_6 \neq 0$, y $\mathcal{O}$, el punto al infinito.

Los puntos en E forman un grupo abeliano aditivo con el punto al infinito $\mathcal{O}$, como el elemento identidad,

Para que esto pueda ocurrir, la suma de dos puntos se define de la siguiente manera: Sean $P, Q \in E$ dos puntos, entonces se define un tercer punto $P + Q$, de la manera que se ilustra en la figura 2.2, tomada de [36]:

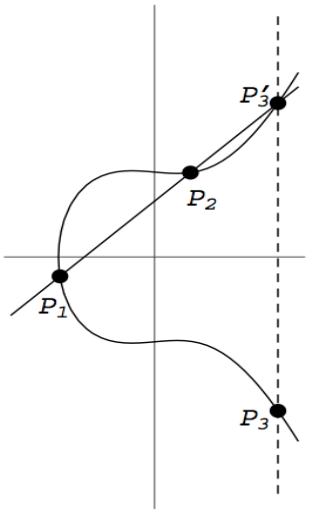


Figura 2.2: Suma de puntos en una curva elíptica

2.6.1. Aritmética sobre curvas elípticas

Definición 2.21 *Dados dos puntos sobre una curva elíptica E, descrita por la ecuación $y^2 = x^3 + Ax + B$*

$$P_1 = (x_1, y_1),\ P_2 = (x_2, y_2)$$

Definimos ahora un tercer punto P_3 de la siguiente manera: Si dibujamos una línea entre los puntos P_1 y P_2, en la figura 2.2 podemos ver que intersecta a E en un tercer punto $P_{3'}$. Si reflejamos este punto alrededor del eje de las x, o en otras palabras, cambiamos de signo a su coordenada y, obtenemos el punto P_3, el cual podemos definir como la suma de P_1 y P_2:

$$P_1 + P_2 = P_3.$$

Para el caso de la ecuación 2.4, la suma queda definida de la siguiente manera. El inverso de $P = (x_1, y_1) \in E$ es $-p = (x_1, -y_1)$. Si $Q \neq P$, entonces $P + Q = (x_3, y_3)$, donde,

$$x_3 = \lambda^2 - x_1 - x_2 \tag{2.7}$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \tag{2.8}$$

La definición de λ depende de P y Q:

Si P $\neq$ Q

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \tag{2.9}$$

Si P $=$ Q

$$\lambda = \frac{3x_1^2 + a}{2y_1} \tag{2.10}$$

En el caso de la ecuación 2.5, el inverso de $P = (x_1, y_1) \in E$ es $-P = (x_1, -y_1 - a_3)$. Si $Q \neq P$, entonces $P + Q = (x_3, y_3)$, donde,

Si P $\neq$ Q

$$x_3 = (\frac{y_1 + y_2}{x_1 + x_2})^2 + x_1 + x_2 \tag{2.11}$$

$$y_3 = (\frac{y_1 + y_2}{x_1 + x_2})(x_1 + x_3) + y_1 + a_3 \tag{2.12}$$

Finalmente, en el caso de la ecuación 2.6, el inverso de $P = (x_1, y_1) \in E$ *es* $-P = (x_1, -y_1 + x_1)$. *Si* $Q \neq -P$, *entonces* $P + Q = (x_3, y_3)$, *donde,*

Si P $\neq$ Q

$$x_3 = (\frac{y_1 + y_2}{x_1 + x_2})^2 + (\frac{y_1 + y_2}{x_1 + x_2})x_1 + x_2 + a_2 \tag{2.13}$$

$$y_3 = (\frac{y_1 + y_2}{x_1 + x_2})(x_1 + x_3) + x_3 + y_1 \tag{2.14}$$

Si P = Q

$$x_3 = (\frac{a_6}{x_1^2}) + x_1^2 \tag{2.15}$$

$$y_3 = (\ x_1^2 + (x_1 + \frac{y_1}{x_1})x_3 + x_3 \tag{2.16}$$

Definición 2.22 *Puntos en Curvas Elípticas. Sea* $\mathbb{F}_q$ *un campo para algún* $q > 3$. *A menos que se indique otra cosa, siempre se definen curvas sobre un campo de orden primo y con característica mayor a tres. Las curvas elípticas que se pueden implementar sobre campos con característica 2 y 3 permiten múltiples optimizaciones, pero así mismo, son susceptibles de ataques especializados relacionados con el problema del logaritmo discreto.*

Recordemos que una curva elíptica sobre un campo $\mathbb{F}_q$ *es una ecuación de la forma:*

$$E : Y^2 = X^3 + aX + b$$

Donde $a, b \in \mathbb{F}_q$.

Definición 2.23 *Sea K un campo finito con característica q, de tal manera que $K = \mathbb{F}_{q^m}$ para algún número natural m. Sea E una curva elíptica definida sobre K. Sea $n = \#E(K)$. Supongamos que $P \in E(K)$ satisface que $rP = \mathcal{O}$, de tal forma que P tiene orden r o un factor de r. A P se le llama* r-punto de torsión*. Al conjunto de r-puntos de torsión en $E(K)$ por E(K)[r].*

En síntesis, las curvas elípticas tienen un papel muy importante en el estudio de los apareamientos bilineales, como se verá en el siguiente capítulo. El conjunto de puntos G sobre una curva elíptica junto con la operación de suma de puntos, forman un grupo abeliano. De manera independiente en 1985, Neal Koblitz y Victor Miller propusieron su uso para diseñar sistemas criptográficos de clave pública.
De acuerdo con [15], se deben hacer varias elecciones al implementar un sistema criptográfico de curva elíptica, entre las que se encuentran:

- El campo finito, la representación de los elementos del campo, así como los algoritmos para llevar a cabo la aritmética de campo.
- Una curva elíptica, la representación para los puntos de la curva elíptica y los algoritmos con que se llevará a cabo la aritmética de curvas elípticas.
- Y obviamente, el protocolo a implementar y sus algoritmos.

Capítulo 3

Apareamientos Bilineales

Más valen dos juntos que uno solo,

porque es mayor la recompensa de su esfuerzo.

Libro del Eclesiastés 4:9

3.1. Introducción

Los apareamientos bilineales fueron introducidos en la comunidad criptográfica por Meneses, Okamoto y Vanstone con su ataque MOV [28], donde describen cómo los apareamientos se pueden usar para transportar el problema del logaritmo discreto en una cierta clase de curvas elípticas sobre un campo finito al problema del logaritmo discreto en un campo finito más pequeño, donde se puede usar un ataque de cálculo de índice sub-exponencial para atacar el problema. Fue sin embargo, la publicación de Boneh y Franklin [7] de un esquema de encriptación basado en identidad, basado en apareamientos bilineales lo que en realidad despertó el interés de los apareamientos entre la comunidad criptográfica. Cabe señalar que los sistemas propuestos serían muy difíciles de construir utilizando primitivas criptográficas más convencionales. Actualmente la Criptografía basada en apareamientos es un campo muy activo de investi-

gación, baste mencionar que desde 2007, anualmente se lleva a cabo una conferencia internacional relativa al tema. La cuarta conferencia, *Pairing 2010* recibió 64 artículos de 17 países, de los que se aceptaron y publicaron 25 de ellos en [21].

Definición 3.1 *De acuerdo con [8], un* apareamiento bilineal *con utilidad en la Criptografía es un mapeo bilineal no degenerado, eficiente de calcular:* $e : G_1 \times G_2 \to G_T$ *donde tanto* G_1, G_2 *como* G_T *son grupos cíclicos y del mismo orden primo* p*. En algunos casos el apareamiento es* simétrico: $G_1 = G_2$*. Cuando* $G_1 \neq G_2$ *se dice que el apareamiento es* asimétrico*.*

Los apareamientos más comúnmente utilizados se derivan de la teoría de curvas elípticas, donde G_1 y G_2 son los subgrupos de puntos de una curva elíptica sobre un campo finito, mientras que G_T es un subgrupo del grupo multiplicativo de un campo finito. Habitualmente, la operación del grupo de curvas elípticas se denota por una suma por lo que escribe G_1 y G_2 de forma aditiva, mientras que se escribe G_T de forma multiplicativa.
Expliquemos primero algunas de las propiedades que se requieren de un apareamiento bilineal.

Definición 3.2 *Sea* $G_1 = \langle P_1 \rangle$ *y* $G_2 = \langle P_2 \rangle$*,* no degenerado *significa que si* $e(P, Q)$ *es el elemento identidad de* G_T*, entonces* P *es el elemento identidad de* G_1 *o* Q *es el elemento identidad de* G_2*.*

En cuanto a que sea eficiente de calcular, significa que hay un algoritmo de tiempo polinomial que calcular el mapeo e. Típicamente el logaritmo base 2 de las dimensiones de G_1, G_2 y G_T estará acotado por arriba por un polinomio de una cantidad que se conoce como el parámetro de seguridad. De esta manera, las representaciones de los elementos de G_1, G_2 y G_T estarán acotados por un polinomio en el parámetro

de seguridad y el cual a su vez implica que el tiempo de ejecución del algoritmo para calcular e también estará acotado por un polinomio en el parámetro de seguridad. Cabe señalar que esta noción asintótica de eficiencia no es suficiente, pues un algoritmo en estos términos puede tener como cota superior un polinomio con un parámetro de seguridad muy alto, lo que lo hace impráctico para su implementación con propósitos comerciales, por ejemplo. Esto ha movido a la comunidad criptográfica a realizar un gran trabajo de investigación para encontrar algoritmos que calculen apareamientos de forma eficiente.

Definición 3.3 *La* bilinealidad *significa que el mapeo e es lineal en ambos componentes, lo que se refiere a las siguientes dos propiedades:*

$$e(R_1 + R_2, Q) = e(R_1, Q)e(R_2, Q) \tag{3.1}$$

$$e(R, Q_1 + Q_2) = e(R, Q_1)e(R; Q_2) \tag{3.2}$$

Una consecuencia de esas dos propiedades es el siguiente hecho, que es justamente la novedad algebraica sobre la que están fundamentados todos los esquemas de criptografía basados en apareamientos.

Sean $a, b \in \mathbb{Z}_p$

$$e(aP_1, bP_2) = e(bP_1, aP_2) = e(P_1, P_2)^{ab} \tag{3.3}$$

Cabe señalar, que debe haber un requerimiento adicional sobre el mapeo e, para su uso en Criptografía. Ciertos problemas computacionales asociados con los elementos de G_1, G_2 y G_T deben ser computacionalmente difíciles. Con esto nos referimos al hecho de que hasta el día de hoy no se conocen algoritmos probabilísticos para resolverlos. En la comunidad se tiene la creencia de que tales problemas de hecho no se pueden resolver

en tiempo polinomial, pero actualmente, se está muy lejos de probar tal afirmación. A partir de los conceptos matemáticos del capítulo anterior, los ampliaremos con conceptos importantes.

3.2. Exponenciación en Grupos Generales Cíclicos

Definición 3.4 *Sea $\langle g \rangle$ un grupo cíclico multiplicativo de orden q. La exponenciación en $\langle g \rangle$ se refiere a la siguiente operación: dado un $a \in \mathbb{Z}_q$, calcular $h = g^a$.*
Sea $a = a_{n-1} \ldots a_o$, donde $n = \lceil \log_2 q \rceil$ y cada a_i sea un bit. Hay dos métodos simples denominados eleva al cuadrado y multiplica *para llevar a cabo este cálculo. Uno de ellos trabaja de derecha a izquierda, esto es desde a_0 hasta a_{n-1} y el otro en sentido contrario, esto es desde a_{n-1} hasta a_0. El primero de ellos, de derecha a izquierda, se explica con el siguiente algoritmo:*

$n \leftarrow 1, h \leftarrow g^{a_0}$

$n \leftarrow 2, h \leftarrow g^{2a_1+a_0} \leftarrow (g^2)^{a_1} \times g^{a_0}$

$t \leftarrow g, r \leftarrow g^{a_0}$

$t \leftarrow t^2, r \leftarrow^{a_1} \times r; h \leftarrow r$

en el i^{esimo} paso, elevar t al cuadrado,

if $a_i = 1$ **then**

$t \times r$

end if

El segundo, de izquierda a derecha, funciona de la siguiente manera:

$n \leftarrow 1, h \leftarrow g^{a_0}$

$n \leftarrow 2, h \leftarrow g^{2a_1+a_0} \leftarrow (g^2)^{a_1} \times g^{a_0}$

$r \leftarrow g^{a_1}$

$r \leftarrow r^2 \times g^{a_0} h \leftarrow r$

en el i^{esimo} paso, elevar r al cuadrado,

if $a_{n-i} = 1$ **then**

$r \times g$

end if

La importancia del segundo método, radica en que la multiplicación siempre involucra a g, la cual es constante durante todo el proceso, lo que puede proporcionar ciertos beneficios en la eficiencia para su implementación.

Definición 3.5 *(El problema del logaritmo discreto) El problema del logaritmo discreto sobre un grupo cíclico $\langle g \rangle$ consiste en lo siguiente: Dados g y $h \in \langle g \rangle$, encontrar a tal que $h = g^a$. A esta a se le conoce como el* logaritmo discreto *de h respecto a la base g y se escribe como* $\log_a h = a$.

Supongamos que n es el logaritmo en base 2 del tamaño de $\langle g \rangle$. Para que un grupo cíclico tenga utilidad en Criptografía, el problema del logaritmo discreto sobre el grupo debería ser computacionalmente difícil. Recordemos que esto significa que no existe un algoritmo con tiempo polinomial de ejecución en n. Por fuerza bruta, podríamos revisar cada uno de los elementos de $\langle g \rangle$ para encontrar al a requerido, pero esto tiene un costo en el tiempo de ejecución de $O(2^n)$, es decir, exponencial. Existen algoritmos genéricos para encontrar logaritmos discretos como el *rho de Pollard* que se ejecuta en tiempo $O(2^{n/2})$, el cual es mucho mejor que el de fuerza bruta, pero aún así es exponencial.

Se cree que el problema del logaritmo discreto sobre ciertos grupos que se obtienen de las curvas elípticas, es computacionalmente difícil y por lo tanto se utilizan para la implementación de operaciones criptográficas. Cabe señalar otros problemas computacionales importantes. Uno de ellos, el problema *Diffie-Hellman* (CDH), la instancia consiste de una triada (g, g^a, g^b) y el requerimiento es calcular g^{ab}. El otro, llamado

problema de decisión de Diffie-Hellman (DDH), es distinguir entre dos distribuciones (g, g^a, g^b, g^{ab}) y (g, g^a, g^b, g^c) para a, b y c.

3.3. El apareamiento Tate

Definición 3.6 *Sea K un campo finito y E una curva elíptica sobre K. Sea L una extensión finita de K. El grupo de divisores de $E(L)$ es el grupo abeliano libre generado por los puntos de $E(L)$. Por lo tanto, cualquier divisor D sobre L es de la forma*

$$D = \sum_{P \in E(L)} n_p(P) \tag{3.4}$$

Donde $n_P \in \mathbb{Z}$ y $n_P = 0$ excepto por una cantidad finita muy grande de P. El grado de un divisor se define como $\sum n_p$ y se dice que el divisor es de grado cero si $\sum n_p = 0$

Definición 3.7 *Una función racional sobre una curva elíptica. Para la forma de Weierstrass, se puede entender como el cociente de dos polinomios sobre la curva. Por lo que el divisor de una función racional f sobre L se define como*

$$div(f) = \sum_{P \in E(L)} ord_P(f)(P) \tag{3.5}$$

Donde $ord_P(f)$ es el orden del polo/cero que f tiene en P. Un divisor D se dice que es principal *si $D = div(f)$ para una función racional f.*

Hay una relación importante entre los divisores principales y las funciones racionales. Un divisor (sobre L) $D = \sum_{P \in E(L)} n_P(P)$ es principal si y solo si $\sum n_P = 0$ y $\sum n_P P = \mathcal{O}$. Se dice que dos divisores D_1 y D_2 son equivalentes ($D_1 \sim D_2$) si $D_1 - D_2$ es principal. Por otro lado, cualquier divisor $D = \sum n_P(P)$ de grado cero es equivalente a un único divisor de la forma $\langle Q \rangle - langle\mathcal{O}\rangle$ para algún $\mathcal{Q} \in E(L)$. Si $P = (x, y)$ entonces por

$f(P)$ se quiere indicar que $f(x, y)$.

Si L es una extensión de grado k de K, entonces el conjunto $\mu_r(L)$ se define como el subgrupo cíclico de L^* de orden r, que es primo y $r|(\#L - 1)$. El conjunto de los n-puntos de torsión L-racionales de E sobre K se define como el conjunto:

$$E(L)[n] = \{P \in E(L) : nP = \mathcal{O}\}. \tag{3.6}$$

Se define a $E(L)/rE(L)$ como la colección de todas las clases laterales de $E(L)$ módulo $rE(L)$ y $f_{s,P}$ como una función L-racional con divisor:

$$div(f_{s,P} = s(P) - ([s]P)) - (s - 1)(\mathcal{O}). \tag{3.7}$$

Sea E una curva elíptica definida sobre $\mathbb{F}_q$, sea r coprimo a q y $r|\#E\mathbb{F}_q$. El grado inherente de E con respecto a r se define, bajo ciertas condiciones, como el entero positivo más pequeño k tal que $r|(q^k - 1)$. En este caso, K es también el entero positivo más pequeño tal que el campo $\mathbb{F}_{q^k}$ que contiene todas las r^{esimas} raíces de unidad.

Definición 3.8 *Sea k el grado inherente de $E/\mathbb{F}_q$ con respecto a r. El* apareamiento Tate *(reducido y normalizado) se define de la siguiente manera:*

$$e: \ E(\mathbb{F}_q)[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \to \mu_r(\mathbb{F}_{q^k}) \tag{3.8}$$

Está dado por:

$$e(P, Q) = f_{r,P}(Q)^{(q^k-1)/r}, \tag{3.9}$$

Donde

- *P es un $r-$punto de torsión de $E(\mathbb{F}_{q^k})$*
- *Q es un punto en una clase lateral en $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$*
- *El resultado es un elemento de $\mathbb{F}_{q^k}$ de orden r.*

Cabe señalar que P pertenece a $E(K)$, mientras que Q pertenece a E(L). Donde K es un campo finito y L es una extensión de grado k de L. Puesto que P es un r-punto de torsión, se sigue que $rP = \mathcal{O}$ y por lo tanto:

$$div(f_{r,P}) = r(P) - ([r]P) - (r-1)(\mathcal{O}) \tag{3.10}$$

$$= r(P) - r(\mathcal{O}). \tag{3.11}$$

3.4. Cálculo del apareamiento Tate y el algoritmo de Miller

El cálculo de $f_{s,Pp}$ se lleva a cabo mendiante un algoritmo de *doblado y suma* similar al de la multiplicación escalar. Supongamos que E está dada en la forma de *Weierstrass*. Sean P y R dos puntos en E. Se definen las siguientes funciones racionales y sus divisores.

1. $l_{P,R}(R \neq P)$ es la línea que pasa a través de P, R y $-(P+R)$ $div(l_{P,R}) = (P) + (R) + (-(P+R)) - 3(\mathcal{O})$.
2. $l_{R,R}$ es la línea que pasa a través de R y de $-2R$ $div(l_{R,R}) = 2(R)+(-2R)-3(\mathcal{O})$
3. $l_{R,R}$ es la línea que pasa a través de R y $-R$ $div(l_{R,-R}) = (R) + (-R) - 2(\mathcal{O})$
4. $h_{P,R}$ para $R \neq P$ se define como $h_{P,R} = l_{P,R}/l_{T,-T}$ donde $T = R + P$ $div(h_{P,R}) = (l_{P,R}) - (l_{T,-T})$
5. $h_{R,R}$ se define como $h_{R,R} = l_{R,R}/l_{T,-T}$ donde $T = 2R$

$$div(h_{R,R}) = (l_{R,R}) - (l_{T,-T})$$

Es de notar, que $div(f_1, P) = (P) - (P) = 0$ y así mismo $f_1, P = 1$. Una ecuación de recurrencia para $f_{s,P}$ se puede obtener como sigue:

$$\begin{aligned} div(f_{2m,P}) &= 2m(P) - (2mP) - (2m-1)(\mathcal{O}) \\ &= 2(m(P) - (mP) - (m-1)(\mathcal{O})) + 2(mP) - (2mP) - (\mathcal{O}) \\ &= 2div(f_{m,P}) + 2(mP) + (-2mP) - 3(\mathbb{O}) \\ &\quad -((2mP) + (-2mP) - 2(\mathcal{O})) \\ &= 2div(f_{m,P}) + div(l_{mP,mP}) - div(l_{2mP,-2mP}) \\ &= 2div(f_{m,P}) + div(h_{mP,mP}). \end{aligned}$$

$$\begin{aligned} div(f_{2m+1,P}) &= (2m+1)(P) - ((2m+1)P) - 2m(\mathcal{O}) \\ &= 2(m(P) - (2mP) - (2m-1)(\mathcal{O})) + (P) + 2(mP) \\ &\quad -((2m+1)P) - (\mathcal{O}) \\ &= 2div(f_{2m,P}) + (P) + (2mP) + (-(2m+1)P) - 3(\mathcal{O}) \\ &\quad -(((2m+1)P) + (-(2m+1)P) - 2(\mathcal{O})) \\ &= div(f_{2m,P}) + div(l_{2mP,P}) - div(l_{(2m+1)P,-(2m+1)P}) \\ &= div(f_{2m,P}) + div(h_{P,2mP}). \end{aligned}$$

De esta manera, tenemos que $div(f_{2m,P}) = 2div(f_{m,P}) + div(h_{mP,mP})$, de lo que obtenemos: $f_{2m,P} = f_{m.P}^2 \times h_{mP,mP}$

De forma similar, $div(f_{2m+1,P}) = f_{2m,P} \times h_{P,2mP}$

El cálculo del apareamiento Tate, se reduce a la siguiente tarea:

Dado $P \in E(K)$ y $Q \in E(L)$, calcular $f_{r,P}(Q)$. Esto se lleva a cabo mediante el algoritmo de Miller.

Definición 3.9 *Sean $r_{t-1}r_{t-2}\ldots r_0$ la expansión binaria de r*

$f \leftarrow 1$

Calcular rP de izquierda a derecha usando doblar y sumar

Sea R la entrada de la i-ésima iteración

$f \leftarrow f^2 \times h_{R,R}(Q); R \leftarrow 2R$

if $r_{n-i} = 1$ **then**

$f \leftarrow f \times h_{R,P}(Q)$

end if

$R \leftarrow R + P$

A esto se le conoce como la operación de Miller*. La R final que se obtiene después de la iteración completa del ciclo se eleva a la potencia $(q^k - 1)/r$ para obtener un elemento único en $\mu_r(L)$. Esta es la parte de la exponenciación final en el apareamiento reducido de Tate.*

A manera de síntesis, cabe recordar que los apareamientos bilineales son funciones que mapean un par de puntos de una curva a un elemento del grupo multiplicativo de un campo finito y han sido utilizados en varios contextos. El primero de ellos fue el apareamiento Weil y posteriormente el apareamiento Tate aunque de mayor complejidad, es más eficiente, pues solo requiere de una aplicación del algoritmo de Miller, en lugar de las dos que requiere el apareamiento Weil.

Actualmente, los mejores métodos para el cálculo de los apareamientos bilineales se basan en el algoritmo de Miller, que de hecho, es un estándar. La optimización del ciclo principal y la invocación de la exponenciación final son temas actuales de investigación.

Capítulo 4

Desarrollo e Implementación

Primero aprende Computación y toda la teoría.

Después desarrolla un estilo de programación.

Entonces, olvídalo todo y hackea.

George Carrette

4.1. Introducción

En este capítulo se explican los detalles de implementación del sistema realizado.

Se han estudiado métodos para implementar esquemas criptográficos de curvas elípticas e hiper-elípticas de forma eficiente y segura por más de veinte años [14]. Aún cuando la aritmética en curvas hiper-elípticas es en ciertos casos, conceptualmente simple y por lo tanto fácil de entender, se siguen realizando descubrimientos que mejoran el rendimiento de manera significativa. Por ejemplo, uno estos casos hace uso de las coordenadas de Edwards para acelerar la suma y el doblado de puntos en una curva elíptica. [4]

Dado que la aritmética de los apareamientos es considerablemente más complicada que la aritmética de las curvas elípticas convencionales, no sorprende que haya

múltiples investigaciones actualmente para mejorar el rendimiento de los protocolos basados en apareamientos. Existen numerosas propuestas para definir y llevar a cabo apareamientos adecuados para su uso con fines criptográficos. $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Esto tiene como consecuencia que para su implementación, los desarrolladores se enfrenten a una extraordinaria cantidad de parámetros que se pueden establecer. Entre estas opciones se encuentran el grado inherente, el género y el tipo de curva (ordinarias o supersingulares), la característica del campo subyacente, sin olvidar los grupos de orden primo $\mathbb{G}_1$ y $\mathbb{G}_2$.

Así mismo, los métodos de implementación para el cálculo de los apareamientos utilizados han sido fuente de varias investigaciones, lo que ha tenido como consecuencia, mejoras importantes en el rendimiento, un ejemplo, se describe en [35].

A continuación se comentará brevemente sobre las bibliotecas utilizadas en la implementación de los criptosistemas implementados en la presente tesis.

4.2. Hardware utilizado

El hardware utilizado es una máquina MacBook Pro con las siguientes características:

- Procesador Intel Core 2 DUO.
- 4 GB de RAM, a 1067 MHz DDR3.

4.3. Software adicional utilizado

La computadora anteriormente descrita, dispone de:

- Sistema Operativo Mac OS X 10.6.8.
- Compilador gcc versión 4.2.1, incluido en el entorno de desarrollo integrado Xcode versión 3.2.5.

- La herramienta de control de generación de ejecutables GNU Make versión 3.81.

4.4. La biblioteca GMP

La biblioteca GNU GMP es una biblioteca para manejo de Aritmética de Precisión Múltiple sobre enteros, números racionales y números de punto flotante [34]. También provee la aritmética más veloz para aquellas aplicaciones que requieren una precisión mayor a la que está directamente soportada por los tipos básicos en el lenguaje C. Esto es particularmente cierto en aplicaciones criptográficas. La biblioteca GMP está diseñada para proveer con un muy buen rendimiento en este tipo de aplicaciones y otras más, al elegir algoritmos basados en el tamaño de los operandos y mantener el consumo de recursos en general al mínimo.

4.4.1. Procedimiento de instalación

Se descargó el código fuente de la página http://gmp.org. La versión utilizada es la 5.0.1, disponible desde febrero de 2010 como un archivo comprimido: gmp-5.01.tar.bz2
El archivo descargado se descomprimió y generó un directorio de nombre gmp-5.01.
De acuerdo con el manual de GMP [34], la biblioteca tiene un sistema de configuración basado en las herramientas autoconf/automake/libtool. En un sistema UNIX, como lo es la plataforma anteriormente descrita se llevó a cabo la siguiente secuencia de comandos, dentro del directorio recientemente creado.

```
./configure
```

```
make
```

```
make check
```

Finalmente, se realizó la instalación de la biblioteca en el directorio /usr/local mediante el siguiente comando:

```
make install
```

4.4.2. Archivos de encabezado y compilación

Toda aplicación que haga uso de la biblioteca debe incluir el archivo de encabezado gmp.h a través de la siguiente declaración para el pre-procesador de C:

```
#include <gmp.h>
```

Por otro lado, para compilar un programa se debe indicar al compilador que se utilizará la biblioteca gmp, mediante la opción -gmp:

```
gcc mi_aplicacion.c -lgmp
```

4.4.3. Ejemplo de uso

La mejor manera de explicar brevemente el uso de las funciones de la biblioteca GMP es a través de uno de los ejemplos incluidos en el subdirectorio demo, cuyo código se presenta en la figura 4.1.

Los puntos importantes a señalar son los siguientes:

- La inclusión del archivo gmp.h en la línea 4.
- La declaración de la variable `n` de tipo `mpz_t`, en la línea 19. Este es el tipo de dato que se utiliza para almacenar un entero de precisión múltiple de acuerdo con la biblioteca GMP.

```c
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <gmp.h>

char *nomprog;

voidprint_usage_and_exit ( ){
  fprintf (stderr, "uso: %s -q nnn\n", nomprog);
  fprintf (stderr, "uso: %s nnn ...\n", nomprog);
  exit (-1);
}

int main (int argc, char **argv){
  mpz_t n;
  int i;
  nomprog = argv[0];
  if (argc < 2)
    print_usage_and_exit ();
  mpz_init (n);
  if (argc == 3 && strcmp (argv[1], "-q") == 0){
      if (mpz_set_str (n, argv[2], 0) != 0)
        print_usage_and_exit ();
      exit (mpz_probab_prime_p (n, 5) == 0);
    }

  for (i = 1; i < argc; i++){
      int class;
      if (mpz_set_str (n, argv[i], 0) != 0)
        print_usage_and_exit ();
      class = mpz_probab_prime_p (n, 5);
      mpz_out_str (stdout, 10, n);
      if (class == 0)
        puts (" es un numero compuesto");
      else if (class == 1)
        puts (" es un primo probable");
      else /* class == 2 */
        puts (" es un primo");
    }
  exit (0);
}
```

Figura 4.1: Ejemplo de uso de la biblioteca GMP

- La invocación de la función `mpz_init` en la línea 27. Esta función inicializa a su argumento, que en este caso es la variable `n`, a cero.
- La invocación de la función `mpz_set_str` en la línea 39. Esta función establece el valor de la variable `n`, con el valor de la cadena que toma como segundo argumento, para nuestro caso es el primer argumento de la línea de comando que se pasó en la invocación del programa, como el tercer argumento es un cero, indica que se debe tomar la base a partir de los primeros caracteres leídos, esto nos permite pasar números escritos en notación binaria, octal, hexadecimal, o bien decimal, si no se hace ninguna indicación especial
- La invocación de la función `mpz_probab_prime` en la línea 41. Esta función determina si un número es primo y puede regresar uno de tres resultados posibles: Devuelve un 2 cuando el número examinado es primo de manera concluyente, un 1 si es probablemente primo y 0 si es un número compuesto. Este proceso se lleva a cabo al realizar unas cuantas divisiones y a continuación base en la prueba de primalidad probabilística de Miller-Rabin. El segundo argumento controla cuantas de estas pruebas se llevan a cabo, de 5 a 10 es un número razonable, y si se especifica un número mayor, entonces se reducirán las probabilidades de que un número compuesto se identifique como *probable primo*.
- La invocación de la función `mpz_out_str` en la línea 42, despliega su argumento en el flujo de salida, en este caso la salida estándar, en la base especificada, que para el ejemplo, es 10.

4.4.4. Funciones Importantes

La biblioteca GMP proporciona varias funciones sobre enteros de precisión múltiple, en diferentes categorías:

- Funciones de inicialización
- Funciones de asignación
- Funciones de conversión
- Funciones Aritméticas
- Funciones de división
- Funciones para exponenciación
- Funciones para extracción de raíces
- Funciones relacionadas con la Teoría de Números
- Funciones de comparación
- Funciones lógicas y de manipulación de bits
- Funciones de entrada y salida
- Funciones de generación de números aleatorios
- Funciones misceláneas y especiales

4.5. La biblioteca PBC

PBC es una biblioteca libre escrita en lenguaje C que permite el prototipado rápido de sistemas criptográficos basados en apareamientos[26]. Provee una interfaz abstracta a un grupo cíclico con un apareamiento bilineal. La biblioteca PBC está construida encima de la biblioteca GMP. Su API está fuertemente influenciada por la API correspondiente de GMP.

4.5.1. Instalación

De la página http://crypto.stanford.edu/pbc/ se obtuvo el archivo pbc-0.5.11.tar.gz, se descomprimió el archivo, lo que generó un directorio, donominado pbc-0.5.11. Mediante una consola de UNIX se llevó a cabo la instalación del sistema con los siguientes

comandos:

```
./configure

make

make install
```

4.5.2. Ejemplo de uso

Uno de los ejemplos de uso, es una calculadora que permite experimentar con los apareamientos. En la figura 4.2 aparece el ejemplo que se llevó a cabo.

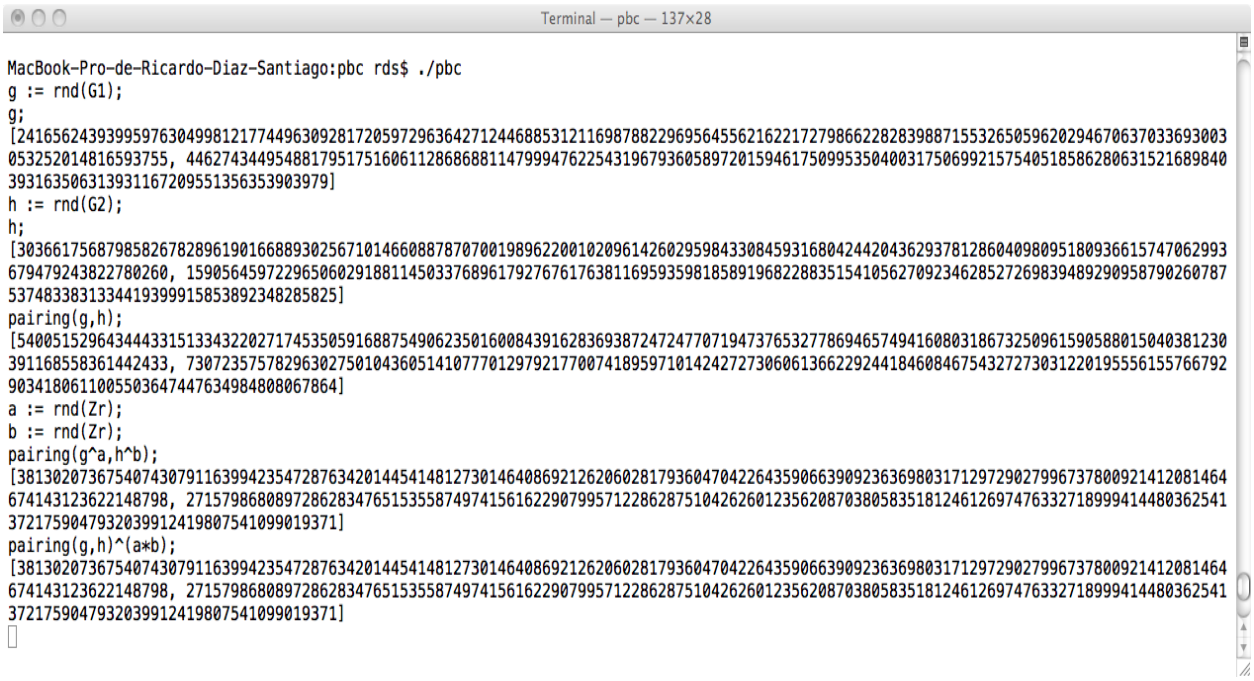


Figura 4.2: Ejemplo de uso de la calculadora de apareamientos

La secuencia de pasos realizados es:

- Se genera un elemento aleatorio g del grupo G1, al que se llama g.
- Se asigna a h, un elemento aleatorio, ahora del grupo G2.

- Se realiza el apareamiento.
- Cabe señalar que el orden tanto de g como de h, es r.
- Se generan dos números aleatorios entre 1 y r, los cuales se asignan a a y b, respectivamente.
- Para comprobar, la bilinealidad del apareamiento, se ejecutan las dos siguientes operaciones:
- `pairing(g\^a,h^b);`
- `pairing(g,h)^(a*b);`

4.6. El Protocolo tripartita de Diffie Hellman

Este protocolo tiene como intención llevar a cabo una generalización del protocolo de Diffie-Hellman, que involucre tres participantes, A, B, C, requiere una sola ronda de comunicación y permite la construcción de un secreto común $K_{A,B,C}$. Esta variante fué presentada por primera vez por Joux [19], y en [20] se presentó una versión revisada y actualizada.

4.6.1. Planteamiento general

La idea con este protocolo es similar al protocolo ordinario de Diffie-Hellman, involucra tres participantes A, B, C, requiere que en un solo paso de comunicación se logre obtener un secreto compartido $K_{A,B,C}$.A partir de una curva elíptica E y algunos l-puntos de torsión P, los tres participantes a los que se denominará A, B y C, cada uno elige un número aleatorio (a, b, c), calcula $P_A = aP$, $P_B = bP$ $P_C = cP$ y se difunden estos valores. A partir de ellos, cada uno calcula F(a,P_B,P_C), F(b,P_A,P_C) y F(c,P_A,P_B), donde la función F se debe elegir de tal manera que asegure que estos números serán iguales y que este valor común $K_{A,B,C}$, será un valor difícil de calcular

dados P_A, P_B y P_C. Claramente, el problema es definir esta F. La figura 4.3, tomada de [5], ilustra el proceso.

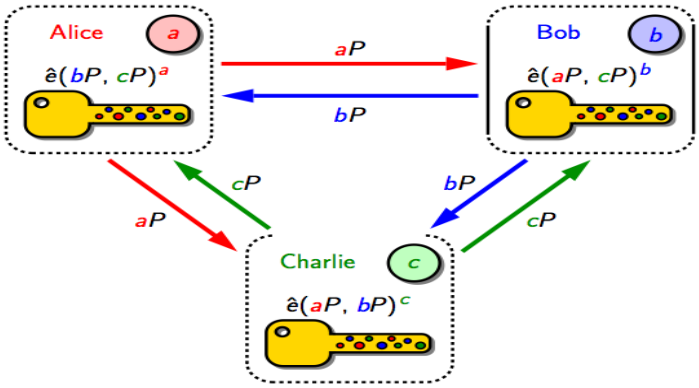


Figura 4.3: Intercambio de claves tripartita

4.6.2. Una solución a partir del apareamiento Tate

En su artículo, Joux [19], inicialmente propone el uso del apareamiento Weil, y posteriormente utiliza el apareamiento Tate, por ser más eficiente. En este caso, la función F se define de la siguiente manera

$$F_T(x, D_1, D_2) = t_l(D_1, D_2)^x \tag{4.1}$$

Se eligen dos puntos independientes P y Q en la $l-$torsión de la curva E. Los tres participantes A, B y C calculan y envian (P_A, Q_A), (P_B, Q_B), (P_C, Q_C), respectivamente, ahora se definen dos divisores para cada usuario.La técnica más simple es que por cada punto X en la curva, se defina $D_X = (X) - (0)$. Ahora, A, B y C cada uno calcula

respectivamente:

$$F_T(a, D_{P_B}, D_{Q_C}) = F_T(a, D_{P_C}, D_{Q_B}), \tag{4.2}$$

$$F_T(b, D_{P_A}, D_{Q_C}) = F_T(a, D_{P_C}, D_{Q_A}), \tag{4.3}$$

$$F_T(c, D_{P_B}, D_{Q_A}) = F_T(a, D_{P_A}, D_{Q_B}). \tag{4.4}$$

Cabe señalar, sin embargo, que se debe ser cuidadoso al evaluar F_T. De hecho, se debe elegir un representante diferente por cada una de las clases del divisor, pues de lo contrario, en algún punto durante el cálculo del apareamiento Tate, se dividiría cero entre cero, provocando que el cálculo falle y devuelva un resultado indefinido. Para evitar esto, se reemplaza D_Q por el divisor equivalente $(Q + R) - (R)$, donde R es algún punto elegido de forma aleatoria.

Un planteamiento diferente para evitar esta dificultad es elegir diferentes divisores. Para un usuario U, se define

$$D_1^U = (P_U) - (Q_U) \equiv D_{P_U} - D_{Q_U}, \tag{4.5}$$

$$D_2^U = (P_U + Q_U) - (0) \equiv D_{P_U} + D_{Q_U}. \tag{4.6}$$

Ahora, tanto A, B y C calculan respectivamente

$$F_T(a, D_1^{(B)}, D_2^{(C)}) = F_T(a, D_1^{(C)}, D_2^{(B)}), \tag{4.7}$$

$$F_T(b, D_1^{(A)}, D_2^{(C)}) = F_T(b, D_1^{(C)}, D_2^{(A)}), \tag{4.8}$$

$$F_T(c, D_1^{(B)}, D_2^{(A)}) = F_T(c, D_1^{(A)}, D_2^{(B)}). \tag{4.9}$$

Por la bilinealidad del apareamiento, los tres números son iguales.

Es importante revisar que el apareamiento no sea degenerado. Esto se logra al revisar

que $t_l(D_P, D_Q)$ o $t_l(D_P - D_Q, D_P + D_Q)$no es igual a 1. De esta manera, el valor común está uniformemente distribuido. En el segundo artículo de Joux [20], explica las razones por las que es más rápido evaluar la función F_T basada en el apareamiento Tate, en lugar del apareamiento Weil.

4.7. Implementación en C

En la figuras 4.4 y 4.5 se muestra la codificación del protocolo de Joux para compartir un secreto tripartita.

4.8. Resultados

En las figuras 4.6 y 4.7, se muestran dos ejecuciones de la codificación del Protocolo de Joux para el secreto tripartita

```c
#include <pbc.h>
#include <pbc_test.h>
int main(int argc, char **argv) {
  pairing_t pairing;
  double time1, time2;
  element_t P, a, b, c, Ka, Kb, Kc, t1, t2, t3, t4, t5, t6;
  pbc_demo_pairing_init(pairing, argc, argv);
  if (!pairing_is_symmetric(pairing))
     pbc_die("El apareamiento debe ser simetrico");
  element_init_G1(P, pairing);
  element_init_G1(t1, pairing);
  element_init_G1(t2, pairing);
  element_init_G1(t3, pairing);
  element_init_Zr(a, pairing);
  element_init_Zr(b, pairing);
  element_init_Zr(c, pairing);
  element_init_GT(t4, pairing);
  element_init_GT(t5, pairing);
  element_init_GT(t6, pairing);
  element_init_GT(Ka, pairing);
  element_init_GT(Kb, pairing);
  element_init_GT(Kc, pairing);
  time1 = pbc_get_time();
  printf("Protocolo compart. de clave entre A, B y C.\n");
  element_random(P);
  element_random(a);
  element_random(b);
  element_random(c);
  element_mul_zn(t1, P, a);
  printf("A le envia a B y a C: aP\n");
  element_printf("aP = %B\n", t1);
  element_mul_zn(t2, P, b);
  printf("B le envia a A y a C: bP\n");
  element_printf("bP = %B\n", t2);
  element_mul_zn(t3, P, c);
  printf("C le envia a  A y a B: cP\n");
  element_printf("cP = %B\n", t3);
```

Figura 4.4: Codificación del protocolo tripartita de Diffie-Hellman en C

```
1   element_pairing(t4, t2, t3);
2   element_pow_zn(Ka, t4, a);
3   element_printf("Ka = %B\n", Ka);
4   element_pairing(t5, t1, t3);
5   element_pow_zn(Kb, t5, b);
6   element_printf("Kb = %B\n", Kb);
7   element_pairing(t6, t1, t2);
8   element_pow_zn(Kc, t6, c);
9   element_printf("Kc = %B\n", Kc);
10
11
12  printf("Secreto compartido K = Ka = Kb = Kc\n");
13  time2 = pbc_get_time();
14  printf("Tiempo total = %fs\n", time2 - time1);
15
16  element_clear(P);
17  element_clear(a);
18  element_clear(b);
19  element_clear(c);
20  element_clear(Ka);
21  element_clear(Kb);
22  element_clear(Kc);
23  element_clear(t1);
24  element_clear(t2);
25  element_clear(t3);
26  element_clear(t4);
27  element_clear(t5);
28  element_clear(t6);
29  pairing_clear(pairing);
30  return 0;
31 }
```

Figura 4.5: Codificación del protocolo tripartita de Diffie-Hellman en C

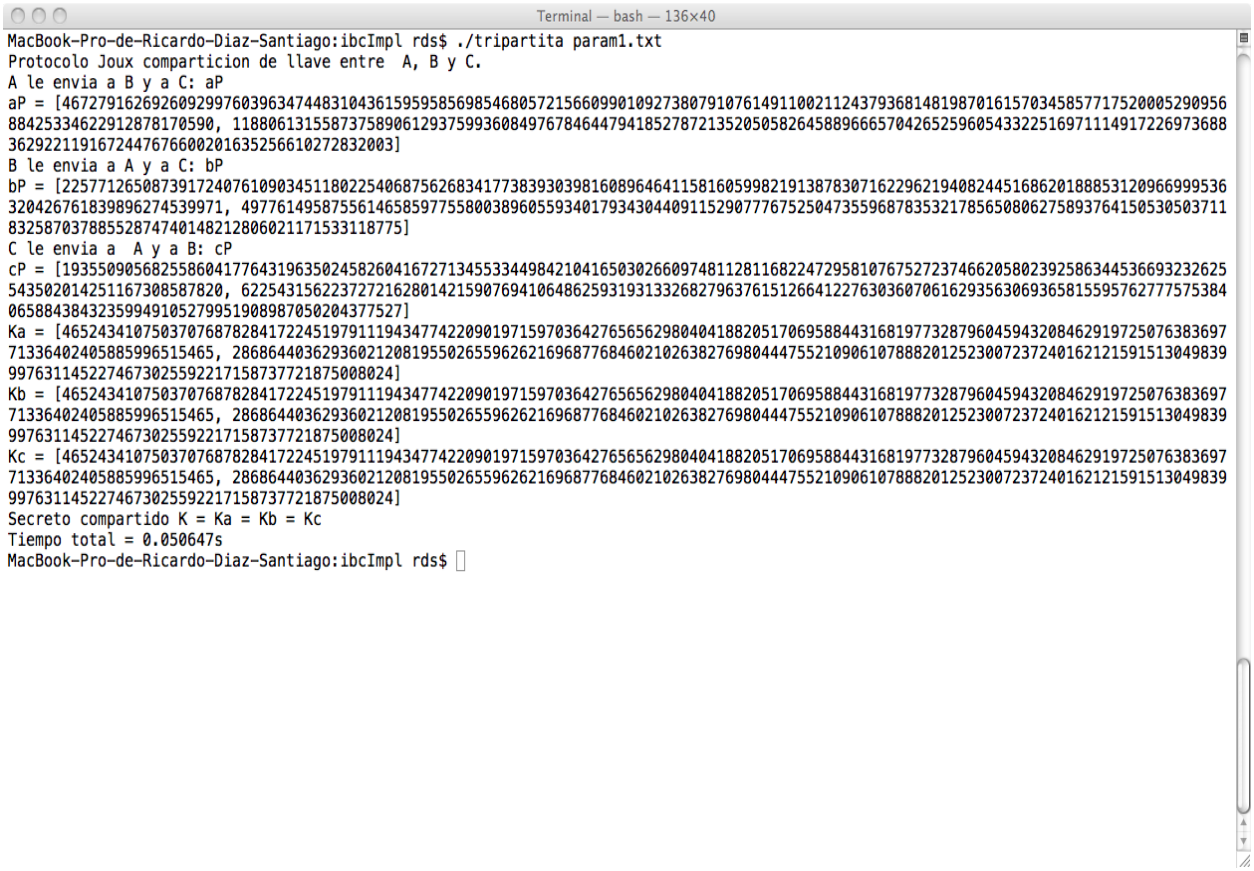


Figura 4.6: Primera ejecución del protocolo de secreto tripartita

Figura 4.7: Segunda ejecución del protocolo de secreto tripartita

En ambos casos, se utilizó la curva $y^2 = x^3 + x$ sobre el campo F_q con un q diferente en cada caso. El apareamiento que se utilizó es simétrico, tanto G_1 como G_2 está formado por el grupo de puntos $E(F_q)$. En este caso ocurre que $\#E(F_q) = q+1$ y $\#E(F_q)^2 = (q+1)^2$. Por lo tanto el grado de seguridad es 2 y G_T es un subgrupo de F_{q^2}.

Para el primero de los casos, se utilizó la curva $y^2 = x^3 + x$ sobre el campo F_q con

q=878071079966331252243778198475404981580688319941420821102865339926647563088022295707862517942266222142315585876958231745927771336731748132492512999822479 1.

En la segunda ejecución,

q=485128758963037524997122772545896285164193521882945211981895675110

0907315811504536129483934709931589896004539852468200733416492853159479914910054803644576011091315742065569036189129085844136080715824725946050134344919971253282806394000868374004850098044198971373968965561057845838812693424263055739761877653925 9.

Para probar que el cifrado es seguro, sería recomendable como trabajo a futuro realizar ataques tales como: Ataques con texto plano escogido (CPA por sus siglas en inglés). Ataques con texto cifrado (CCA por sus siglas en inglés).

Por otro lado, es importante la elección de la curva elíptica, el orden de los grupos que se utilizan en el apareamiento, el grado de encajamiento, también denominado grado de seguridad, y sobre todo, está la suposición de que resolver el problema del logaritmo discreto es inherentemente difícil sobre estos grupos.

Esto se puede presentar cuando un texto se pasa a un punto (a un elemento de los grupos G_1 o G_2

A manera de epílogo, cabe hacer énfasis en las ventajas de los apareamientos bilineales en Criptografía:

- Permiten resolver problemas de manera conceptualmente más sencilla, como el secreto compartido entre tres participantes en una sola ronda de comunicación.
- Al estar basados en curvas elípticas, requieren una cantidad menor de bits, para lograr el mismo nivel de seguridad que otros métodos comparables ya establecidos.

El uso de bibliotecas permite explorar nuevos protocolos criptográficos, pues se cuenta con código modularizado que es posible parametrizar, por el tipo de curva y el campo finito del cual se tomarán los puntos.

Capítulo 5

Conclusiones y Trabajo a Futuro

Vas a salir de tu incertidumbre a una graz paz y libertad.

Texto en una galleta de la suerte

5.1. Conclusiones

La Criptografía es una disciplina dinámica, continuamente encuentra uso en nuevas aplicaciones, lo que implica nuevos desarrollos basados en diferentes paradigmas matemáticos. Uno de estos casos son los apareamientos bilineales.

El objetivo general planteado al inicio de esta tesis, se cumplió, al implementar el protocolo tripartita de Joux, para el intercambio de claves públicas en un solo paso de comunicación, proporcionó un excelente ejemplo del uso de los apareamientos bilineales para resolver un problema que de otra manera requiere de un proceso más complejo, que requiere al menos dos pasos de comunicación entre los participantes.

Para poder llevarlo a cabo, se realizó el estudio de la Matemática necesaria para tratar con los *apareamientos bilineales*, que incluyó el estudio de las *curvas elípticas*, con lo que se cumplió el primero de los objetivos especiíficos.

El segundo de los objetivos específicos se cumplió al utilizar el lenguaje C, la bibliote-

ca GMP que proporciona funciones optimizadas para el manejo de números grandes y es de uso común en aplicaciones criptográficas, así como la biblioteca PBC que proporcionó las funciones para el manejo de apareamientos bilineales.

5.2. Trabajo a Futuro

- Implementación de apareamientos en dispositivos programables.
- Uso de curvas hiper-elípticas, Kawazoe [23] plantea formas de construcción de estas curvas para su uso con apareamientos.
- Aplicación de los apareamientos a formas novedosas de la Criptografía tales como las firmas anónimas. Al usar una firma anónima, cualquier persona que tenga un mensaje firmado, puede convertir la firma en una firma anónima, a través de un tercer agente [13]. Esto puede ser útil, por ejemplo, en el caso de una publicación de un documento gubernamental, que se debe hacer público, en el caso de México, por una petición avalada por el artículo 3 de la Ley Federal de Transparencia y Acceso a la Información Pública [1]. En este caso, el personal responsable de publicar la información, requiere proteger aquellos datos que sean personales, sin embargo, si el documento se trata de un contrato entre el gobierno y un individuo, el requiriente de la información podría requerir comprobar que la firma de los contratos es auténtica. Para resolver esta contradicción, es posible utilizar una de estas firmas anónimas.
- Otra posible aplicación de los apareamientos bilineales es en la creación de las llamadas *firmas ciegas* tal como se propone en [30]. Las firmas ciegas pueden jugar un papel muy importante en aplicaciones tales como el voto electrónico, o los sistemas de efectivo electrónico, también conocidos como *e-cash*.

Bibliografía

[1] *Ley Federal de Transparencia y Acceso a la Información Pública*. Diario Oficial de la Federación, 2002. Última reforma publicada el 06 de julio de 2010. Fecha de consulta: 10 de octubre de 2010.

[2] AMIPCI. *Estudio AMPICI de Comercio Electrónico.* 2009. Disponible en http://estudios.amipci.org.mx. Fecha de consulta: 07 de febrero de 2011.

[3] F. Ayres Jr. and L. R. Jaisingh. *Theory and problems of Abstract Algebra*. Mc Graw Hill, New York, NY, 2nd. edition, 2004.

[4] D. J. Bernstein and T. Lange. Faster addition and doubling on elliptic curves. In *ASIACRYPT 2007, LNCS 4833*, pages 29–50, New York, NY, 2007. Springer.

[5] J.-L. Beuchat. An Introduction to Pairing-based Cryptography. Disponible en http://cs.cinvestav.mx. Fecha de Consulta: 10 de marzo 2011, 2010.

[6] I. F. Blake, G. Seroussi, and N. P. Smart, editors. *Advances in Elliptic Curve Theory*. Cambridge University Press, Cambridge, UK, 2005.

[7] D. Boneh and M. Franklin. Identity Based Encryption from the Weil Pairing. In *Proceedings of Crypto 2001. Lecture Notes in Computer Science*, volume 2139, pages 213–229, New York, NY, 2001. Springer-Verlag.

[8] S. Chatterjee and P. Sarkar. *Identity-Based Encryption*. Springer-Verlag, New York, NY, 2011.

[9] T. S. Denis and S. Johnson. *Cryptography for Developers*. Syngress Publishing Inc., Rockland, MA, 2007.

[10] T. S. Denis and G. Rose. *BigNum Math. Implementing Cryptographic Multiple Precision Arithmetic*. Syngress Publishing Inc., Rockland, MA, 2006.

[11] W. Diffie and M. Hellman. *New Directions in Cryptography*. *IEEE Transactions on Information Theory*, 22:644–654, 1976.

[12] A. Enge. *Elliptic Curves and their application to Cryptography. An introduction*. Kluwer Academic Publishers, Norwell, Masachussetts, 1999.

[13] H. Fumitaka, K. Tetsutaro, and S. Koutarou. Anonymizable Signature and Its Construction from Pairings. In M. Joye, A. Miyajo, and A. Otsuka, editors, *Pairing Based Cryptography 2010. 4th International Conference*, volume LNCS 64897. Springer, 2010.

[14] D. Hankerson, A. Meneses, and M. Scott. *Identity-Based Cryptography*, chapter Software Implementation of Pairings. IOS Press, Amsterdam, The Netherlands, 2009.

[15] D. Hankerson, A. Meneses, and S. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer, New York, NY, 2004.

[16] Harvey E. Rose. *A Course on Finite Groups*. Springer, Bristol, UK, 2009.

[17] I. N. Herstein. *Abstract Algebra*. Prentice Hall, New York, NY, 1996.

[18] Jeffrey Hoffstein and Jill Pipher and Joseph H Silverman. *An Introduction to Mathematical Cryptography*. Undergraduate Texts in Mathematics. Springer, New York, NY, 2008.

[19] A. Joux. A one round protocol for Tripartite Diffie-Hellman. In *Proceedings of ANTS IV, Lecture Notes in Computer Science*, volume 1838, pages 385–393, 2000.

[20] A. Joux. A one round protocol for Tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–276, 2004.

[21] M. Joye, A. Miyajo, and A. Otsuka, editors. *Pairing Based Cryptography 2010. 4th International Conference*, volume LNCS 64897. Springer, 2010.

[22] M. Joye and G. Neven, editors. *Identity-Based Cryptography*. IOS Press, Amsterdam, The Netherlands, 2009.

[23] M. Kawazoe and T. Takahasi. Pairing Friendly Hiperelliptic Curves with Ordinary Jacobians of Type $x^2 = x^5 + ax$. In S. D. Galbraith and K. G. Patterson, editors, *Pairing Based Cryptography 2008. 2nd. International Conference*, volume LNCS 5209. Springer, 2008.

[24] H. Kopka and P. W. Daly. *A Guide to LaTeX. Document Preparation for Beginners and Advanced Users*. Addison Wesley, Harlow, England, 3rd. edition, 1999.

[25] S. Kottwitz. *LaTeX Beginner's Guide*. Packt Publishing Ltd., Birmingham, UK, 2011.

[26] B. Lynn. *PBC library manual*. Stanford University, Palo Alto California, 0.5.11 edition, 2006.

[27] B. Lynn. *On the implementation of pairing-based cryptosystems*. PhD thesis, Stanford University, Palo Alto California, 2007.

[28] A. Meneses, S. Vanstone, and T. Okamoto. Reducing elliptic curve logarithms to logarithms in a finite field. In *STOC 91: Proceedings of the twenty-third annual ACM symposium on Theory of Computing*, pages 80–89, New York, NY, 1991. ACM Press.

[29] N. R. Reilly. *Introduction to Applied Algebraic Systems*. Oxford University Press, New York, NY, 2009.

[30] R. Shakerian, T. MohammadPour, S. H. Kamail, and M. Hedayati. An Identity Based Public Key Cryptography Blind Signature Scheme from Bilinear Pairings. In *IEEE International Conference on Computer Science and Information Technology*, 2010.

[31] A. Shamir. *Identity-based cryptosystems and signature schemes*. In CRYPTO, volume 196 of Lecture Notes in Computer Science, pages 47–53. Springer, 1984.

[32] R. Shirey. *Internet Security Glosary, Request for Coments 2828.* Mayo 1996. Disponible en http://www.ietf.org/rfc/rfc2828.txt. Fecha de Consulta: 7 de Febrero de 2011.

[33] D. R. Stinson. *Cryptography Theory and Practice*. Chapman & Hall/CRC, Boca Raton, Fl., 3rd. edition, 2006.

[34] Torbjörn Granlund and the GMP development team. *The GNU Multiple Precision Arithmetic Library*. GNU Free Software Foundation, 5.0.1 edition, Febrero 2010.

[35] M. Wang, G. Dai, and H. Hu. An efficient generation method of elliptic curve

for pairing-based cryptosystems. In *IEEE International Conference on Machine Vision and Human-Machine Interface*, pages 676–678, 2010.

[36] L. C. Washington. *Elliptic Curves: Number Theory and Cryptography*. Chapman & Hall/CRC, Boca Raton, Fl., 2008.