



UNIVERSIDAD CENTRAL DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA DE INGENIERIA CIVIL EN COMPUTACION E INFORMÁTICA

EFICIENCIA DE LA CRIPTOGRAFÍA DE CURVA ELÍPTICA Y RSA PARA ENFRENTAR LOS NUEVOS REQUERIMIENTOS DE SEGURIDAD EN INTERNET

**CRISTÓBAL CARBONELL JIMÉNEZ
RODRIGO DÍAZ MUÑOZ
PABLO MEJÍAS OSORIO**

Profesor Guía: ERIC DONDEERS ORELLANA

Santiago de Chile, 2007



UNIVERSIDAD CENTRAL DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

ESCUELA DE INGENIERIA CIVIL EN COMPUTACION E INFORMÁTICA

EFICIENCIA DE LA CRIPTOGRAFÍA DE CURVA ELÍPTICA Y RSA PARA ENFRENTAR LOS NUEVOS REQUERIMIENTOS DE SEGURIDAD EN INTERNET

CRISTÓBAL CARBONELL JIMÉNEZ

RODRIGO DÍAZ MUÑOZ

PABLO MEJÍAS OSORIO

Memoria para optar al título de
Ingeniero Civil en Computación
e Informática.

Profesor Guía: ERIC DONDERS ORELLANA

Santiago de Chile, 2007



UNIVERSIDAD CENTRAL DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

ESCUELA DE INGENIERIA CIVIL EN COMPUTACION E INFORMÁTICA

EFICIENCIA DE LA CRIPTOGRAFÍA DE CURVA ELÍPTICA Y RSA PARA ENFRENTAR LOS NUEVOS REQUERIMIENTOS DE SEGURIDAD EN INTERNET

CRISTÓBAL CARBONELL JIMÉNEZ

RODRIGO DÍAZ MUÑOZ

PABLO MEJÍAS OSORIO

Memoria preparada bajo la supervisión de
la comisión integrada por los profesores:

ERIC DONDEERS ORELLANA

RODRIGO ARRIAGADA

CRISTIAN MARTINEZ

Quienes recomiendan que sea aceptada para completar
las exigencias del Título de Ingeniero Civil en
Computación e Informática.

Santiago de Chile, 2007

AGRADECIMIENTOS

Debemos agradecer a varias personas que nos brindaron apoyo y soporte a lo largo de nuestra carrera, tales como Cristián Martínez, Sergio Quezada, Rodrigo Arriagada, Lorena Paredes, Orlando Cavieres y nuestro profesor guía Eric Donders.

Nos gustaría agradecer también a Rubén y Sandrita, por toda la ayuda y grata compañía que nos brindaron.

Agradecemos también a nuestros compañeros y amigos con quienes pudimos compartir muy buenos momentos y la vez, con quienes pudimos aprender a trabajar en equipo para cumplir nuestros objetivos.

DEDICATORIA

*A nuestras familias, cuyo
apoyo fue vital para conseguir
nuestros objetivos.*

RESUMEN

Hoy en día, el manejo de información usando TI tanto para las personas como para las empresas se ha hecho indispensable, transferencias de datos o envío de correos electrónicos por Internet son parte de las tareas que se ejecutan cotidianamente dentro de cualquier organización. Como consecuencia, la información en todas sus formas se ha convertido en un activo de altísimo valor, la cual se debe proteger y asegurar, al punto en que se han creado políticas de aseguramiento de la información tanto a nivel físico en donde el acceso a los equipamientos computacionales es resguardado como a nivel lógico donde entra en juego el ámbito de la criptografía.

Dentro de los métodos criptográficos de cifrado encontramos los sistemas simétricos y los sistemas asimétricos, los cuales difieren en el manejo de las claves que hacen posible sus respectivos procesos de cifrado y descifrado. Ambos tipos criptográficos logran satisfacer algunos de los principios básicos de la seguridad informática como la confidencialidad e integridad, sin embargo, para el caso de los sistemas asimétricos también se logra satisfacer el principio básico de la irrefutabilidad de la información.

Existen dos métodos criptográficos asimétricos que destacan en esta memoria, la criptografía RSA y la criptografía de curvas elípticas. RSA es el método criptográfico más famoso y usado en la actualidad y ha logrado establecerse en la mayoría de los sistemas informáticos seguros, sin embargo, el método de curvas elípticas hoy en día se presenta como una alternativa eficaz para reemplazar a futuro lo que RSA ofrece. La ventaja del método de curvas elípticas por sobre RSA radica en la capacidad de optimizar y reducir los tiempos de procesamiento matemático ofreciendo un mismo nivel de seguridad con menos cálculos, lo que se traduce a cargas de datos menores y mayor rapidez para ejecutar todo tipo de transacciones. Tanto RSA como curvas elípticas se encuentran respaldados por muchos estudios de alto nivel que acreditan ser métodos criptográficos eficientes. Poseen además normativas que regulan su utilización en aplicaciones tanto a nivel gubernamental como comercial, como por ejemplo las normativas aplicadas por los estándares fijados por la IEEE, NIST, ANSI entre otras.

Finalmente se ha desarrollado un estudio del comportamiento de los métodos de cifrado que afirma concluyentemente el impacto en la optimización de equipos computacionales se ve favorecido con el uso de curvas elípticas.

Palabras Claves: Criptografía, RSA, Curvas Elípticas, Estándares, Estudios, Internet.

INDICE

AGRADECIMIENTOS.....	i
DEDICATORIA.....	ii
RESUMEN.....	iii
INDICE.....	iv
INDICE DE FIGURAS	vii
INDICE DE TABLAS.....	ix
I. INTRODUCCIÓN.....	1
II. DESARROLLO DEL TEMA.....	2
2.1 Antecedentes y motivación.....	2
2.2 Descripción del problema.....	3
2.3 Solución propuesta	4
2.4 Objetivos y alcances del proyecto	5
2.4.1 Objetivo general	5
2.4.2 Objetivos específicos.....	5
2.4.3 Alcance del proyecto	6
2.5 Estado del arte	6
2.6 Metodologías y herramientas a utilizar	7
2.7 Desarrollo del aporte ingenieril	9
2.8 Resultados obtenidos	9
2.9 Organización del documento	9
III. FUNDAMENTOS TEÓRICOS	11
3.1 La importancia de la información.....	11
3.2 Seguridad informática.....	11
3.3 Criptografía.....	13
3.4 Criptosistemas	14
3.4.1 Sistemas simétricos	15
3.4.2 Sistemas asimétricos.....	18
3.5 Criptosistema RSA	21
3.5.1 Algoritmo RSA.....	21
3.5.2 Cifrado RSA	23
3.5.3 Descifrado RSA.....	23

3.5.4	Teorema del resto chino para RSA.....	24
3.6	Criptografía de curvas elípticas.....	25
3.6.1	Fundamentos matemáticos de curvas elípticas.....	25
3.6.2	Adición de curvas elípticas.....	26
3.6.3	Curvas elípticas sobre cuerpos finitos	28
3.6.4	Adición de curvas elípticas sobre cuerpos finitos	29
3.6.5	Curvas elípticas sobre F_{2^m}	29
3.6.6	Adición para curvas elípticas sobre F_{2^m}	31
3.6.7	Algoritmo - ECIES	31
IV.	ESTADO DEL ARTE	34
4.1	Seguridad en comercio electrónico.....	34
4.2	Seguridad en correos electrónicos	36
4.3	Seguridad wireless.....	37
4.4	Seguridad en tarjetas magnéticas.....	38
4.5	Seguridad en PDAs y otros dispositivos móviles.....	39
4.6	Seguridad en redes y VPN.....	39
V.	ESTUDIOS.....	41
5.1	NIST	41
5.1.1	Metodología de verificación de algoritmos de NIST	47
5.2	Certicom	49
5.2.1	Certicom y FIPS 140-2	55
5.3	Sun Microsystems	57
5.4	IEEE	62
5.4.1	Criptografía asimétrica tradicional	63
5.4.2	Criptografía asimétrica basada en problemas fuertes en Lattices.....	64
5.4.3	Criptografía asimétrica basada en passwords.....	65
5.4.4	Criptografía asimétrica basada en identidad usando pares.....	66
5.5	RSA Laboratories	66
5.6	Otros Estudios	72
VI.	ESTÁNDARES	75
VII.	BENCHMARK USANDO HERRAMIENTA DE CIFRADO.....	76
7.1	Cifrado y descifrado del algoritmo de curvas elípticas	76
7.2	Cifrado y descifrado del algoritmo de RSA con clave de descifrado rápida..	79
7.3	Cifrado y descifrado del algoritmo de RSA con clave de descifrado lenta....	82

7.4	Curvas elípticas versus RSA	85
VIII.	VENTAJAS ECONÓMICAS USANDO CURVAS ELIPTICAS.....	87
8.1	Dispositivos móviles	87
8.2	Costos asociados de diseño para sistemas embebidos.....	88
8.2.1	Poder de procesamiento.....	88
8.2.2	Compuertas lógicas.....	89
8.2.3	Desgaste de baterías	90
8.2.4	Ancho de banda y protocolos	90
8.3	Reducción de gastos	91
IX.	CONCLUSIONES.....	92
X.	BIBLIOGRAFÍA.....	96
XI.	ANEXOS.....	99
	ANEXO A – MANUAL DE USO DE LA HERRAMIENTA DE CIFRADO.....	100
	ANEXO B – LIBRERIAS USADAS EN LA HERRAMIENTA	114
	ANEXO C – CÓDIGO DE LA LIBRERIA RSA USADA EN JAVA.....	118
	ANEXO D – CÓDIGO DE ENVÍO Y RECEPCION DE DATOS EN JAVA.....	124

INDICE DE FIGURAS

FIGURA 2.1: “ENVÍO DE MENSAJES POR UN CANAL INSEGURO”	3
FIGURA 2.2: “SOLUCIÓN PROPUESTA”	4
FIGURA 2.3: “METODOLOGÍA DE DESARROLLO DE LA MEMORIA”	7
FIGURA 3.2: “SISTEMAS DE ENCRIPCIÓN SIMÉTRICA”	15
FIGURA 3.3: “DISTRIBUCIÓN DE CLAVES SIMÉTRICAS”	17
FIGURA 3.4: “DISTRIBUCIÓN DE CLAVES ASIMÉTRICAS”	18
FIGURA 3.5: “SISTEMAS DE ENCRIPCIÓN ASIMÉTRICOS”	19
FIGURA 3.6: “SISTEMA CRIPTOGRÁFICO RSA”	24
FIGURA 3.7: “CURVA ELÍPTICA EN UN PLANO CARTESIANO”	26
FIGURA 3.8: “ADICIÓN DE PUNTOS EN CURVAS ELÍPTICAS”	27
FIGURA 3.9: “PUNTOS DE UNA CURVA ELÍPTICA DENTRO DE UN CUERPO FINITO”	28
FIGURA 3.10: “PUNTOS DE UNA CURVA ELÍPTICA DENTRO DE UN CUERPO FINITO BINARIO”	30
FIGURA 3.11: “SISTEMA CRIPTOGRÁFICO ECIES”	33
FIGURA 4.1 “ESQUEMA SSL ENTRE CLIENTE Y SERVIDOR”	35
FIGURA 4.2: “PLATAFORMA DE ENCRIPCIÓN PGP”	36
FIGURA 4.3: “SMART CARD, CON CIRCUITOS INTERNOS”	38
FIGURA 4.4: “TECNOLOGÍAS DE CIFRADO USADOS POR IPSEC”	40
FIGURA 5.1: “LOGOTIPO DE NIST”	41
FIGURA 5.2: “LOGOTIPO DE ANSI”	47
FIGURA 5.3 “PRODUCTOS DE CERTICOM”	49
FIGURA 5.4: “GRÁFICO COMPARATIVO ENTRE EL TIEMPO DE RESPUESTA DE UN SERVIDOR AL USAR ALGORITMOS CRIPTOGRÁFICOS RSA Y CURVAS ELÍPTICAS SEGÚN PETICIONES DE TRANSACCIONES POR SEGUNDO”	50
FIGURA 5.5: “COMPARACIÓN DEL TIEMPO DE RESPUESTA DE UN SERVIDOR AL USAR ALGORITMOS CRIPTOGRÁFICOS RSA Y CURVAS ELÍPTICAS SEGÚN TAMAÑO DE CLAVE APLICADA”	52
FIGURA 5.6: “COMPARACIÓN DE NIVELES DE SEGURIDAD BRINDADOS POR RSA Y CURVAS ELÍPTICAS”	53

FIGURA 5.7: “PROCEDIMIENTO DE VALIDACIÓN DE ALGORITMOS Y MÓDULOS”	55
FIGURA 5.8: “ARQUITECTURA DE SEGURIDAD PARA EL GOBIERNO”	57
FIGURA 5.9: “NIVEL DE CUMPLIMIENTO VERSUS REUTILIZACIÓN DE SESIONES”	61
FIGURA 5.10: “FÓRMULA DE THROUGHPUT SEGÚN LA REALIZACIÓN DE SESIONES”	61
FIGURA 5.11: “ESQUEMA DE ADAPTACIÓN DE LA TECNOLOGÍA CRIPTOGRÁFICA”	62
FIGURA 5.12: “LOGO DE IEEE”	63
FIGURA 5.13: “EJEMPLO DE LATTICE”	65
FIGURA 7.1: “GRÁFICO DE CIFRADO ECC TIEMPO VERSUS TAMAÑO DE CLAVE”	78
FIGURA 7.2: “GRÁFICO DE DESCIFRADO ECC TIEMPO VERSUS TAMAÑO DE CLAVE”	78
FIGURA 7.3: “GRÁFICO DE CIFRADO RSA CON CLAVE RÁPIDA TIEMPO VERSUS TAMAÑO DE CLAVE”	81
FIGURA 7.4: “GRÁFICO DE DESCIFRADO RSA CON CLAVE RÁPIDA TIEMPO VERSUS TAMAÑO DE CLAVE”	81
FIGURA 7.5: “GRÁFICO DE CIFRADO RSA CON CLAVE LENTA TIEMPO VERSUS TAMAÑO DE CLAVE”	84
FIGURA 7.6: “GRÁFICO DE DESCIFRADO RSA CON CLAVE LENTA TIEMPO VERSUS TAMAÑO DE CLAVE”	84
FIGURA 7.7: “GRÁFICO COMPARATIVO DE CIFRADO RSA VERSUS CCE” ...	85
FIGURA 7.7: “GRÁFICO COMPARATIVO DE DESCIFRADO RSA VERSUS CCE”	86

INDICE DE TABLAS

TABLA 3.1: “PRINCIPIOS BÁSICOS DE LA SEGURIDAD INFORMÁTICA”	12
TABLA 3.2: “ALGORITMOS MÁS USADOS DE CIFRADO SIMÉTRICO”	16
TABLA 3.3: “ALGORITMOS MÁS USADOS DE CIFRADO ASIMÉTRICO”	20
TABLA 5.2: “TAMAÑO DE CLAVES Y ALGORITMOS PARA CADA TIPO DE USO”	43
TABLA 5.3: “IDENTIFICADOR ANSI PARA LAS CURVAS APROBADAS POR NIST”	47
TABLA 5.4: “COMPARACIÓN DE PERFORMANCE DE RSA Y CURVAS ELÍPTICAS”	51
TABLA 5.5: “EVOLUCIÓN DE LOS PROCESADORES SEGÚN LA CANTIDAD DE INSTRUCCIONES POR SEGUNDOS QUE REALIZAN”	54
TABLA 5.6: “COMPARACIÓN DE SEGURIDAD DE ALGORITMOS”	58
TABLA 5.7: “COMPARACIÓN OPERACIONES EN SSL”	59
TABLA 5.8: “ESTÁNDARES PKCS”	67
TABLA 5.9: “ALGORITMOS HASH USADOS CON RSA”	70
TABLA 5.10: “REQUERIMIENTOS DE SISTEMA PARA ECC Y RSA”	71
TABLA 5.11: “COMPARACIÓN ENTRE ECC Y RSA EN UNIDADES DE TIEMPO PARA PROCESOS CRIPTOGRÁFICOS”	71
TABLA 5.12: “COMPARACIÓN DE COMPLEJIDAD COMPUTACIONAL EN ECC PARA OPERACIONES BÁSICAS”	72
TABLA 5.13: “COMPARACIÓN DE NÚMERO DE OPERACIONES BÁSICAS EN ECC”	73
TABLA 5.14: “TIEMPOS DE EXPONENCIACIÓN DE CURVAS ELÍPTICAS”	73
TABLA 5.15: “MEDICIÓN DE TIEMPOS DE PROCESOS DE CIFRADO Y DESCIFRADO EN RSA”	74
TABLA 6.1: “ALGUNOS ESTÁNDARES DE ECC”	75

I. INTRODUCCIÓN

Uno de los conceptos más escuchados hoy en día en el ámbito de la informática es la seguridad de la información. Se sabe que la información es un bien que se debe resguardar y proteger, ya que si cae en manos de personas ajenas a nuestro núcleo informativo podría ser un peligro latente. Es por esto que se presenta en esta memoria a los procesos criptográficos como un concepto de estudio de relevancia y que toma mayor importancia mientras avanzan las tecnologías basadas en la comunicación vía Internet. Los procesos criptográficos cumplen un rol fundamental en la protección de la información que se envía por la red, sin embargo, la aplicación de estos procesos trae consigo una carga de transacciones extra, por lo que se debe tener cuidado en que los procesos de criptografía no ocupen excesivos recursos que lleguen a disminuir el poder de procesamiento de las transacciones primarias para un equipo o dispositivo computacional. Es por esto que se ha propuesto el uso de la criptografía de curvas elípticas como una opción válida para reemplazar al método RSA, las curvas elípticas presentan varias optimizaciones sobre RSA que lo hace un candidato válido para ser usado masivamente, hay que recordar que RSA hoy en día es el método criptográfico más difundido en el mundo.

Dentro de la presente memoria se dan a conocer tanto ventajas como desventajas de ambos sistemas criptográficos, se presentan estudios que avalan a ambos y se presentan además pruebas de rendimiento en donde se deja en evidencia la performances que alcanzan ambos métodos criptográficos funcionando sobre el mismo equipamiento de hardware, lo que entrega pruebas contundentes de las ventajas reales que la criptografía de curvas elípticas posee para ser aplicadas sobre las transacciones que requieran de seguridad por Internet.

II. DESARROLLO DEL TEMA

2.1 Antecedentes y motivación

La propuesta de esta memoria se encuentra inmersa dentro del área de la seguridad informática, precisamente en la aplicación de algoritmos para realizar procesos criptográficos que logren resguardar la información de forma segura y confidencial. Los ataques para vulnerar el nivel de seguridad aportado por los algoritmos criptográficos es el principal motivo para que se dispare el desarrollo e investigación de nuevos métodos criptográficos, donde idealmente se busca el desarrollo de un método criptográfico inviolable que logre asegurar efectivamente la disponibilidad, confidencialidad, disponibilidad e irrefutabilidad de la información que se desea proteger.

El avance de las tecnologías es un arma de doble filo para la efectividad de la seguridad informática, debido a que por un lado acelera los procesos informáticos en general, pero a su vez esta velocidad permite vulnerar los sistemas criptográficos en un menor tiempo cuando es atacado. Esto último despierta la necesidad de establecer niveles de seguridad para el cifrado de información y pensar en realizar la comparación entre distintos métodos de cifrado para establecer cual es el más útil en cada caso.

Existen diversos tipos de algoritmos criptográficos, los podemos encontrar en diversos sistemas de información y sistemas tecnológicos, como por ejemplo, el acceso a sitios de Internet seguros bajo el protocolo HTTPS o en el acceso a la configuración de routers en el área de redes informáticas, sin embargo, las necesidades de seguridad no necesariamente serán las mismas para cada sistema, por lo que no se puede asegurar que un método criptográfico es mejor que otro si no son aplicados al mismo servicio. Es por esto que se ha optado por comparar dos métodos de cifrado para usarlos con el mismo fin, el envío de información cifrada desde un usuario hacia otro. Los métodos a comparar son: El cifrado usando el algoritmo de RSA (Rivest, Shamir, Adleman) y El cifrado usando el algoritmo de Curvas Elípticas, las métricas más importantes a considerar serán el tiempo que se emplea para los procesos de cifrado y descifrado de información y el tamaño de las claves empleadas bajo un mismo nivel de seguridad.

2.2 Descripción del problema

Se desea realizar el envío de un mensaje o información desde un usuario que actúa como emisor hacia un usuario que actúa como receptor mediante el uso de un canal considerado como inseguro. Durante el paso del mensaje o información por el canal inseguro, se corre el peligro de que exista un tercer ente interceptando y/o recibiendo la información destinada al receptor que se encuentra a la escucha de forma legible y decodificada. Quizás en muchos casos los mensajes o información enviada puede que no sea un gran secreto, sin embargo, cuando sí se requiere del envío de información confidencial este proceso inseguro puede ser un gran problema.

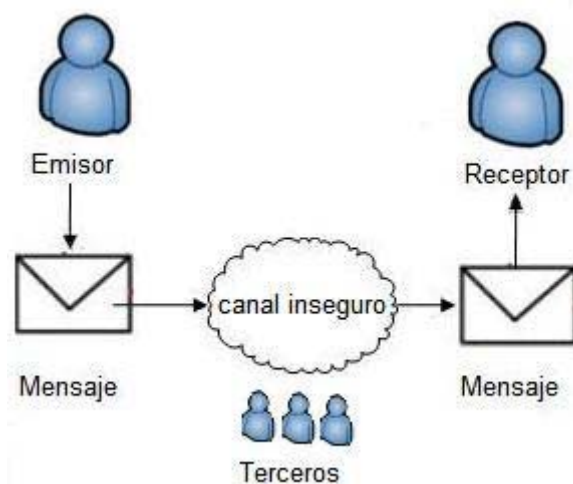


Figura 2.1: “Envío de mensajes por un canal inseguro”

2.3 Solución propuesta

La solución para mantener la información segura sin que ésta sea interceptada de forma legible por un tercer ente es la adopción del envío de un mensaje a modo de criptograma. En esta ocasión se realizará mediante la adopción de algoritmos criptográficos basados en RSA y basados en Curvas Elípticas.

La idea es demostrar la efectividad de los procesos criptográficos que protegen el mensaje o información de forma segura entre el emisor y el receptor sin que haya sido interceptada por un tercer ente durante su camino. El emisor emite un mensaje, el cual entra en la etapa de cifrado criptográfico para la creación de un criptograma, el ente interceptador si procede a realizar un ataque solo obtendrá información codificada ilegible para él, luego cuando el mensaje llega al receptor, se procede a descifrar el criptograma para obtener el mensaje original intacto.

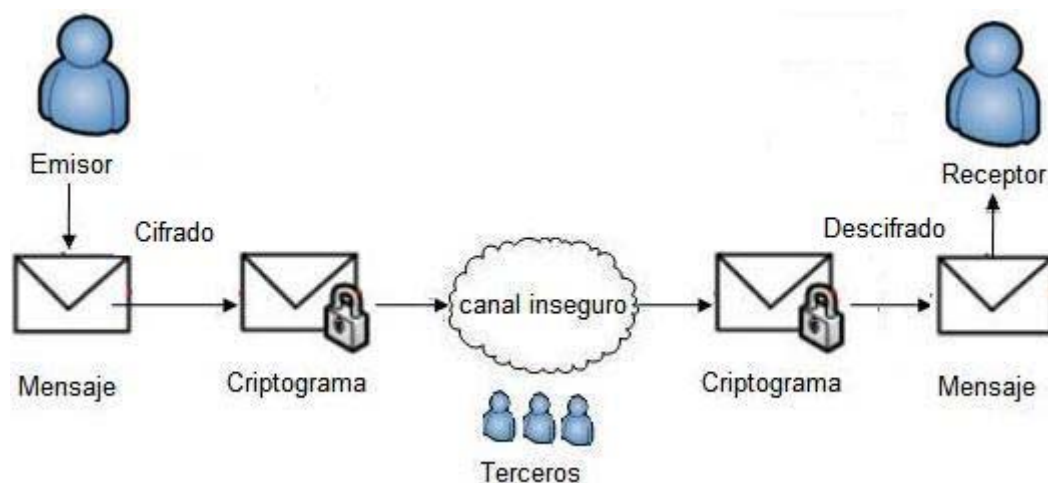


Figura 2.2: “Solución propuesta”

2.4 Objetivos y alcances del proyecto

2.4.1 Objetivo general

Los objetivos generales planteados para este proyecto de título son:

- Evaluar a la criptografía de curva elíptica como un método que permite optimizar los procesos de confidencialidad en la infraestructura de las organizaciones, respecto de la tradicional criptografía RSA usada ampliamente hoy en día en un contexto abierto como lo es Internet.

2.4.2 Objetivos específicos

Los objetivos específicos planteados para este proyecto son:

- Mostrar estudios comparativos de Curvas Elípticas y RSA con el propósito de analizar las ventajas y desventajas de cada uno en un ambiente abierto como “Internet”.
- Presentar un análisis costo/beneficio de lo que significaría en el ámbito económico de las organizaciones el migrar de una tecnología criptográfica a otra.
- Desarrollar una herramienta de cifrado utilizando algoritmos de Curvas Elípticas y RSA, con el fin de estudiar a nivel práctico los tiempos que se requieren para cifrar o descifrar información.
- Estudiar las metodologías de las principales organizaciones que dedican su tiempo a desarrollar estándares, o bien, a realizar estudios que permitan generar avances en materia criptográfica.
- Proponer una metodología preliminar de trabajo que permita a futuro realizar nuevas comparaciones de algoritmos criptográficos, indicando los fundamentos teóricos que los sustentan, entregar como conclusión las ventajas y desventajas que presenta cada uno, con el fin de ser un aporte en la toma de decisiones.

2.4.3 Alcance del proyecto

Se debe considerar como parte del proyecto el estudio de los criptosistemas basados en los algoritmos de cifrado RSA y de curvas elípticas, la recopilación de información sobre algunos estudios comparativos previos y el desarrollo de una herramienta que ejecuta los algoritmos antes mencionados. No se debe considerar como parte del proyecto, la implementación de algoritmos de cifrado simétricos independientes a los incorporados a los procesos internos de cifrado por curvas elípticas, ni el desarrollo de soluciones para implementar firmas digitales. Tampoco se debe considerar dentro del alcance del proyecto, estudios de cómo implementar los algoritmos de Curva Elíptica y RSA a nivel de hardware.

2.5 Estado del arte

Actualmente, los algoritmos criptográficos son usados en diversos sistemas informáticos y en nuevas tecnologías tales como: sistemas bancarios, sistemas comerciales, redes inalámbricas, VPN's, entre otras. El uso de algoritmos simétricos por ejemplo, es utilizado para realizar el acceso a la configuración de routers, o bien como parte de una subrutina para el funcionamiento de algún algoritmo de encriptación de tipo asimétrico.

Dentro del ámbito del comercio electrónico, aún existe desconfianza por parte de los usuarios respecto a la seguridad de este tipo de sistemas, se considera que es muy inseguro debido al mal uso que se podría dar al autorizar transacciones con tan solo ingresar los números de una tarjeta de crédito bancaria. Actualmente y debido a que el comercio electrónico ya es una realidad, se sigue trabajando en el desarrollo de algoritmos criptográficos robustos que logren proveer seguridad, rapidez y eficacia en su uso.

2.6 Metodologías y herramientas a utilizar

La metodología que es usada para el desarrollo de esta memoria es la que se describe en la figura 2.3.

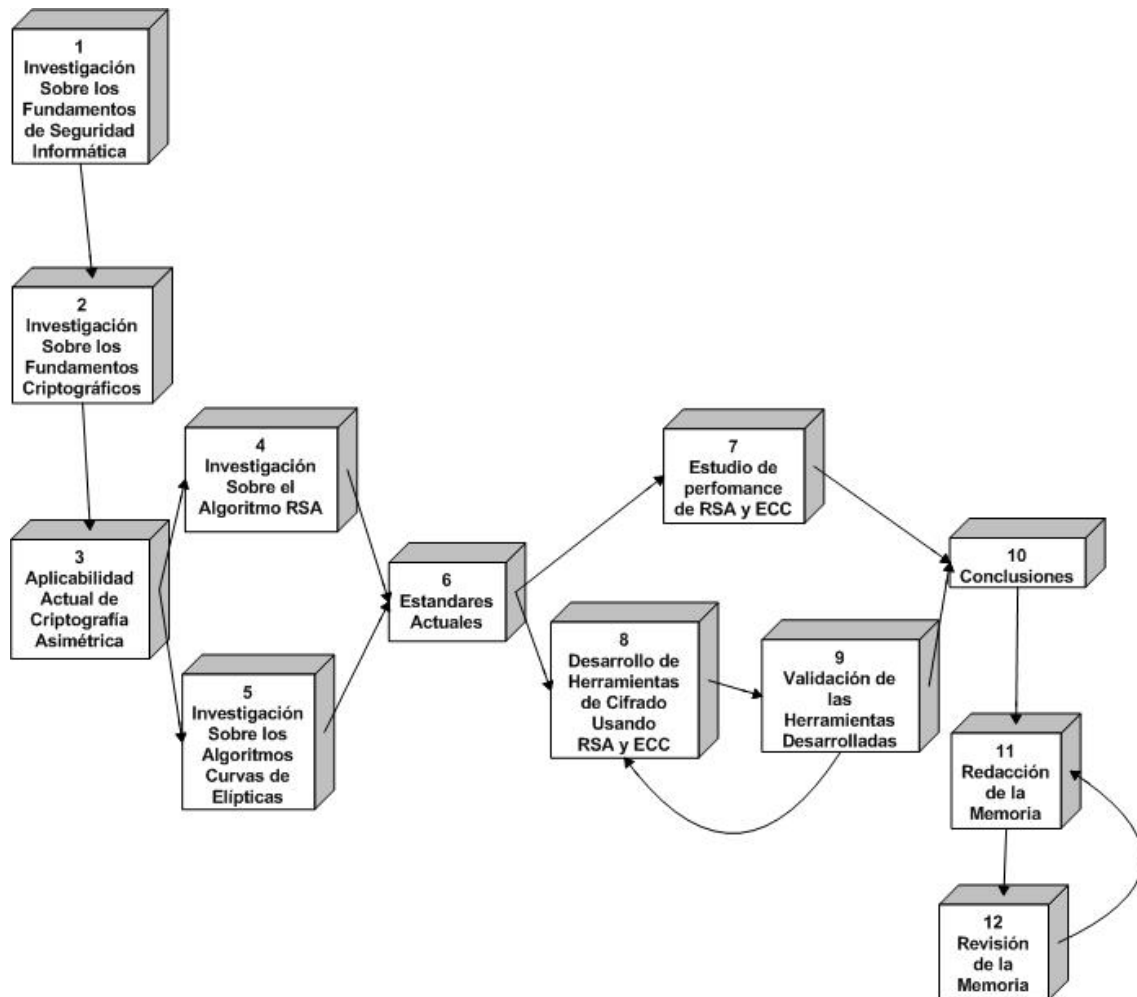


Figura 2.3: “Metodología de desarrollo de la memoria”

La descripción de cada una de las etapas es la que se presenta a continuación:

1. **Investigación sobre los fundamentos de seguridad informática.** En este proceso se ha investigado en profundidad los fundamentos básicos de la seguridad informática (Confidencialidad, Disponibilidad, Integridad, Irrefutabilidad) sobre los tipos de seguridad existentes tanto en la seguridad física como en la seguridad lógica.
2. **Investigación sobre los fundamentos criptográficos.** En este proceso se ha investigado en profundidad los fundamentos básicos de los principales métodos criptográficos existentes (criptografía simétrica, criptografía asimétrica).
3. **Aplicabilidad actual de la criptografía asimétrica.** En este proceso se ha investigado sobre el estado del arte en criptografía en donde se ocupa actualmente la criptografía asimétrica, como por ejemplo HTTPS ó PGP.
4. **Investigación sobre el algoritmo RSA.** En este proceso se ha investigado todo lo relacionado al algoritmo de RSA, tanto su historia, como su funcionamiento.
5. **Investigación sobre el algoritmo ECC.** En este proceso se ha investigado todo lo relacionado al algoritmo de ECC, tanto su historia, como su funcionamiento.
6. **Estándares actuales.** En este proceso se ha investigado los estándares actuales de cada criptosistema, se presentan estudios realizados por diferentes organismos especialistas a nivel mundial.
7. **Estudio de performance de RSA y CEE.** En este proceso se ha investigado sobre el desarrollo de cada tipo de criptosistema, en cuanto a su tiempo de cifrado y su tiempo de descifrado.
8. **Desarrollo de herramientas de cifrado usando RSA y CEE.** En este proceso se ha diseñado una herramienta de software en la cual se cifran mensajes para enviarlos mediante un canal inseguro como Internet, usando cualquiera de los dos criptosistemas antes mencionados.
9. **Validación de las herramientas desarrolladas.** En este proceso se realizaron pruebas de calidad para confirmar que los resultados que esta generando la herramienta desarrollada, presenta los valores esperados.
10. **Conclusiones.** En este proceso se presentan las conclusiones que se han obtenido durante el desarrollo de esta memoria.

11. Redacción de la memoria. En este proceso se ha realizado la redacción de la memoria, este proceso se realizó en forma paralela al desarrollo de esta memoria.

12. Revisión de la memoria. En este proceso se revisa la redacción de la memoria y se corrige los problemas tanto de redacción, puntuación, y formato. Es un desarrollo permanente hasta la entrega final.

2.7 Desarrollo del aporte ingenieril

Además del desarrollo de una metodología de trabajo para futuras comparaciones de algoritmos criptográficos, se desarrollo una herramienta de cifrado de datos mediante el uso de los métodos criptográficos de curvas elípticas y RSA la cual se ha utilizado para realizar pruebas de rendimiento entre un método y otro en relación al tiempo de respuesta entregado luego de los procesos de cifrado y descifrado. La idea es poder corroborar lo que los estudios de las instituciones de investigación criptográfica dicen en relación a la eficacia de estos métodos de forma experimental.

2.8 Resultados obtenidos

Luego de haber ejecutado la herramienta criptográfica desarrollada, se realizaron pruebas las cuales se encuentran detalladas más adelante.

2.9 Organización del documento

La memoria está dividida en once apartados, incluyendo bibliografía y anexos. Se ha ordenado de la siguiente forma:

En el primer capítulo, se presenta la introducción al tema en general y al trabajo realizado.

En el segundo capítulo, se presenta el desarrollo del tema en general, se indica cuál es el problema presentado y cuál es la solución que satisface el problema a nivel general. Se estipulan los objetivos generales y específicos de la memoria, los cuales son satisfechos dentro de los capítulos siguientes, además se indican los alcances del proyecto donde se estipula lo que debe o no ser considerado como parte fundamental de la memoria. También se considera la explicación de la metodología seguida para completar los contenidos de esta memoria.

En el tercer capítulo se indican los fundamentos teóricos que avalan los contenidos de la memoria, en este caso la introducción a la seguridad informática, criptografía y criptosistemas para finalizar con la presentación de los métodos criptográficos RSA y curvas elípticas.

En el cuarto capítulo se presenta el estado del arte, indicando algunas aplicaciones de la seguridad informática en ámbitos como el comercio electrónico, e-mails, conectividad wireless entre otras.

En el quinto capítulo se presentan algunas entidades de importancia en relación a estudios criptográficos. Se incluyen por ejemplo la empresa líder en estudios de curvas elípticas "Certicom" y "RSA Labs" para los estudios del método criptográfico RSA.

En el sexto capítulo se mencionan algunos estándares desarrollados para la criptografía de curvas elípticas.

En el séptimo capítulo se presentan los resultados del benchmark realizado a la herramienta desarrollada de cifrado presentando gráficos para cada prueba realizada.

En el octavo capítulo se presentan algunas ventajas económicas que se pueden deducir del uso de la criptografía de curvas elípticas versus usar criptografía RSA, específicamente en dispositivos móviles o embebidos.

En el noveno capítulo se presentan las conclusiones referentes al uso de los métodos criptográficos desarrollados.

En el décimo capítulo se encuentra el apartado donde se incluye la bibliografía asociada a la memoria.

Finalmente en el undécimo capítulo se adjuntan anexos propios de algunos contenidos de la memoria.

III. FUNDAMENTOS TEÓRICOS

3.1 La importancia de la información

El éxito de una empresa dependerá de la calidad de la información que genera y gestiona. Así, una empresa tendrá una información de calidad si ésta posee entre otras características, las de confidencialidad, de integridad, de disponibilidad y de irrefutabilidad.

La implantación de políticas y medidas de seguridad informática en una empresa se comienzan a tener en cuenta sólo a finales de la década pasada. En este nuevo siglo, es un factor estratégico en el desarrollo y éxito de la misma. Después de atentados terroristas, incendios, huracanes y diversas amenazas, muchas empresas han desaparecido por no haber sido capaces de recuperarse tras haber perdido toda su información.

En toda empresa, la preocupación permanente por la mejora de la administración, las finanzas y la producción han conducido a la rápida adopción de sistemas automáticos capaces de facilitar tareas mecánicas y rutinarias, evitar errores y mejorar el control de la cartera de clientes y con el incremento consiguiente de la calidad.

3.2 Seguridad informática

La seguridad es un estado en cual se tiene un nivel de certeza de que un sistema se encuentre sin riesgos o vulnerabilidades. En el caso de la informática, se dice que son procedimientos que permiten asegurar que los activos de información de un sistema informático sean utilizados de manera adecuada, y bajo las restricciones acordadas.

Hoy en día las redes de computadores se pueden encontrar en cualquier lugar. Para que estas redes sean seguras es importante que se cumplan los principios básicos de la seguridad informática:

Tabla 3.1: “Principios básicos de la seguridad informática”

Principios básicos de la seguridad informática	
Confidencialidad	Consiste en asegurar que la información sea legible solamente por usuarios autorizados dentro del sistema informático.
Integridad	Consiste en asegurar que la información no pueda ser modificada por quienes no posean la autorización necesaria para realizarlo.
Disponibilidad	Consiste en asegurar que los recursos de información estén disponibles cuando se necesiten.
Irrefutabilidad	Conocido también como “no repudio”, su objetivo es garantizar que cuando un usuario envía una información, éste no pueda negar el contenido de tal acto.

Según los tipos de amenazas podemos clasificar la seguridad de un sistema de información en:

- **Seguridad Física:** La seguridad física consiste en la aplicación de procedimientos para establecer barreras físicas, medidas de control de accesos y prevención sobre los recursos de información. Estas medidas están orientadas a proteger el hardware y los medios de almacenamiento de datos en relación a amenazas tales como son los desastres naturales y las amenazas ocasionadas por el hombre como los sabotajes y disturbios.
- **Seguridad Lógica:** Al igual que la seguridad física, esta también consiste en aplicación de barreras, pero a nivel de software, de tal manera que se resguarden el acceso a los datos y sólo se les permita acceder a ellos a las personas autorizadas. La seguridad lógica está orientada al control de acceso a los datos, implementando la identificación y autenticación de usuarios e implementación de perfiles para administrar el nivel de acceso a la información para cada usuario válido en un sistema.

3.3 Criptografía

El Hombre desde sus inicios ha necesitado comunicarse con los demás y ha tenido la necesidad de que algunos de sus mensajes sólo sean conocidos por las personas a las cuales se les hayan destinado. Desde tiempos remotos, se han usado sistemas de cifrado de mensajes en los cuales un mensaje es cifrado para lograr ser entendido solamente por aquellos que conozcan el método para descifrarlo.

La criptografía se puede definir como el arte o la ciencia de cifrar y descifrar información utilizando como por ejemplo las técnicas matemáticas de tal forma que hagan posible el intercambio de mensajes de manera que sólo puedan ser leídos por las personas a quienes van dirigidos, por lo que podemos decir que la criptografía ha existido desde muchos siglos naciendo innatamente desde las antiguas civilizaciones. La diferencia entre los procesos criptográficos del pasado con los actuales radican simplemente en la evolución de las comunicaciones, la vida de las personas hoy en día mantienen el uso de las nuevas tecnologías como pilar fundamental de sus comunicaciones, incluyendo el mundo de la informática e Internet por lo que los métodos criptográficos para este tipo de medios se hace un tema de alta importancia para la conformidad de la confidencialidad de información para cada persona.

La finalidad de la criptografía es, en primer lugar, garantizar el secreto en la comunicación entre dos entidades (ya sean personas, organizaciones u otros) y, en segundo lugar, asegurar que la información que se envía es auténtica en un doble sentido: que el remitente sea realmente quien dice ser y que el contenido del mensaje enviado no haya sido modificado en su tránsito.

El proceso criptográfico consiste en transformar un mensaje legible en otro que no lo es (mediante claves que sólo el emisor y el destinatario conocen), para después devolverlo a su forma original, sin que nadie que vea el mensaje cifrado sea capaz de entenderlo. El mensaje cifrado recibe el nombre de “Criptograma”.

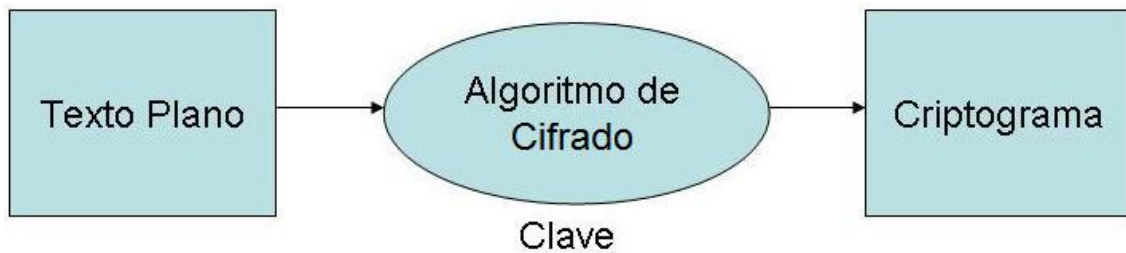


Figura 3.1: “Transformación de un texto plano a Criptograma”

En la actualidad, la criptografía no sólo se utiliza para transferir información de forma segura ocultando su contenido a posibles intrusos. Una de las ramas de la criptografía que más ha revolucionado el panorama actual de las tecnologías informáticas es el de la firma digital: tecnología que busca asociar al emisor de un mensaje con su contenido de forma que aquel no pueda posteriormente repudiarlo.

3.4 Criptosistemas

Un Criptosistema, o sistema criptográfico, se puede definir como los fundamentos y procedimientos de operación o algoritmo que participan en el cifrado y descifrado de un mensaje.

Todo sistema criptográfico consta de cinco componentes:

- a. El conjunto de todos los mensajes a transmitir.
- b. El conjunto de todos los mensajes cifrados.
- c. El conjunto de claves a utilizar.
- d. El conjunto de todos los métodos de cifrado.
- e. El conjunto de todos los métodos de descifrado.

A lo largo de la historia, se han utilizado cientos de criptosistemas diferentes, pero, a grandes rasgos se pueden dividir solamente en dos, los sistemas simétricos y los sistemas asimétricos.

3.4.1 Sistemas simétricos

Los sistemas simétricos, son aquellos en los cuales se necesita una única clave para cifrar y descifrar mensajes entre el emisor y el receptor del mensaje, dicha clave debe ser acordada con anterioridad a la emisión del mismo para un correcto funcionamiento del sistema.

Los sistemas simétricos basan su seguridad en el tamaño de la clave a aplicar, es decir, a mayor tamaño de la clave usada, mayor seguridad se otorga.

En la figura 3.2 se puede ver el funcionamiento del sistema criptográfico simétrico:

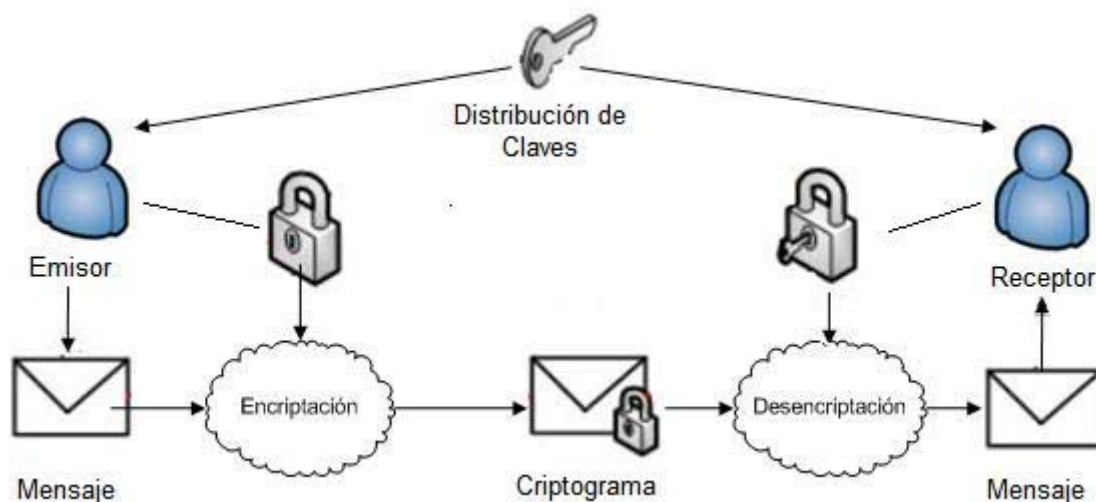


Figura 3.2: “Sistemas de encriptación simétrica”

El emisor envía un mensaje, el cual entra al proceso de cifrado usando la clave simétrica previamente distribuida. Al final del proceso de cifrado obtenemos el mensaje codificado conocido como criptograma, el cual para conocer su contenido se debe pasar por el proceso de descifrado aplicando la misma clave usada para el cifrado, al finalizar el descifrado, el receptor recuperará el mensaje inicial.

En la tabla 3.2 se presentan reseñas para los algoritmos más usados de cifrado simétrico:

Tabla 3.2: “Algoritmos más usados de cifrado simétrico”

ALGORITMO	CLAVE (BITS)	CARACTERÍSTICA BÁSICA
Feistel	Por lo general 64	Es el padre de los cifradores simétricos en bloques.
Lucifer	128	Es un algoritmo del tipo Feistel y posteriormente dará paso a DES.
DES	56	Algoritmo tipo Feistel que se convirtió en estándar, actualmente no se usa debido a su vulnerabilidad.
Loki	64	Algoritmo similar a DES.
RC5	Variable	Algoritmo muy rápido debido a su arquitectura simple, sus bajos requisitos de memoria y alta seguridad.
CAST	64	Inmune a ataques por criptoanálisis diferencial y lineal.
Blowfish	Variable	Algoritmo de tipo Feistel propuesto por Bruce Schneier.
IDEA	128	Algoritmo que es inmune ante un criptoanálisis diferencial.
Skipjack	80	Algoritmo usado para comunicaciones oficiales en USA, es inseguro ya que posee Backdoor.
Rijndael o AES	128 o más	Algoritmo mas usado y es actualmente un estándar mundial.

Los sistemas simétricos son más rápidos en la operación de cifrado y descifrado de mensajes en comparación con los sistemas asimétricos, esto es debido a la simplicidad de sus operaciones, las cuales pueden ser por sustitución de caracteres o algunas operaciones básicas de matemáticas como por ejemplo el operador lógico XOR.

Los sistemas simétricos fueron los más usados en la antigüedad, sin embargo, presentan las siguientes debilidades:

- Una mala gestión de claves: para un número de usuarios n son necesarias $\sum_{i=0}^{n-1} i$ claves distintas, es decir para seis usuarios distintos el sistema debería manejar quince claves distintas, es decir cinco claves para cada usuario y cada clave distinta una de la otra, esto se puede apreciar de mejor manera en la figura 3.3.

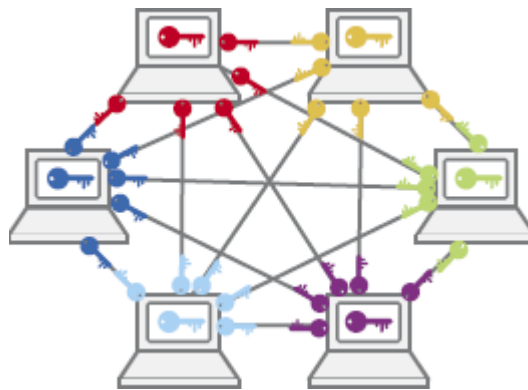


Figura 3.3: “Distribución de claves simétricas” – Fuente Certicom

- La mala forma de distribuir la clave a ocupar para cifrar y descifrar mensajes, esto es debido a que para la transmisión de claves se ocupa un método de transmisión inseguro ya sea Internet, teléfono, correo o otro método similar, la manera más segura de transmitir una clave que se ocupará para cifrar y descifrar mensajes es entregarla personalmente, pero en ese caso es preferible entregar inmediatamente el mensaje y no realizar el método de cifrado y descifrado.
- La inexistencia de una manera fácil y sencilla de firmar mensajes, aunque estos se pueden llegar a autenticar mediante marcas.

3.4.2 Sistemas asimétricos

Los sistemas asimétricos, son aquellos en los cuales se necesitan dos claves para cifrar y descifrar mensajes entre el emisor y el receptor del mensaje, tanto el emisor como el receptor poseen un par de claves, de éstas una será de tipo público donde da lo mismo que todo el mundo la conozca y la otra será de tipo privado (la cual se tiene que proteger) y para enviar mensajes el emisor tiene que cifrar el mensaje con la clave publica del receptor, para que así el receptor sea el único que pueda descifrar el mensaje usando su clave privada, cabe destacar que tanto como la clave publica como la privada son inversas entre si dentro de un cuerpo multiplicativo, en la figura 3.4 se muestra como funciona el intercambio de claves en los sistemas asimétricos.



Figura 3.4: “Distribución de claves asimétricas” – Fuente Certicom

Los sistemas asimétricos basan su seguridad en las funciones matemáticas del tipo unidireccional, es decir que son fáciles de calcular en un sentido pero en el otro sentido complica mucho el cálculo.

En la figura 3.5 se puede ver el funcionamiento del sistema criptográfico asimétrico:

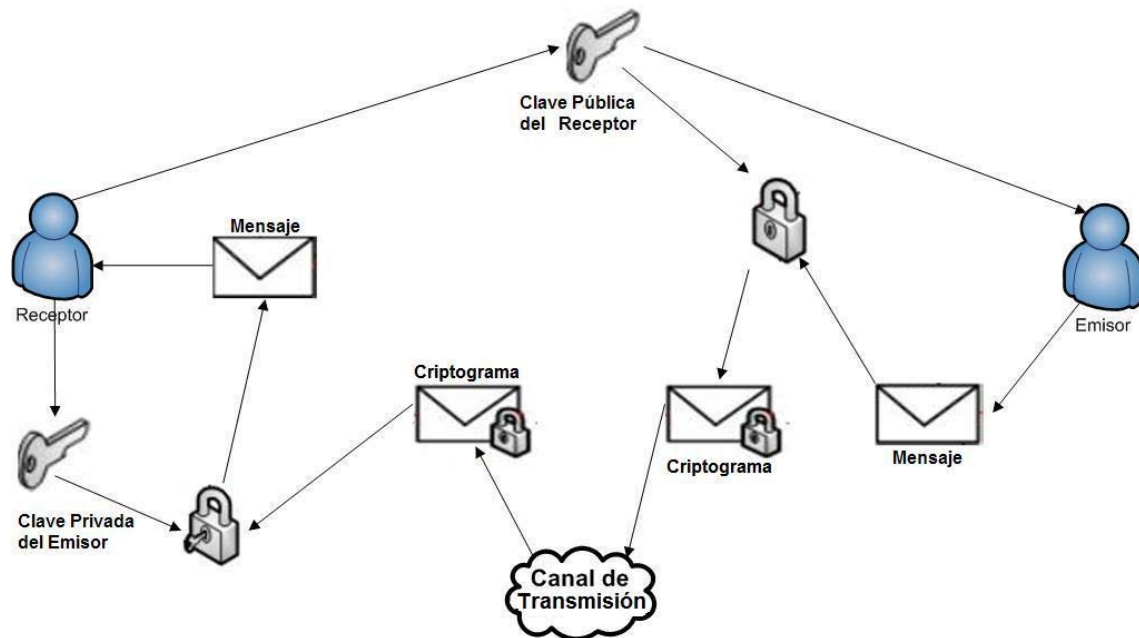


Figura 3.5: “Sistemas de encriptación asimétricos”

El receptor publica su clave pública mediante cualquier medio, ya sea por Internet o publicado en algún diario u otro medio similar y es tomado por el emisor del mensaje y cifra el mensaje con esta clave, luego envía el criptograma mediante cualquier canal de transmisión, el receptor toma el criptograma lo descifra con su clave privada y obtiene así el mensaje.

En la tabla 3.3 se presentan reseñas para los algoritmos más usados de cifrado asimétrico:

Tabla 3.3: “Algoritmos más usados de cifrado asimétrico”

ALGORITMO	CLAVE (BITS)	CARACTERÍSTICA BÁSICA
DSA	Cualquier factor de 64 entre 512 y 1024.	Algoritmo con el cual firmaba documentos el gobierno federal de USA.
El Gamal	No trabaja con tamaños de bits, sino que trabaja bajo el cuerpo de un número primo grande.	Algoritmo basado en Diffie-Hellman, también basado en el problema del logaritmo discreto.
Pohlig y Hellman	--	Algoritmo de cifra de clave secreta y que basa su seguridad en el problema del logaritmo discreto, cabe destacar que salio un mes antes que el RSA.
Diffie-Hellman	No trabaja con tamaños de bits, sino que se selecciona un grupo multiplicativo (con inverso) p y un generador α de dicho primo, ambos valores públicos.	Algoritmo de intercambio de claves de sesión.
RSA	Sobre 1024 bits y se basa en la distribución de la elección de los primos p y q .	Algoritmo de cifrado de bloque más usado en la actualidad.
ECC	Se recomienda un tamaño de clave sobre los 80 bits.	Algoritmo que es una variante de los sistemas asimétricos ya que ocupan las curvas elípticas para cifrar.

Los sistemas asimétricos son los más usados en la actualidad, sin embargo, se le pueden asociar las siguientes debilidades:

- Para una misma longitud de clave y mensaje se necesita mayor tiempo de procesamiento de información, en comparación con un algoritmo de cifrado simétrico.
- Las claves asimétricas deben ser de mayor tamaño que las claves simétricas, para así garantizar la autenticidad e integridad del mensaje.
- El mensaje cifrado ocupa más espacio que el original.
- Se cifran números, no se cifran mensajes, por lo mismo los mensajes hay que pasarlos por alguna función de HASH como SHA-1 o MD5 antes de cifrar el mensaje.

3.5 Criptosistema RSA

RSA, fue creado por Ronald Rivest, Adi Shamir, y Leonard Adleman y se propuso en el año 1977. En sus primeros años fue patentado por Massachusetts Institute of Technology bajo patente de RSA Labs. Sin embargo su uso comercial sólo empezó desde el año 2000.

El problema de RSA se basa en factorizar números grandes compuestos por números primos, de los cuales se puede obtener información de las claves.

3.5.1 Algoritmo RSA

Si bien, la intención al ocupar estos algoritmos es cifrar mensajes, lo que realmente se está haciendo es cifrar números los cuales tienen su representación en texto.

El algoritmo RSA trabaja con 2 tipos de claves por cada usuario:

- Clave privada: Esta clave es conocida tan sólo por el propio usuario.
- Clave pública: Esta clave la puede conocer cualquier persona.

Estas claves se obtienen a partir de operaciones matemáticas que se presentan a continuación:

- a. El usuario debe escoger un número n el cual sea el producto de dos números primos p y q .

$$n = p * q$$

- b. Se calcula la función de Euler a n , es decir se debe calcular $\phi(n)$.

La función de Euler entrega el número de elementos del *Conjunto Reducido de Restos*, es decir, entrega la cantidad de números en el rango de $[1, n-1]$ los cuales no son factores de n . Es este caso, al ser n un número compuesto por dos factores primos, el resultado de la función de Euler será:

$$\phi(n) = (p - 1)(q - 1)$$

- c. Se elige la clave privada e , la cual corresponde a un número entero en el rango $]1, n[$ tal que e y $\phi(n)$ sean primos relativos, o sea que el mínimo común múltiplo entre ambos sea igual a 1. Es decir:

$$1 < e < n / \text{mcm}(e, \phi(n)) = 1$$

- d. Obtener la clave privada d , al calcular el inverso de e dentro del cuerpo finito Z_n (números enteros entre 0 y n). Es decir:

$$d / e * d \equiv 1 \text{ mod } n$$

También denotado de la siguiente forma:

$$d = \text{INV}[e, \phi(n)]$$

Una vez finalizadas estas operaciones, se publica la clave pública e y el número n .

3.5.2 Cifrado RSA

La operación de cifrado de RSA es simple. Si se quiere cifrar un número N , habiendo elegido una clave pública e y calculado la clave privada d mediante la elección de n , se procede a calcular C , que corresponde a la salida de la operación de cifrado. Dependiendo de para qué se quiera usar el cifrado la operación será diferente.

Si lo que se quiere es la confidencialidad, ciframos el número N con la clave pública e del receptor, utilizando también el n del receptor.

$$C = N^e \bmod n$$

En cambio, si se quiere el no repudio, utilizamos el mecanismo de firma digital, utilizando tanto clave privada d como el n del emisor.

$$C = h(N)^d \bmod n$$

En este último caso, $h(N)$ corresponde a la utilización de un algoritmo hash tal como MD5 o SHA-1 para el número N . Luego se tendrá que enviar tanto el mensaje N , como C para poder realizar la comprobación del mensaje.

3.5.3 Descifrado RSA

El descifrado clásico opera muy similar al cifrado. La diferencia es que se utilizan las claves contrarias al proceso de cifrado. Para que el receptor pueda obtener el número N a través de C deberá realizar la siguiente operación:

$$N = C^d \bmod n$$

En este caso, al igual que en el cifrado, se utiliza tanto la clave como el n del receptor.

Si el mensaje fue enviado con firma digital (N , C), el descifrado y posterior comprobación se realizará calculando:

$$h(N) = C^e \bmod n$$

Luego, se procederá a calcular $h(N)$ a través del mensaje N . Si los dos $h(N)$ son congruentes, quiere decir que el mensaje no ha sido alterado, y el emisor ha sido comprobado.

En la figura 3.6 se puede un ejemplo del funcionamiento de este sistema:

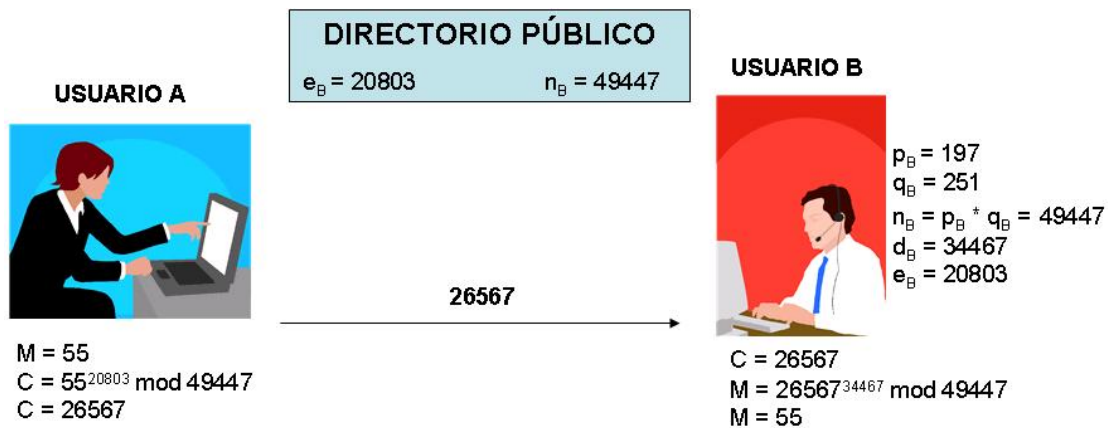


Figura 3.6: "Sistema criptográfico RSA"

En este caso el usuario A quiere enviarle el mensaje $M = 55$ al usuario B, para lo cual obtiene la clave pública de B, es decir $e_B = 20803$ y el grupo finito con el que se trabaja, es decir $n_B = 49447$. Luego se aplica las operaciones de cifrado y descifrado mencionadas en los párrafos anteriores.

3.5.4 Teorema del resto chino para RSA

Este teorema se utiliza para poder realizar la operación de descifrado con mayor rapidez que el descifrado clásico, ya que utiliza los valores de p y q en vez de n , que en este tipo de operaciones son de tamaño considerablemente menor.

Para realizar el descifrado, este teorema realiza la siguiente operación:

$$N = \{A_p[C_p^{d_p} \bmod p] + A_q[C_q^{d_q} \bmod q]\} \bmod n$$

Donde:

$$A_p = q [(q-1) \bmod p] \text{ o bien, } q^{p-1} \bmod n$$

$$A_q = p [(p-1) \bmod q] \text{ o bien, } p^{q-1} \bmod n$$

$$C_p = C \bmod p$$

$$C_q = C \bmod q$$

$$dp = d \bmod (p-1)$$

$$dq = d \bmod (q-1)$$

3.6 Criptografía de curvas elípticas

Este tipo de algoritmo fue propuesto en 1985 por Neil Koblitz y Victor Millar, cada uno de manera independiente.

Si bien, este algoritmo también basa su dificultad en la resolución del logaritmo discreto, su resolución es mucho más complicada que los demás algoritmos, ya que en vez de usar números enteros, ocupa puntos de una curva elíptica.

Las curvas elípticas en sí aparecieron para calcular el perímetro de una elipse, es de ahí que se adoptó el nombre para esta modalidad de criptografía.

3.6.1 Fundamentos matemáticos de curvas elípticas

La curva elíptica dentro de los reales, son un conjunto de puntos que satisfacen la ecuación:

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

Como se puede apreciar, al cambiar los coeficientes a y b alteraría toda la curva elíptica.

Una curva elíptica puede formar un grupo sólo si se cumple lo siguiente:

$$4a^3 + 27b^2 \neq 0$$

Dentro de los reales una curva elíptica puede ser representada como se muestra en la siguiente figura.

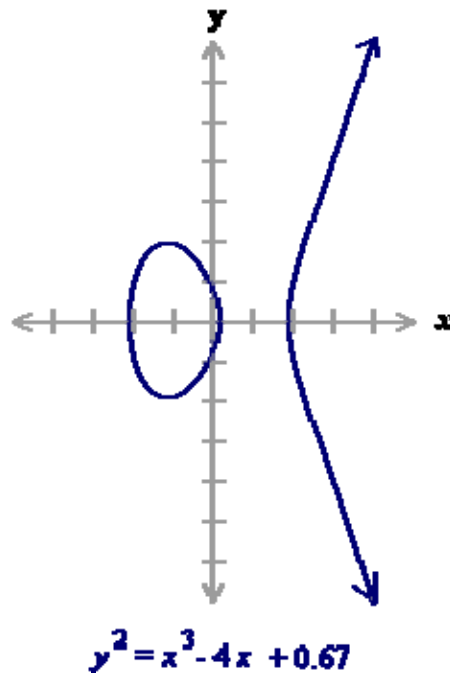


Figura 3.7: “Curva elíptica en un plano cartesiano” – Fuente Certicom

La figura 3.7 corresponde la representación de la ecuación con exponentes $a=-4$ y $b=0,67$.

3.6.2 Adición de curvas elípticas

La adición dentro de las curvas elípticas no es como se está acostumbrado en la suma de puntos de un plano cartesiano.

La adición dos puntos dentro de un grupo formado por una curva elíptica está definida geoméricamente.

Se define como el negativo de un punto $P=(x_P, y_P)$ al reflejo del mismo dentro del eje x , es decir

$$\boxed{-P=(x_P, -y_P)}$$

Si se quiere sumar dos puntos diferentes P y Q , teniendo en cuenta que Q es distinto de $-P$, definiremos $-R$ como el punto de intersección de la recta \overline{PQ} con la curva, como se ilustra en la figura 3.8:

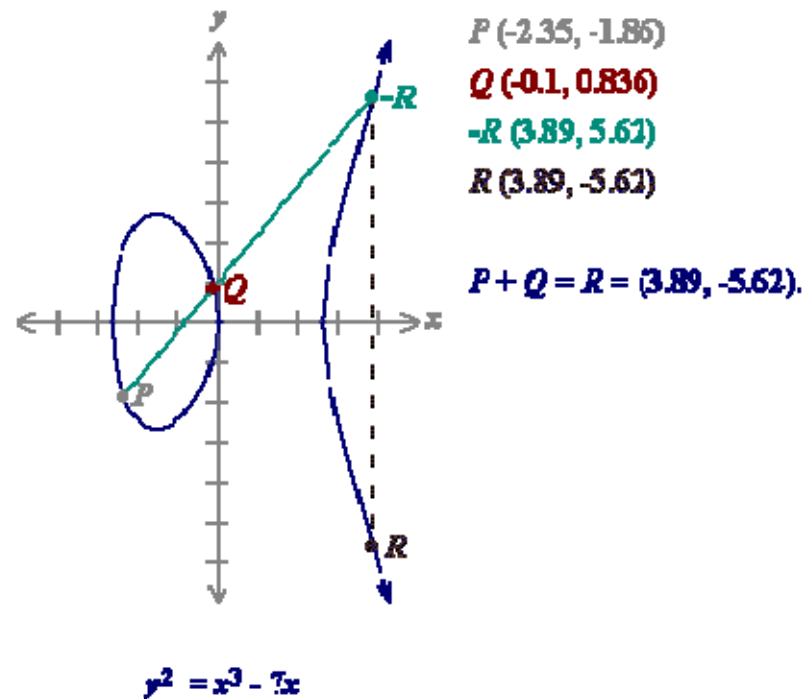


Figura 3.8: “Adición de puntos en curvas elípticas” – Fuente Certicom

En este caso $-R = (3.89, 5.62)$ y su reflejo respecto al eje x , vendría siendo el resultado de la suma de los puntos P y Q , es decir $R = (3.89, -5.62)$.

Se define como *punto O*, al punto en el infinito. Este punto actúa como neutro aditivo al momento de ser operado con un punto P dentro de la curva. Es decir:

$$P + O = P$$

Si $Q = -P$ entonces:

$$P + Q = O$$

Si $P = Q$ entonces:

$$P + Q = -R$$

También podemos decir que si queremos doblar P :

$$P + (-P) = O$$

$$P + P = 2P = R$$

3.6.3 Curvas elípticas sobre cuerpos finitos

Al igual que RSA, la criptografía de curvas elípticas trabaja en cuerpos finitos. Definimos F_p como al grupo de números enteros entre 0 y $p-1$.

La ecuación de la curva elíptica sobre cuerpos finitos estaría representada de la siguiente forma:

$$y^2 \bmod p = x^3 + ax + b \bmod p$$

La representación gráfica de este grupo es muy diferente al visto anteriormente.

Por ejemplo si queremos representar la ecuación $y^2 = x^3 + x$ dentro de F_{23} , la ecuación cambiaría a $y^2 \bmod 23 = x^3 + x \bmod 23$, el gráfico de dicha ecuación se puede apreciar en la figura 3.9.

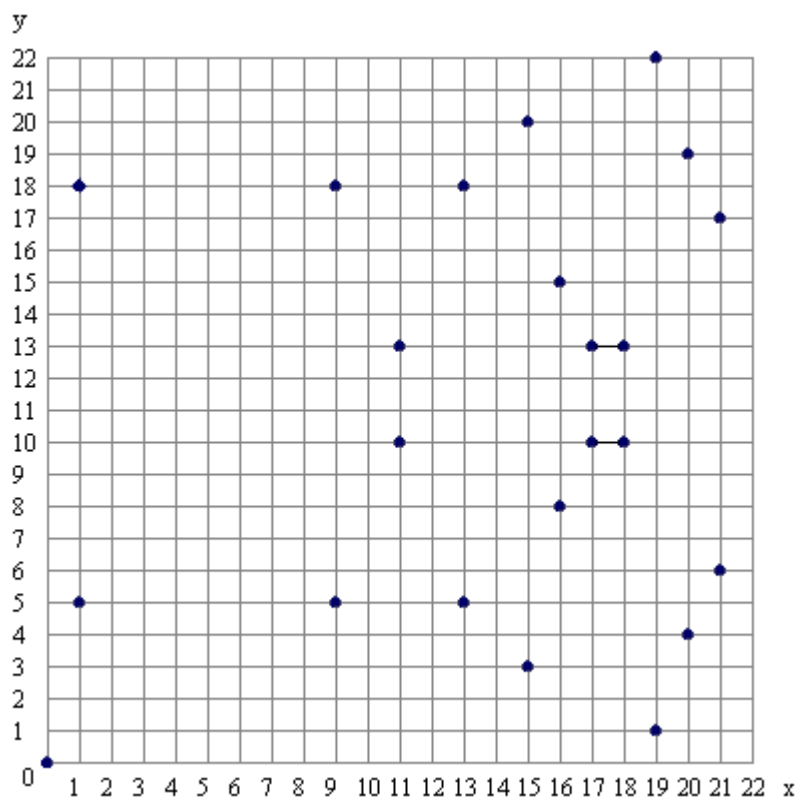


Figura 3.9: “Puntos de una curva elíptica dentro de un cuerpo finito”- Fuente Certicom

3.6.4 Adición de curvas elípticas sobre cuerpos finitos

En este caso, es más fácil realizar la adición a nivel de fórmula. Si queremos obtener el negativo de un punto P :

$$-P = (x_P, -y_P \bmod p)$$

Si $P \neq Q$, para obtener el valor de $P+Q=R=(x_R, y_R)$ se debe operar de la siguiente forma:

Calculamos

$$s = (y_P - y_Q) / (x_P - x_Q) \bmod p$$

Luego obtenemos (x_R, y_R) con las siguientes ecuaciones:

$$x_R = s^2 - x_P - x_Q \bmod p$$

$$y_R = -y_P + s(x_P - x_R) \bmod p$$

Para obtener $2P = R$

Calculamos

$$s = (3x_P^2 + a) / (2y_P) \bmod p$$

Y operamos con las siguientes ecuaciones:

$$x_R = s^2 - 2x_P \bmod p$$

$$y_R = -y_P + s(x_P - x_R) \bmod p$$

3.6.5 Curvas elípticas sobre F_{2^m}

Al trabajar en F_{2^m} estamos operando con cadenas de m bits, y para trabajar en representación binaria, las ecuaciones de curvas elípticas tendrán la siguiente forma:

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

La condición en este caso es que b sea diferente de 0.

En F_{2^4} diremos que $g = (0010)$ es generador del cuerpo, ya que al elevarlo a los exponentes desde 0 hasta 2^{m-1} , es decir 15 podemos obtener todos los valores dentro del cuerpo. Es decir:

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

Para obtener estos resultados, fue necesario ocupar en algunos casos el polinomio irreducible $f(x) = x^4 + x + 1$.

Un ejemplo de curva dentro de este cuerpo finito es $y^2 + xy = x^3 + g^4x^2 + 1$, en donde los valores de a y b son g^4 y 1 respectivamente. Cabe destacar que para operar cadenas de m bits, no se toma en consideración los acarreos de bits.

Volviendo a la ecuación, podemos señalar que hay 15 puntos que la satisfacen, estos son:

$$\begin{aligned}
&(1, g^{13}), (g^3, g^{13}), (g^5, g^{11}), (g^6, g^{14}), (g^9, g^{13}), (g^{10}, g^8), (g^{12}, g^{12}), \\
&(1, g^6), (g^3, g^8), (g^5, g^3), (g^6, g^8), (g^9, g^{10}), (g^{10}, g), (g^{12}, 0), (0, 1)
\end{aligned}$$

Su representación gráfica puede verse en la figura 3.10:

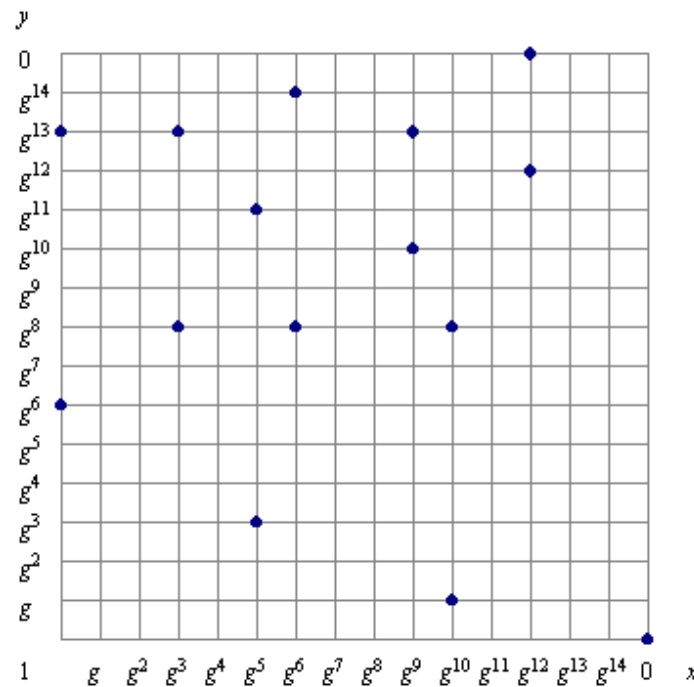


Figura 3.10: “Puntos de una curva elíptica dentro de un cuerpo finito binario” –

Fuente Certicom

3.6.6 Adición para curvas elípticas sobre F_{2^m}

En este caso, es más fácil realizar la adición a nivel de fórmula. Si queremos obtener el negativo de un punto P :

$$-P = (x_P, x_P + y_P)$$

Si $P \neq Q$, para obtener el valor de $P+Q=R=(x_R, y_R)$ se debe operar de la siguiente forma:

Calculamos

$$s = (y_P + y_Q) / (x_P + x_Q)$$

Luego obtenemos (x_R, y_R) con las siguientes ecuaciones:

$$x_R = s^2 + s + x_P + x_Q + a$$

$$y_R = s(x_P + x_R) + x_R + y_P$$

Para obtener $2P = R$

Calculamos

$$s = x_P + y_P / x_P$$

Y operamos con las siguientes ecuaciones:

$$x_R = s^2 + s + a$$

$$y_R = x_P^2 + (s + 1) * x_R$$

3.6.7 Algoritmo - ECIES

Existen varios algoritmos basados en curvas elípticas, pero en este caso se hará referencia al llamado “*Elliptic Curve Integrated Encryption Scheme*”.

Suponiendo que un usuario A quiere mandar un mensaje M usando este algoritmo a un usuario B , primero se debe publicar algunos parámetros. Estos son:

- Los parámetros de la curva (a y b).
- Un punto P perteneciente a la curva.
- El orden n de P en relación a la curva.

Luego, se procede a realizar los siguientes cálculos:

- a. El usuario B escoge un entero d que será su clave privada, y calcula $Q_B = dP$.
- b. El usuario B publica los puntos *su clave pública* Q_B .
- c. El usuario A obtiene los puntos P y Q_B .
- d. El usuario A escoge un entero k que usará solo una vez, y calcula Q_A de la misma manera que el usuario B .
- e. El usuario A calcula $z = kQ_B$ que será el secreto compartido entre A y B , y luego utilizando una “Key Derivation Function” obtendrá un par claves k_1 y k_2 , las cuales se ocuparán para cifrar el mensaje con un algoritmo simétrico y para autenticar el mensaje mediante un algoritmo MAC.
- f. El usuario A cifra el mensaje M utilizando la clave k_1 y un algoritmo simétrico como AES, obteniendo C .
- g. El usuario A aplica la función MAC con la clave k_2 , obteniendo como resultado r .
- h. El usuario A envía el trío (Q_A, C, r) a B .
- i. El usuario B calcula $z = dQ_A$, y luego genera k_1 y k_2 de la misma manera que el usuario A .
- j. El usuario B comprueba que al utilizar k_2 en el algoritmo MAC elegido, la salida sea igual a r .
- k. El usuario B utilizando el algoritmo simétrico elegido, descifra el mensaje M , a través de C y la clave k_1 generada.

El procedimiento anterior puede ser representado con la figura 3.11.

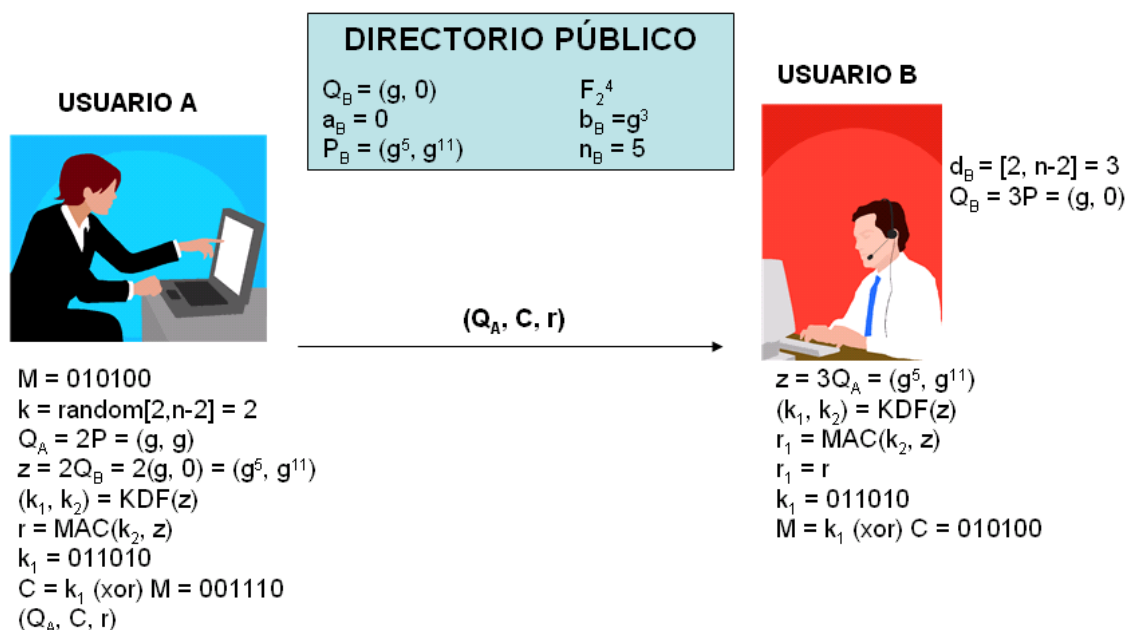


Figura 3.11: “Sistema criptográfico ECIES”

En este caso, el usuario A quiere enviarle el mensaje binario $M = 010100$ al usuario B, para lo cual extrae la información pública de B, es decir, la llave pública Q_B , los coeficientes de la ecuación de la curva elíptica a_B y b_B , el cuerpo finito F_2^4 con el que se trabaja, el punto P y el orden n_B , el cual se obtiene al encontrar el coeficiente de P con que la suma es igual a O , es decir $5P$. El algoritmo simétrico escogido es la función XOR. Una vez obtenidos estos datos, se procede a hacer los cálculos indicados anteriormente en la descripción del algoritmo.

IV. ESTADO DEL ARTE

4.1 Seguridad en comercio electrónico

Para garantizar la seguridad de los sitios de comercio electrónico en Internet, existe una solución comúnmente usada, llamada SSL¹.

SSL fue diseñado y propuesto en 1994 por Netscape Communications Corporation, pero como todo nuevo sistema necesita correcciones y revisiones constantes, no fue hasta la versión 3.0 que se hizo robusto y famoso. Actualmente es soportado por la mayoría de los browsers famosos tales como Internet Explorer y Mozilla Firefox.

Su uso para el comercio electrónico en Internet corresponde a lo que se llama HTTPS (Hypertext Transfer Protocol over SSL), el cual ocupa algoritmos simétricos como DES, 3DES, RC2, RC4, IDEA o AES para cifrar los datos intercambiados entre el servidor y el cliente, así como también ocupa algoritmos asimétricos como RSA para cifrar e intercambiar la clave usada por los algoritmos simétricos, y finalmente ocupa un algoritmo hash tal como MD5 o SHA-1 para verificar la integridad del mensaje.

SSL tiene varios pasos a seguir, lo cual se representa en la figura 4.1.

¹ SSL, siglas de Secure Socket Layer.

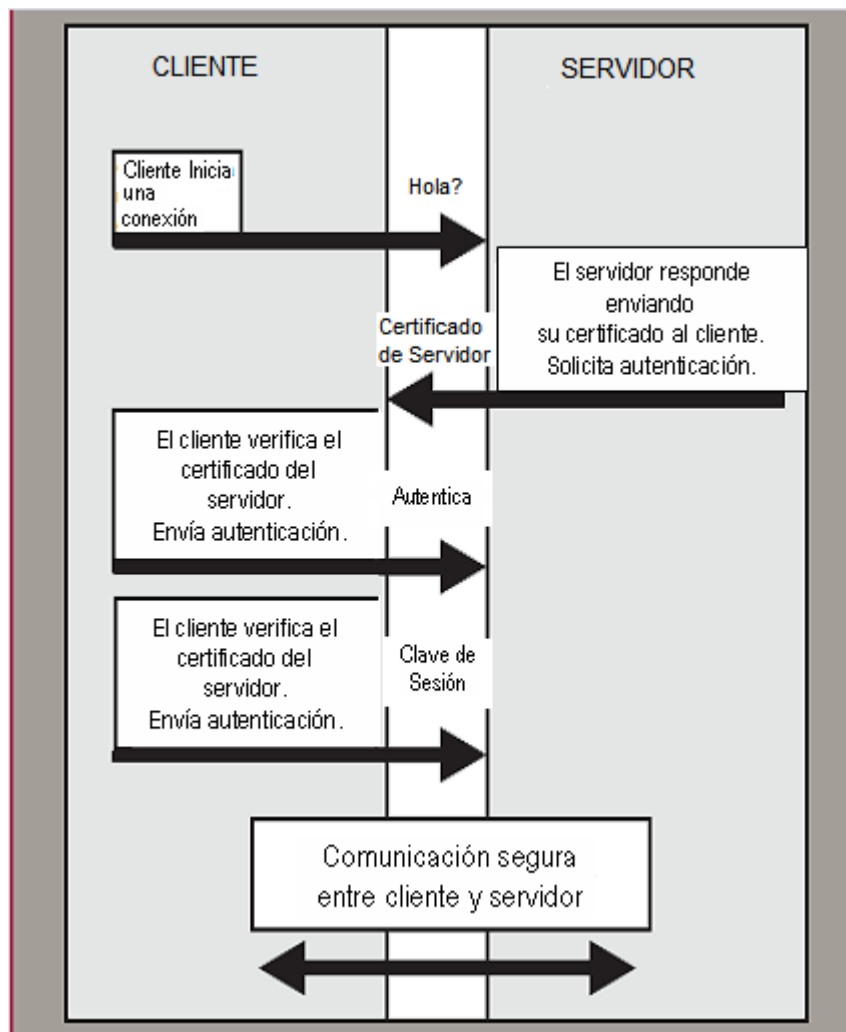


Figura 4.1 “Esquema SSL entre cliente y servidor”

En el comercio electrónico, los servidores manejan lo que se llama “Certificados Digitales” emitidos generalmente por empresas acreditadas en este rubro. Estos certificados contienen información sobre la empresa dueña del servidor y de los algoritmos usados para la comunicación segura.

En la figura 4.1 el cliente se conecta al sitio seguro y el cliente envía un saludo de comunicación con el servidor, luego el servidor responde enviando el certificado con sus datos, el cliente revisa que el certificado este en orden para luego realizar la autenticación necesaria. Una vez cumplida esa etapa, el cliente utiliza el algoritmo asimétrico informado en el certificado (generalmente RSA) para enviar la clave de sesión generada, y así utilizar el algoritmo simétrico para una comunicación rápida y segura entre ambos.

Las entidades bancarias utilizan este protocolo contratando servicios de certificados con empresas extranjeras como VeriSign o Thawte. También existen en Chile empresas tal como E-Cert, ONCE, que emiten certificados para este tipo de comercio.

4.2 Seguridad en correos electrónicos

Proteger la información es un asunto de preocupación para la mayoría de las organizaciones. En el caso de los correos electrónicos, sin protección, los mensajes pueden ser destruidos o suplantados sin el consentimiento del remitente.

PGP (Pretty Good Privacy) es un programa con diversas funciones que ayudan a proteger la información cifrándola mediante el uso de criptografía de llave pública. Con PGP uno puede cifrar mensajes y archivos, con la opción de asegurar que el documento procede de la persona que dice remitirlo, utilizando firma digital.

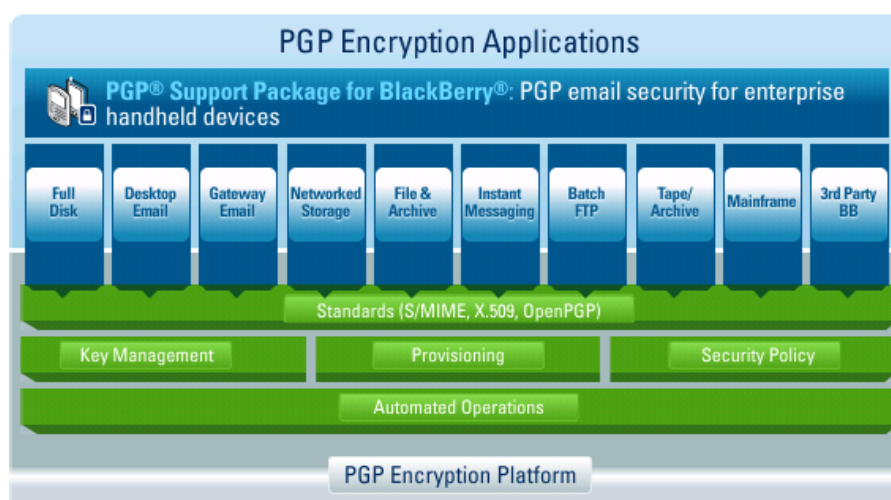


Figura 4.2: “Plataforma de encriptación PGP” – Fuente PGP

PGP ha desarrollado una gama de productos relacionados con la protección de la información, entre ellos:

- PGP Whole Disk Encryption: En muchas organizaciones, las personas transportan sus equipos o discos fuera del recinto protegido, por lo que están expuestos a robos. Esta aplicación cifra toda la información del

disco, evitando así, pérdidas mayores que puedan significar una catástrofe para la organización.

- PGP SDK: Grupo de herramientas que permiten a desarrolladores integrar PGP a sus aplicaciones.
- PGP Universal Gateway Email: Provee un manejo centralizado de la seguridad de email, sin necesidad que los clientes posean el software necesario para el cifrado.
- PGP Support Package for BlackBerry: Email seguro para equipos de mano.

Existe también la versión GNU de PGP, que es una alternativa gratuita para todos aquel que quiera un producto similar, sin embargo, solo se podrá obtener algunos de los productos ofrecidos por la empresa PGP.

4.3 Seguridad wireless

La evolución de la calidad de vida y la disminución de los precios en la tecnología de Laptops ha permitido a las personas adquirir con mayor facilidad estos dispositivos. La seguridad en los accesos a redes inalámbricas es un asunto que tiene que ir evolucionando rápidamente para poder satisfacer los requerimientos de las personas.

La seguridad en wireless está enfocada de diversas formas:

- Acceso y transporte de datos fuertemente protegidos.
- VPNs.
- Protección de acceso con contraseña.
- Cifrado de datos.
- SSL e IPSec.
- WEP y WPA.
- Protocolos de autenticación de usuarios (EAP).

El sistema de cifrado WEP permite el cifrado de la información mediante la utilización de una contraseña. Si bien no es considerado un método muy seguro, es un primer paso para la seguridad de la red.

Si se utiliza protocolos avanzados tales como EAP-TTS o TTLS, que son protocolos basados en autenticar dispositivos mediante el uso de criptografía asimétrica y/o certificados digitales, se puede mejorar la débil seguridad de WEP.

WPA es la evolución de WEP, y ofrece metodologías más robustas para el cifrado y autenticación.

4.4 Seguridad en tarjetas magnéticas

Las Smart Cards son tarjetas de bolsillo con circuitos integrados, utilizados en diversos ámbitos en organizaciones como bancos, operadoras de teléfono y corporaciones de seguros entre otras.

La gran ventaja de utilizarlas, respecto a las tarjetas magnéticas comunes es que permite ejecutar algoritmos criptográficos en su circuitería interna.



Figura 4.3: “Smart card, con circuitos internos”

Una de las empresas que ha tenido una participación muy activa en este tipo de tarjetas, es Motorola, quienes incorporaron un microprocesador de clave pública para poder trabajar con algoritmos como RSA.

También incluyen librerías de firmware que incluyen algoritmos tanto de llave secreta como privada (DES, RSA, DSS, SHA).

Existen Smart Cards que almacenan la llave a ocupar, utilizando algoritmos como DES o 3DES, y otras que generan llaves, utilizando comúnmente el algoritmo RSA.

También se puede almacenar certificados digitales y la llave privada, para firmar documentos o autenticarse utilizando la Smart Card.

4.5 Seguridad en PDAs y otros dispositivos móviles

La tecnología actualmente nos ofrece diversos aparatos móviles para motivos de entretenimiento, comercio y/o trabajo. Pero cada nueva tecnología lleva a preocuparse por la seguridad ofrecida para estos aparatos.

El acceso a Internet vía WAP por ejemplo, es algo que pueden realizar celulares, PDA, Blackberrys y otros equipos similares. A través de WAP uno puede realizar transacciones comerciales que necesitan ser seguras, es decir, también necesitan una gran infraestructura en donde la criptografía es uno de sus principales componentes.

El riesgo de pérdida o hurto de los dispositivos móviles está siempre presente, por lo que proteger la información sensible para una organización es de vital importancia. Esta información que vendedores, distribuidores de productos, encuestadores pueden almacenar en sus aparatos móviles es posible cifrarla, utilizando por ejemplo, en el caso de las PDA, la aplicación PDA Defense.

Los aparatos mencionados por lo general, no tienen un rendimiento como el de un computador de escritorio, por lo que necesitan algoritmos que se puedan ejecutar lo más rápidamente posible.

4.6 Seguridad en redes y VPN

Las VPN (Virtual Private Network) son sistemas que permiten a un computador en una red pública, conectarse a una red local.

Como las redes públicas son inseguras, se necesita un protocolo que pueda satisfacer las necesidades de seguridad de las conexiones. Uno de ellos es IPSec (Internet Protocol Security) el cual es un conjunto de extensiones de la familia del protocolo IP.

IPSec provee servicios de autenticación, integridad, control de acceso y confidencialidad, combinando algoritmos de clave pública, privada, hash y certificados digitales.



Figura 4.4: “Tecnologías de cifrado usados por IPSec”

Los algoritmos asimétricos participan dentro del protocolo de gestión de claves Internet Key Exchange (IKE) el cual permite negociar claves entre 2 entes.

V. ESTUDIOS

Debido a la problemática actual comentado anteriormente, es necesario buscar algoritmos que permitan proteger nuestra información anticipando los avances tecnológicos.

Existen un sin número de productos y servicios que de una u otra forma dependen de la tecnología.

Tanto organismos gubernamentales, como organismos comerciales están preocupados por la rapidez con la que crece la tecnología en contraste a la seguridad actual de los sistemas de información. Entre estos se pueden mencionar a NIST, Certicom, Sun Microsystems, IEEE entre otros.

5.1 NIST

Uno de los organismos que trabaja constantemente buscando estándares que permiten resguardar nuestros activos de información es el NIST (National Institute of Standards and Technology). Es un organismo federal que forma parte del departamento de comercio de EEUU cuyo fin es elaborar y promover patrones de medición, normas y tecnologías para aumentar la productividad, facilitar el comercio y mejorar la calidad de vida. Las áreas que dependen netamente del NIST en EEUU son biotecnología, nanotecnología, tecnologías de la información y fabricación avanzada.



Figura 5.1: “Logotipo de NIST”

Como los algoritmos asimétricos son utilizados generalmente para el intercambio de claves entre dos entes, para luego cifrar la información mediante un algoritmo simétrico, NIST ha realizado estudios sobre el tamaño de las claves que deben usar algunos algoritmos asimétricos en contraste a los tamaños de claves del algoritmo simétrico AES.

Tabla 5.1: “Tamaños para claves públicas para usar con AES” – Fuente NIST

NIST guidelines for public key sizes for AES			
ECC KEY SIZE (Bits)	RSA KEY SIZE (Bits)	KEY SIZE RATIO	AES KEY SIZE (Bits)
163	1024	1 : 6	
256	3072	1 : 12	128
384	7680	1 : 20	192
512	15 360	1 : 30	256

Supplied by NIST to ANSI X9F1

Como se puede observar en la tabla 5.1, para proteger la información utilizando una clave de 128 bits para el algoritmo AES, se puede realizar mediante la utilización de un algoritmo basado en curvas elípticas con clave de 256 bits, o bien utilizar el algoritmo RSA con clave de 3072 bits.

A medida que se requiere mayor seguridad, el ratio entre el tamaño de clave de un algoritmo basado en curvas elípticas y el algoritmo RSA se va haciendo más evidente.

Otro estudio realizado por NIST describió los tamaños de claves mínimos que tendrán que usarse para algunos algoritmos antes y después del año 2010.

Tabla 5.2: “Tamaño de claves y algoritmos para cada tipo de uso” – Fuente NIST

PIV Key Type	Período para uso	Algoritmos y tamaños de claves
PIV authentication key	31/12/2010	RSA (1024, 2048 ó 3072 bits) ECDSA(Curvas recomendadas de 224-283 bits)
	después del 31/12/2010	RSA (2048 o 3072 bits) ECDSA(Curvas recomendadas de 224-283 bits)
Card authentication key	31/12/2010	2TDEA 3TDEA AES-128, AES-192, AES-256 RSA(1024, 2048 ó 3072 bits) ECDSA(Curvas recomendadas de 224-283 bits)
	después del 31/12/2010	3TDEA AES-128, AES-192, AES-256 RSA(2048 ó 3072 bits) ECDSA(Curvas recomendadas de 224-283 bits)
Digital signature key	31/12/2008	RSA(1024, 2048 ó 3072 bits) ECDSA(Curvas recomendadas de 244-283 bits)
	después del 31/12/2008	RSA(2048 ó 3072 bits) ECDSA(Curvas recomendadas de 224-283 bits)
Key management key	31/12/2008	RSA key transport (1024, 2048 ó 3072 bits) ECDH ó ECC MQV (curvas recomendadas 224-283 bits)
	después del 31/12/2008	RSA key transport (2048 ó 3072 bits) ECDH ó ECC MQV (curvas recomendadas 224-283 bits)

La tabla 5.2 vuelve a describir lo antes comentado, de la diferencia de los tamaños de claves entre utilizar el algoritmo RSA para diversos usos en contraste a algún algoritmo basado en curvas elípticas.

La tabla también recomienda usar algunas curvas específicas para el gobierno federal, las que están descritas en el documento FIPS186-3. Las curvas con las que se trabaja fueron las siguientes:

Degree 163 Binary Field
$p(t) = t^{163} + t^7 + t^6 + t^3 + 1$
$n = 5846006549323611672814742442876390689256843201587$
$a = 1$
$b = 2\ 0a601907\ b8c953ca\ 1481eb10\ 512f7874\ 4a3205fd$
$P_x = 3\ f0eba162\ 86a2d57e\ a0991168\ d4994637\ e8343e36$
$P_y = 0\ d51fbc6c\ 71a0094f\ a2cdd545\ b11c5c0c\ 797324f1$

Degree 233 Binary Field
$p(t) = t^{233} + t^{74} + 1$
$n = 690174634679056378743475586227702555583981273734501355\backslash$ 5379383634485463
$a = 1$
$b = 066\ 647ede6c\ 332c7f8c\ 0923bb58\ 213b333b\ 20e9ce42\ 81fe115f\ 7d8f90ad$
$P_x = 0fa\ c9dfcbac\ 8313bb21\ 39f1bb75\ 5fef65bc\ 391f8b36\ f8f8eb73\ 71fd558b$
$P_y = 100\ 6a08a419\ 03350678\ e58528be\ bf8a0bef\ f867a7ca\ 36716f7e\ 01f81052$

Degree 283 Binary Field
$p(t) = t^{283} + t^{12} + t^7 + t^5 + 1$
$n = 7770675568902916283677847627294075626569625924376904889$ $109196526770044277787378692871$
$a = 1$
$b = 27b680a\ c8b8596d\ a5a4af8a\ 19a0303f\ ca97fd76\ 45309fa2$ $a581485a\ f6263e31\ 3b79a2f5$
$P_x = 5f93925\ 8db7dd90\ e1934f8c\ 70b0dfec\ 2eed25b8\ 557eac9c$ $80e2e198\ f8cdbcdd\ 86b12053$
$P_y = 3676854\ fe24141c\ b98fe6d4\ b20d02b4\ 516ff702\ 350eddb0$ $826779c8\ 13f0df45\ be8112f4$

Degree 409 Binary Field
$p(t) = t^{409} + t^{87} + 1$
$n = 6610559687902485989519153080327710398284046829642812192$ $84648798304157774827374805208143723762179110965979867288$ 366567526771
$a = 1$
$b = 021a5c2\ c8ee9feb\ 5c4b9a75\ 3b7b476b\ 7fd6422e\ flf3dd67$ $4761fa99\ d6ac27c8\ a9a197b2\ 72822f6c\ d57a55aa\ 4f50ae31$ $7b13545f$
$P_x = 15d4860\ d088ddb3\ 496b0c60\ 64756260\ 441cde4a\ fl771d4d$ $b01ffe5b\ 34e59703\ dc255a86\ 8a118051\ 5603aeab\ 60794e54$ $bb7996a7$
$P_y = 061b1cf\ ab6be5f3\ 2bbfa783\ 24ed106a\ 7636b9c5\ a7bd198d$ $0158aa4f\ 5488d08f\ 38514f1f\ df4b4f40\ d2181b36\ 81c364ba$ $0273c706$

Degree 571 Binary Field
$p(t) = t^{571} + t^{10} + t^5 + t^2 + 1$
$n = 3864537523017258344695351890931987344298927329706434998$ $65723525145151914228956042453614399938941577308313388112$ $19269444862468724628168130702345282883033324113931911052$ 85703
$a = 1$
$b = 2f40e7e\ 2221f295\ de297117\ b7f3d62f\ 5c6a97ff\ cb8ceffl$ $cd6ba8ce\ 4a9a18ad\ 84ffabbd\ 8efa5933\ 2be7ad67\ 56a66e29$ $4afd185a\ 78ff12aa\ 520e4de7\ 39baca0c\ 7ffeff7f\ 2955727$
$P_x = 303001d\ 34b85629\ 6c16c0d4\ 0d3cd775\ 0a93d1d2\ 955fa80a$ $a5f40fc8\ db7b2abd\ bde53950\ f4c0d293\ cdd711a3\ 5b67fb14$ $99ae6003\ 8614f139\ 4abfa3b4\ c850d927\ e1e7769c\ 8eec2d19$
$P_y = 37bf273\ 42da639b\ 6dccfffe\ b73d69d7\ 8c6c27a6\ 009cbbca$ $1980f853\ 3921e8a6\ 84423e43\ bab08a57\ 6291af8f\ 461bb2a8$ $b3531d2f\ 0485c19b\ 16e2f151\ 6e23dd3c\ 1a4827af\ 1b8ac15b$

Podemos notar que en todas estas curvas, el coeficiente de a es 1, lo que NIST describe como una ecuación seudo aleatoria de una curva elíptica, en donde el cofactor es 2, es decir, el número de puntos pertenecientes a este cuerpo es, $2n$.

Existen curvas que usa otros coeficientes para el parámetro a y b los cuales son llamados curvas de Koblitz, pero mantienen el tamaño de la clave sugeridas por el NIST.

Estas son las curvas recomendadas para trabajar en cuerpos binarios, sin embargo existen otras curvas recomendadas para trabajar en cuerpos con números primos que son equivalentes a en nivel de seguridad a las recomendadas por el NIST, las cuales tienen tamaño de clave de 192, 224, 256, 384 y 521 bits.

Para representar las curvas como objetos, NIST creó la siguiente tabla:

Tabla 5.3: “Identificador ANSI para las curvas aprobadas por NIST” – Fuente NIST

Asymmetric Algorithm	Object Identifier
Curve P-224	ansip224r1 ::= { iso(1) identified-organization(3) certicom(132) curve(0) 33 }
Curve K-233	ansit233k1 ::= { iso(1) identified-organization(3) certicom(132) curve(0) 26 }
Curve B-233	ansit233r1 ::= { iso(1) identified-organization(3) certicom(132) curve(0) 27 }
Curve P-256	ansip256r1 ::= { iso(1) member-body(2) us(840) ansi-X9-62(10045) curves(3) prime(1) 7 }
Curve K-283	ansit283k1 ::= { iso(1) identified-organization(3) certicom(132) curve(0) 16 }
Curve B-283	ansit283r1 ::= { iso(1) identified-organization(3) certicom(132) curve(0) 17 }

Se puede ver que el nombre de representación de los objetos, provienen del “American National Standards Institute” (ANSI), organismo sin fines de lucro que supervisa el desarrollo de estándares para producto, servicios y sistemas en EEUU.



Figura 5.2: “Logotipo de ANSI”

5.1.1 Metodología de verificación de algoritmos de NIST

La metodología que utiliza NIST para hacer las pruebas de los algoritmos y módulos criptográficos se puede resumir de la siguiente manera:

- a. **Validación de la implementación de los algoritmos y módulos criptográficos**, NIST desarrolló documentos en los que se especifican como realizar las mediciones necesarias para testear los diferentes tipos de algoritmos y módulos criptográficos en los llamados “Cryptographic Algorithm Validation Program” (CAVP) y “Cryptographic Module Validation Program” (CMVP).
- b. **Acreditación de los laboratorios independientes**. NIST tiene muchos colaboradores independientes, pero para poder certificar que los estudios realizados en sus laboratorios sean de alta calidad, se definió una serie de procedimientos y requerimientos que deben cumplir los algoritmos

como mínimo, los cuales son especificados en checklists y en formularios, los que son constantemente revisados.

- c. **Desarrollo de herramientas para testing.** Para cada tipo de testing, es necesario desarrollar una herramienta que entregue de manera clara la información de las métricas necesarias a estudiar.
- d. **Proveer de soporte técnico a foros de la industria.** Una vez realizado los estudios, las empresas interesadas en implementar dicha tecnología, reciben soporte técnico por especialistas del área a través de foros, en donde se discuten las problemáticas e intereses de las organizaciones.
- e. **Programa de educación.** NIST también realiza un programa educacional, para enterarse de las nuevas tecnologías que surgen en el área de la criptografía.
- f. **Programa de Entrenamiento.** Luego se procede a entrenar al personal en la materia correspondiente a los estudios realizados, para su continuo mejoramiento en el área.
- g. **Programa de Extensión.** Una vez realizada las etapas anteriores, se procede a publicar masivamente la información de los estudios, con el fin de que las organizaciones tomen las medidas correspondientes, en el caso de tener que hacerlo, o bien, puedan mejorar la infraestructura tecnológica que posean.

5.2 Certicom

Certicom es la empresa comercial líder en lo que se refiere a investigación y desarrollo de criptografía de curva elíptica. Fue fundada en 1985 por Gordon Agnew, docente de la Universidad de Waterloo en Ontario, Canadá. Posee alrededor de 350 patentes en las cuales cubre muchos aspectos de la criptografía de curva elíptica.

La arquitectura de seguridad de Certicom contiene varias aplicaciones criptográficas que ofrecen servicios como SSL, IPsec, entre otros.

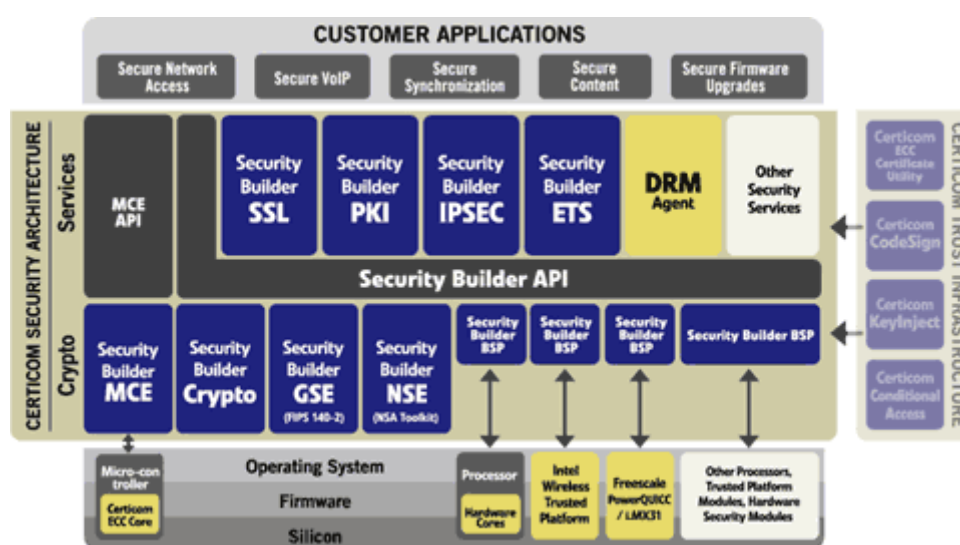


Figura 5.3 “Productos de Certicom” – Fuente Certicom

Todos estos componentes comparten una API lo que permite a desarrolladores maximizar la portabilidad, así como también volver a usar el código en otros productos.

Unos de los mayores complejos que las organizaciones tienen es pensar que la migración de un sistema criptográfico a otro es muy complicado, sin embargo, los productos de Certicom permiten fácilmente migrar de sistemas que usan RSA por ejemplo a uno que cumpla con los estándares actuales para criptografía de curva elíptica.

Algunos estudios de Certicom son expuestos en su newsletter “Code and Cipher”, en donde se habla de muchas de las ventajas que poseen los sistemas basados en criptografía de curva elíptica en comparación a otros sistemas como RSA.

Cada vez se conectan más dispositivos a Internet, y también se conectan dispositivos de menor tamaño, los cuales requieren un procesamiento de datos rápido y eficiente. La criptografía de curva elíptica usa menos ancho de banda que los algoritmos comúnmente usados para protocolos de SSL/TLS.

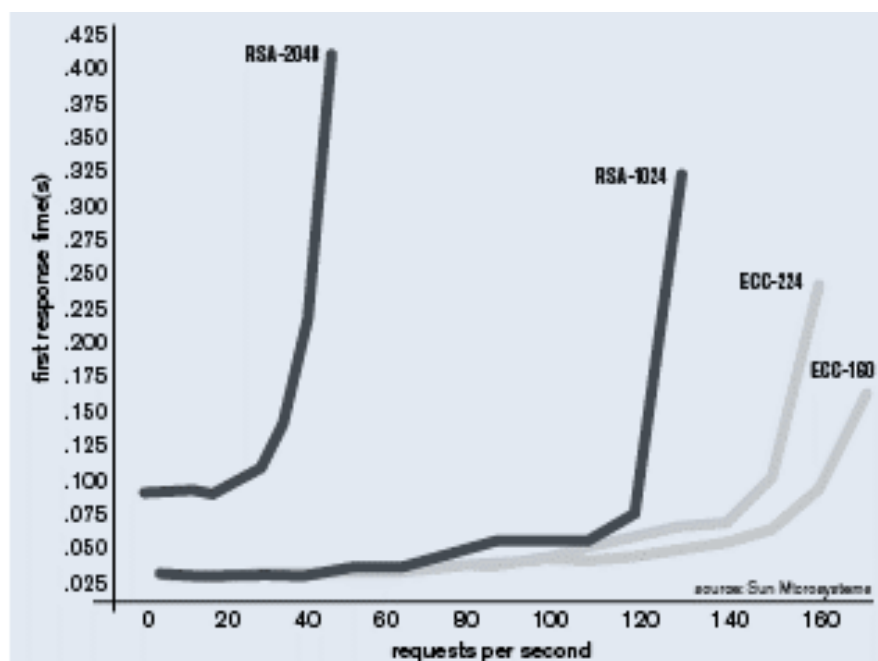


Figura 5.4: “Gráfico comparativo entre el tiempo de respuesta de un servidor al usar algoritmos criptográficos RSA y curvas elípticas según peticiones de transacciones por segundo” – Fuente Certicom

En la figura 5.4 se ve un gráfico comparativo de los tiempos de respuesta del servidor entre usar un algoritmo basado en curva elíptica y RSA. Como se ve claramente, usar el algoritmo RSA para mantener la seguridad que la tecnología hoy en día nos exige se va haciendo más complicado, en cambio al usar curvas elípticas, el impacto es mucho menor, y por lo tanto, el tiempo de respuesta es más rápido.

Podemos notar que al usar curvas elípticas aumenta las peticiones satisfactorias desde un 30% hasta un 270% en comparación a RSA.

Otro tema importante es comparar las operaciones que se hacen entre los diferentes sistemas. En la tabla 5.4, se muestra una comparativa que respalda lo mostrado en el gráfico anterior.

Tabla 5.4: “Comparación de performance de RSA y curvas elípticas” – Fuente Certicom

	ECC-160	RSA-1024	ECC-224	RSA-2048
Tiempo (ms)	3.69	8.75	5.12	56.18
Operaciones/seg	271.3	114.3	195.5	17.8
Ratio de Performance	2.4 : 1		11 : 1	
Ratio de tamaño de clave	1 : 6.4		1 : 9.1	

Aquí se exponen los sistemas con sus tamaños de claves equivalentes, mostrando tiempo de respuesta, operaciones por segundos, el ratio de rendimiento entre ellos y el ratio entre los tamaños de claves. Claramente se nota la superioridad en todos estos aspectos de la criptografía de curva elíptica respecto a RSA, incluso se observa que si se utiliza una curva con clave de 224 bits (equivalente a RSA de 2048), sigue siendo más rápido que usar RSA de 1024, con lo que aumentaríamos la seguridad y la velocidad actual de muchos sistemas basados en RSA.

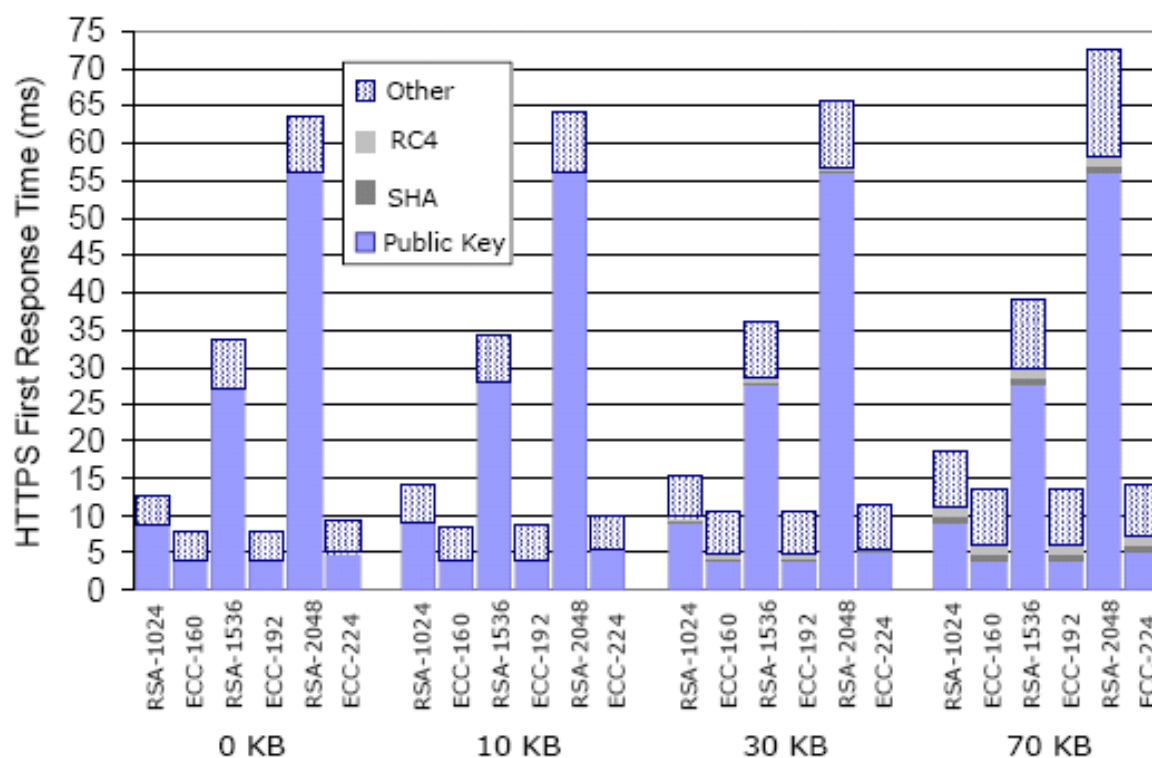


Figura 5.5: “Comparación del tiempo de respuesta de un servidor al usar algoritmos criptográficos RSA y curvas elípticas según tamaño de clave aplicada” – Fuente Certicom

En la figura 5.5 se muestra una comparación del tiempo de respuesta de un Web Server UltraSparc III de 900 MHz corriendo Apache 2.0.45, usando RSA y curvas elípticas con distintos tamaños de claves. Se observa que el tiempo de procesamiento al usar curvas elípticas se reduce desde un 29% hasta un 85% respecto a RSA.

Otro estudio importante es el tiempo que se demora un equipo en romper el nivel de seguridad de los algoritmos. A continuación se presenta un gráfico comparativo entre los tamaños de claves de los algoritmos RSA y curvas elípticas en contraste a los tiempos necesarios para romper su seguridad.

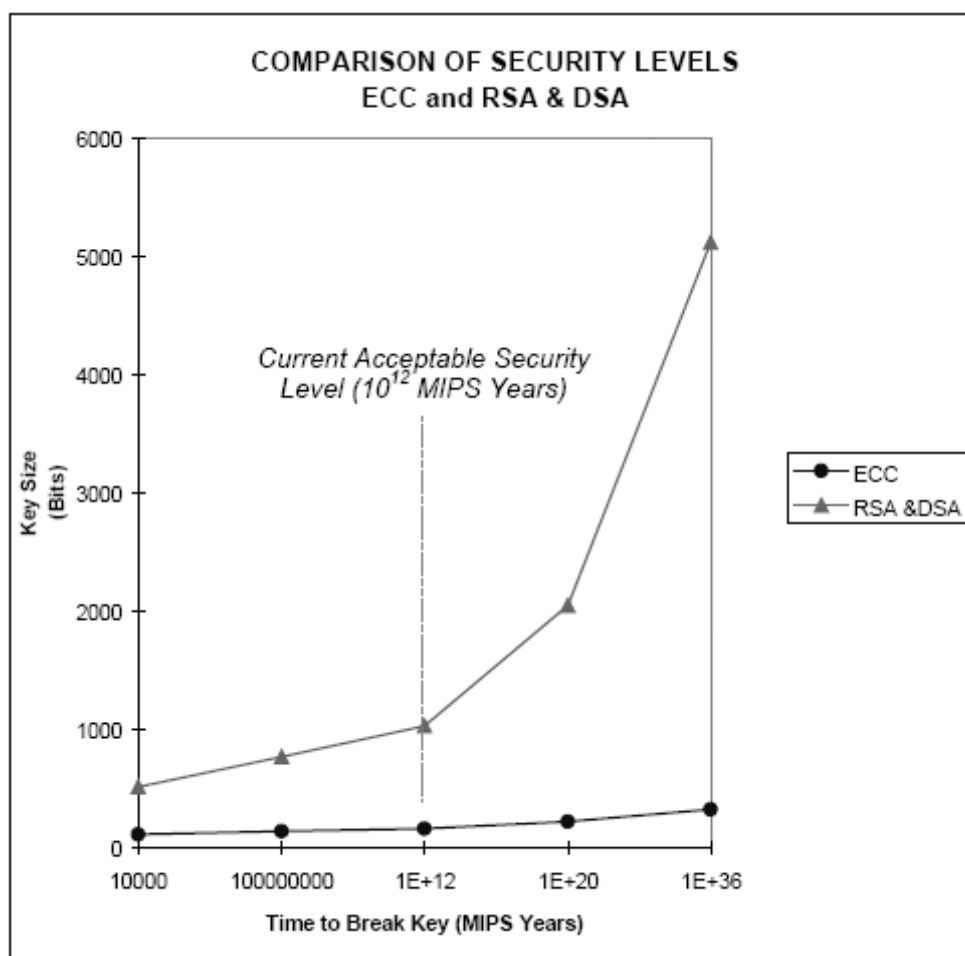


Figura 5.6: “Comparación de niveles de seguridad brindados por RSA y curvas elípticas” – Fuente Certicom

Actualmente un tiempo aceptable para medir la seguridad de los algoritmos según el tamaño de su clave es que se demore en romper aproximadamente en 10^{12} MIPS-años, es decir, un computador de 10^{12} millones de instrucciones por segundo funcionando durante un año, algo difícil teniendo en consideración la potencia de los computadores actuales. En la tabla 5.5 se muestra como han ido evolucionando la potencia de los procesadores:

Tabla 5.5: “Evolución de los procesadores según la cantidad de instrucciones por segundos que realizan” – Fuente Wikipedia

Procesador	IPS	Reloj	Año
Intel 8080	640 KIPS	2 MHz	1974
Intel 8086	800 KIPS	4,77 MHz	1974
Motorola 68000	1 MIPS	8 MHz	1979
Intel 486DX	54 MIPS	66 MHz	1992
PowerPC 600s (G2)	35 MIPS	33 MHz	1994
ARM 7500FE	35,9 MIPS	40 MHz	1996
PowerPC G3	525 MIPS	233 MHz	1997
ARM10	400 MIPS	300 MHz	1998
Zilog eZ80	80 MIPS	50 MHz	1999
Pentium 4 Extreme Edition	9726 MIPS	3,2 GHz	2003
ARM Cortex A8	2000 MIPS	1,0 GHz	2005
Xbox360 IBM “Xenon” Single Core	6400 MIPS	3,2 GHz	2005
AMD Athlon 64	8400 MIPS	2,8 GHz	2005
AMD Athlon FX-57	12000 MIPS	2,8 GHz	2005
AMD Athlon 64 Dual Core	18500 MIPS	2,2 GHz	2005
AMD Athlon 64 3800+ X2 (Dual Core)	18900 MIPS	2,2 GHz	2005
Overclocked AMD Athlon 64 3800+ X2 (Dual Core)	25150 MIPS	2,8 GHz	2006
Cell	6400 MIPS	3,2 GHz	2006
Procesador Cell de la PlayStation 3	21800 MIPS	3,2 GHz	2006
AMD Athlon FX-60 (Dual Core)	22150 MIPS	2,6 GHz	2006
Overclocked AMD Athlon FX-60 (Dual Core)	24300 MIPS	2,8 GHz	2006
Overclocked AMD Athlon FX-60 (Dual Core)	27100 MIPS	3,0 GHz	2006

En estos momentos, esto quiere decir que para RSA, el tamaño de clave mínima aceptable sea de 1024 bits, y para Curvas Elípticas, tan solo 163 bits. Sin embargo, existen estudios que indican que para el 2010, el tamaño de clave mínima aceptable deberá incrementarse, todo a raíz del avance de la tecnología.

5.2.1 Certicom y FIPS 140-2

Certicom también ha trabajado para satisfacer las necesidades de requerimientos para la seguridad a nivel de gobierno, la cual está regulada por las publicaciones FIPS (Federal Information Processing Standards). Una de las publicaciones más importantes es FIPS 140-2 la cual describe los requerimientos de la criptografía para el gobierno federal de los Estados Unidos, que tanto el hardware como el software deben tener para proteger la información sensible. La validación de los productos para cumplir con esta publicación es necesaria para poder optar a comercializar con el gobierno. Otros países como el Reino Unido y Canadá han adoptado lo señalado en la publicación, así como también lo han hecho industrias financieras y médicas.

NIST se encarga de validar los algoritmos y productos mediante una metodología muy cuidadosa. Respecto al FIPS 140-2 especifica detalladamente los módulos a tratar, puertos, interfaces, seguridad física, generación de números pseudo aleatorios entre otros.

Para recibir la validación un módulo criptográfico debe cumplir con:

- Tener bien definido el alcance criptográfico para que toda la información sensible sea debidamente protegida por el producto.
- Usar al menos un algoritmo aprobado por las publicaciones FIPS correctamente implementado.

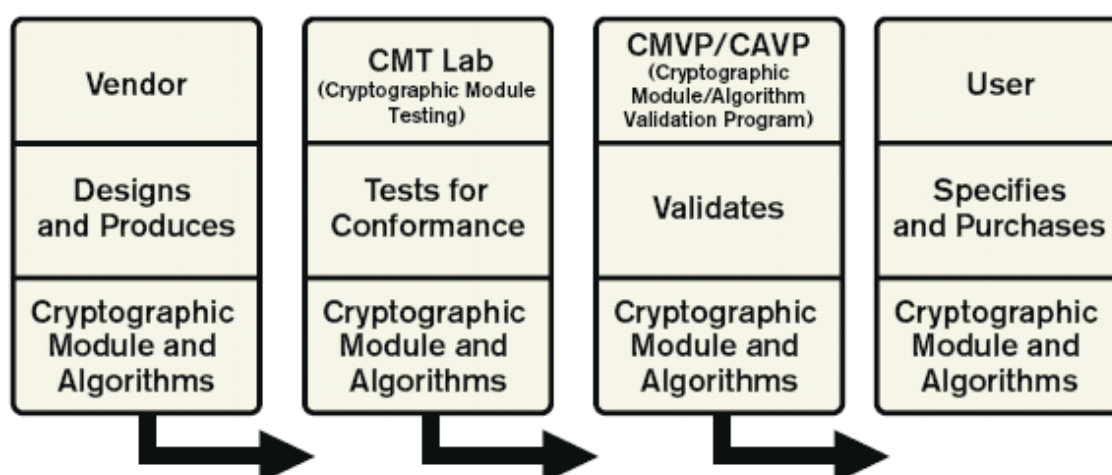


Figura 5.7: “Procedimiento de Validación de algoritmos y módulos” – Fuente Certicom

Para Certicom, si bien el mercado de gobierno le es atractivo y lucrativo, todo el proceso de validación para los productos ciertamente es una gran barrera. Este proceso de validación toma aproximadamente entre 8 y 12 meses, lo que es bastante tiempo, sin contar la gran cantidad de desarrolladores que se necesitarán. En términos monetarios, Certicom destaca que someter un módulo al proceso de validación cuesta entre 50000 y 100000 dólares, lo que es bastante dinero, además de estar actualizando los productos constantemente debido a las actualizaciones de las publicaciones.

Sin embargo, existe una manera para poder ahorrar tiempo y costo para desarrollar productos con su validación. Esta es incorporar un módulo pre-validado en los nuevos productos, pero teniendo las siguientes consideraciones:

- **Arquitectura Extensible:** Se refiere a una plataforma que comparte una misma API para poder cambiar fácilmente de módulos si se estima conveniente, con un mínimo esfuerzo en escritura de código
- **Flexibilidad de Actualización:** Al usar un módulo pre-validado permite mantener la certificación, por lo tanto es conveniente utilizar los mismos productos e ir actualizándolos.
- **Flexibilidad de Plataforma:** Si uno quiere estar certificado para cualquier plataforma, se debe validar para el primer nivel de seguridad, ya que los niveles más avanzados sólo permiten utilizarse en plataformas aprobadas por el Common Criteria.
- **Uso de Cables vs Wireless:** Se refiere a que existen algunos módulos que han sido optimizados para ciertos entornos, lo que su uso permitiría evitar algunos problemas específicos.

Teniendo el debido cuidado señalado anteriormente, Certicom desarrolló sus productos y en Octubre del año 2003, la NSA (National Security Agency) compró los derechos de licencia de muchos de los productos de Certicom para proteger la información de seguridad nacional, dando así un gran paso en lo que es la nueva era de la criptografía de algoritmos asimétricos. Con la frase “La próxima generación de la criptografía para proteger la información del gobierno de los Estados Unidos será el uso de la criptografía de curva elíptica”.

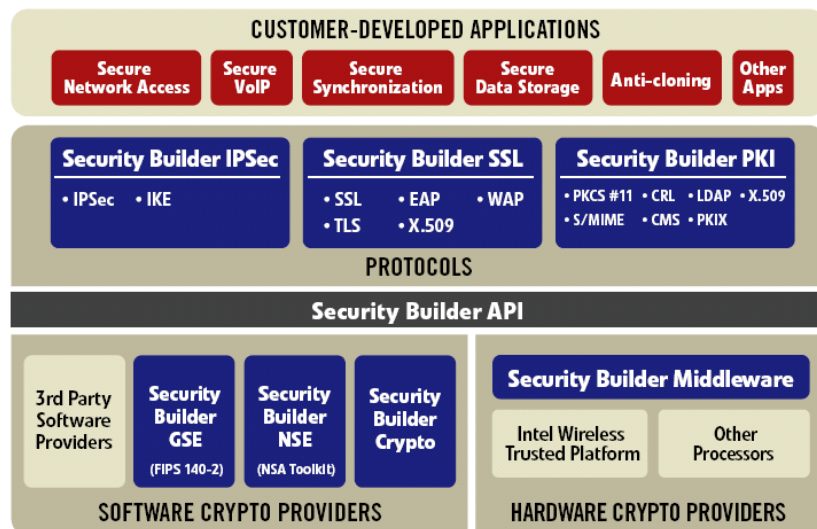


Figura 5.8: “Arquitectura de Seguridad para el Gobierno” – Fuente Certicom

En la figura 5.8, se presenta la gama de productos que ofrece Certicom para el gobierno, que incluye muchos aspectos, de los que destacan módulos para SSL, PKI, IPsec, Voz sobre IP entre otros.

5.3 Sun Microsystems

Sun Microsystems es una empresa que provee soluciones de infraestructura computacional que incluye tanto hardware como software, teniendo en cuenta también que ofrecen servicios. Uno de sus principales productos es la plataforma Java, el sistema operativo Solaris y los procesadores StorageTek y UltraSPARC.

En lo que concierne a criptografía, Sun realiza constantemente estudios, los cuales publica en su página de proyectos de estudios para la criptografía. Al igual que NSA, Sun denomina a la criptografía de curva elíptica como “criptografía de la próxima generación”, con lo cual se comprueba este fenómeno soportada por grandes organizaciones.

Los propósitos de Sun al realizar sus estudios son:

- Implementación de una librería de criptografía de curva elíptica y de arquitecturas de seguridad para varias plataformas, desde sensores pequeños hasta servidores web de alto rendimiento.
- Implementación de una arquitectura de hardware común para acelerar tanto ECC como RSA.

- Promover ampliamente el uso de la criptografía de curva elíptica en industrias, realizando lo siguiente:
 - a. Promover la estandarización de ECC en conjunto con SSL
 - b. Contribuir la tecnología de ECC a OpenSSL y NSS/Mozilla, las dos librerías de código abierto más famosas

Lo anterior permitiría a servidores Web Apache, que posee alrededor del 60% del mercado, comunicarse de manera segura y eficiente con dispositivos de menor capacidad utilizando tecnología basada en criptografía de curva elíptica.

Tabla 5.6: “Comparación de seguridad de algoritmos” – Fuente Sun Microsystems

Sistema Asimétrico	Ejemplos	Problema Matemático	Mejor método para resolver el problema matemático (tiempo de ejecución)
Factorización de Enteros	RSA, Rabin-Williams	Dado un número n , encontrar sus factores primos	Number Field Sieve: $\exp[1.923(\log n)^{1/3} (\log \log n)^{2/3}]$ (Sub-exponencial)
Logaritmo Discreto	Diffie-Helman (DH), DSA, El Gamal	Dado un primo n , y los números g y h , encontrar x tal que $h=g^x \bmod n$	Number Field Sieve: $\exp[1.923(\log n)^{1/3} (\log \log n)^{2/3}]$ (Sub-exponencial)
Logaritmo Discreto de Curva Elíptica	ECDH, ECDSA	Dada una curva elíptica E , y los puntos P y Q dentro de E , encontrar x tal que $Q = xP$	Algoritmo Pollar-rho: \sqrt{n} (Full Exponencial)

Sun dentro de sus publicaciones, comparó algunos algoritmos en conjunto con el problema matemático en que se basa su seguridad, y luego de analizar mejores métodos conocidos actualmente para romper esa seguridad, recalcó que los algoritmos basados en curva elíptica corren con ventaja frente a RSA, DH y DSA, ya que en caso de curva elíptica, su mejor método, es decir el algoritmo Pollar-rho tiene orden full exponencial, en contraste a los otros cuyo orden es sub-exponencial. Es por esto que ECC posee mayor seguridad por bit que RSA, esto es porque por cada 2 bits adicionales para la clave, atacarla requerirá 2 veces más de esfuerzo.

Sun ocupa gran parte de su esfuerzo en estudiar el comportamiento de ECC en relación a SSL, es por esto, que compara las operaciones que se realiza en los procesos de cliente y servidor, como lo indica la tabla 5.7.

Tabla 5.7: “Comparación operaciones en SSL” – Fuente Sun Microsystems

	RSA	ECDH-ECDSA
Cliente	$RSA_{Verificar} + RSA_{Encriptar}$	$ECDSA_{Verificar} + ECDH_{op}$
Servidor	$RSA_{Desencriptar}$	$ECDH_{op}$

En SSL participa un cliente y un servidor, el cliente en el caso de RSA, busca el certificado, verifica que sea correcto y luego utiliza la clave pública del servidor para cifrar un mensaje, después el servidor descifra el mensaje utilizando su clave privada. Cabe señalar que lo común es que los servidores tengan una clave pública pequeña, para que los clientes no demoren mucho en cifrar los mensajes, por lo que la clave privada del servidor es muy grande, y es en el proceso de descifrado donde el servidor ocupa más tiempo para terminar con el intercambio de claves.

A diferencia de RSA, las operaciones de SSL utilizando ECC, demora menos, ya que, el proceso de intercambio de claves se basa en operaciones ECDH, es decir, tanto el cliente como el servidor calculan el secreto compartido con puntos de curva elíptica, lo que es mucho más rápido que la operación de descifrado de RSA.

SSL se ocupa en varios tipos de sitios Web. Sun ha estudiado principalmente dos tipos de sitios:

- Sitios basados en carros de compra, donde la información que se debe proteger generalmente corresponde a números de tarjetas de crédito e información personal del cliente. Un ejemplo de esto es el sitio de compras Amazon, en donde SSL es usado en la finalización de la compra y no cuando uno esta navegando en el sitio.
- Sitios financieros, en donde en la página de inicio hay un módulo de inicio de sesión protegido por SSL, y una vez logueado, toda la navegación está protegida.

Tal como Certicom, Sun estudió algunos tipos de métricas interesantes como son:

- First-Response Time, el cual corresponde al tiempo de espera entre la inicialización de SSL y recibir el primer paquete de respuesta HTTPS.
- Fetches per second, mide el nivel de cumplimiento de las solicitudes de páginas Web.

Sun comprobó que el uso de ECC optimiza el rendimiento de los servidores en relación a RSA, al medir sus resultados con herramientas públicas como son `http_load` y el comando `speed` dentro de SSL.

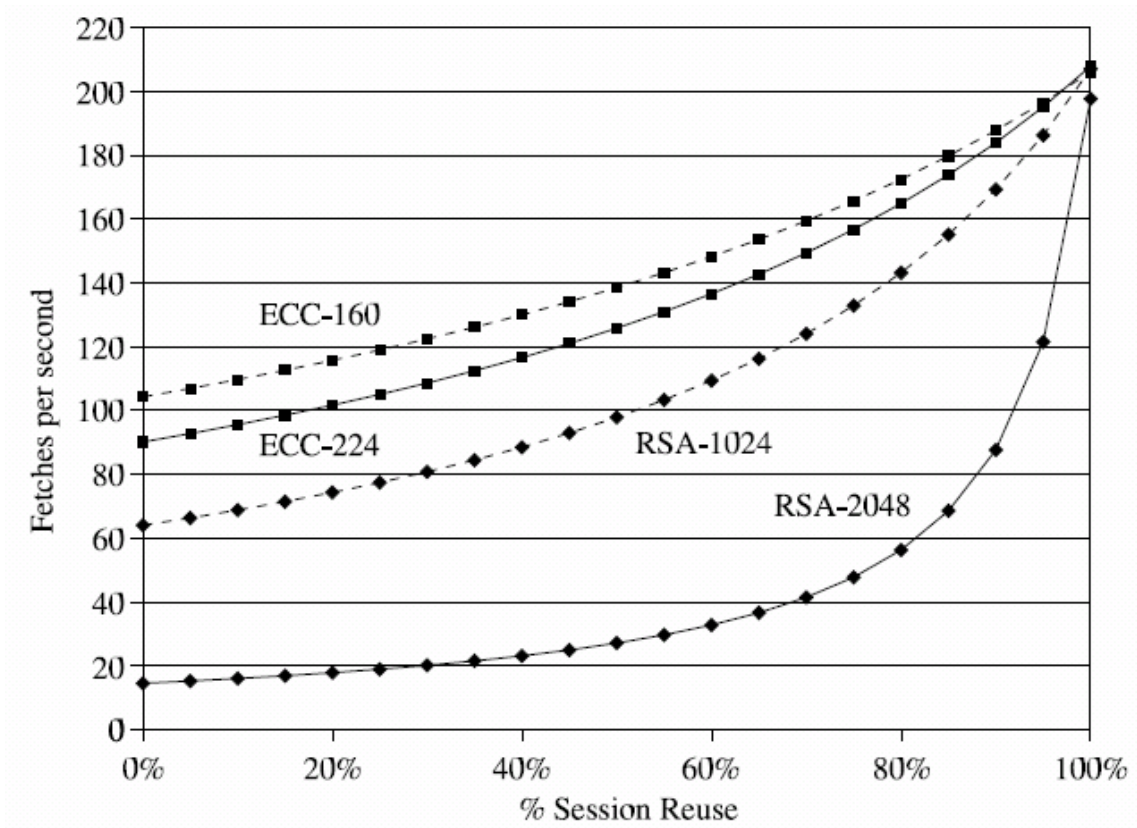


Figura 5.9: “Nivel de cumplimiento versus reutilización de sesiones” – Fuente Sun Microsystems

La figura 5.9 fue obtenida midiendo el máximo rendimiento utilizando `http_load` para páginas de 30 KB. Los rendimientos que no se midieron con `http_load`, fueron calculados analíticamente usando la fórmula que se muestra en la figura 5.10.

$$T_r = \frac{1}{(1 - \frac{r}{100})/T_0 + (\frac{r}{100})/T_{100}}$$

Figura 5.10: “Fórmula de throughput según la realización de sesiones”

En la fórmula, T_r es el throughput para un porcentaje r de reutilización de las sesiones. Tal como se esperaba, al ir aumentando el porcentaje de sesiones rechazadas disminuía el impacto de rendimiento de escoger uno u otro sistema asimétrico. Pero la diferencia se nota cuando los valores de reutilización son altos, como 90% (una sesión nueva por cada 10 obtenidas), en donde ECC puede manejar alrededor de 11% mas peticiones que RSA con claves pequeñas y de 110% aproximadamente con claves grandes.

Sun quiere adaptar toda la tecnología actual a un esquema que integre la nueva generación criptográfica. Esto se puede resumir en la figura 5.11.

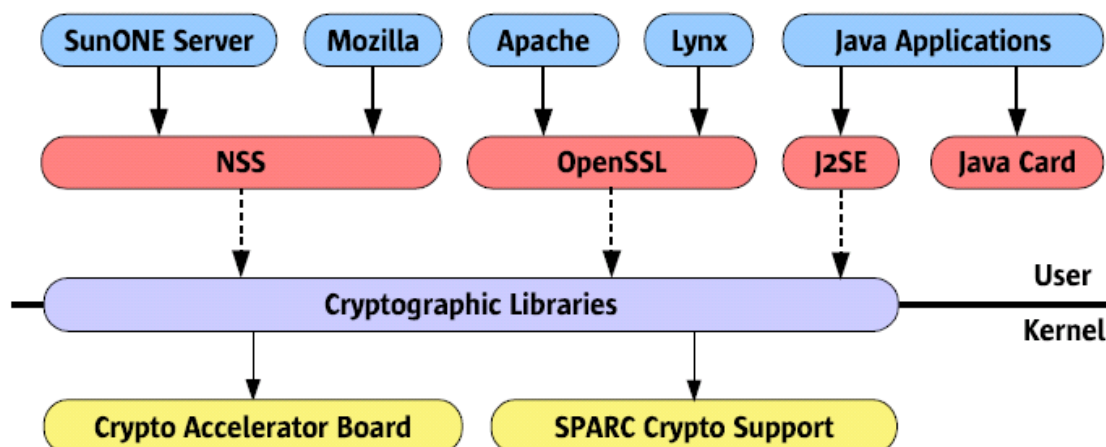


Figura 5.11: “Esquema de adaptación de la tecnología criptográfica”– Fuente Sun Microsystems

Lo que se pretende es que integrar las librerías con aceleradores que permitan realizar las operaciones de manera más rápida. A la vez, integrar a los navegadores usuales como Mozilla, con tecnología de curva elíptica al usar SSL, tanto en servidores web Apache como en otros comerciales. No hay que dejar de lado las aplicaciones basadas en Java, como J2SE utilizado mucho en empresas, y las Java Card, que necesitan integrarse a la nueva generación, ya que como dice Sun, “La seguridad ya no es una opción, se esta convirtiendo en una parte integral de una arquitectura de sistema”.

5.4 IEEE

IEEE (Institute of Electrical and Electronics Engineers), es una organización mundial, sin fines de lucro, que se dedica al desarrollo de estándares. Se creó en el año 1884, por grandes personalidades como Thomas Alva Edison, Alexander Gram. Bell y Franklin Leonard Pope.



Figura 5.12: “Logo de IEEE”

Cuando se fusionaron el AIEE (American Institute of Electrical Engineers) y el IRE (Institute of Radio Engineers) en el año 1963, se cambió el nombre por el actual.

Sus estándares representan prestigio y calidad en las áreas derivadas de la eléctrica, tales como la ingeniería computacional, tecnologías biomédicas, telecomunicaciones, etc.

La criptografía es una rama considerada fundamental en los procesos de las organizaciones, es por esto que se preocupan por el desarrollo de estándares en esta materia. Una de los estándares más estudiados por las grandes organizaciones, precisamente es el llamado “Standard Specifications For Public-Key Cryptography” creado por IEEE en su proyecto P1363.

Los jefes del proyecto han ido cambiando en este proyecto que se originó en 1994. Primero estuvo Buró Kaliski perteneciente a RSA Security, luego Ari Singer de NTRU, y actualmente es William Whyte, de NTRU Cryptosystems Inc también.

El alcance del proyecto abarca los siguientes temas:

- Criptografía asimétrica tradicional.
- Criptografía asimétrica basada en problemas fuertes en Lattices.
- Criptografía asimétrica basada en passwords.
- Criptografía asimétrica basada en identidad usando pares.

5.4.1 Criptografía asimétrica tradicional

Esta especificación trabaja con 3 tipos de esquemas:

- **Intercambio de claves:** Se especifican los algoritmos para el intercambio de claves de los sistemas. Estos son:
 - **DL/ECKAS-DH1 - DL/ECKAS-DH2:** Algoritmos basados en el problema de logaritmo discreto, o bien logaritmo discreto de curva elípticas. Es decir los algoritmos de Diffie-Hellman y su similar para curvas elípticas.

- **DL/ECKAS-MQV:** También basado en el problema de logaritmo discreto y logaritmo discreto de curva elíptica, utilizando el algoritmo de Menezes-Qu-Vanstone.
- **Firmas:** Especifica los algoritmos para utilizar en firmas digitales. Estos son:
 - **DL/ECSSA:** Utiliza los algoritmos DSA, ECDSA, Nyberg-Rueppel, y Nyberg-Rueppel para curva elíptica.
 - **IFSSA:** Basado en factorización de enteros. Utiliza variantes de RSA como Rabin Williams, ESIGN.
 - **DL/ECSSR:** Similar a DL/ECSSA pero con recuperación.
 - **DL/ECSSR-PV:** Similar a DL/ECSSR ocupando la versión de Pintsov-Vanstone.
 - **IFSSR:** Similar a IFSSA pero con recuperación.
- **Cifrado:** Especifica esquemas de cifrado para el Standard. Estos son:
 - **IFES:** Basados en factorización de enteros. Principalmente RSA.
 - **DL/ECIES:** Basado en el problema de logaritmo discreto de curva elíptica. Elliptic Curve Integrated Encryption Écheme (ECIES). Variante de ElGamal, DHAES.
 - **IFES/EPOC:** Similar a IFES, ocupando la versión EPOC.

5.4.2 Criptografía asimétrica basada en problemas fuertes en Lattices

Especifica técnicas criptográficas basadas en problemas fuertes en Lattices, que son un método de análisis de datos que toma como base una matriz especificando un set de objetos y atributos.

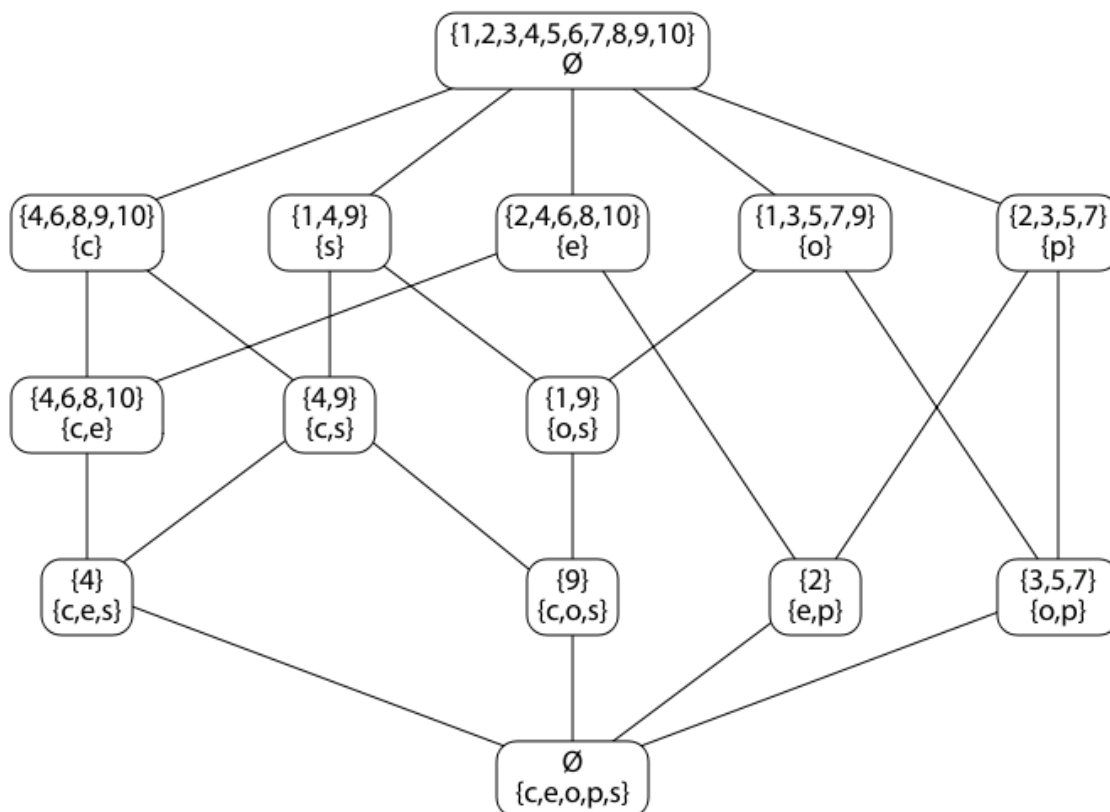


Figura 5.13: “Ejemplo de Lattice” – Fuente Wikipedia

El propósito es incluir primitivas matemáticas para derivación de claves secretas, cifrado asimétrico, identificación, firmas digitales y esquemas criptográficos basados en lattices.

5.4.3 Criptografía asimétrica basada en passwords

El objetivo de este documento es incluir esquemas para intercambio de claves con autenticación vía password. Esto incluye:

- **BPKAS-PAK:** Corresponde a un esquema balanceado, versión PAK.
- **BPKAS-PPK:** Similar a la anterior pero utiliza versión PPK.
- **BPKAS-SPEKE:** Similar a los anteriores, utilizando intercambio de claves simple exponencial.
- **APKAS-AMP:** Corresponde a un esquema aumentado para el intercambio de claves, versión AMP.
- **APKAS-BSPEKE2:** Similar al anterior pero versión BSPEKE2.
- **APKAS-PAKZ:** Similar al anterior pero versión PAKZ.

- **APKAS-SRP3 - APKAS-SRP6:** Similar al anterior pero utiliza password seguro remoto.
- **APKAS-WSPEKE:** Similar al anterior pero versión WSPEKE.
- **PKRS-11:** Corresponde a un esquema de recuperación de la llave de autenticación para el password.

5.4.4 Criptografía asimétrica basada en identidad usando pares

Este documento especifica esquemas basados en mapas bilineales sobre curvas elípticas, es decir, puntos de curva elíptica. Define algoritmos para el cómputo de los puntos, parámetros adecuados de curva, etc.

5.5 RSA Laboratories

RSA Laboratories es un centro de estudios perteneciente a RSA, la división de seguridad de EMC.



Figura 5.14: “Logo de RSA Laboratories”

RSA, es la empresa que patentó el algoritmo del mismo nombre y un gigante de servicios de seguridad basados en criptografía. Otros productos famosos son sus librerías criptográficas BSAFE y mecanismos de autenticación como SecurID.

Para realizar sus estudios, RSA Laboratories tiene dos áreas principales:

- **Técnicas avanzadas para la autenticación de usuarios:** Se dedican a estudiar los nuevos paradigmas de autenticación con el fin de proveer seguridad y a la vez, ser eficiente en su uso.

- **Privacidad y Seguridad para la identificación mediante Radio**

Frecuencia: Se dedican a estudiar mecanismos de autenticación para sistemas de radio frecuencia y tecnologías wireless.

Pero sin duda, los trabajos de estudios más famosos de esta organización, son sus iniciativas de estándares PKCS (Public Key Cryptography Standards), los cuales son utilizados en muchos sistemas de información. En la tabla 5.8 se puede ver un resumen de la iniciativa de estándares.

Tabla 5.8: “Estándares PKCS” – Fuente RSA Laboratories

ID	NOMBRE	DESCRIPCIÓN
PKCS #1	Estándar criptográfico RSA.	Estándar que define el formato para el cifrado utilizando el algoritmo RSA.
PKCS #3	Estándar de intercambio de claves Diffie-Helman.	Estándar que describe una metodología para implementar el mecanismo de Diffie-Helman.
PKCS #5	Estándar de cifrado basado en passwords.	Estándar que provee recomendaciones para la implementación de criptografía basada en passwords.
PKCS #6	Estándar de sintaxis de certificados extendidos.	Estándar que describe la sintaxis consistente en certificados y atributos para la utilización de certificados extendidos.
PKCS #7	Estándar sobre la sintaxis del mensaje criptográfico.	Estándar que describe la sintaxis para la información que debe ser cifrada, como las firmas digitales.
PKCS #8	Estándar sobre la sintaxis de la información de clave privada.	Estándar que describe la información de clave privada y sus atributos para ciertos algoritmos, con el fin de crear confianza respecto a la información cifrada.
PKCS #9	Tipos de atributos seleccionados.	Estándar que define los tipos de atributos para usar en los estándares PKCS #6, PKCS #7, PKCS #8 y PKCS #10.
PKCS #10	Estándar de solicitud de	Estándar que describe la sintaxis para la

	certificación.	solicitud de un certificado digital de una llave pública, un nombre, y otros atributos.
PKCS #11	Estándar de Interfaz de token criptográfico.	Estándar que especifica una API, para utilizar en dispositivos que almacenan información criptográfica y realizan operaciones criptográficas.
PKCS #12	Estándar de sintaxis de intercambio de información personal.	Estándar que especifica un formato para almacenamiento o transporte de las claves privadas de los usuarios, certificados, etc.
PKCS #13	Estándar de criptografía de curva elíptica.	Estándar en desarrollo. Incluirá parámetros, generación de claves, validación, firmas digitales, cifrado, intercambio de claves, etc.
PKCS #15	Estándar de formato de información de token criptográfico.	Estándar que permite a los usuarios utilizar tokens para identificarse frente a múltiples aplicaciones.

Cabe señalar que los estándares PKCS #2 y PKCS #4 quedaron obsoletos al incluirse dentro de PKCS #1. En cuanto a PKCS #14 se dice que está en desarrollo, y que corresponde a un estándar para la generación de números pseudo-aleatorios, sin embargo, no aparece información oficial en sitio de los PKCS.

El estándar PKCS #1 se usa en la mayoría de los sistemas de cifrado a nivel comercial, todo tipo de entidades hace uso de sistemas basados en este estándar. Un ejemplo es la página de correos electrónicos GMAIL, que usa un certificado de GOOGLE. En la figura 5.15 se muestra el uso del estándar:

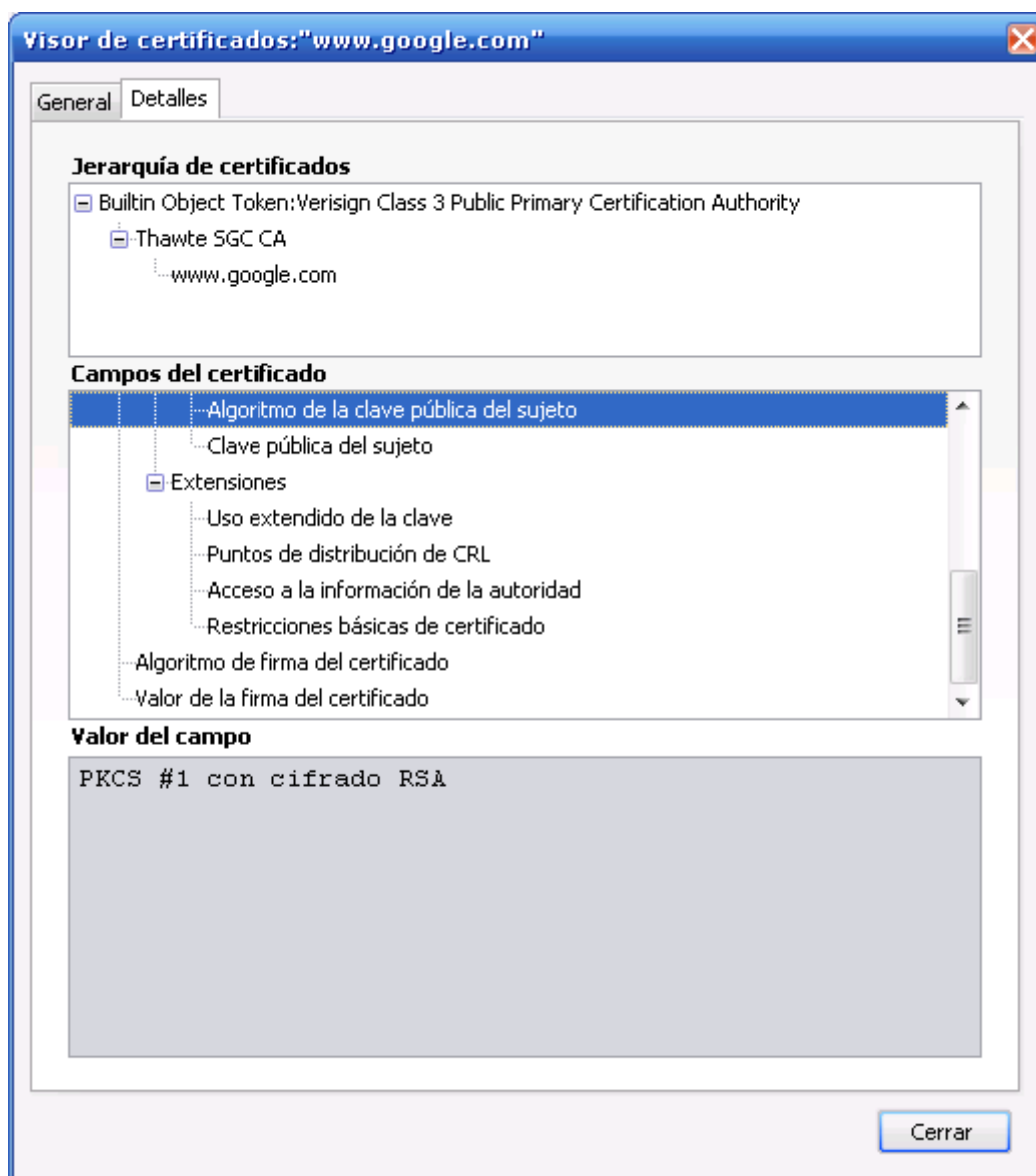


Figura 5.15: "Información sobre certificado digital de GMAIL"

Dentro del documento del estándar PKCS #1, podemos encontrar la descripción de los tipos de claves que ocupa, es decir la pública y la privada, además de las primitivas criptográficas para el cifrado y descifrado, descripción de los algoritmos hash a utilizar en firmas digitales, tanto en operaciones de generación de la firma, como también en la verificación de la misma, lo que podemos ver en la tabla 5.9.

Tabla 5.9: “Algoritmos hash usados con RSA” – Fuente RSA Laboratories

Algoritmo Hash	IDENTIFICADOR
MD2	Md2WithRSAEncryption ::= {pkcs-1 2}
MD5	Md5WithRSAEncryption ::= {pkcs-1 4}
SHA-1	Sha1WithRSAEncryption ::= {pkcs-1 5}
SHA-256	Sha256WithRSAEncryption ::= {pkcs-1 11}
SHA-384	Sha384WithRSAEncryption ::= {pkcs-1 12}
SHA-512	Sha512WithRSAEncryption ::= {pkcs-1 13}

Se puede ver que existen varios algoritmos hash para utilizar según el estándar, sin embargo, existen algunos que se usan en menor proporción ya que a medida que avanza el tiempo, se han encontrado diversas vulnerabilidades que hacen posible la colisión de estos algoritmos, como es el caso de MD2 y MD5. Últimamente, se está recomendando la utilización de SHA-1 como algoritmo hash, ya que por el momento cumple con las necesidades requeridas por los sistemas comerciales. La utilización de SHA-256, SHA-384 o SHA-512 es en menor proporción, ya que si bien, hacen los sistemas más seguros, todo depende del tipo de información que se maneja, porque también significa mayor cantidad de operaciones, lo que hace bajar el rendimiento de las máquinas.

Respecto a publicaciones de RSA Laboratories en relación a criptografía de curva elíptica, podemos ver muchos artículos en la página de RSA Laboratories. Principalmente explican en que consiste la criptografía de curva elíptica, que tan seguro es, en donde es útil su aplicación.

En el artículo “Overview of Elliptic Curve Cryptosystems”, cuya última revisión fue en el año 1997, se defiende mucho al algoritmo RSA. Si bien, habla de algunas ventajas de la criptografía de curva elíptica, también recalca algunos aspectos en que RSA es superior. En la tabla 5.10 se puede apreciar uno de ellos:

Tabla 5.10: “Requerimientos de sistema para ECC y RSA” – Fuente RSA Laboratories

	ECDSA y ECIES en grupos finitos primos	RSA 1024 bits y clave pública $e = 2^{16} + 1$
Parámetros de sistema	$(4 * 160) + 1 = 641$	0
Clave pública	$160 + 1 = 161$	$1024 + 17 = 1041$
Clave privada	160 (801 con parámetros de sistema)	2048 (o 2560 con información CRT)

Este estudio informa que para usar criptografía de curva elíptica se necesitan varios parámetros iniciales, pero sin embargo, pueden usarse por múltiples usuarios. En su contraparte, RSA no requiere de parámetros de sistema, ya que su primera etapa consiste en el cómputo de la clave pública y privada, pero aunque estas operaciones sean intensas computacionalmente, no lo es tanto en comparación al cómputo de los parámetros de sistema para algoritmos basados en curva elíptica.

Otro punto que defiende mucho RSA Laboratories, es el hecho que para el proceso de cifrado de datos, RSA es mejorable en cuanto a unidades de tiempo en que se demora, esto es porque la clave pública para cifrar, puede ser un número significativamente más pequeño que los usados por algoritmos basados en criptografía de curva elíptica. En la tabla 5.11 podemos ver una comparación para estos procesos:

Tabla 5.11: “Comparación entre ECC y RSA en unidades de tiempo para procesos criptográficos” – Fuente RSA Laboratories

	ECDSA y ECIES en grupos finitos primos	RSA 1024 bits y clave pública $e = 2^{16} + 1$	Sistemas basados en el problema de logaritmo discreto con 1024 bits
Cifrado	120	17	480
Descifrado	60	384	240
Firma	60	384	240
Verificación	120	17	480

El proceso de cifrado para RSA es más rápido que ECC, ya que la clave pública es la mínima aceptada, sin embargo, el proceso de descifrado aumenta considerablemente. Los otros sistemas basados en el problema de logaritmo discreto, ocupan mucho más tiempo que RSA y ECC, es por esto que no se ocupan mucho comercialmente, a excepción de DSA, que se ocupa para firmar documentos.

En el estándar en desarrollo PKCS #13, RSA Laboratories buscará integrarse a la nueva generación, debido a las necesidades actuales que se han ido generando.

5.6 Otros Estudios

Existen otro tipo de estudios que ya no están relacionadas a organizaciones, sino a personas mismas, como es el caso de la tesis doctoral de José Luis Salazar Riaño, titulada “Ampliación del espacio de mensajes en criptosistemas de curvas elípticas mediante técnicas de isomorfía y entrelazado” realizada en la Universidad de Zaragoza, España, en donde muestra muchos aspectos teóricos de la criptografía de curva elíptica y las operaciones que realiza.

En la tabla 5.12 se muestra la complejidad computacional de operaciones básicas que se realizan en criptografía de curva elíptica:

Tabla 5.12: “Comparación de complejidad computacional en ECC para operaciones básicas” – Fuente José Luis Salazar Riaño

Complejidad computacional	Fp	F2m
suma y resta	$O(\log(p))$	$O(m)$
Productos	$O(\log^2(p))$	$O(m)$
Cálculo de inversos	$O(\log^2(p))$	$O(m^2)$
Cálculo de raíces cuadradas	$O(\log^4(p))$	---

Estas operaciones se realizan tanto en el cifrado como en el descifrado, y podemos notar que las operaciones más simples son las sumas y restas, que vistas a nivel de bits, significan unas cuantas operaciones XOR e intercambios de bits.

Otra comparación interesante es el número de operaciones básicas realizada en la aritmética de puntos de curvas elípticas, lo que podemos ver en la tabla 5.13.

Tabla 5.13: “Comparación de número de operaciones básicas en ECC” – Fuente José Luis Salazar Riaño

	F_p				F_2^m		
	Sumas	Productos	Inversos	Raíces	Sumas	Productos	Inversos
Exponenciación	0	$3(\log p)/2$	0	0	0	$3m/2$	0
Sumar puntos	10	3	1	0	7	2	1
Doblar puntos	16	4	1	0	4	3	1
Producto de número por punto	$21(\log p)$	$11(\log p)/2$	$3(\log p)/2$	0	$15m/2$	$4m$	$3m/2$
Asignar mensaje a punto	3	5	0	1	3	2	1
Asignar punto a mensaje	1	1	0	0	2	0	0

Si bien, el número de operaciones no es fijo, es una buena aproximación para ver el comportamiento interno de las operaciones de cifrado y descifrado. En este caso, se utilizó la esperanza matemática de la cantidad de unos que tenga en base binaria el número por el que se multiplique el punto, para hacer las mediciones.

El resto del estudio se basa en comparaciones utilizando curvas elípticas y el cifrado de ElGamal, lo que nos hace salir de nuestro alcance.

Otro estudio lo ha hecho Microsoft Corporation, con el fin de ver las ventajas que se obtienen en el ambiente wireless al ocupar ECC en contraste con RSA.

Se realizaron mediciones de los tiempos de exponenciación de curvas elípticas en grupos de primos con diferentes procesadores. El resultado se refleja en la tabla 5.14.

Tabla 5.14: “Tiempos de exponenciación de curvas elípticas” – Fuente Microsoft

Procesador	MHz	163-bit	192-bit	256-bit	384-bit	521-bit
Ultra Sparc II	450	6,1	8,7	-	-	-
StrongARM	200	22,9	37,7	-	-	-
Pentium II	400	-	18,3	42,4	136,4	310,4
Pentium II	400	-	2,1	5,1	16,4	27,8

En la primera y segunda medición no se realizó ningún tipo de optimización, y se trabajó con curvas cuya seguridad se compara a la actual utilizada por RSA. En la tercera y cuarta medición se usaron rutinas especializadas para la reducción modular para los primos propuestos por el NIST. Todo esto es comparable con lo que aparece en la tabla 5.15, que muestra procesos similares para el algoritmo RSA:

Tabla 5.15: “Medición de tiempos de procesos de cifrado y descifrado en RSA” –

Fuente Microsoft

Procesador	MHz	1024-RSA_d	1024-RSA_e	2048-RSA_d	2048-RSA_e
Ultra Sparc II	450	32,1	1,7	205,5	6,1
StrongARM	200	188,7	10,8	1273,8	39,1
ARM7TDMI	1	12070	1180	-	-

En la tabla 5.15 se muestra un muestreo de tiempos de los procesos de cifrado y descifrado para RSA en milisegundos. Podemos notar que la “d” bajo el tamaño de clave corresponde a la clave privada, es decir representa al proceso de descifrado, y “e” representa a la clave pública, lo que corresponde al proceso de cifrado.

En el proceso de cifrado, RSA parece ser un levemente mas rápido que ECC, sin embargo, el proceso de descifrado alcanza una distancia enorme entre ambos, ya que el tamaño de clave privada es mucho mas grande.

VI. ESTÁNDARES

Existen muchos aportes de estándares para el uso de criptografía de curva elíptica. En la tabla 6.1 podemos ver varios de estos:

Tabla 6.1: “Algunos estándares de ECC” – Fuente Certicom

Grupo	Estándar
IEEE	P1363-2000 · P1363a · P1363.2
CEN	TC331 WG3 (DPM)
NESSIE	ECDSA · PSEC
SECG	SEC1 · SEC2
ANSI X9F	X9.24 Key management · X9.37 Check Image Exchange · X9.57 Cert management · X9.59 Payment · X9.62 ECDSA · X9.63 Key establishment · X9.68 Compressed certificates · X9.73 CMS · X9.84 Biometrics · X9.90 IRD · X9.92 ECPVS · X9.95 Time stamps · X9.96 XML CMS
FIPS	FIPS 186-2 Signatures (ECDSA) · SP 800-56 Key establishment · SP 800-57 Key management
FAA Security	Next Generation ATN · Secure ACARS
ISO	14888 · 15946 · 9796 · 18033 ...
IETF	PKIX · SMIME · IPSec (IKE) · TLS
CE 1394	Consumer Electronics DTCP
OMA	WTLS · WPKI · WMLScript ...

VII. BENCHMARK USANDO HERRAMIENTA DE CIFRADO

Para probar la herramienta desarrollada, se realizaron pruebas para comprobar la eficacia del algoritmo de curva elíptica en comparación al algoritmo de RSA, para eso se realizaron cinco pruebas de cifrado y descifrado de la palabra “ABCDEFGHJKLMNOP”.

7.1 Cifrado y descifrado del algoritmo de curvas elípticas

Para la prueba de Cifrado y Descifrado del Algoritmo de Curva Elíptica se usaron los siguientes valores:

163 Bits	Privada de A.	5846006549323611672814742442876390689256843201584
	Privada de B.	5846006549323611672814742442876390689256843201583
233 Bits	Privada de A.	69017463467905637874347558622770255558398127373450135553 79383634485460
	Privada de B.	69017463467905637874347558622770255558398127373450135553 79383634485459
283 Bits	Privada de A.	77706755689029162836778476272940756265696259243769048891 09196526770044277787378692868
	Privada de B.	77706755689029162836778476272940756265696259243769048891 09196526770044277787378692867
409 Bits	Privada de A.	66105596879024859895191530803277103982840468296428121928 46487983041577748273748052081437237621791109659798672883 66567526768
	Privada de B.	66105596879024859895191530803277103982840468296428121928 46487983041577748273748052081437237621791109659798672883 66567526767
571 Bits	Privada de A.	38645375230172583446953518909319873442989273297064349986 57235251451519142289560424536143999389415773083133881121 92694448624687246281681307023452828830333241139319110528 5700

	Privada de B.	38645375230172583446953518909319873442989273297064349986 57235251451519142289560424536143999389415773083133881121 92694448624687246281681307023452828830333241139319110528 5699
--	------------------	--

Con los valores ya mencionados se obtuvieron los siguientes resultados:

Tamaño de Clave	Tiempo en Pruebas de Cifrado	Tiempo Promedio de Cifrado	Tiempo en Pruebas de Descifrado	Tiempo Promedio de Descifrado
163 Bits	00:00:00,20	00:00:00,19	00:00:00,19	00:00:00,19
	00:00:00,19		00:00:00,19	
	00:00:00,17		00:00:00,19	
	00:00:00,19		00:00:00,19	
	00:00:00,19		00:00:00,19	
233 Bits	00:00:00,44	00:00:00,44	00:00:00,45	00:00:00,49
	00:00:00,42		00:00:00,58	
	00:00:00,42		00:00:00,45	
	00:00:00,48		00:00:00,53	
	00:00:00,42		00:00:00,42	
289 Bits	00:00:00,98	00:00:01,06	00:00:01,11	00:00:01,05
	00:00:01,03		00:00:00,98	
	00:00:00,97		00:00:00,98	
	00:00:01,13		00:00:01,03	
	00:00:01,17		00:00:01,14	
409 Bits	00:00:02,16	00:00:01,95	00:00:02,41	00:00:02,00
	00:00:01,89		00:00:01,95	
	00:00:01,92		00:00:01,89	
	00:00:01,88		00:00:01,88	
	00:00:01,89		00:00:01,89	
571 Bits	00:00:07,81	00:00:07,33	00:00:06,99	00:00:07,14
	00:00:07,56		00:00:07,56	
	00:00:07,98		00:00:07,83	
	00:00:06,64		00:00:06,67	
	00:00:06,66		00:00:06,66	

A partir de los valores obtenidos se puede graficar, como lo muestra las figuras 7.1 y 7.2.

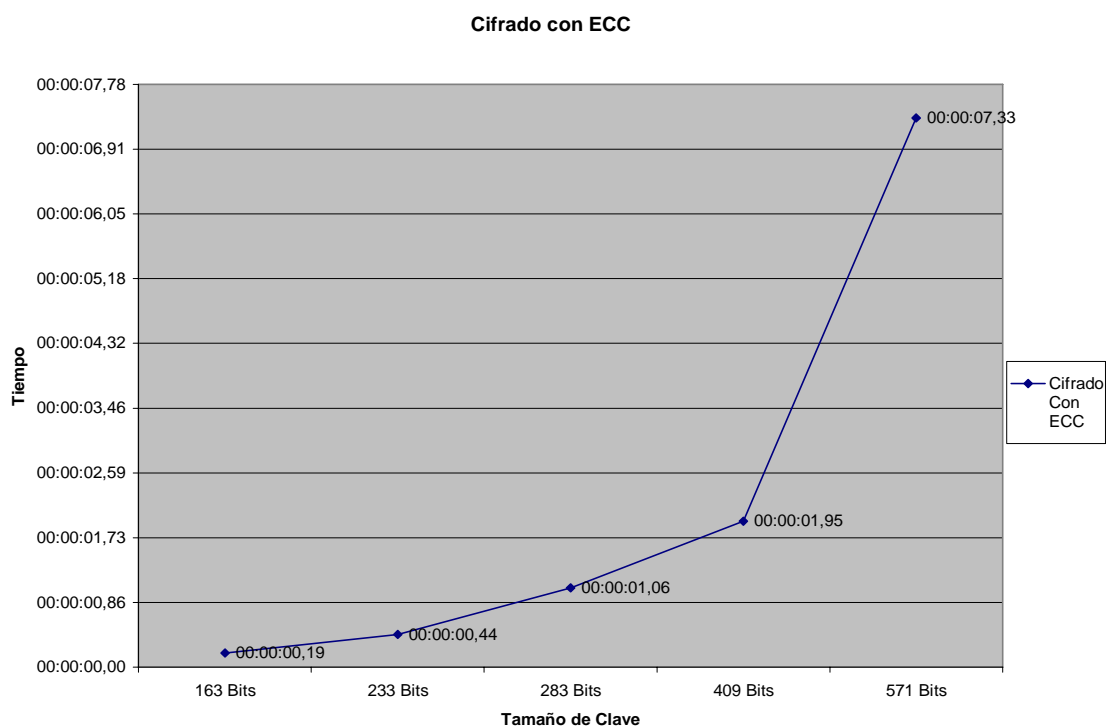


Figura 7.1: “Gráfico de cifrado ECC tiempo versus tamaño de clave”

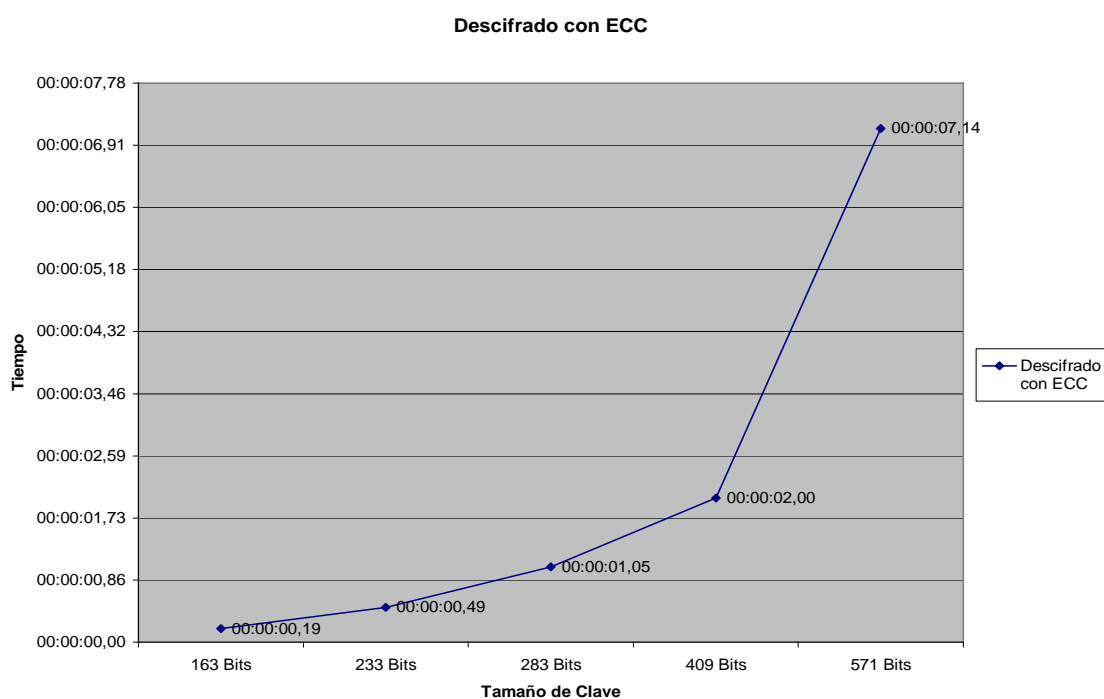


Figura 7.2: “Gráfico de descifrado ECC tiempo versus tamaño de clave”

Como se puede apreciar, claramente en las figuras 7.1 y 7.2, tanto para cifrar como para descifrar el algoritmo de curva elíptica no se demora más de 10 segundos en cada proceso, por lo mismo se asegura la rapidez de cálculos que realiza este tipo de algoritmo.

7.2 Cifrado y descifrado del algoritmo de RSA con clave de descifrado rápida

Para la prueba de Cifrado y Descifrado del Algoritmo de RSA con clave de descifrado rápida se usaron los siguientes valores:

1024 Bits.	Valor de P.	Número de 309 Dígitos.
	Valor de Q.	Número de 309 Dígitos.
	Valor de Clave Privada.	65537
1024 Bits.	Valor de P.	Número de 675 Dígitos.
	Valor de Q.	Número de 675 Dígitos.
	Valor de Clave Privada.	65537
3072 Bits.	Valor de P.	Número de 925 Dígitos.
	Valor de Q.	Número de 925 Dígitos.
	Valor de Clave Privada.	65537
7680 Bits.	Valor de P.	Número de 2312 Dígitos.
	Valor de Q.	Número de 2312 Dígitos.
	Valor de Clave Privada.	65537
15360 Bits.	Valor de P.	Número de 4624 Dígitos.
	Valor de Q.	Número de 4624 Dígitos.
	Valor de Clave Privada.	65537

Con los valores ya mencionados se obtuvieron los siguientes resultados:

Tamaño de Clave	Tiempo en Pruebas de Cifrado	Tiempo Promedio de Cifrado	Tiempo en Pruebas de Descifrado	Tiempo Promedio de Descifrado
1024 Bits	00:00:04,52	00:00:04,37	00:00:00,03	00:00:00,03
	00:00:04,34		00:00:00,03	
	00:00:04,33		00:00:00,03	
	00:00:04,33		00:00:00,03	
	00:00:04,34		00:00:00,03	
2048 Bits	00:00:43,33	00:00:43,23	00:00:00,16	00:00:00,15
	00:00:43,13		00:00:00,14	
	00:00:43,33		00:00:00,16	
	00:00:43,13		00:00:00,16	
	00:00:43,23		00:00:00,14	
3072 Bits	00:01:50,17	00:01:49,94	00:00:00,30	00:00:00,30
	00:01:49,86		00:00:00,30	
	00:01:49,83		00:00:00,28	
	00:01:49,91		00:00:00,30	
	00:01:49,92		00:00:00,30	
7680 Bits	00:32:00,09	00:29:32,01	00:00:02,52	00:00:01,92
	00:32:19,36		00:00:01,77	
	00:27:47,41		00:00:01,77	
	00:27:46,58		00:00:01,77	
	00:27:46,63		00:00:01,75	
15360 Bits	03:38:40,02	03:40:05,55	00:00:06,99	00:00:06,99
	03:39:38,75		00:00:06,98	
	03:41:48,08		00:00:07,02	
	03:41:21,39		00:00:07,03	
	03:38:59,53		00:00:06,95	

Y a partir de los valores obtenidos se puede graficar como lo muestra las figuras 7.3 y 7.4:

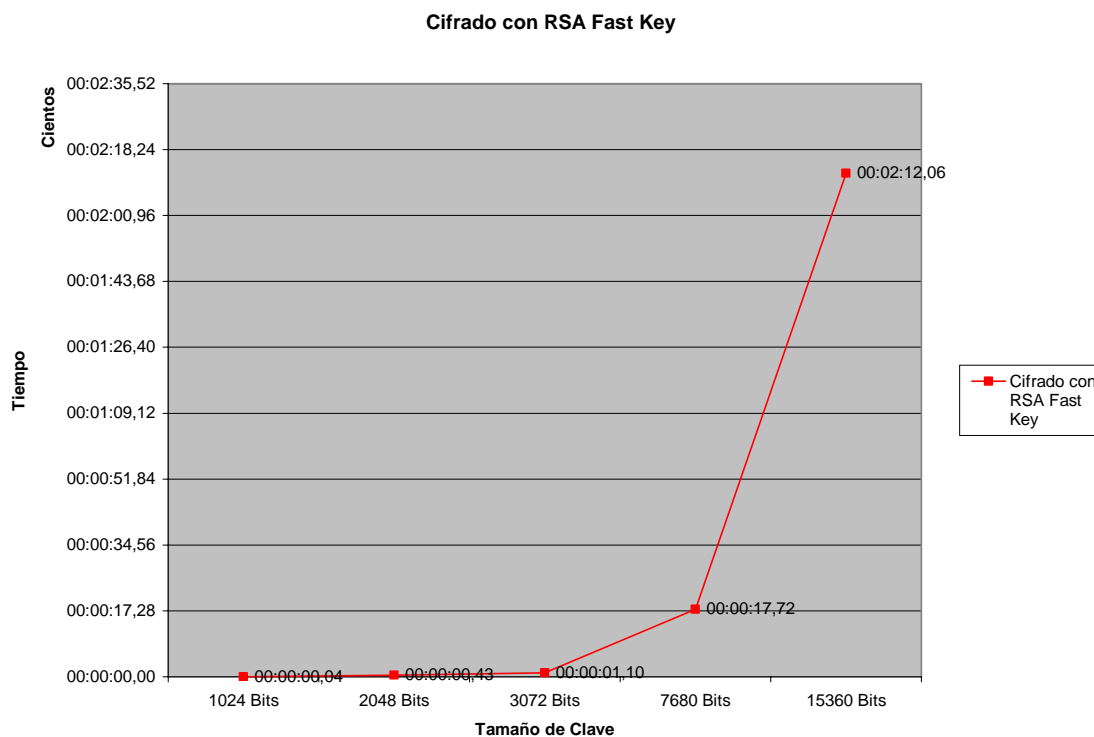


Figura 7.3: “Gráfico de cifrado RSA con clave rápida tiempo versus tamaño de clave”

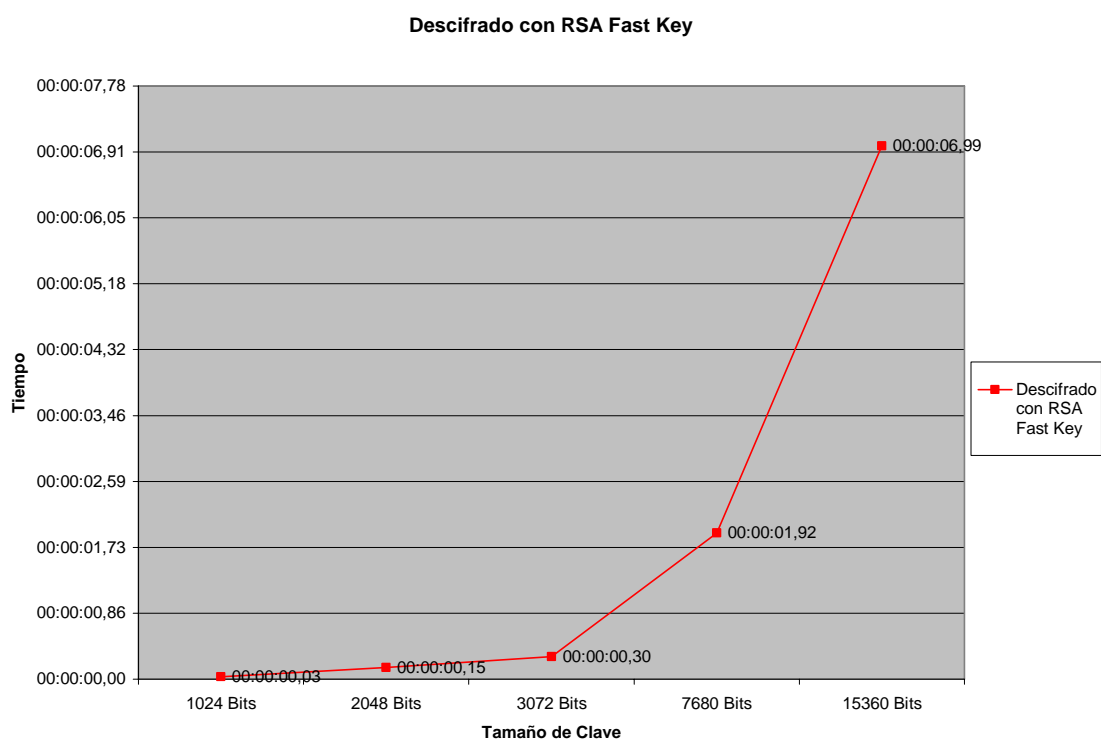


Figura 7.4: “Gráfico de descifrado RSA con clave rápida tiempo versus tamaño de clave”

Como se aprecia en las figuras 7.3 y 7.4, claramente el proceso de cifrado requiere bastante tiempo de procesamiento a diferencia del proceso de descifrado el cual se realiza en cosa de segundos, esto es debido a que la clave privada es de un menor tamaño.

7.3 Cifrado y descifrado del algoritmo de RSA con clave de descifrado lenta

Para la prueba de Cifrado y Descifrado del Algoritmo de RSA con clave de descifrado lenta se usaron los siguientes valores:

1024 Bits.	Valor de P.	Número de 309 Dígitos.
	Valor de Q.	Número de 309 Dígitos.
	Valor de Clave Privada.	Número de 616 Dígitos.
1024 Bits.	Valor de P.	Número de 675 Dígitos.
	Valor de Q.	Número de 675 Dígitos.
	Valor de Clave Privada.	Número de 1349 Dígitos.
3072 Bits.	Valor de P.	Número de 925 Dígitos.
	Valor de Q.	Número de 925 Dígitos.
	Valor de Clave Privada.	Número de 1850 Dígitos.
7680 Bits.	Valor de P.	Número de 2312 Dígitos.
	Valor de Q.	Número de 2312 Dígitos.
	Valor de Clave Privada.	Número de 4624 Dígitos.
15360 Bits.	Valor de P.	Número de 4624 Dígitos.
	Valor de Q.	Número de 4624 Dígitos.
	Valor de Clave Privada.	Número de 9247 Dígitos.

Con los valores ya mencionados se obtuvieron los siguientes resultados:

Tamaño de Clave	Tiempo en Pruebas de Cifrado	Tiempo Promedio de Cifrado	Tiempo en Pruebas de Descifrado	Tiempo Promedio de Descifrado
1024 Bits	00:00:04,31	00:00:04,29	00:00:04,25	00:00:04,25
	00:00:04,28		00:00:04,30	
	00:00:04,31		00:00:04,23	
	00:00:04,27		00:00:04,25	
	00:00:04,27		00:00:04,24	
2048 Bits	00:00:43,13	00:00:42,43	00:00:42,31	00:00:42,30
	00:00:42,17		00:00:42,25	
	00:00:42,34		00:00:42,33	
	00:00:42,23		00:00:42,25	
	00:00:42,30		00:00:42,38	
3072 Bits	00:01:47,72	00:01:47,53	00:01:47,53	00:01:47,49
	00:01:47,45		00:01:47,38	
	00:01:47,61		00:01:47,50	
	00:01:47,48		00:01:47,61	
	00:01:47,38		00:01:47,42	
7680 Bits	00:27:40,83	00:27:33,27	00:27:30,95	00:27:32,20
	00:27:30,52		00:27:31,58	
	00:27:31,02		00:27:31,63	
	00:27:31,83		00:27:35,63	
	00:27:32,17		00:27:31,19	
15360 Bits	03:37:52,78	03:38:13,20	03:36:42,56	03:38:35,97
	03:37:53,94		03:37:51,63	
	03:37:57,39		03:40:18,75	
	03:38:13,34		03:38:14,17	
	03:39:08,55		03:39:52,72	

A partir de los valores obtenidos se ha graficado como se ve en las figuras 7.5 y 7.6, cabe destacar que el gráfico esta en una escala logarítmica para apreciar de mejor forma los valores expuestos en el gráfico.

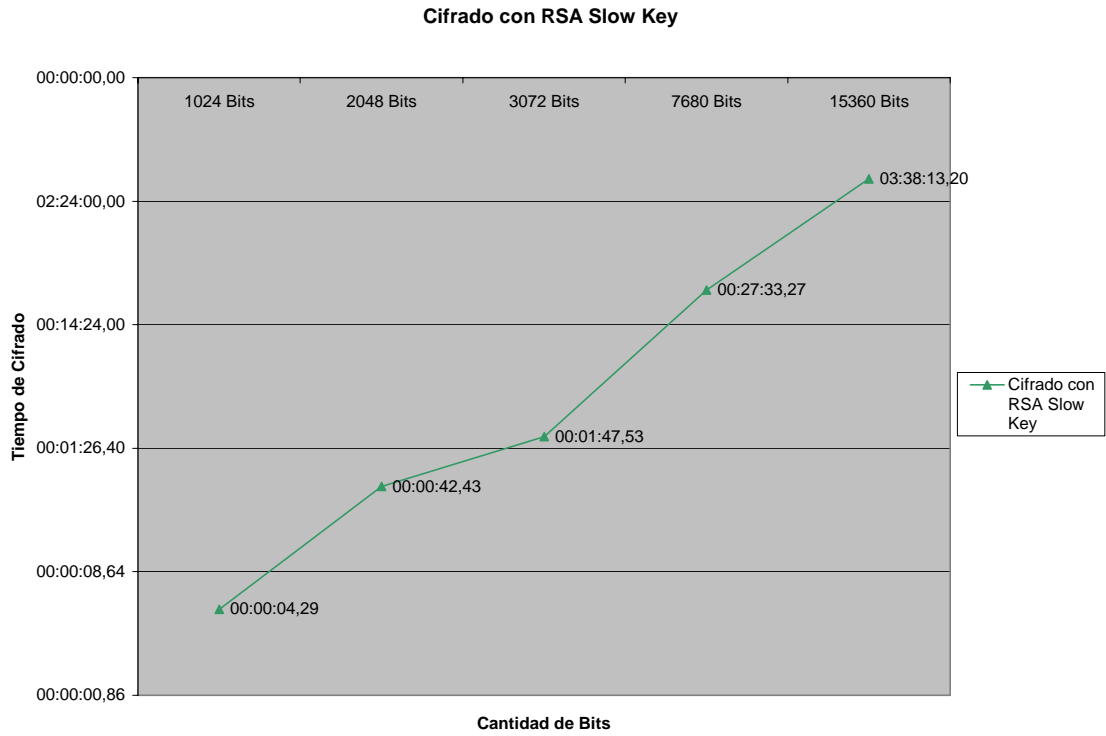


Figura 7.5: “Gráfico de cifrado RSA con clave lenta tiempo versus tamaño de clave”

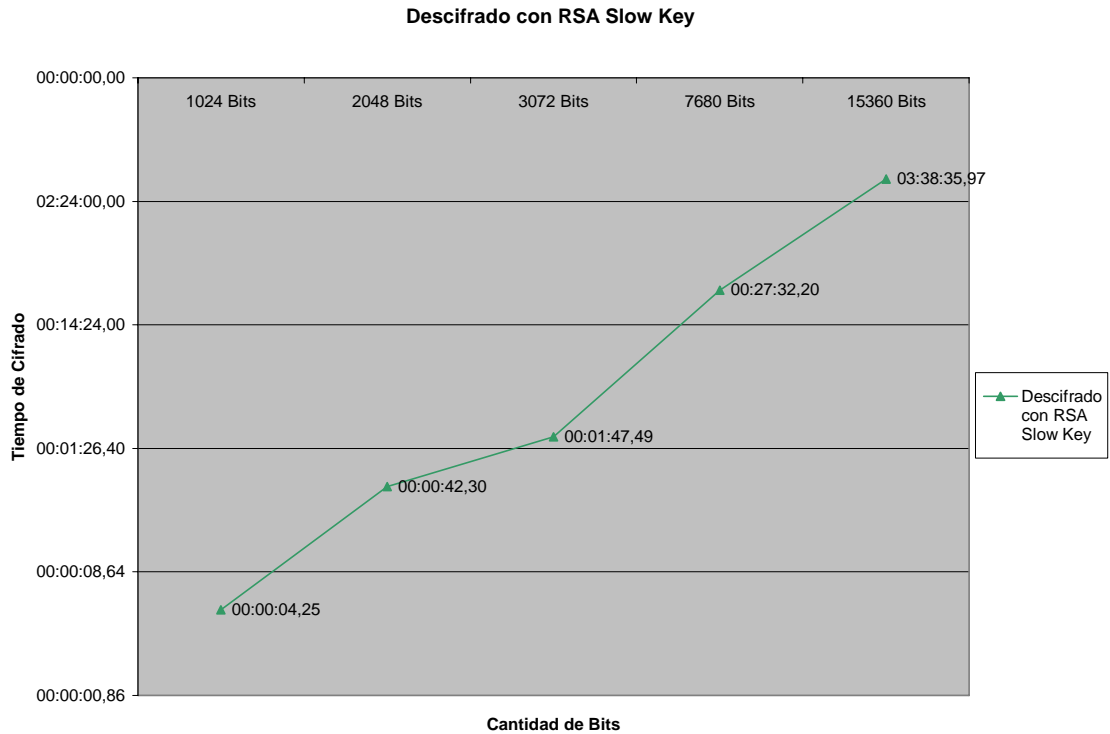


Figura 7.6: “Gráfico de descifrado RSA con clave lenta tiempo versus tamaño de clave”

Como se observa en las figuras 7.5 y 7.6, el proceso de cifrado requiere bastante tiempo de procesamiento y el proceso de descifrado requiere una cantidad de tiempo de procesamiento parecida al proceso de cifrado.

7.4 Curvas elípticas versus RSA

Finalmente se presenta un gráfico comparativo de los 3 tipos de cifrado y descifrado los cuales se pueden apreciar en las figuras 7.7 y 7.8, cabe destacar que para la construcción de este gráfico se usó una escala logarítmica para poder apreciar de mejor forma las diferencias.

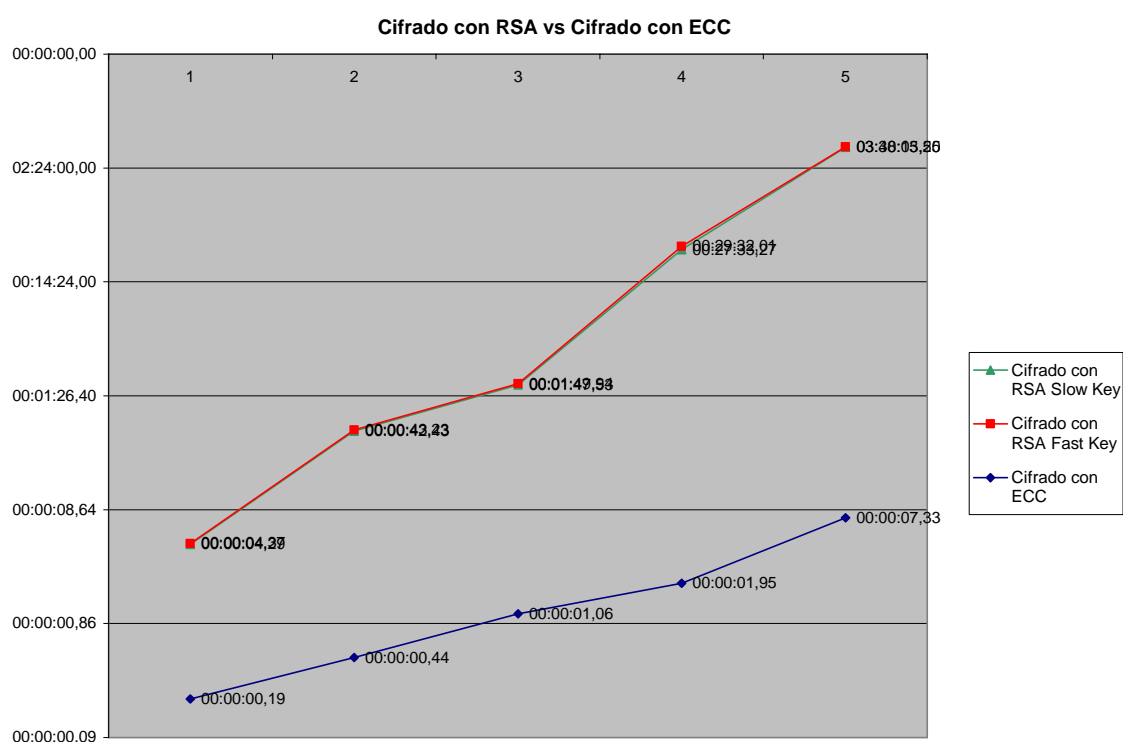


Figura 7.7: “Gráfico comparativo de cifrado RSA versus ECC”

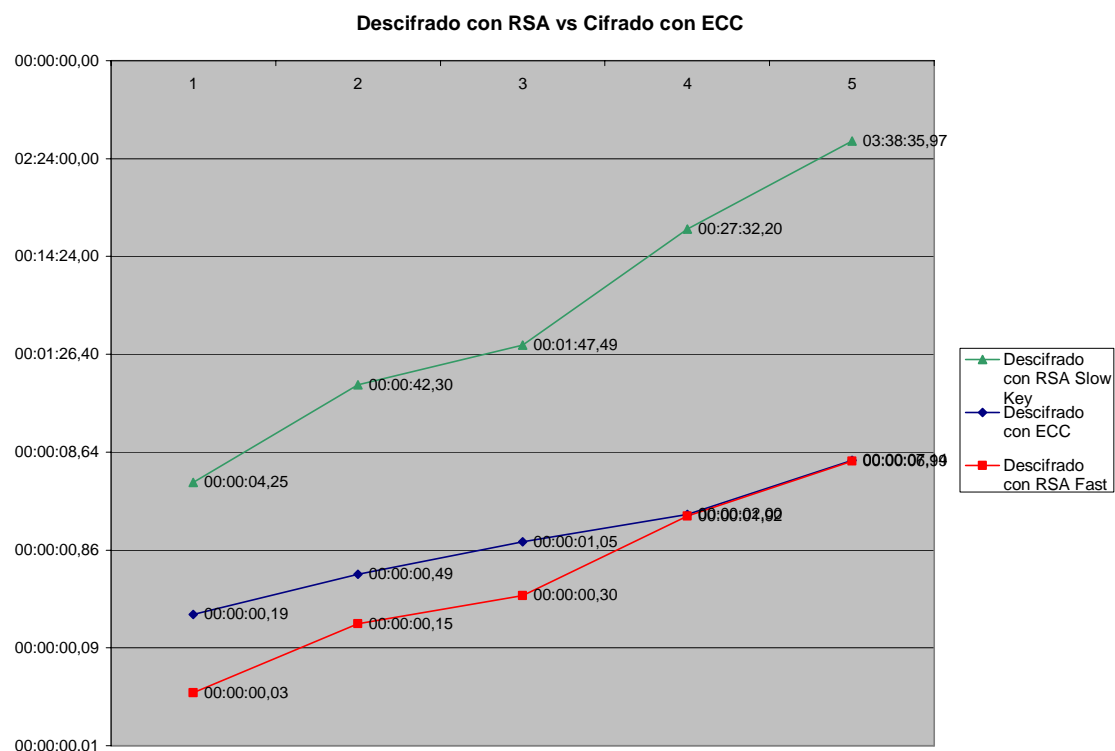


Figura 7.8: “Gráfico comparativo de descifrado RSA versus ECC”

Como se puede apreciar en los gráficos de las figuras 7.7 y 7.8, el proceso de cifrado realizado por el algoritmo de curvas elípticas requiere un menor tiempo para cifrar los mensajes, a diferencia del algoritmo de RSA que requiere un tiempo mucho mayor para realizar el proceso de cifrado.

Para el proceso de descifrado no hay mucha diferencia entre el algoritmo de curva elíptica y el algoritmo de RSA cuando se selecciona una clave privada pequeña, se realiza el proceso de descifrado de una manera rápida y con una clave privada de mayor tamaño el tiempo de descifrado se asemeja mucho al tiempo de cifrado.

VIII. VENTAJAS ECONÓMICAS USANDO CURVAS ELIPTICAS

8.1 Dispositivos móviles

Los dispositivos móviles son aquellos equipos diseñados específicamente para realizar funciones específicas y de forma portable, como por ejemplo, teléfonos celulares, PDAs, palms, entre otros. El uso de este tipo de tecnologías va creciendo día a día convirtiéndose en herramientas fundamentales para las personas y su relación con empresas u organizaciones de diferentes tamaños. Hoy en día se puede revisar el correo electrónico y calendario personal fuera de la oficina en un pequeño teléfono inteligente o salir de viaje de negocios con un dispositivo pocket PC inalámbrico con capacidad GPS que ayude a orientarse en ubicaciones poco conocidas.

Sea cual sea la utilización que se le otorgue a estos dispositivos, cada uno de ellos posee características en relación a la cantidad de memoria que usan o la capacidad de procesamiento de información que aplican para su funcionamiento. Si hablamos de la implementación de procesos de cifrado para potenciar la seguridad que brindan, se puede entrar en un conflicto de valoración de un producto móvil, debido a que aplicar un proceso de cifrado implica una mayor carga de transacciones y por ende se requiere mayor poder de procesamiento por parte del equipo móvil, esto traducido a términos económicos significa dispositivos móviles con memorias y procesadores evidentemente más caros.

Para muchos dispositivos móviles que requieren de algún nivel de seguridad, la implementación de un proceso de cifrado usando RSA se hace casi impracticable, la cantidad de memoria adicional que se requiere y los tiempos de cpu que se deben dedicar al proceso son altísimos. Si se implementa un sistema como este, el valor del dispositivo subiría considerablemente simplemente por el hecho de poseer un procesador más potente y memoria dedicada para este proceso para no interferir con la que memoria que usa para desarrollar su propio desempeño funcional. Cabe recordar que la aplicación de un proceso de cifrado de información dentro de un dispositivo móvil es una función de soporte la cual no debe interferir con las funciones principales de cada dispositivo, no se puede pensar que el proceso de cifrado use una mayor porción de procesamiento de datos que la función primaria de un dispositivo.

Usando curvas elípticas significa que un dispositivo embebido puede:

- Usar un procesador más barato y pequeño o aplicar mayor procesamiento a las funciones primarias del dispositivo; en casos donde las operaciones de seguridad son integradas a un chipset, usando curvas elípticas significará chips menos costosos.
- Aplicar menos ciclos de procesador porque el dispositivo está creando menos calor, por ende menos escape energético, dando como resultado alargar la vida útil de sus baterías.
- Requerir menos ancho de banda para sus transacciones debido a que protocolos mas eficientes.

8.2 Costos asociados de diseño para sistemas embebidos

En esta sección se examinan los factores asociados a los costos que los desarrolladores, fabricantes y vendedores deberían considerar cuando implementan seguridad en sus diseños embebidos.

8.2.1 Poder de procesamiento

En la determinación de las operaciones potenciales de cualquier dispositivo, la selección de chips es una de las consideraciones más importantes: un procesador más caro permite a un dispositivo hacer más, además al usar métodos eficientes se le permite a un procesador más barato realizar las mismas funciones que uno más caro. Es aquí donde las características de la criptografía de curvas elípticas se presentan como un método de eficiencia que permite el ahorro de dinero ante el uso de RSA.

Por ejemplo, una smart card para una transacción financiera se debe autenticar a un lector. Realizando esta transacción usando curvas elípticas requerirá menos procesamiento e incluirá dos posibles beneficios:

- a. la transacción ocurrirá más rápido significando que el sistema puede procesar mayor cantidad de transacciones y generar mayores ingresos.

- b. un procesador más barato (y mas lento) puede ser usado en la smart card para realizar la misma transacción en el mismo tiempo que lo haría un procesador más rápido implementando AES o RSA por ejemplo.

Otros dispositivos móviles como PDAs o smartphones muestran resultados similares. Un procesador típico para este tipo de dispositivos es el ARM SA1110, 206MHZ. Al mismo nivel de seguridad de 128 bit de AES, ECC-256 provee de excelentes tiempos de respuesta, RSA-3072 ofrece buena respuesta solamente en verificaciones de firmas; los tiempos para generar claves y firmas que presenta son inaceptables.

Como conclusión se puede afirmar que para obtener un buen resultado usando RSA se requiere del uso de un procesador más caro que el usado por curvas elípticas para obtener resultados similares.

8.2.2 Compuertas lógicas

Como ya ha sido dicho, la criptografía de curvas elípticas ofrece mejoras en software, sin embargo, también puede ser muy eficiente en hardware. Los beneficios de curvas elípticas pueden incrementar dramáticamente su eficiencia en comparación a RSA en ambientes basados en hardware. Los diseños optimizados basados en chips han demostrado que pueden ser hasta treinta y siete veces más rápidos que el mismo diseño implementado en software.

Esta ventaja se ve reflejada en la cantidad de compuertas lógicas usadas en un diseño electrónico (lo que indica el espacio usado en un chip) y en performance en relación a RSA. Como todo diseñador electrónico sabe, más compuertas lógicas dentro del diseño electrónico significan más dinero.

Como se puede apreciar en la tabla 8.1, la criptografía de curvas elípticas sobresale en el espacio utilizado y en velocidad dentro de un diseño electrónico. Siguiendo las tendencias tecnológicas, cuando se optimiza por espacio la tecnología más nueva es diez veces menor (3260 compuertas lógicas en contra de 34000) y aún así presenta mejores tiempos de respuesta que su competencia RSA. Cuando se optimiza por velocidad, la criptografía de curvas elípticas es siete veces más rápida usando los mismos tamaños de claves y más de ochenta veces más rápido cuando se usan claves que se requerirán a futuro para garantizar la seguridad.

Tabla 8.1: “Tiempo de procesamiento y cantidad de compuertas lógicas usadas en hardware para la implementación de los algoritmos RSA y Curvas Elípticas”

Algoritmo	Optimización	Tiempo	Cant. de compuertas lógicas
RSA – 1024	Optimización de espacio	4.90 ms	34.000
ECC – 163		0.66 ms	3.260
RSA – 1024	Optimización de velocidad	2.60 ms	150.000
ECC – 163		0.35 ms	48.400
RSA – 3072	Optimización de espacio	184 ms	50.000
ECC – 283		29 ms	6.660
RSA – 3072	Optimización de velocidad	110 ms	189.200
ECC – 283		1.3 ms	80.100

Por estas razones, la criptografía de curvas elípticas es una opción clara y obvia al momento de implementar este tipo de seguridad en hardware. Cualquier dispositivo que use RSA requerirá de mayor poder de procesamiento para un microprocesador del que debería usar.

8.2.3 Desgaste de baterías

Como consecuencia de una implementación eficiente de seguridad en hardware según lo planteado en el uso de compuertas lógicas, se requiere de menos ciclos de procesamiento y menos trabajo por parte de los microprocesadores, lo que finalmente se traduce en el uso de menos energía y en una disipación de calor menor. Esto es un punto crítico dentro de los dispositivos móviles, donde el factor limitante para el uso en muchos casos es la duración de la batería.

Como el uso de ECC reduce los factores mencionados, se dice que el uso de ECC es parte de la solución para aumentar la vida de los dispositivos móviles.

8.2.4 Ancho de banda y protocolos

Usando menos ancho de banda puede afectar a los tiempos y capacidad de las transacciones, significando que puede transferir más datos en el mismo tiempo.

8.3 Reducción de gastos

Existen costos que se pueden reducir en diferentes etapas del desarrollo de las nuevas tecnologías, la aplicación de hardware específico optimizado para utilizar la menor cantidad de compuertas lógicas o la implementación de los algoritmos de cifrado por medio de software entre otras opciones. En ambas se ha demostrado que el uso de la criptografía de curvas elípticas presenta los mejores resultados.

Un ejemplo donde la criptografía de curvas elípticas ofrecen beneficios claros es en la utilización de las smart cards. Una de las principales funciones del uso de las smart cards es la criptografía. Son generalmente usadas porque pueden ser usadas para accesos seguros, transacciones financieras, identificación y todas aquellas actividades donde se requiera de seguridad.

Como en RSA los largos de claves se mueven después de los 1024 bits, es requerido un coprocesador matemático para lograr una performance aceptable, lo que trae consigo una mayor complejidad y por ende un valor comercial más elevado. Usando curvas elípticas por otro lado, no se requiere de un coprocesador, a lo más un pequeño set de hardware reducido, pero aún así las transacciones con smart cards se realizan mucho más rápido.

IX. CONCLUSIONES

Se han estudiado diversos tipos de investigaciones realizadas por organizaciones de alto prestigio y cuyas metodologías de trabajo son consideradas de alto nivel, de acuerdo a esto se han sacado conclusiones respecto al futuro de los algoritmos de cifrado RSA y de Curva Elíptica.

Una gran interrogante que existe es: ¿Los métodos basados en curvas elípticas reemplazarán o coexistirán con RSA?

Si bien, existen muchos estudios y estándares que respaldan el uso de algoritmos basados en curva elíptica, no podemos obviar la influencia que tiene RSA como empresa en el mercado de la seguridad de la información.

Si se compara el poder de la empresa RSA respecto a su competidor Certicom en cuanto a su influencia en el mercado, se podría decir que es similar a la competencia que tiene Microsoft y Linux. Como se sabe, las distribuciones Linux tienen muchas ventajas sobre el sistema operativo Windows de Microsoft, sin embargo, existe una amplia diferencia en cuanto al uso de estos sistemas operativos en el mundo, y podemos atribuir esa diferencia, a la solidez y prestigio que ha generado Microsoft en los años que lleva en el mercado.

Como se ha visto, la criptografía de curva elíptica lleva varios años dando vuelta en la mente de investigadores y desarrolladores, pero gran parte de la diferencia respecto a RSA, es que no se ha promovido de manera masiva la idea de cambiar a una mejor tecnología, esto se debe a la traba de pensar que cambiar de tecnología es sinónimo de costos muy elevados para las organizaciones. Otra traba existente es, que al no usarse regularmente en productos, existen pocos intentos de ataque que busquen romper la seguridad de los algoritmos basados en curva elíptica, por lo que es difícil confiar en algo que no ha pasado la prueba de la sociedad por decirlo así, ya que los hackers suelen atacar sistemas con el fin de obtener algún provecho, pero al no ocuparse ECC en sistemas de organizaciones, los intentos para investigar sobre ataques a ECC son escasos.

Volviendo a la interrogante, va a ser un proceso similar al vivido por Windows y Linux pero con algunas diferencias.

Todos los estudios revelan que en algunos años más, los sistemas de las organizaciones estarán sobrecargados al usar RSA, pero aún así, se seguirá usando durante varios años más, hasta que se publique el estándar PKCS #13 que está siendo desarrollado por RSA Laboratories. Al estar listo el estándar, las empresas tendrán un sólido soporte de una organización que lleva años y prestigio en el negocio, por lo que ya existiría la confianza para migrar finalmente de tecnología. En ese momento, empezará la real competencia de Certicom, que será la de ofrecer mejores soluciones que RSA para la protección de todos los medios de las organizaciones.

Se cree que alrededor del año 2010 algunas empresas innovadoras en el ámbito de la tecnología, empezarán a estudiar la factibilidad de migrar de tecnología, por lo que habrá coexistencia de los algoritmos durante varios años, hasta que se consolide el uso de algoritmos basados en curva elíptica.

Otra interrogante de gran importancia es la siguiente: ¿Cuándo es conveniente migrar de tecnología?

Se han analizado algunos aspectos económicos sobre las ventajas que posee la utilización de ECC sobre RSA, sin embargo, pueden existir casos en que la inversión no genere las expectativas de las empresas, o bien, que no convenga en ese momento.

Las grandes empresas necesitan que sus decisiones estén respaldadas por estudios de alto nivel, por lo que atreverse a cambiar de tecnología tempranamente es difícil, aún cuando una organización como la NSA haya decidido usar ECC logrando excelentes resultados.

Se piensa que esta tecnología podría ser aprovechada por las PYMES, ya que el riesgo que estas empresas corren es menor que el riesgo que corren las grandes empresas y a su vez pueden ofrecer un servicio de comercio electrónico con un nivel de seguridad más alto.

Otro aspecto importante al decidir la migración es hacer una evaluación de proyecto al momento de adquirir una máquina que utilice tecnología de cifrado, ya que dependiendo de la cantidad de años que se utilice en el proyecto sería necesario migrar de una tecnología a otra. Por ejemplo, si se compra un servidor que utiliza cifrado RSA, hay que tener en consideración que a medida que pasen los años, el nivel de seguridad

tendrá necesariamente que aumentar, lo que significaría mayor carga para el servidor. Si cambiamos la máquina cada cierto tiempo razonable, el aumentar la seguridad no significaría aumento en el tiempo que se demore cifrar, pero aún así aumentará. Si el servidor que se adquiere, se ocupará durante gran cantidad de años, la necesidad de migrar de tecnología se hará evidente.

Otro aspecto a considerar para decidir de migrar de tecnología es la clasificación que le damos a la información, es decir, pública, secreta, top-secret, etc. Esto se debe a que, dependiendo de la clasificación de la información, es el nivel de seguridad mínimo que deben tener los sistemas que manejan esa información. Por lo general todo esto está ligado a seguridad de tipo gubernamental o militar, pero es considerado por grandes empresas también. Comúnmente se utiliza el nivel de seguridad mínimo para los sistemas, para no disminuir la eficiencia de los procesos, pero cuando hablamos de información militar, se necesitan muchos niveles de seguridad, por lo que al utilizar el algoritmo RSA para proteger información clasificada como top-secret significa una sobrecarga al sistema que reduce la eficiencia de todos los procesos en un porcentaje elevado.

Se piensa que eso fue lo que motivó a la NSA a migrar de tecnología, pero no sin antes realizar los estudios de calidad que les asegure que lo que están adquiriendo realmente cumpla su función en todo aspecto. Un proceso que se prolongó por varios años, pero que actualmente cuenta con toda una gama de productos para poder cumplir sus funciones.

Las organizaciones que quieren seguir ocupando RSA se preguntarán: ¿Existe alguna manera para que no disminuya tanto la capacidad de procesamiento?

Constantemente se ha estado trabajando en dispositivos que aceleren algunas operaciones básicas de cifrado, mediante la implementación de algunos algoritmos específicos, pero la diferencia no es tan significativa. Otra manera es balancear la carga de cifrado y descifrado, es decir, usar claves públicas más grandes, lo que significaría claves privadas mas pequeñas. El problema de esto es estarían transfiriendo la carga de procesamiento a los usuarios del sistema, que en muchas ocasiones poseen computadores de bajo nivel de procesamiento. Si bien, el sistema en sí tendría menos carga de trabajo, las transacciones al final serían más lentas, debido a lo que se demoraría el computador del usuario en cifrar la información, por lo que la calidad de servicio disminuiría notablemente, y esto puede que cause pérdidas mayores al gasto de

máquinas que soporten más procesamiento o bien al proceso de migración de tecnología.

Lo cierto es que la nueva tendencia es la utilización de dispositivos cada vez más pequeños para realizar tareas que faciliten la vida laboral y cotidiana. Esto quiere decir que el acceso a Internet está aumentando de manera exponencial, por lo que garantizar un servicio de calidad y seguro se convirtió en una necesidad clara, y las empresas fabricantes de los dispositivos móviles se encuentra hoy en día constantemente estudiando como mejorar el servicio debido a la gran competencia que existe. Nos referimos a empresas como IBM, Sun Microsystems, Microsoft, Hewlett-Packard o Motorola entre otras.

El cambio de sistema criptográfico es inminente, y si bien tomará algunos en incorporarse de lleno en el mercado, ya se logró un gran paso, que es la aprobación de la NSA para su uso en procesos gubernamentales. Se espera que pronto, otras entidades empiecen a usar la nueva generación de criptografía.

Chile tendrá que esperar a que las sedes primarias de las empresas en países extranjeros adopten el uso de la criptografía de curva elíptica, y luego exijan su uso en las sedes secundarias en el resto del mundo.

A nivel de gobierno, seremos uno de los primeros países en Latinoamérica en migrar de tecnología, ya que actualmente Chile es uno de los países con mayor crecimiento a nivel tecnológico de la región, lo que implica sólidos proyectos innovadores y aprovechar los recursos de manera eficiente.

X. BIBLIOGRAFÍA

- Libro Electrónico de “Seguridad Informática y Criptografía” versión 4.1, Sexta edición 1 de Marzo de 2006. Autor Dr. Jorge Ramió Aguirre Universidad Politécnica de Madrid. Colaboración de Josep María Miret Biosca.
- “Introducción a los criptosistemas de curva elíptica”, Gabriel Belingueres.
- “Elliptic Curve Cryptography, an implementation guide”, Anoop MS.
- “Seguridad en Redes inalámbricas para sistemas multimedia de tiempo real”, Tesis de Luis de Jesús González Noriega, Centro de investigación y de estudios avanzados del Instituto Politécnico Nacional de México.
- “Elliptic Curve Cryptography Tutorial”, Certicom.
- “Ampliación del espacio de mensajes en criptosistemas de curvas elípticas mediante técnicas de isomorfia y entrelazado”, José Luis Salazar Riaño, Universidad de Zaragoza España.
- “ECC Classroom”, Certicom.
- “Standards For Efficient Cryptography, SEC 1: Elliptic Curve Cryptography”, Certicom.
- “IEEE P1363 / D13 - Standards Specifications for Public Key Cryptography”, Institute of Electrical and Electronics Engineers, Inc.
- “ECC in Action” – Certicom.
- “Remarks on the security of Elliptic Curve Cryptosystem” – Certicom.
- “Sun ECC Server Performance” – Certicom.
- “An Elliptic Curve Cryptography (ECC) Primer” – Certicom.
- “Using Elliptic Curve Cryptography (ECC) for Enhanced Embedded Security” – Certicom.
- “NIST Computer Security HandBook Cryptography” – NIST.
- “Federal Agency Use of Public Key Technology for Digital Signaturas and Authentication” – NIST.
- “Recommendation for Key Management – Part 1: General (Revised)” – NIST.

- “Recommendation for Key Management – Part 2: Best Practices for Key Management Organizations” – NIST.
- “Elliptic Curve Cryptosystem for Smart Cards” – Certicom.
- “Federal Information Processing Standards Publication – Digital Signature Standard (DSS)” – NIST.
- “The advantages of Elliptic Curve Cryptography for Wireless Security” – Microsoft.
- “Conexiones IPsec en Múltiples Plataformas” – Alex Flores, Rodrigo Henriquez – Universidad Católica de Temuco.
- “Code & Cipher vol 1” – Certicom.
- “Code & Cipher vol 2” – Certicom.
- “Code & Cipher vol 3” – Certicom.
- “Embedded security in action” – Certicom.
- “FAQ – The NSA ECC License Agreement” – Certicom.
- “Benefits of ECC on Server Performance” – Certicom.
- “Financial Advantages of ECC over RSA or Diffie-Hellman” – Certicom.
- “Computer Security Resource Center – csrc.nist.gov” – NIST.
- “Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs” – Sun Microsystems.
- “Elliptic Curve Cryptography: The Next Generation of Internet Security” – Sun Microsystems.
- “Speeding up Secure Web Transactions using Elliptic Curve Cryptography” – Sun Microsystems.
- “Elliptic Curve Cryptography: How it Works” – Sun Microsystems.
- “Securing the Next Generation Internet” – Sun Microsystems.
- “Integrating Elliptic Curve Cryptography (ECC) into the Web’s Security Infrastructure” – Sun Microsystems.
- “Securing the Web with the Next-Generation Public-Key Cryptosystem” – Sun Microsystems.
- “Sizzle: SSL on Motes” – Sun Microsystems.
- “Enabling the Web with Next Generation Cryptographic Technologies” – Sun Microsystems.

- “The Advantages of Elliptic Curve Cryptography for Wireless Security” – Microsoft Corporation.
- “Public-Key Cryptography Standards (PKCS)” – RSA Laboratories.
- “CryptoBytes Technical Newsletter” – RSA Laboratories.

XI. ANEXOS

ANEXO A – MANUAL DE USO DE LA HERRAMIENTA DE CIFRADO

Manual de Uso

Ventana Principal del programa





La ventana tiene 3 principales opciones de menú las cuales se indican en la siguiente tabla:

Menú	Descripción		
Archivo	Salir	Sale del prototipo.	
Cifrado	Cifrado por RSA	RSA Cliente A.	Ejecuta el cliente A del algoritmo de RSA.
		RSA Cliente B.	Ejecuta el cliente B del algoritmo de RSA.
		RSA Ambos Clientes.	Ejecuta ambos clientes (A y B) del algoritmo de RSA al mismo tiempo.
	Cifrado por ECC	ECC Cliente A.	Ejecuta el cliente A del algoritmo de ECC.
		ECC Cliente B.	Ejecuta el cliente B del algoritmo de ECC.
		ECC Ambos Clientes.	Ejecuta ambos clientes (A y B) del algoritmo de ECC al mismo tiempo.
Ayuda	Información	Muestra la información del prototipo desarrollado.	
	Acerca de	Muestra la información acerca de los desarrolladores de este prototipo.	

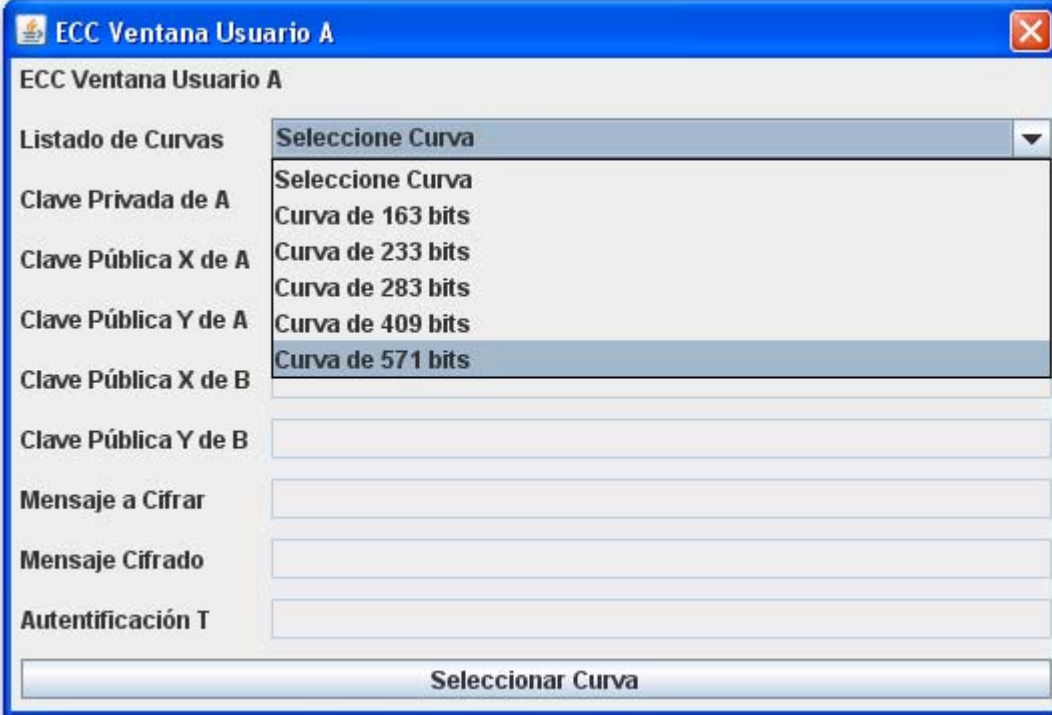
Cientes de algoritmos

Para todos los clientes, antes de partir el programa se solicita indicar la ubicación del otro cliente del algoritmo seleccionado.

Nombre	Descripción	Imagen
Local	Se indica que la dirección IP del otro cliente es una dirección local.	
Remoto	Se indica la dirección IP remota del otro cliente.	

Una vez seleccionado la dirección elegida se abrirá la ventana del algoritmo seleccionado.

Cientes del algoritmo ECC



ECC Ventana Usuario A

Listado de Curvas Seleccione Curva

Clave Privada de A Seleccione Curva

Clave Pública X de A Curva de 163 bits

Clave Pública Y de A Curva de 233 bits

Clave Pública X de B Curva de 283 bits

Clave Pública Y de B Curva de 409 bits

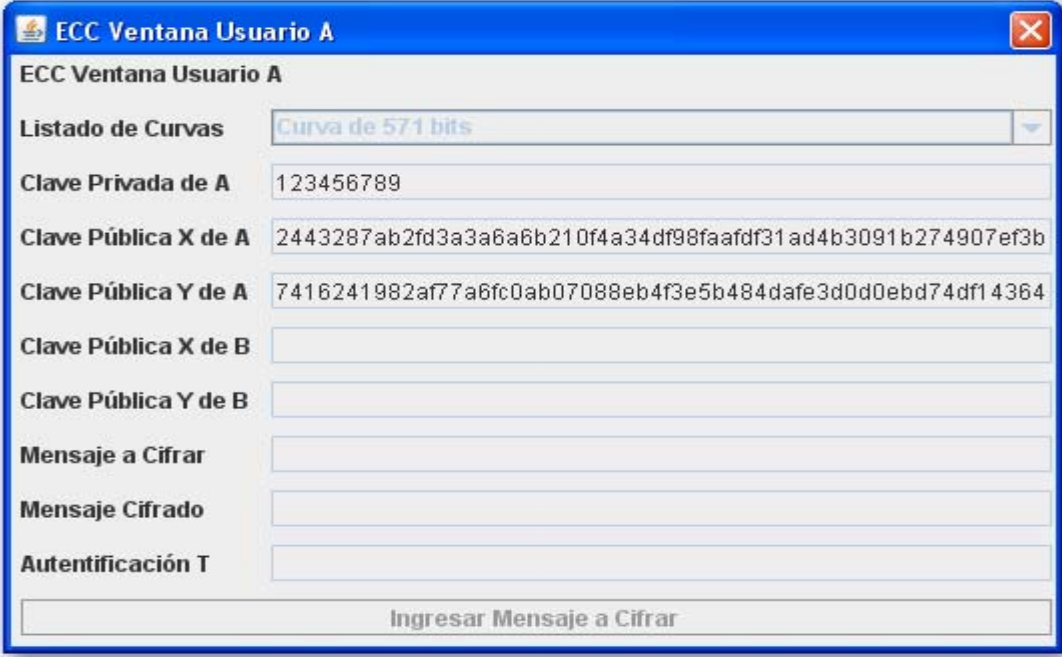
Mensaje a Cifrar

Mensaje Cifrado

Autenticación T

Seleccionar Curva

Esta es la ventana principal del usuario A en la cual lo primero que se debe hacer es seleccionar de la lista la curva con la que se desea trabajar; una vez elegida la curva se debe elegir cuál va a ser su clave privada. Luego de ingresada la clave privada se procede a calcular automáticamente la clave pública de este usuario y se envía al otro usuario como se indican en las siguiente figuras.



ECC Ventana Usuario A

Listado de Curvas Curva de 571 bits

Clave Privada de A 123456789

Clave Pública X de A 2443287ab2fd3a3a6a6b210f4a34df98faafdf31ad4b3091b274907ef3b

Clave Pública Y de A 7416241982af77a6fc0ab07088eb4f3e5b484dfe3d0d0ebd74df14364

Clave Pública X de B

Clave Pública Y de B

Mensaje a Cifrar

Mensaje Cifrado

Autenticación T

Ingresar Mensaje a Cifrar

Una vez recibida la clave pública del usuario A, el usuario B procede a realizar los mismos cálculos que ha realizado anteriormente el usuario A. Cabe destacar que la curva ha sido preseleccionada y solamente se debe aceptar y elegir la clave privada, generar a partir de ella su clave pública y proceder a enviársela al usuario A.

ECC Ventana Usuario B

Listado de Curvas: Curva de 571 bits

Clave Privada de B:

Clave Pública X de B:

Clave Pública Y de B:

Clave Pública X de A: 2443287ab2fd3a3a6a6b210f4a34df98faafdf31ad4b3091b274907ef3b

Clave Pública Y de A: 7416241982af77a6fc0ab07088eb4f3e5b484d4fe3d0d0ebd74df14364

Mensaje Cifrado:

Autenticación T:

Mensaje Descifrado:

Seleccionar Curva

Una vez que el usuario A recibe la clave pública del usuario B, procede a ingresar el mensaje que se desea cifrar como indica en la siguiente figura.

ECC Ventana Usuario A

Listado de Curvas: Curva de 571 bits

Clave Privada de A: 123456789

Clave Pública X de A: 2443287ab2fd3a3a6a6b210f4a34df98faafdf31ad4b3091b274907ef3b

Clave Pública Y de A: 7416241982af77a6fc0ab07088eb4f3e5b484d4fe3d0d0ebd74df14364

Clave Pública X de B: 7fa07f2a93bf35c9a1ffb59345850cbcf75f815ee649efa9440015980629

Clave Pública Y de B: 98ce7cbc4332c0ca193cb64a50127a0e9e48c31f1c744fe5c10476cbfe

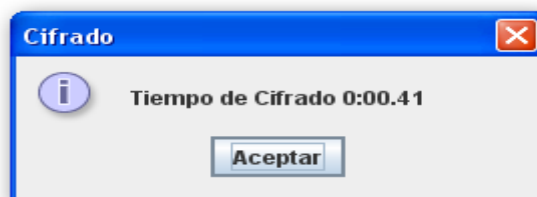
Mensaje a Cifrar: Mensaje Secreto

Mensaje Cifrado:

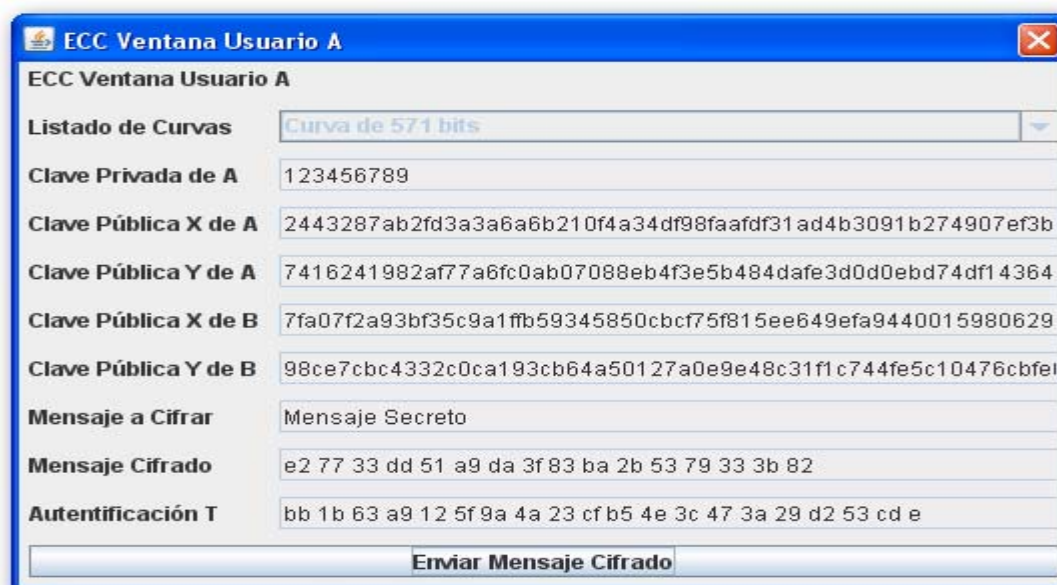
Autenticación T:

Ingresar Mensaje a Cifrar

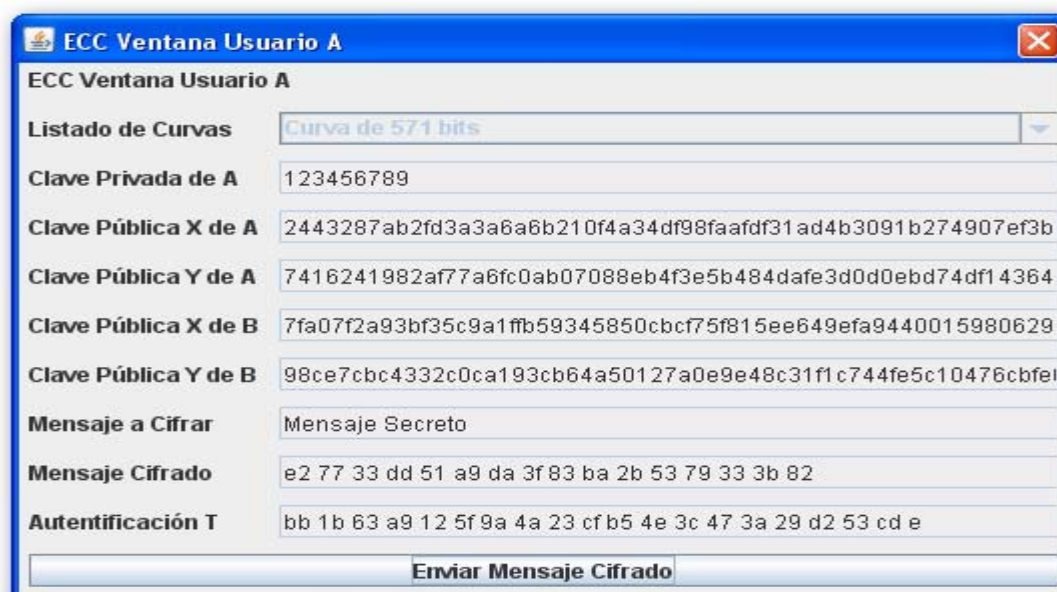
Una vez cifrado el mensaje se muestra un mensaje que indica cuál es el tiempo que se demoró en cifrar el mensaje, esto se ve claramente en la siguiente imagen.



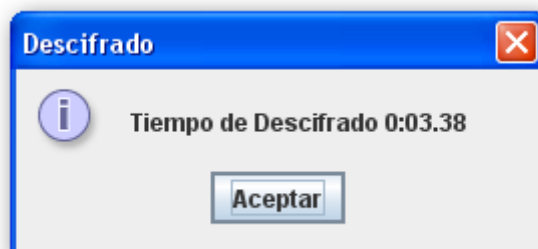
Y una vez hecho esto se procede a enviar el mensaje cifrado al usuario B.



El usuario B recibe el mensaje cifrado de A y procede a descifrarlo como se ve en la siguiente figura.



Y también se indica el tiempo que se demoró el usuario B en descifrar el mensaje cifrado del usuario A y esto se ve en la siguiente figura.



Y de esta manera el usuario B recibe y descifra sin problemas el mensaje obtenido por el usuario A. Esto se ve claramente en la siguiente figura.

ECC Ventana Usuario B	
Listado de Curvas	Curva de 571 bits
Clave Privada de B	987654321
Clave Pública X de B	7fa07f2a93bf35c9a1ffb59345850cbcf75f815ee649efa9440015980629
Clave Pública Y de B	98ce7cbc4332c0ca193cb64a50127a0e9e48c31f1c744fe5c10476cbfe1
Clave Pública X de A	2443287ab2fd3a3a6a6b210f4a34df98faafdf31ad4b3091b274907ef3b
Clave Pública Y de A	7416241982af77a6fc0ab07088eb4f3e5b484dfe3d0d0ebd74df14364
Mensaje Cifrado	e2 77 33 dd 51 a9 da 3f 83 ba 2b 53 79 33 3b 82
Autenticación T	bb 1b 63 a9 12 5f 9a 4a 23 cf b5 4e 3c 47 3a 29 d2 53 cd e
Mensaje Descifrado	Mensaje Secreto
Descifrar Mensaje	

Cientes del algoritmo RSA

RSA Ventana Usuario A

RSA Ventana Usuario A

Listado de Bits: Seleccione Nivel de Seguridad

Valor de P de A: Seleccione Nivel de Seguridad

Valor de Q de A: Nivel de Seguridad de 1024 bits

Valor de N de A: Nivel de Seguridad de 2048 bits

Valor de Phi de N de A: Nivel de Seguridad de 3072 bits

Clave Privada de A: Nivel de Seguridad de 7680 bits

Clave Pública de A: Nivel de Seguridad de 15360 bits

Clave Pública de B:

Valor N de B:

Mensaje a Cifrar:

Mensaje Cifrado:

Seleccionar la Cantidad de Bits para P

Esta es la ventana principal del usuario A en la cual lo primero que se debe hacer es seleccionar de la lista el nivel de seguridad con la cuál se desea trabajar, una vez elegido el nivel de seguridad se calculará el valor primo P. De la misma forma se calculará el primo Q. Una vez hecho esto se calcula los valores de N ($P \cdot Q$) y $\phi(n)$ ($(P-1) \cdot (Q-1)$), después de realizar estos cálculos aparece una pantalla con la que se pide la forma de ingresar la clave privada como se muestra en la siguiente imagen.

Forma de Ingreso de la Clave Privada

☒ Generación Automática

☐ Ingreso Manual

Aceptar

Existen dos formas de ingresar la clave privada:

- **Generación Automática:** genera de manera automática un número válido para ser usado como clave privada.
- **Ingreso Manual:** permite ingresar de manera manual un número que cumpla todas las condiciones para ser clave privada.

Una vez ingresada o generada la clave privada se procede a calcular la clave pública de este usuario y se envía al otro usuario como se indica en la siguiente figura.

The screenshot shows a window titled "RSA Ventana Usuario A" with a blue border and a close button in the top right corner. The window contains the following fields and values:

RSA Ventana Usuario A	
Listado de Bits	Nivel de Seguridad de 3072 bits
Valor de P de A	3681263745275566245779257332363424855516359349096946679
Valor de Q de A	4357087843464488767290536164870378744634800813433647530
Valor de N de A	1603958951312672403354034227556684494555869633357425941
Valor de Phi de N de A	1603958951312672403354034227556684494555869633357425941
Clave Privada de A	
Clave Pública de A	
Clave Pública de B	
Valor N de B	
Mensaje a Cifrar	
Mensaje Cifrado	
Generar Clave Privada	

Una vez recibida la clave pública del usuario A, el usuario B procede a realizar los mismos cálculos que ha realizado anteriormente el usuario A. La única diferencia es que esta vez se ingresará una clave de manera manual.

RSA Ventana Usuario B	
Listado de Bits	Nivel de Seguridad de 3072 bits
Valor de P de B	5505154373608074820534133962988949807231645425116204271
Valor de Q de B	6983482900299486779835263279933196244012741129244050701
Valor de N de B	0204792470726366416251314481186321457990214617796743971
Valor de Phi de N de B	7716155196818804815881917238264175406745828063436489000
Clave Privada de B	65537
Clave Pública de B	
Clave Pública de A	1414668272152561582725184116994032847399609066178031786
Valor N de A	1603958951312672403354034227556684494555869633357425941
Mensaje Cifrado	
Mensaje Descifrado	
Ingresar Clave Privada	

Una vez ingresada se procede a generar la clave pública de B y enviarla al usuario A.

RSA Ventana Usuario B	
Listado de Bits	Nivel de Seguridad de 3072 bits
Valor de P de B	5505154373608074820534133962988949807231645425116204271
Valor de Q de B	6983482900299486779835263279933196244012741129244050701
Valor de N de B	0204792470726366416251314481186321457990214617796743971
Valor de Phi de N de B	7716155196818804815881917238264175406745828063436489000
Clave Privada de B	65537
Clave Pública de B	1254221550167372839911687182632925616257336815573827814
Clave Pública de A	1414668272152561582725184116994032847399609066178031786
Valor N de A	1603958951312672403354034227556684494555869633357425941
Mensaje Cifrado	
Mensaje Descifrado	
Descifrar Mensaje	

Una vez que el usuario A recibe la clave pública del usuario B, procede a ingresar el mensaje que se quiere cifrar como indica en la siguiente figura.

RSA Ventana Usuario A	
Listado de Bits	Nivel de Seguridad de 3072 bits
Valor de P de A	3681263745275566245779257332363424855516359349096946679
Valor de Q de A	4357087843464488767290536164870378744634800813433647530
Valor de N de A	1603958951312672403354034227556684494555869633357425941
Valor de Phi de N de A	1603958951312672403354034227556684494555869633357425941
Clave Privada de A	1402528275497083133663219103372628786441814997605999747
Clave Pública de A	1414668272152561582725184116994032847399609066178031786
Clave Pública de B	1254221550167372839911687182632925616257336815573827814
Valor N de B	1513300030070127470392183715746709982374893364549992700
Mensaje a Cifrar	Mensaje Secreto
Mensaje Cifrado	
Cifrar Mensaje	

Una vez cifrado el mensaje se muestra un mensaje que indica cuál es el tiempo que se demoró en cifrar el mensaje, esto se ve claramente en la siguiente imagen.

Cifrado con RSA

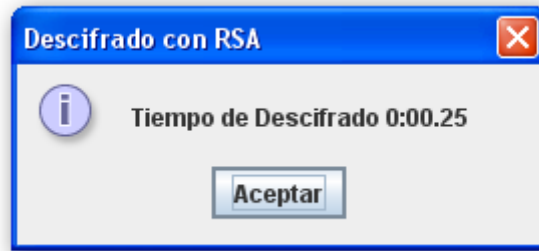
Tiempo de Cifrado 1:38.59

Aceptar

Y una vez hecho esto se procede a enviar el mensaje cifrado al usuario B, el usuario B recibe el mensaje cifrado de A y procede a descifrarlo como se ve en la siguiente figura.

RSA Ventana Usuario B	
Listado de Bits	Nivel de Seguridad de 3072 bits
Valor de P de B	5505154373608074820534133962988949807231645425116204271
Valor de Q de B	6983482900299486779835263279933196244012741129244050701
Valor de N de B	0204792470726366416251314481186321457990214617796743971
Valor de Phi de N de B	7716155196818804815881917238264175406745828063436489000
Clave Privada de B	65537
Clave Pública de B	1254221550167372839911687182632925616257336815573827814
Clave Pública de A	1414668272152561582725184116994032847399609066178031786
Valor N de A	1603958951312672403354034227556684494555869633357425941
Mensaje Cifrado	38ADEDE4674DEF3D15B4C30921900F103139CA7C6AFA3B9FE85E
Mensaje Descifrado	
Descifrar Mensaje	

Y también se indica el tiempo que se demoro el usuario B en descifrar el mensaje cifrado del usuario A y esto se ve en la siguiente figura.



Y de esta manera el usuario B recibe y descifrar sin problemas el mensaje obtenido por el usuario A. y esto se ve claramente en la siguiente figura.

 A window titled "RSA Ventana Usuario B" with a blue title bar. It contains several input fields and a button. The fields are:

- Listado de Bits:** A dropdown menu showing "Nivel de Seguridad de 3072 bits".
- Valor de P de B:** 5505154373608074820534133962988949807231645425116204271
- Valor de Q de B:** 6983482900299486779835263279933196244012741129244050701
- Valor de N de B:** 0204792470726366416251314481186321457990214617796743971
- Valor de Phi de N de B:** 7716155196818804815881917238264175406745828063436489000
- Clave Privada de B:** 65537
- Clave Pública de B:** 1254221550167372839911687182632925616257336815573827814
- Clave Pública de A:** 1414668272152561582725184116994032847399609066178031786
- Valor N de A:** 1603958951312672403354034227556684494555869633357425941
- Mensaje Cifrado:** 38ADEDE4674DEF3D15B4C30921900F103139CA7C6AFA3B9FE85E
- Mensaje Descifrado:** Mensaje Secreto

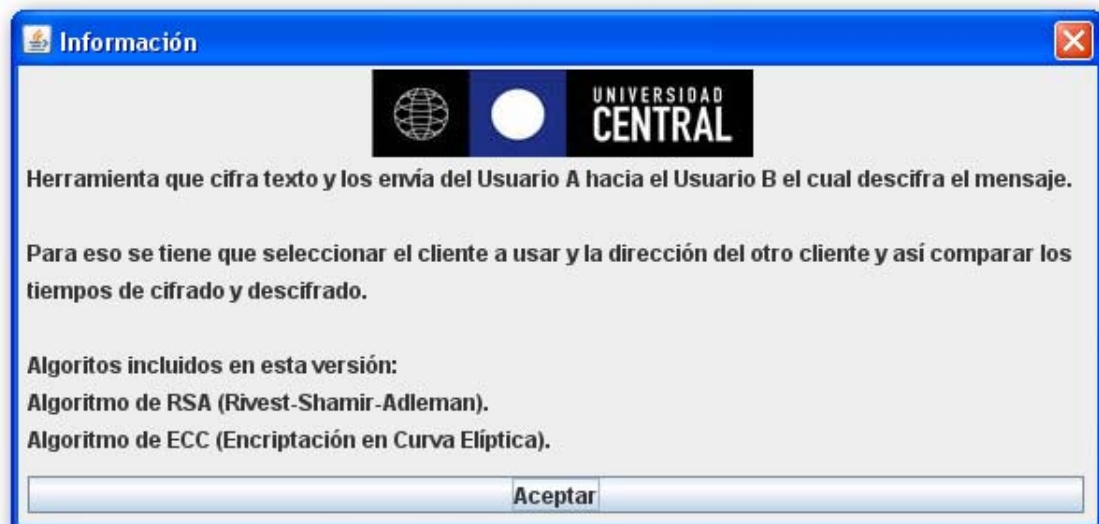
 At the bottom of the window is a button labeled "Descifrar Mensaje".

Ventana Acerca de:



En esta ventana se da a conocer los desarrolladores del prototipo de software de cifrado y de descifrado, además se da a conocer al profesor guía que ayudo a la creación de este software.

Ventana Información:



En esta ventana se da a conocer información de cómo es el funcionamiento de este prototipo de software de cifrado y de descifrado, también indica los algoritmos que incluye esta versión.

Ventana Tabla Comparativa:

Nº	ECC	RSA
0	163	1024
1	233	2048
2	283	3072
3	409	7680
4	571	15360

Aceptar

En esta ventana se da a conocer una tabla comparativa aproximada de ambos tipos de cifrado incluidos en esta herramienta.

ANEXO B – LIBRERIAS USADAS EN LA HERRAMIENTA

Librería de ECC

Jborzoi es una librería de java que implementa la criptografía de curvas elípticas definidas sobre campos binarios finitos de características F_{2^m} , Posee los siguientes métodos para cifrar mensajes:

Método	Descripción
ECKAS-DH1	Esquema de convenio de claves de curva elíptica de Diffie-Hellman.
ECDSA	Algoritmo de firma digital en curva elíptica.
ECIES	Esquema de cifrado y descifrado integrado en curva elíptica.

Como se ha descrito anteriormente, el método elegido para trabajar con curvas elípticas es el algoritmo de ECIES.

Cabe destacar que cualquiera de los tres métodos de esta librería ocupa una serie de curvas aprobadas por NIST (National Institute of Standards and Technology) las cuales están en aprobadas en el documento FIPS-186-2. Las curvas que propone NIST se describen en la siguiente tabla.

Curva NIST	Descripción de la curva
NIST_B_163	Curva Binaria de 163 Bits.
NIST_B_233	Curva Binaria de 233 Bits.
NIST_B_283	Curva Binaria de 283 Bits.
NIST_B_409	Curva Binaria de 409 Bits.
NIST_B_571	Curva Binaria de 571 Bits.

Además esta librería ocupa las siguientes clases para su funcionamiento que se describen en la siguiente tabla:

Nombre de la clase	Descripción
DerIOBuffer	Cifrado/ Descifrado DER.
ECC	Funciones de criptografía de curvas elípticas.
ECDomainParameters	Los parámetros de dominio de las curvas elípticas especifican la curva elíptica usada.
ECDSA	Algoritmo de firma por curvas elípticas tal como es especificado en ANSI X9.62, FIPS 186-2 y en IEEE P1363.
ECIES	Esquema de encriptación integrado de curvas elípticas tal como es especificado en ANSI X9.63 y en IEEE P1363a Draft.
ECPoint	Clase abstracta de un punto de una curva elíptica representando un punto en una curva como dos elementos de un campo finito x e y .
ECPointF2m	Clase de un punto en una curva elíptica representando un punto en la curva como en una curva como dos elementos binarios de un campo finito x e y .
ECPrivKey	Llaves privadas de curvas elípticas consistentes en dos variables: dp y los parámetros de dominio de curvas elípticas con s . La llave privada es la que se debe mantener en secreto.
ECPubKey	Llaves públicas de curvas elípticas consistentes en dos variables: dp y los parámetros de dominio de curvas elípticas con W . La llave pública es un punto en la curva.
ECurve	Curva elíptica $E(F_q) : y^2 + xy = x^3 + a_4x^2 + a_6$.
ECurveF2m	Curva elíptica $E(F_{2^m}) : y^2 + xy = x^3 + a_4x^2 + a_6$.
F2m	Esta es una clase de campos finitos binarios de multi-precisión (F2m).
Fq	Esta es una clase abstracta de campos finitos multi-precisión (Fq).

Cabe destacar que esta librería ocupa el algoritmo simétrico de AES para realizar el cifrado del mensaje.

Librería de RSA

La librería RSA que se usó en esta memoria, es de creación propia, toma como base la librería Jborzoi y usa las librerías estándar de java, en especial la clase java.math de la cual la subclase BigInteger es la más importante para el propósito de este trabajo, su característica principal es suministrar números enteros de precisión arbitraria y poder realizar operaciones matemáticas sin perder la precisión numérica.

Cabe destacar además que la subclase de java BigInteger se puede trabajar con la exponenciación modular de manera directa, es decir no se tuvo la necesidad de implementar el algoritmo de exponenciación rápida, ni el teorema del resto chino.

Además esta librería ocupa las siguientes clases para su funcionamiento que se describen en la siguiente tabla:

Nombre de la clase	Descripción
RSA	Clase principal de la librería la cual se encarga de cifrar y descifrar los mensajes.
RSAGeneratePrimes	Genera de manera aleatoria un número primo con la cantidad de bits indicada.
RSAPrivKey	Clase que crea una clave privada.
RSAPubKey	Clase que crea una clave publica a partir de la clave privada que se ingresa.

ANEXO C – CÓDIGO DE LA LIBRERIA RSA USADA EN JAVA

RSA.java

```
package libreriaRSA;
import java.math.BigInteger;

public class RSA
{
    protected BigInteger[] encriptado;
    public RSA(String mensaje, RSAPubKey E, BigInteger N)
    {
        int i;
        byte[] temp = new byte[1];
        byte[] digitos = mensaje.getBytes();
        BigInteger[] bigdigitos = new BigInteger[digitos.length];

        for(i=0; i<bigdigitos.length;i++){
            temp[0] = digitos[i];
            bigdigitos[i] = new BigInteger(temp);
        }

        BigInteger[] encriptado = new BigInteger[bigdigitos.length];

        for(i=0; i<bigdigitos.length; i++)
            encriptado[i] = bigdigitos[i].modPow(E.givemeE(),N);

        this.encriptado=encriptado;
    }

    public RSA(BigInteger[] cifrado)
    {
        this.encriptado=cifrado;
    }

    public BigInteger[] cifrado()
    {
        return encriptado;
    }

    public String decrypt(RSAPrivKey D, BigInteger N)
    {
        BigInteger[] descriptado = new BigInteger[encriptado.length];

        for(int i=0; i<descriptado.length; i++)
            descriptado[i] = encriptado[i].modPow(D.givemeD(),N);

        char[] charArray = new char[descriptado.length];

        for(int i=0; i<charArray.length; i++)
            charArray[i] = (char) (descriptado[i].intValue());
    }
}
```

```
return(new String(charArray));
    }

    public String toString()
    {
        String salida =new String();
        for(int i=0; i<encriptado.length; i++)
        {
            salida+=encriptado[i].toString(16).toUpperCase();
            if(i != encriptado.length-1) salida+=" ";
        }
        return salida;
    }
}
```

RSAGeneratePrimes.java

```
package libreriaRSA;
import java.math.BigInteger;
import java.util.Random;

public class RSAGeneratePrimes
{
    protected BigInteger primeP, primeQ, N, phiN;

    public RSAGeneratePrimes(BigInteger p,int sizePrime)
    {
        BigInteger q,phi;

        do q = new BigInteger(sizePrime, 10, new Random());
        while(q.compareTo(p)==0);

        phi = p.subtract(BigInteger.valueOf(1));
        phi = phi.multiply(q.subtract(BigInteger.valueOf(1)));

        this.primeP=p;
        this.primeQ=q;
        this.N=p.multiply(q);
        this.phiN=phi;
    }

    public RSAGeneratePrimes(int sizePrimeP)
    {
        BigInteger p;
        p = new BigInteger(sizePrimeP, 10, new Random());
        this.primeP=p;
    }
    public String toString(BigInteger num)
    {
        return num.toString();
    }

    public BigInteger givemeP() {return(primeP);}
    public BigInteger givemeQ() {return(primeQ);}
    public BigInteger givemePhiN() {return(phiN);}
    public BigInteger givemeN() {return(N);}
}
```

RSAPrivKey .java

```
package libreriaRSA;
import java.math.BigInteger;
import java.util.Random;

public class RSAPrivKey
{
    protected BigInteger privkeyD;

    public RSAPrivKey(int sizePrime, BigInteger phiN)
    {
        BigInteger d;
        do d = new BigInteger(2 * sizePrime, new Random());
        while((d.compareTo(phiN) != -1) ||
        (d.gcd(phiN).compareTo(BigInteger.valueOf(1)) != 0));

        this.privkeyD=d;
    }

    public RSAPrivKey(BigInteger d)
    {
        this.privkeyD=d;
    }

    public String toString()
    {
        return privkeyD.toString();
    }

    public BigInteger givemeD() {return(privkeyD);}
}
```

RSAPubKey .java

```
package libreriaRSA;
import java.math.BigInteger;

public class RSAPubKey
{
    protected BigInteger pubkeyE;
    public RSAPubKey(RSAPrivKey d, BigInteger phiN)
    {
        BigInteger e;
        e = d.givemeD().modInverse(phiN);
        this.pubkeyE=e;
    }

    public RSAPubKey(BigInteger e)
    {
        this.pubkeyE=e;
    }

    public String toString()
    {
        return pubkeyE.toString();
    }

    public BigInteger givemeE() {return(pubkeyE);}
}
```

ANEXO D – CÓDIGO DE ENVÍO Y RECEPCION DE DATOS EN JAVA

enviar.java

```
package comun;
import java.io.DataOutputStream;
import java.net.Socket;

public class enviar
{
    public enviar(String word,String ip, int socket)
    {
        Socket sckt = null;
        DataOutputStream data = null;
        String palabra=new String();
        palabra=word;
        try
        {
            sckt = new Socket(ip, socket);
            data = new DataOutputStream(sckt.getOutputStream());
            data.writeBytes(palabra);
            data.close();
            sckt.close();
        }
        catch(Exception e)
        {
            System.err.println("Se ha producido la excepción : " +e);
        }
    }
}
```

usuario.java

```

package User;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

public class usuario
{
    public ServerSocket ss;
    public boolean on=true;
    public usuario (frameUsuario user)
    {
        ServerSocket ss = null;
        Thread hilo1;
        try
        {
            ss = new ServerSocket(35557);
            this.ss=ss;
        }
        catch (IOException ioe)
        {
            System.err.println("Error al abrir el socket de servidor : "
+ ioe);
            System.exit(-1);
        }
        while(on)
        {
            try
            {
                Socket sckt = ss.accept();
                hilo1= new Thread(new servidorUsuario (sckt,
user));
                hilo1.start();
            }
            catch(Exception e)
            {
                System.err.println("Se ha producido la excepción :
" +e);
            }
        }
    }
}

```

servidorUsuario.java

```

package User;
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.net.Socket;
import com.dragongate_technologies.borZoi.ECDomainParameters;
import com.dragongate_technologies.borZoi.ECPubKey;
import com.dragongate_technologies.borZoi.ECPointF2m;
import com.dragongate_technologies.borZoi.F2m;

public class servidorUsuario extends Thread
{
    protected static Socket sux;
    protected static int flag=0;
    protected static String temp1,temp2, temp3;

    public servidorUsuario (Socket SCKT, frameUsuario usuario)
    {
        sux=SCKT;
        this.user=usuario;
    }

    public String leerPalabra(Socket sckt)
    {
        String palabra=new String();
        try
        {
            BufferedReader StreamEntrada = new BufferedReader(new
InputStreamReader(sckt.getInputStream()));
            palabra=StreamEntrada.readLine();
            StreamEntrada.close();
        }
        catch(Exception io)
        {
            }
        return palabra;
    }

    public void run()
    {
        Socket sckt=sux;
        String word=new String();
        try
        {
            word=leerPalabra(sckt);
            sckt.close();
        }
        catch(Exception e)
        {
            }
    }
}

```

