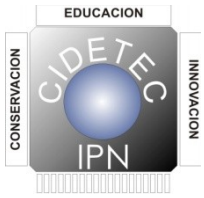



*INSTITUTO POLITÉCNICO NACIONAL*

*CENTRO DE INNOVACIÓN Y DESARROLLO TECNOLÓGICO EN CÓMPUTO*

# *Criptosistema híbrido basado en Triple-DES y ElGamal aplicado en imágenes*

## *Tesis*

*Para obtener el grado de maestría en tecnología de cómputo*

*Presenta:*

*Lic. Sergio Mabel Juárez Vázquez*

*Bajo la dirección de:*

*Dr. Carlos Rentería Márquez*
*Dr. Rolando Flores Carapia*

*noviembre de 2011*

SIP-14-BIS

# INSTITUTO POLITÉCNICO NACIONAL

## SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

*ACTA DE REVISIÓN DE TESIS*

En la Ciudad de México, D.F. siendo las 12:30 horas del día 11 del mes de Octubre del 2011 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del CIDETEC para examinar la tesis titulada:

"Criptosistema híbrido basado en Triple- DES y ElGamal aplicado en imágenes"

Presentada por el alumno:

| Juárez | Vázquez | Sergio Mabel |
|---|---|---|
| Apellido paterno | Apellido materno | Nombre(s) |

Con registro:

| B | 0 | 9 | 1 | 3 | 5 | 8 |
|---|---|---|---|---|---|---|

aspirante de:

Maestría en Tecnología de Cómputo

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

LA COMISIÓN REVISORA
Directores de tesis

DR. ROLANDO FLORES CARAPIA
Primer Vocal

DR. CARLOS RENTERÍA MÁRQUEZ
Segundo Vocal

DRA, HIND TAUD
Presidente

DR. VÍCTOR MANUEL SILVA GARCÍA
Secretario

M. EN C. EDUARDO RODRÍGUEZ ESCOBAR
Tercer Vocal

PRESIDENTE DEL COLEGIO DE PROFESORES

DR. VÍCTOR MANUEL SILVA GARCÍA

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

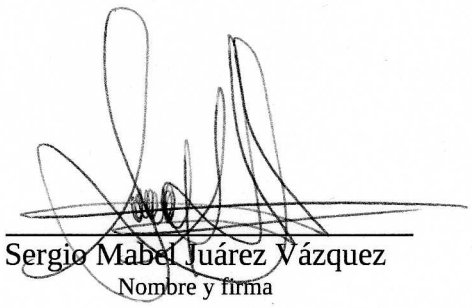
# *INSTITUTO POLITÉCNICO NACIONAL*

***SECRETARÍA DE INVESTIGACIÓN Y POSGRADO***

## *CARTA CESIÓN DE DERECHOS*

En la Ciudad de México D.F., el día 10 del mes noviembre del año 2011, el que suscribe Sergio Mabel Juárez Vázquez alumno del Programa de Maestría en Tecnología de Cómputo con número de registro B091358, adscrito a CIDETEC, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección de Dr. Carlos Rentería Márquez y Dr. Rolando Flores Carapia y cede los derechos del trabajo intitulado "Criptosistema híbrido basado en Triple-DES y ElGamal aplicado en imágenes", al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

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



Sergio Mabel Juárez Vázquez

Nombre y firma

# Criptosistema híbrido basado en Triple-DES y ElGamal aplicado en imágenes

Juárez Vázquez Sergio M.

noviembre de 2011

# Resumen

El objetivo general de este trabajo es aplicar la criptografía a información en forma de imágenes digitales. En términos más particulares la tesis documentará el desarrollo de un programa de cómputo basado en un criptosistema híbrido utilizando Triple-DES para cifrar y descifrar las imágenes y ElGamal para la propagación de las claves logrando una transmisión segura de éstas, es importante mencionar que se incluirán los fundamentos teóricos que sustentan ambos criptosistemas a utilizar y que propondremos una forma innovadora de implementar ElGamal.

# Abstract

The overall objective of this work is to apply cryptography to digital images. In specific terms the thesis will document development of a computer program based on a hybrid cryptosystem using Triple-DES to encrypt and decrypt images and ElGamal will be apply to key propagation finally make a secure transmission, it is important to note that will include the theoretical foundations that support both cryptosystems used and we will make an innovative propouse to implement ElGamal.

# Dedicatoria

**A mi Madre:**

Quisiera saber cómo sabes ser mi madre, quisiera encontrar un teorema con condiciones necesarias y suficientes para agradecerte porque sé tu corazón siempre converge a quererme. Por ahora sólo se me ocurre mejorar a cada instante y que un día alguien más sea quien quiera agradecerte.

**A 091309081510 (a Abril):**

Que cruces sin dolor y en paz nos dejes atrás, te quiero y espero que tu alma vuelva allá a ser una estrella, sólo hasta que nos volvamos a encontrar y te arrulle la vida, cuantas noches tu imagen en sueños me arropó.

A mi Angel **Marlen.**

A mis hermanos: **Konky, Faty, La chata, Giova. Victor y Daniel.**

A mis sobrinos: **Jesús, Axel, Raúl, Yael, Aldair y el Chato.**

A la **China.**

A Las **Matemáticas:** Con tan sólo un poquito que he comprendido de ustedes he sido feliz y llegado a donde estoy, quien fuese el sabio que las domina a perfección.

# Agradecimientos

A los doctores Carlos Rentería Márquez, Rolando Flores Carapia, Victor Manuel Silva García por su interés en este trabajo así como por sus valiosas aportaciones a mi persona.

A la Dra. Flor de María Correa Romero, gracias por sus consejos y la guia que ha sido.

Al Instituto Politécnico Nacional por la oportunidad, la instalaciones y los profesores con los que estudié desde el nivel bachillerato hasta el nivel maestría. Gracias a la ESFM.

Al CONACYT y al IPN por las becas otorgadas, que sin duda fueron una pieza clave para la culminación y desarrollo de mis estudios.

A mi Madre y a mi Familia.

# Planteamiento del problema

En la actualidad el manejo y tráfico de información por medios digitales ha cobrado gran importancia, ya que todos los sistemas modernos de convivencia humana emplean computadoras para ordenar su información, las imágenes son un tipo de información bastante útil, en algunos asuntos imprescindibles. Así como toda la información, las imágenes enfrentan el problema de caer en manos no autorizadas durante el proceso de transmisión.

Implementar un criptosistema híbrido para cifrar imágenes resovería el problema de transmitir imágenes por medios digitales en forma segura. La implementación de el criptosistema ElGamal debe ser cuidadosa en los pasos en donde se generan los elementos primitivos y los números primos grandes a manera que estos dos eslavones no representen debilidades.

# Justificación

Existe software comercial y de licencia GNU que implementa criptosistemas híbridos para el cifrado y transmisión de archivos, sin embargo no existe un programa de cómputo aplicado especificamente a imágenes capaz de cifrarlas de manera eficiente y que además pueda llevar a cabo la propagación de claves que permitan su correcto envio por un una red de computadoras.

Haciendo un análisis de las ventajas y desventajas de los distintos criptosistemas simétricos y asimétricos hemos podido observar que ElGamal y Triple-DES son dos criptosistemas sin patente pero con una alta seguridad, lo que los convierte en dos buenos candidatos para ser incorporados en un criptosistema híbrido y así resolver el problema de encriptar imágenes y hacer una distribución de claves de manera satisfactoria.

Por otra parte analizamos la complejidad de programar ambos algoritmos, y encontramos que la implementación de Triple-DES como algoritmo de cifrado de imágenes no es muy complicada, En el caso de ElGamal la biblioteca GMP nos ayudara a hacer su implementación en el programa de cómputo de manera eficiente y sencilla debido al poder de cálculo aritmetico que esta posee.

Los softwares que ocupan ElGamal casi nunca detallan su implementación así que proponemos una manera original de hacer dicha implementación que reforza la seguridad del criptosistema.

# Objetivos de la tesis

## Objetivos generales

Esta tesis tiene por objetivo diseñar un criptosistema híbrido para cifrar imágenes a manera que su algoritmo pueda ser implementado en un programa de cómputo para hacerlo funcional.

## Objetivos particulares

El objetivo principal es implementar un algoritmo de criptosistema híbrido basado en Triple-DES y ELGamal en un programa de cómputo cuyo código fuente se escriba en lenguaje c++. El programa tendrá que ser capaz de cifrar imágenes en formato de mapa de bits y propagar las claves mediante un cifrado de ElGamal con resultados aceptables comparados con los que ofrece el software existente para propositos similares.

Dentro de los objetivos del trabajo también está el ser didáctico en cada paso de la implementación para aquellos lectores con conocimientos básicos de álgebra abstracta y criptografía, por lo que los fundamentos teóricos se presentan con detalle y se exponen ejemplos representativos de casos sencillos.

Mostrar las ventajas y desventajas del criptosistema tanto teóricas como de desempeño en los programas de cómputo con datos técnicos de tiempo y seguridad.

# Organización de la tesis

La tesis está formada por cinco capítulos. El primero, da una introducción al lector sobre los conceptos básicos del trabajo además de una visión general sobre el mismo.

El capítulo segundo está dedicado a los aspectos teóricos que sustentan el trabajo en su mayoría son conceptos relacionados con teoría de números y teoría de grupos. El apartado en la medida que sea posible pretende ser claro y autosuficiente, aunque varios resultados se tienen que consultar en las referencias bibliográficas de la tesis a trabajos más especializados.

Por otra parte en el capítulo tres se aplica la teoría presentada en el **Capítulo 2**, para diseñar el algoritmo propuesto que da una solución al problema que justifica este trabajo.

El capítulo cuarto muestra los resultados de la implementación del criptosistema en el programa de cómputo, exhibiendo ventajas, desventajas y datos técnicos que nos dan un parámetro de evaluación.

En el último de los capítulos con base a los resultados se exponen algunas conclusiones que el trabajo deja al escritor y las cuales pueden ayudar a reflexiónar al lector sobre el criptosistema elaborado y la implementación del ElGamal propuesta.

# Índice general

# Índice de figuras

# Capítulo 1

# Introducción

## 1.1. Criptografía

No pasó mucho tiempo desde que el ser humano adquirió la habilidad de comunicarse con sus semejantes que tuvo la necesidad de proteger la información y clasificarla así como a las personas que tenía derecho de saberla. Criptografía es una palabra de origen griego compuesta de dos vocablos: *krypto (oculto)* y *graphos (escritura)*, más detalladamente es la ciencia que se encarga del estudio de los métodos para ocultar información escrita cifrandola de diversas formas; siendo los algoritmos matemáticos una de sus principales herramientas para lograr su cometido. La criptografía ha servido para ocultar diferentes formas y contenidos de información, quizá la necesidad por ocultar contenidos de índole militar ha dado los más grandes avances a esta ciencia. Si dos aliados se encuentran distantes y necesitan ponerse de acuerdo sobre como desarrollar una estrategia bélica en contra de su enemigo, es de vital importancia que los mensajes entre ellos se envien por un canal seguro para que su contrario no los intercepte y más importante aún, que si dicho adversario consigue alguno de los mensajes éste no pueda saber la información contenida, es decir que sólo la persona autorizada y a quien va dirigido el mensaje sea capaz de entenderlo.

Como hemos dicho en los parrafos anteriores la criptografía es antiquisima y alrededor de ella existen muchos mitos y anecdotas. Quizá uno de los primeros sistemas de cifrado de los que se tiene registro es el de la scitala, una especie de vara en la que se enrollaba un pergamino y se escribía el mensaje que se quería transmitir, para cuando el escrito era desenrollado, parecía ser un conjunto de

letras desacomodadas y que no formaban un mensaje coherente, la clave para poder entender el mensaje era poseer una scitala del mismo grosor, recibido el pergamino se enrrollaba de nueva cuenta para conseguir el mensaje original, de esta forma los generales espartanos dejaban una scitala en la base y llevaban una de las mismas dimensiones con ellos.

Figura 1.1: Scitala: Sistema criptografico espartano.

Otro de los criptosistemas clásicos y más antiguos es el criptosistema Cesar, nombrado así ya que supuestamente era utilizado por el emperador romano Julio Cesar, éste posee una base más teórica que el anterior, la idea principal del criptosistema es escribir los mensajes en un nuevo alfabeto donde las letras están recorridas tres posiciones módulo el número total de letras en el alfabeto previamente establecido, por el momento la palabra módulo sólo nos dirá que las últimas tres letras en el alfabeto se harán corresponder con las primeras tres.

Ejemplo: Supongamos que queremos escribir un mensaje con el alfabeto español y en este mismo idioma. Escribimos el alfabeto español en forma convencional y bajo éste lo escribimos de nueva cuenta pero con las letras recorridas tres posiciones respecto al orden original del alfabeto, tal y como está escrito en el arreglo siguiente.

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

Ahora supongamos que queremos enviar el mensaje:

"ELEJERCITODELAFRONTERANORTEESTADEBILITADO"

El mensaje escrito en el nuevo alfabeto quedaría como.

"hñhmhuflwrghñdiurpwhudpruwhhvwdghelñlwdgr"

Para volver a tener el mensaje original sólo hay que regresar las letras las tres posiciones que las habiamos recorrido.

Este criptosistema podría modificarse para hacerlo un poco más complicado de vulnerar, recorriendo las letras un número aleatorio de posiciones o permutando las letras del alfabeto más arbitrariamente. Este tipo de criptosistemas se llaman criptosistemas por sustitución pues cada letra del mensaje se sustituye por otra de un alfabeto distinto.

Existen también criptosistemas por transposición, y en este tipo sólo se permutan las letras propias del mensaje sin cambiar los símbolos que lo integran.

Ejemplo: Retomemos el mensaje anterior y esta vez etiquetemos las letras en el mensaje con el número de posición que ocupan, permutemos cada letra en una posición par con la letra inmediata anterio.

El mensaje

"ELEJERCITODELAFRONTERANORTEESTADEBILITADO"

quedaría como

"lejereicoteedalrfnoetarontreetsdabelitidao".

Los criptosistemas antiguos no eran muy complicados en teoría y esto quizá se debía a que los cálculos para cifrar un mensaje eran hechos a mano, así pues se procuraba que los cálculos no fuesen tan complejos, hubo intentos por automatízar estas tareas y tener criptosistemas más complejos pero fáciles de aplicar, uno de estos intentos fue la máquina Enigma[1] creada por los alemanes en la Segunda Guerra Mundial, ésta era una especie de máquina de escribir con algunos cambios para producir mensajes cifrados con un criptosistema simétrico polialfabético por sustitución sin que el emisor se preocupara por los cálculos.

Figura 1.2: Enigma: Máquina de cifrado Nazi.

Los celos sobre secretos y conocimiento en general, han hecho que la criptografía haya evolucionado con la sociedad, hasta ser alcanzada por la era de la computación a principio de los años 70's, la criptografía sigue siendo aplicable a los mensajes que ahora son enviados por canales digitales, a manera que ésta siga procurando:

- Confidencialidad
- Integridad
- Autenticidad

[1]Para ver detalles matemáticos de como funciona el criptosistema montado en Enigma puede consultar [10].

En otras palabras el objetivo primordial de la criptografía es lograr que dos personas se puedan comunicar utilizando un canal al que también pudiera tener acceso un intruso. Plantearemos una situción que sea más representativa. Supongamos que Ivan[2] padre de Alicia, prohibe la relación que ella tiene con Bruno. Alicia quiere comunicarse con Bruno pero por mandato de su padre no puede verlo así que ingeniosamente tendrá que disponer de un medio escrito e intentar encontrar un canal por donde pueda hacer llegar a Bruno los mensajes. Lo ideal sería que Ivan ni siquiera sepa que ellos mantienen comunicanción y en el peor caso si Ivan fuese consciente de la comunicación a Alicia le gustaría que nadie inclusive el propio Ivan pudiese saber el contenido de sus mensajes, pero también desearía que no se pudiera modificar el contenido ni sustituirlo por otro. A lo largo del trabajo usaremos está suposición como punto de partida para explicar algunos conceptos.


Figura 1.3: Alicia quiere comunicarse con Bruno sin que Ivan se entere.

Los canales de transmisión de los mensajes pueden ser muchos: símbolos impresos; cartas escritas en papel, pinturas, grabados, una linea telefónica, sonido, ondas de radio frecuencia, una red de computadoras, etc. La información que Alicia quiere enviar a Bruno y que debe protegerse a su paso por el canal inseguro en la jerga de la criptografía amenudo suele llamarse ***texto plano***, por otra parte el ***cifrado*** es el proceso de convertir con ayuda de una llave o clave y un algoritmo el texto plano en un nuevo mensaje esta vez incomprensible, a este nuevo mensaje se acostumbra llamarle ***texto cifrado***, si se posee la clave adecuada para revertir

[2]Estos personajes usados a lo largo de la tesis fueron creados por el escritor en *www.sp-studio.de* y únicamente son usados con fines académicos lo cual es permitido según el sitio, los personajes originales de la serie South Park® son propiedad de Comedy Central®. Para más información consulte la dirección web.

el proceso se obtiene el mensaje original, y a este último proceso se le denomina ***descifrado***.

Por otra parte las imágenes son una de las principales maneras en que la información puede ser representada y transmitida, de hecho antes de que los seres humanos pudieran desarrollar sistemas escritos o hablados de comunicación eran éstas las que cumplian con dicho objetivo. ¿Quien no ha escuchado aquella frase: "una imagen dice más que mil palabras"?

Las imágenes ahora también son representadas y transmitidas por medios digitales, para esto se necesitó de un modelo matemático el cual facilitara el almacenamiento y la representación de éstas en una computadora, existen varios tipos de estos modelos. Así pues para cifrar una imagen digital se podrían cifrar los números del modelo matemático usado para representarla.

## 1.2. Tipos de Criptosistemas

Un criptosistema es un conjunto de reglas por las que se lleva a cabo un proceso de cifrado y descifrado. Existen distintos tipos de criptosistemas como hemos mencionado la mayoría de ellos basan su algoritmo en mecanismos matemáticos pero éstos pueden clasificarse en tres grupos cuya diferencia entre cada uno radica en los patrones en que se realizan los procesos de cifrado y descifrado, y en las claves que se emplean en dichos procesos, los tres grupos son:

- Criptosistemas Simétricos.
- Criptosistemas Asimétricos o de clave pública.
- Criptosistemas Híbridos.

### Criptosistemas Simétricos

En este tipo de criptosistema las claves con las que se llevan a cabo los procesos de cifrado y descifrado son las mismas o bien las claves se pueden derivar muy fácilmente una de otra. La figura 1.4 muestra la escencia de este tipo de criptosistema.

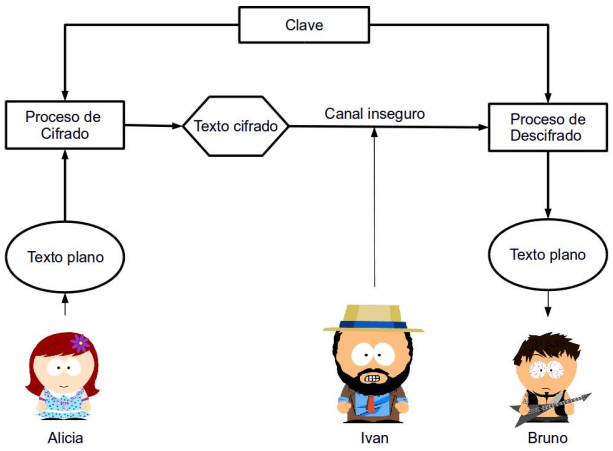



Figura 1.4: Esquema de criptosistema Simétrico

Los criptosistemas de este tipo suelen ser rápidos en sus procesos de cifrado y descifrado pero su principal problema está en que la clave debe mantenerse secreta y estar fuera del alcance de cualquier intruso, el conocimiento de la clave debe ser sólo de las partes que van a cifrar y a descifrar por lo que también se les conoce como criptosistemas de clave secreta, así pues la forma en la que una de las dos partes da a conocer la clave a su contraparte constituye una vulnerabilidad para este tipo de sistemas de cifrado.

### Criptosistemas Asimétricos

Los criptografos estadounidenses Whitfield Diffie (1944- ) y Martin E. Hellman (1945- ) plantearon por primera vez en su artículo titulado *"New Directions in Cryptography"*[15] una solución al incoveniente de distribución de claves que mostraban los criptosistemas simétricos dando así origen a los asimétricos. Regresando al ejemplo de Alicia y Bruno vemos que lo atractivo de este tipo de

criptosistemas es que una vez Bruno genera sus claves cualquier persona puede cifrar mensajes para él pero para descifrar los mensajes se necesita la clave privada que obviamente sólo él conoce. Veamos como es este esquema en la figura 1.5.

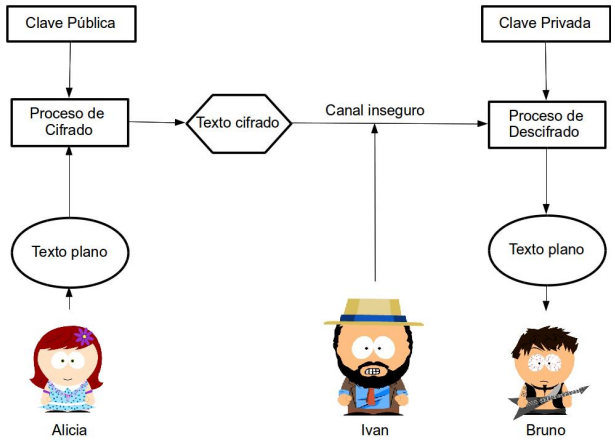



Figura 1.5: smallEsquema de criptosistema Asimétrico

Este tipo de criptosistemas justifican su seguridad en funciones matemáticas que se calculan con una computadora sin complejidad en un sentido pero que son díficiles de resolver en su sentido inverso, hay pocos de estos criptosistemas por que encontrar este tipo de funciones no es nada fácil. Un ejemplo es el siguiente: dados dos números primos grandes hallar su producto con una computadora es rápido pero dado un número grande encontrar quienes son sus factores primos es computacionalmente intratable, esto es en lo que se apoya la seguridad del conocido criptosistema RSA.

Pero no todo es miel sobre ojuelas, estos criptosistemas tienen como desventaja en relación a los simétricos ser más lentos en el proceso de cifrado y descrifrado, generalmente los textos cifrados son más grandes con respecto a los textos planos y las claves también lo son con relación a las claves de los criptosistemas simétricos.

**Criptosistemas Híbridos**

Los criptosistemas híbridos incorporan los dos sistemas anteriores, generalmente se usa el criptosistema simétrico para cifrar el mensaje y después se procede a cifrar la clave de éste con el criptosistema asimétrico para hacer la propagación de la clave más segura.

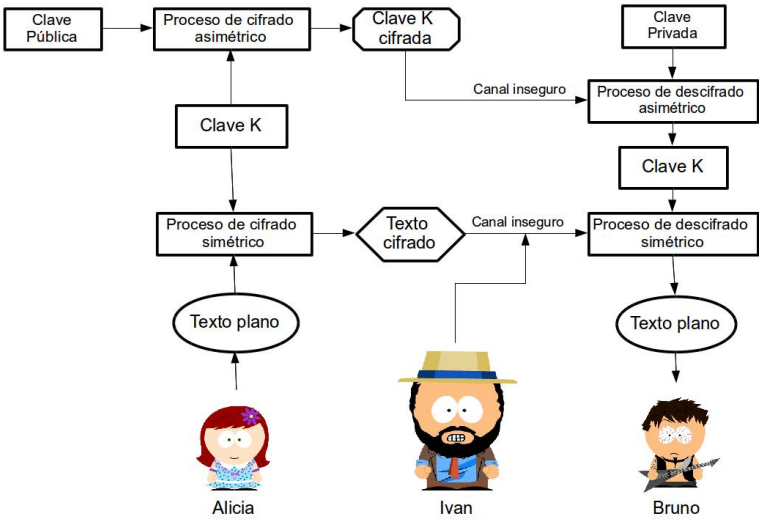



Figura 1.6: Esquema de criptosistema Híbrido

El esquema en la figura 1.6 nos deja observar como aprovechar las ventajas del criptosistema simétrico y asimétrico que componen un criptosistema híbrido.

## 1.3. Criptosistema híbrido para cifrado de imágenes

Ahora supongamos que Alicia quiere enviarle a Bruno por internet una foto de ella estando en la playa por el internet pero Ivan tiene capacidad para poder interceptar todo aquello que Alicia mande o reciba por el modem casero.

Figura 1.7: Foto de Alicia en la playa

Para la forma en que piensa Ivan no le conviene a Alicia que él vea la foto, sin embargo Alicia y Bruno ya saben como construir un criptosistema híbrido para poder realizar la transacción, y lo hacen según el esquema de la figura 1.3.

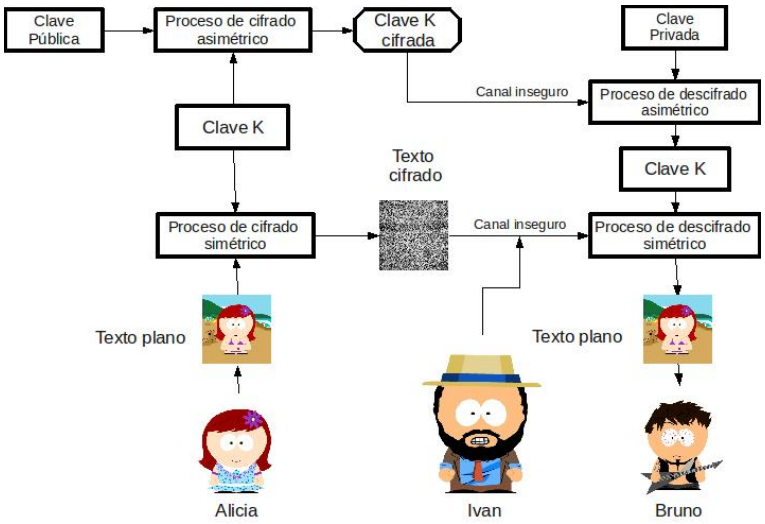



Figura 1.8: Esquema de transmisión de imágenes cifradas

Para que el esquema sea funcional Bruno genera las claves del criptosistema asimétrico y manda a Alicia por cualquier canal incluyendo uno que pudiera ser inseguro la clave pública, luego ella encripta la imagen con un criptosistema simétrico y la clave del criptosistema simétrico digamos k la cifra con la clave pública del criptosistema asimétrico, cuando envia la foto ésta no es más que ruido visual por lo que Ivan no podrá distinguir imagen alguna y junto con la foto

se adjunta la clave k encriptada, después Bruno es el único que puede descifrar la clave secreta k, finalmente con ella descifra la fotografía.

La manera en que Alicia y Bruno pudieron transmitir una imagen por un canal al que tenía acceso un adversario sin que este pudiera identificar dicha imagen es lo que desarrollaremos con formalidad y matemática en lo que resta de este documento.

## 1.4. Estado del arte

Con la era de la computación se vino una nueva era para la criptografía, los criptosistemas podrian ser mucho más robustos ya que las computadoras hacian cálculos complejos en poco tiempo, de esta manera los procesos de cifrado eran más eficientes. Algunos de los criptosistemas modernos más populares son DES,Triple-DES, AES, RSA, ElGamal, Blowfish, CAST5, basados en curvas elípticas[13] y variaciones de estos.

En los años más recientes las comunicaciones digitales y las computadoras han tenido avances a pasos agigantados y la criptografía pretende seguirlos con la misma velocidad, además las computadoras y el internet se han vuelto tan populares y de fácil acceso para la gran mayoría de la población mudial lo que incluye a millones de estudiantes que tienen a la mano una computadora. Las universidades y las empresas privadas dedicadas a esto siguen siendo por razones económicas las que tienen a la mano más poder de cómputo pero es muy diferente a años pasados donde tajantemente eran las únicas con posibilidad de acceder a un ordenador.

La computación cuántica y el procesamiento paralelo están aportando mejoras a los criptosistemas ya inventados, la tarea de crear nuevos criptosistemas y algoritmos de cifrado sigue siendo de los matemáticos y personas dedicadas a las ciencias de cómputo pero con la ayuda de las herramientas que mensionamos, implementar algoritmos complejos y eficientes es cada vez más sencillo.

El procesamiento paralelo se perfila como uno de los mejores aliados para la criptografía pues con su ayuda pretende que los criptosistemas programados sean significativamente más rápidos y eficientes, es materia de trabajo para las personas que se dedican al diseño de estos sistemas poder adecuar los algoritmos originales

de cifrado pensados para correr de forma secuencial en una computadora con una unidad de procesamiento en algoritmos que ocupen en paralelo los varios CPUs o GPUs con que ahora muchas de las computadoras cuentan.

La computación cuántica promete ser para los sistemas digitales en general una revolución que traerá una gran cantidad de beneficios, discrepando de la computación convencional donde la carga es lo que sirve como motor para poder interpretar pulsos eléctricos como unidades escenciales de información (bits), en la cómputacion cuántica se aprovecha lo que se llama el spin del electrón, lo que se refleja como una gran ventaja ya que no sólo se está atado a dos posibles estados como lo hace la carga al ser interpretada como bit, los eléctrones son capaces de estar en varios estados en el mismo instante de tiempo. La criptografía cuántica por su lado es una de las técnicas más prometedoras para los criptosistemás cómputacionales, dando seguiridad informática sustentada con el principio de incertidumbre de Heisenber[26][27].

## 1.5. Criptosistemas híbridos actuales

Los criptosistemas híbridos no tienen un registro histórico como tal pero se puede decir que nacieron casi inmediatamente con los criptosistema de clave pública al estudiar las ventajas y desventajas que éstos exhiben respecto a los simétricos y lo bien que se complementan. Un criptosistema híbrido se forma a paritr de cualesquiera dos criptosistemas con la única condición de que uno sea simétrico y el otro de clave pública. Existen programas de cómputo comerciales, libres y acedémicos que operan criptosistemas de este tipo, como ejemplo están PGP,GPG y un criptosistema basado en CCE (criptosistema de curva elíptica) y mapeos caóticos de caracter académico.

El paquete PGP *(Pretty Good Privacy)* es un asoftware comercial que realiza cifrado de archivos, distribución de claves y firmas digitales, para el cifrado de archivos primero ejecuta un algoritmo de compresión y después sobre el archivo comprimido realiza la encriptación. PGP fue creado por Phil Zimmermann en 1991 y ha venido evolucionando desde entonces, usa como criptosistemas simétricos IDEA y Triple-DES, los asimétricos son reservados. En la actualidad la empresa Symantec ofrece el produco con una licencia de un año para usarse en dos computadoras además de soporte técnico por la cantidad de $2 937 usd.

Por su parte GPG *(GNU Privacy Guard)* es la versión de sofware libre alternativa a PGP, esta vienen incluida en muchos de los sistemas operativos base linux y tienen aplicaciones que son complementos de Firefox y Thunderbird para la seguridad de datos y correos electrónicos en su paso por interner. GPG usa criptosistemas que no están patentados por lo que IDEA no está dentro de su reperterio, en contra parte se apoya de otros como ElGamal, RSA, CAST5, Triple DES, AES y Blowfish, al igual que PGP el cifrado se hace sobre un archivo comprimido basado en el archivo original que se necesita cifrar, también se pueden crear firmas digitales y agregarse un plugin para trabajar con IDEA con un cierto costo.

Kamlesh Gupta y Sanjay Silakari publicaron en el *International Journal of Computer Applications* [28] un criptosistema híbrido basado en mapeos caóticos como criptosistema simétrico y curva elíptica como asiḿetrico, éste está dedicado especificamente a cifrar imágenes y sus tirmpos de cifrado son de alrededor de 12 segundos en sus peores casos.

En cuanto a la encriptación de imágenes no todos los métodos de cifrado se pueden aplicar con exito ya que el cerebro humano es más perspicaz para porder intuir formas que para reconocer texto o leer e interpretar cadenas de digitos binarios, algunos de los sistemas de cifrado convencionales realizan una dispersión y difusión de los valores de los pixeles de una imagen de tal manera que cuando una persona ve la imagen cifrada aun se puede reconocer cual es ésta, es dicir después de los procesos de cifrado se encuentra correlación entre los pixeles.

Las formas de cifrado que han dado mejores resultados en está área son los que se basan en transformadas de Fourier, transformadas discretas de Fourier, sistemas dinámicos(caos,multi-caos,hiper-caos)[18][23][22][25], por reducir a casi nula la correlación de los pixeles en las imágenes cifradas y ser más veloces que los basados en criptosistemas clásicos como AES y Triple-DES y algunos de los convencionales con alteraciones que evitan el problema que expone el parrafo anterior.

# Capítulo 2

# Fundamentos teóricos

Este capítulo está dedicado a la teoría matemática que nos ayudará a cimentar este trabajo. Comenzaremos con resultados básicos relacionados con los números enteros y posteriormente pasaremos a estudiar un poco a cerca de la teoría de grupos, muchas de la demostraciones se omitirán sin embargo intentaremos ser claros en la explicación de los teoremas necesarios, además cuando se requiera el lector será sugerido a estudiar en las referencias bibliográficas que usa este trabajo los temas no alcanzados por la tesis.

## 2.1. Los números enteros

Como primer tema del capítulo estudiaremos los números enteros y algunas de sus propiedades, no con tanto detalle como para hacer una construcción axiomática de los mismos pero si enunciando varios teoremas importantes, partiremos con el hecho de que los números enteros con las operaciones de suma y multiplicación usuales forman un campo. Iniciaremos los conceptos matemáticos con el algoritmo de la división de Euclides.

**Definición 2.1.1** *Dados* $a,b \in \mathbb{Z}$ *con* $b \neq 0$*, siempre existen* $q,r \in \mathbb{Z}$*, únicos tal que* $a = bq + r$ *donde* $0 \leq r < |b|$*. Los números* $q$ *y* $r$ *tienen los nombres de cociente y residuo respectivamente.*

**Definición 2.1.2** *Sean* $a,b \in \mathbb{Z}$ *con* $b \neq 0$*, se dice que* $b$ *divide a* $a$ *o que* $b$ *es un divisor de* $a$ *si al aplicar el algoritmo de la división de Euclides el residuo es cero, es decir existe* $q \in \mathbb{Z}$*, tal que* $a = bq$*. Si* $b$ *divide a* $a$ *se denotará como* $b|a$*.*

**Definición 2.1.3** *Sean* $a,n \in \mathbb{Z}$ *definimos* $a \pmod n$ *como el residuo que resulta de dividir* $a$ *entre* $n$*.*

Notemos que dado $n \in \mathbb{N}$ se tiene que para cualquier $a \in \mathbb{Z}$, el residuo de la división $a/n$ denotado por $a \pmod n$ forzosamente ha de pertenecer al conjunto $\{0, 1, \ldots, n-1\}$, es decir $a \pmod n \in \{0, 1, \ldots, n-1\}$, así pues podriamos agrupar en un mismo conjunto a todos los números enteros cuyo residuo al dividirse por $n$ es $a \pmod n$, y si repetimos el proceso para cada elemento de $\{0, 1, \ldots, n-1\}$ formariamos subconjuntos de enteros disjuntos a pares cuya unión son todos los números enteros, o lo que es lo mismo se induce una partición en $\mathbb{Z}$.

**Definición 2.1.4** *Sea* $n \in \mathbb{N}$ *y sean* $a,b \in \mathbb{Z}$*, definimos la relación* $\mathfrak{R} \subseteq \mathbb{Z} \times \mathbb{Z}$ *llamada relación módulo* $n$ *como*

$$a \mathfrak{R} b \iff a \pmod n = b \pmod n.$$

La relación módulo $n$ es una relación de equivalencia y el conjunto

$$\{0, 1, \ldots, n-1\}$$

es un conjunto completo de representantes de ésta que típicamente se denota como $(\mathbb{Z}/\mathbb{Z}_n)$.

**Definición 2.1.5** *Sean* $a,b,n \in \mathbb{Z}$*, se dice que* $a$ *es congruente con* $b$ *módulo* $n$ *y se denota como* $a \equiv b \, mod \, n$*, si* $n$ *divide* $a - b$*.*

> Nota: A partir de la **Definición 2.1.5** se puede deducir la relación de equivalencia de la **Definición 2.1.4**. Pusimos ambas notaciones por que según nos convenga ocuparemos una u otra en los teoremas en que cada una sea necesaria.

**Definición 2.1.6** *Sean* $a,b \in \mathbb{Z}$ *ambos distintos de cero, se llama máximo común divisor de* $a$ *y* $b$ *denotado por mcd*$(a,b)$ *al número* $d \in \mathbb{Z}$ *con* $d > 0$ *tal que:*

i. $d \mid a$ *y* $d \mid b$

ii. $\forall\ c \in \mathbb{Z}$, *tal que* $c \mid a$ *y* $c \mid b \implies c \mid d$

**Definición 2.1.7** *Sean* $a,b \in \mathbb{Z}$, *decimos que* $a$ *y* $b$ *son primos relativos si mcd* $(a,b) = 1$

**Definición 2.1.8** *Se dice que un número* $p \in \mathbb{Z}$ *con* $p > 1$ *es un número primo si sus únicos divisores son* $-1, 1, -p$ *y* $p$. *Al conjunto de números primos lo denotaremos por* $\mathbb{P}$. *Si un número entero mayor que uno no es primo lo llamaremos compuesto.*

**Teorema 2.1.1** ***(Teorema fundamental de la aritmética)*** *Sea* $n \in \mathbb{Z}^+$ *con* $n > 1$, *entoces* $n$ *se puede escribir de manera única salvo reordenamientos como el producto de potencias de números primos. Es decir existen* $\{p_1, \ldots, p_k\} \subset \mathbb{P}$ *y* $\{\alpha_1, \ldots, \alpha_k\} \subset \mathbb{Z}^+$, *tales que*

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}.$$

**Teorema 2.1.2** ***(Teorema de Dirichlet)*** *Sean* $a,b \in \mathbb{Z}^+$ *tal que mcd*$(a,b) = 1$, *entonces en la progresión aritmética de término general* $\{x_n\} = a\,n + b$ *existe una infinidad de números primos.*

> Nota: El teorema afirma la existencia de una infinidad de números primos en la progresión pero no que todos lo números de esta forma son primos, de hecho en cada progresión de esta forma existen también una infinidad de números compuestos.

## 2.2. Teoría de grupos

Los grupos son una estructura álgebraica que puede considerarse el concepto base del álgebra moderna, ya que los anillos, campos y espacios vectoriales

pueden ser estudiados como grupos dotados con operaciones y axiomas adicionales. Muchos matemáticos famosos, entre ellos Euler, Gauss, Lagrange, Ruffini, Abel y Galois[1] fueron los iniciadores de su estudio, en la actualidad no dejan de ser objeto de investigación hallando numerosos aspectos interesantes y aplicaciones en diversas ramas de las matemáticas como la geometría, la topología, la teoría de números; y en distintas areas como la genética, la química, la física, la criptografía y la informática por citar algunas. Empecemos con lo escencial, la definición de grupo.

**Definición 2.2.1** *Sea $G$ un conjunto no vacío sobre el cual está definida una operación binaria $(\cdot)$ a la que usualmente se le llama multiplicación, la pareja $(G,\cdot)$[2] forma un grupo si $\forall\ \alpha,\beta,\gamma \in G$ se cumple que:*

I. $\alpha \cdot \beta \in G$.

II. $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$.

III. $\exists\ e \in G$, *tal que* $e \cdot \alpha = \alpha \cdot e = \alpha$.

IV. $\exists\ \alpha^{-1} \in G$, *tal que* $\alpha \cdot \alpha^{-1} = \alpha^{-1} \cdot \alpha = e$.

> Nota: En un grupo no necesariamente se debe cumplir que $\alpha\beta = \beta\alpha$ para cualesquiera $\alpha,\beta$ elementos del grupo. Los grupos en los que todos sus elementos conmutan con respecto a la operación definida en ellos reciben el nombre de "Grupos Abelianos[3]".

[1] Leonhard Paul Euler (1707-1783), Johann Carl Friedrich Gauss (1777-1855), Joseph Louis Lagrange (1736-1813), Paolo Ruffini (1765-1825), Niel Henrik Abel (1802-1829), Évariste Galois (1811-1832)

[2] Si no hay motivo a confunción sobre quien es la operación que define al grupo $(G,\cdot)$ a éste se le denotará tan sólo como $G$.

[3] El nombre es en honor al matemático noruego Niel Henrik Abel.

**Lema 2.2.1** *Sea* $p \in \mathbb{N}$ *un número primo, tomemos el conjunto* $(\mathbb{Z}/\mathbb{Z}_p)$ *y definamos* $(\mathbb{Z}/\mathbb{Z}_p)^* := (\mathbb{Z}/\mathbb{Z}_p) - \{0\}$. *Se afirma que la pareja* $((\mathbb{Z}/\mathbb{Z}_p)^*, \cdot)$ *es un grupo abeliano, donde la operación* $(\cdot)$ *está dada por:*

$$\forall\ a,b \in (\mathbb{Z}/\mathbb{Z}_p)^*; \quad a \cdot b = a \cdot b \ (mod\ n).$$

La demostración del lema anterior es muy conocida y mencionada en los textos introductorios al estudio del álgebra moderna, para comprobarlo el lector puede consultar [1][2][3] o probar por propia cuenta que se cumplen las propiedades enlistadas en la **Definición 2.2.1**.

En la definición de grupo está involucrado un conjunto no vacío cualquiera, y puede pasar que algunos de sus subconjuntos hereden la estructura de grupo con respecto a la operación restringida a ellos.

**Definición 2.2.2** *Sea* $(G, \cdot)$ *un grupo y sea H un subconjunto de G, entonces* $(H, \cdot)$ *se dice ser un subgrupo de* $(G, \cdot)$ *denotado por* $H < G$, *si la pareja* $(H, \cdot)$ *forma en sí mismo un grupo.*

La cardinalidad del conjunto a partir del cual se forma un grupo le da propiedades especiales, para nosotros son importantes los grupos cuyo conjunto no tiene un cardinal infinito o en una forma más coloquial cuyo conjunto está formado por un número finito de elementos. A esta característica la llamaremos el orden de $(G, \cdot)$ y la denotaremos por $\circ(G)$.

**Definición 2.2.3** *Si* $(G, \cdot)$ *es un grupo y* $\alpha \in G$, *se define el orden de* $\alpha$ *como el mínimo entero positivo* $n$, *tal que* $\alpha^n = e$.

**Definición 2.2.4** *Un grupo* $(G, \cdot)$ *se dice ser cíclico si* $\exists\ \alpha \in G$, *tal que* $\forall\ \beta \in G$, $\exists\ n \in \mathbb{N}$, *tal que* $\alpha^n = \beta$. *A* $\alpha$ *se le llama elemento primitivo de G o generador de G, además cuando G es generado por* $\alpha$ *se denota como* $G = < \alpha >$.

La demostración del siguiente teorema se omitirá pero puede ser consultada en cualquier libro que trate teoría de grupos en particular en [1].

**Teorema 2.2.1** ***(Teorema de Lagrange)*** *Si* $(G,\cdot)$ *es un grupo finito y* $(H,\cdot)$ *es un subgrupo de* $(G,\cdot)$*, entonces* $\circ(H)$ *es un divisor de* $\circ(G)$*.*

> Nota: Si $k \in \mathbb{Z}$ es un divisor de $\circ(G)$, no necesariamente $(G,\cdot)$ debe tener un subgrupo de orden $k$.

**Teorema 2.2.2** *Si* $(G,\cdot)$ *es un grupo tal que* $\circ(G) = \mathsf{p} \in \mathbb{P}$*, entonces* $(G,\cdot)$ *es un grupo cíclico.*

**Demostración:** Tomemos $\alpha \in G$ y ahora definamos al conjunto $A$ como el conjunto de las potencias de $\alpha$, éste es un subgrupo de $(G,\cdot)$, queremos probar que $A := <\alpha> = (G,\cdot)$, ahora por el teorema de Lagrange se debe tener que $\circ(A)|\circ(G)$ pero como $\circ(G) = \mathsf{p}$ un número primo, entonces $\circ(A) = 1$ o bien $\circ(G) = \mathsf{p}$.

Si $\circ(A) = 1$ implica que $\alpha = \mathsf{e}$ pero como $\alpha$ se tomó en forma arbitraría, se tendría que $G = \{\mathsf{e}\}$ lo que contradice el hecho de que $\circ(G)$ es un número primo, por tanto no queda otra opción más que $\circ(A) = \mathsf{p}$, luego entonces se concluye lo que se quería probar $(G,\cdot) = <\alpha>$ .■

**Definición 2.2.5** *Sea* $(G,\cdot)$ *un grupo finito cíclico de orden* $\mathsf{n}$ *y sean* $\alpha,\beta \in G$*, con* $(G,\cdot) = <\alpha>$*. El logaritmo discreto base* $\alpha$ *de* $\beta$ *denotado por* $\log_\alpha \beta$*, es el único entero* $\mathsf{a} \in \{0,\ldots,\mathsf{n}-1\}$*, tal que* $\alpha^{\mathsf{a}} = \beta$*.*

Estos son los resultados sobre teoría de grupos que ocuparemos, son básicos dentro de un tratado especializado de la teoría pero suficientes para el fin de esta tesis.

## 2.3. Criptosistemas

Hemos hablado ya de la criptografía y los criptosistemas en el **Capítulo 1** y lo narrado en él es una explicación ordinaria de como aplicar la criptografía y de lo que es un criptosistema, pero para nuestros intereses nos convendría tener una definición formal y en lenguaje matemático.

**Definición 2.3.1** *Un criptosistema es una quintupla* $(\mathcal{P}, C, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *donde*

- $\mathcal{P}$ *es un conjunto finito de posibles textos planos.*
- $C$ *es un conjunto finito de posibles textos cifrados.*
- $\mathcal{K}$ *es un conjunto de posibles llaves.*

*y además se cumple que para cada* $K \in \mathcal{K}$*, existe una función de cifrado*

$$e_K \in \mathcal{E} \quad con \quad e_K : \mathcal{P} \longrightarrow C$$

*y una correspondiente función de descifrado*

$$d_K \in \mathcal{D} \quad con \quad d_K : C \longrightarrow \mathcal{P}$$

*tal que*

$$d_K(e_K(x)) = x$$

*para cada texto plano* $x \in \mathcal{P}$*.*

Alicia y Bruno deben establecer cuales serán especificamente los componentes de su criptosistema; a qué conjunto pertenecerán los símbolos de los textos planos, los símbolos de los textos cifrados, cuales son claves validas y las funciones a calcular para los procesos de cifrado y descifrado.

Matemáticamente si Alicia se quiere comunicar con Bruno y desea que él reciba del mensaje $x \in \mathcal{P}$, pero no está disponible un canal seguro para transmitirlo, Alicia puede resuelver el problema de que el mensaje viaje sin ser leido por alguien no autorizado si cálcula

$$y = e_K(x),$$

donde $K \in \mathcal{K}$ y $y \in C$, y luego envia $y$ como mensaje. Cuando Bruno lo recibe, él conoce la clave de descifrado y le basta con revertir el proceso calculando el mensaje original de la forma

$$x = d_K(y).$$

Para cualquier tipo de criptosistema siempre se cumple la igualdad

$$d_K(e_K(x)) = d_K(y) = x$$

sin embargo las funciones de cifrado y descifrado tienen características especiales según se trate de un criptosistema simétrico, asimétrico o híbrido.

En los Criptosistemas simétricos las funciones de cifrado y descifrado son fáciles de deducir una de la otra en ocasiones constan de operaciones que se invierten a sí mismas.

Para el caso de los asimétricos necesitamos funciones llamadas de un sentido, donde las operaciones son fáciles de calcular en una dirección pero cuya operación inversa es díficil de calcular, la idea de esto es que un adversario no pueda deducir la clave privada en base a la clave pública.

La naturaleza de los híbridos hace que una de sus funciones de cifrado tenga propiedades de funciones de cifrado simétrico y la otra tenga propiedades de funciones de cifrado asimétrico, lo mismo ocurre para las funciones de descifrado.

## 2.4. ElGamal

ElGamal es un criptosistema de clave pública, éste basa su seguridad en la teoría de grupos y el problema del logaritmo discreto, fue publicado por el criptógrafo egipcio Taher ElGamal (1955- ) en 1985 en el artículo titulado *A Public key Cryptosystem and A Signature Scheme based on discrete Logarithms* [16].

El problema del logaritmo discreto consta de encontrar el logaritmo de un elemento en un determinado grupo finito, dado un generador de éste.

Tomemos $\gamma \in \mathbb{Z}$ un número primo y consideremos el grupo $((\mathbb{Z}/\mathbb{Z}_\gamma)^*, \cdot)$ de orden $\gamma - 1$, el problema de encontrar logaritmos discretos en este grupo sería: Para un elemento arbitrario $\beta \in (\mathbb{Z}/\mathbb{Z}_\gamma)^*$, hallar el único $\mathsf{a} \in \{0, \ldots, \gamma - 2\}$, tal que $\alpha^{\mathsf{a}} \equiv \beta \bmod \gamma$, donde $\alpha$ es un generador dado de $((\mathbb{Z}/\mathbb{Z}_\gamma)^*, \cdot)$.

A continuación escribiremos la definición formal del criptosistema ElGamal, describiendo los componentes de la quintupla que lo forma.

**Definición 2.4.1** ***(Criptosistema ElGamal)*** *Sea $\gamma \in \mathbb{Z}$ un número primo tal que el problema del logaritmo discreto en $((\mathbb{Z}/\mathbb{Z}_\gamma)^*, \cdot)$ sea computacionalmente intratable, sea $\alpha \in (\mathbb{Z}/\mathbb{Z}_\gamma)^*$ un elemento primitivo de tal grupo multiplicativo y $k \in (\mathbb{Z}/\mathbb{Z}_{\gamma-1})$ un número aleatorio. Construyamos el criptosistema $\mathcal{G} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ donde $\mathcal{P} = (\mathbb{Z}/\mathbb{Z}_\gamma)^*$, $\mathcal{C} = (\mathbb{Z}/\mathbb{Z}_\gamma)^* \times (\mathbb{Z}/\mathbb{Z}_\gamma)^*$, y $\mathcal{K} = \{(a, \beta) \mid \beta \equiv \alpha^a \ mod\ \gamma\}$.*

*Para $(a, \beta) = K \in \mathcal{K}$ definimos la función de cifrado*

$$e_K : (\mathbb{Z}/\mathbb{Z}_\gamma)^* \longrightarrow (\mathbb{Z}/\mathbb{Z}_\gamma)^* \times (\mathbb{Z}/\mathbb{Z}_\gamma)^*$$

*dada por:*

$$\forall\ x \in (\mathbb{Z}/\mathbb{Z}_\gamma)^*; \quad e_K(x) = (y_1, y_2),$$

*donde*

$$y_1 = \alpha^k \ mod\ \gamma$$

*y*

$$y_2 = x\beta^k \ mod\ \gamma.$$

*La función de descifrado $d_K : (\mathbb{Z}/\mathbb{Z}_\gamma)^* \times (\mathbb{Z}/\mathbb{Z}_\gamma)^* \longrightarrow (\mathbb{Z}/\mathbb{Z}_\gamma)^*$ la definimos como:*

$$\forall\ (y_1, y_2) \in (\mathbb{Z}/\mathbb{Z}_\gamma)^* \times (\mathbb{Z}/\mathbb{Z}_\gamma)^*; \quad d_K(y_1, y_2) = y_2(y_1^a)^{-1} \ mod\ \gamma.$$

La definición anterior nos obliga a conocer al menos quien es alguno de los elementos primitivos de nuestro grupo para esto usaremos el siguiente teorema el cual da las condiciones suficientes y necesarias para identificar a un elemento del grupo como primitivo.

**Teorema 2.4.1** *Sea $\gamma \in \mathbb{Z}$ un número primo con $\gamma > 2$ y $\alpha \in (\mathbb{Z}/\mathbb{Z}_\gamma)^*$, entonces $\alpha$ es un elemento primitivo de $((\mathbb{Z}/\mathbb{Z}_\gamma)^*, \cdot)$ si y sólo si $\alpha^{(\gamma-1)/q} \not\equiv 1\ mod\ \gamma$, $\forall q \in \mathbb{P}$ tal que $q \mid (\gamma - 1)$.*

**Demostración:** Supongamos primero que $\alpha$ es un elemento primitivo modulo $\gamma$, entonces $\alpha^i \not\equiv 1 \bmod \gamma\ \forall\ i \in \{1, \ldots, \gamma - 2\}$, de esta manera se obtiene la primera implicación.

Supongamos ahora que $\alpha \in (\mathbb{Z}/\mathbb{Z}_\gamma)^*$ no es un elemento primitivo módulo $\gamma$, sea $k$ el orden de $\alpha$, entonces por el teorema de Lagrange se tiene que $k \mid (\gamma - 1)$

y $k < \gamma - 1$ ya que $\alpha$ no es un elemento primitivo, por tanto se debe cumplir que $(\gamma - 1)/k \in \mathbb{Z}$ y además $(\gamma - 1)/k > 1$. Sea $\mathsf{q}$ un número primo divisor de $(\gamma - 1)/k$, entonces $k$ es un divisor de $(\gamma - 1)/\mathsf{q}$. Sabemos que $\alpha^k \equiv 1 \bmod \gamma$ y que $k|(\gamma - 1)/\mathsf{q}$ y por tanto se tiene que $\alpha^{(\gamma-1)/\mathsf{q}} \equiv 1 \bmod \gamma$.■

Para hacer más transparente el funcionamiento de ElGamal escribiremos un ejemplo de como se cifra y descifra usandolo.

Ejemplo: Imaginemos que Alicia tiene que mandarle el mensaje $\mathsf{x} = 1299$ a Bruno, éste genera la claves públicas y la privada obteniendo los valores

$$\gamma = 2579,$$
$$\alpha = 2,$$
$$\mathsf{a} = 765,$$

recordemos que $\gamma$ es un número primo $\alpha$ un generador de $((\mathbb{Z}/\mathbb{Z}_\gamma)^*, \cdot)$ y $\mathsf{a} \in (\mathbb{Z}/\mathbb{Z}_\gamma)^*$ es elegido al azar, mientras que.

$$\beta := \alpha^a \bmod \gamma = 2^{765} \bmod 2579 = 949$$

Bruno ahora tiene que publicarle a Alicia las claves $\alpha, \beta, \gamma$ y conservar privada la clave $\mathsf{a}$. Para cifrar ella selecciona un número aleatorio en $((\mathbb{Z}/\mathbb{Z}_\gamma)^*, \cdot)$ digamos $k = 853$ y después calcula

$$\begin{aligned} y_1 &= \alpha^k \bmod \gamma \\ &= 2^{853} \bmod 2579 \\ &= 435 \end{aligned}$$

y

$$\begin{aligned} y_2 &= \mathsf{x}\beta^k \bmod \gamma \\ &= 1299\ (949^{853}) \bmod 2579 \\ &= 2396 \end{aligned}$$

cuando Bruno recibe el mensaje cifrado $y = (y_1, y_2) = (435, 2396)$, usa su llave privada en los siguientes cálculos.

$$\begin{aligned}
x &= y_2(y_1^a)^{-1} \bmod \gamma && (2.1)\\
&= x\beta^k((\alpha^k)^a)^{-1} \bmod \gamma && (2.2)\\
&= x\beta^k((\alpha^{ka})^{-1} \bmod \gamma && (2.3)\\
&= x\beta^k((\alpha^a)^k)^{-1} \bmod \gamma && (2.4)\\
&= x\beta^k((\beta)^k)^{-1} \bmod \gamma && (2.5)\\
&= 2396\ (435^{765})^{-1} \bmod 2579 && (2.6)\\
&= 1299. && (2.7)
\end{aligned}$$

Cabe mencionar que el número $k$ que Alicia usa para cifrar el mensaje hace que con muy poca probabilidad dos textos cifrados sean iguales no importando si se trata del mismo texto plano, además como lo muestra la cadena de igualdades (2.1) a (2.7), el número $k$ no tiene relevancia a la hora descifrar el mensaje.

## 2.5. DES

Las necesidades por cifrar información digital culminaron en que el 15 de mayo de 1973 [9] se hiciera por parte de la National Bureau of Standars (NBS) la publicación de una solicitud de propuestas para un algoritmo de cifrado que cumpliera con rigurosos criterios de diseño, pero ninguna de las propuestas recibidas parecía buena por lo que el 27 de agosto de 1974 se lanzó una nueva solicitud, esta vez IBM presentó un algoritmo de nombre Lucifer creado por Horst Feistel, para que más tarde el 17 de marzo de 1975 el Data Encryption Standard (DES) un criptosistema de cifrado por bloques que actua sobre textos planos compuestos por digitos binarios, fue publicado en el registro federal y después de bastantes discusiones alrededor de él el 15 de enero de 1977 fue adoptado como estandar para cifrar datos no clasificados en el Federal Information Processing Standards publicación 46 (FIPS-46), ha sido confirmado en el estandar en 1983, en 1988 (FIPS-46-1), en 1993 (FIPS-46-2); en 1998 se definió en el FISP-46-3 el algoritmo Triple-DES una variante de DES que incrementaba la seguridad, finalmente para el año de 2002 DES fue reemplazado por el Advanced Encryption Standard (AES) al probar que se podia realizar un ataque por fuerza bruta mediante una

computadora de arquitectura dedicada en un tiempo poco menor a dos días. Sin embargo Triple-DES sigue siendo uno de los algoritmos simétricos más utilizados.

Hubo controversia acerca del criptosistema por parte de varios retractores que afirman que se construyó una puerta trasera al algoritmo de modo que el gobierno pudiese tener control sobre él, la National Security Agency (NSA) participó activamente al modificar algunos aspectos. El criptosistema padre Lucifer empleaba claves de 128 bits de logitud mientras que en DES las claves son de 64 bits, de los cuales 8 bits son descartados en el proceso, reduciendo así las claves a 56 bits, la consecuencia de esto es que el espacio de posibles claves fue reducido de $2^{127}$ a $2^{55}$ y estadísticamente se sabe que no es necesario probar las $2^{55}$ posibles claves, con probar tan sólo un poco más de la mitad de ellas la probabilidad de hallar la correcta es alta, volviéndolo así más vulnerable a un ataque por fuerza bruta, adicional a esto se presume que las cajas S que sirven para hacer difusión en el orden de los bits sufrieron importantes cambios. A pesar de todo esto es quizá el más popular de los criptosistemas simétricos modernos pues fue el primero que sirvió como estandar para cifrar información.

El cifrado de DES es un cifrado por bloques, los textos planos son cadenas de 64 bits al igual que los textos cifrados y las clves. El algoritmo de cifrado con DES [17][9][11][12] es el siguiente.

1. Inicio.
2. Leer un texto plano $\mathsf{x}$ de 64 bits.
3. Conseguir la clave $K$.
4. Permutar los bits de $\mathsf{x}$ según una permutación inicial IP(fija) y ponerlos en $\mathsf{x}_0$.
5. Dividir la cadena $\mathsf{x}_0$ de 64 bits sin alterar el orden en dos de 32 bits nombradas $L_0$ y $R_0$, de tal manera que $L_0$ es la cadena de 32 bits más a la izquierda y $R_0$ los más a la derecha.
6. Para $i = 1, \ldots, 16$ hacer el ciclo:
   - $L_i = R_{i-1}$

- $R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$

7. Aplicar la permutación inversa de IP a los bits de la cadena $R_{16}L_{16}$ y almacenarlos en y
8. Imprimir y.
9. Fin

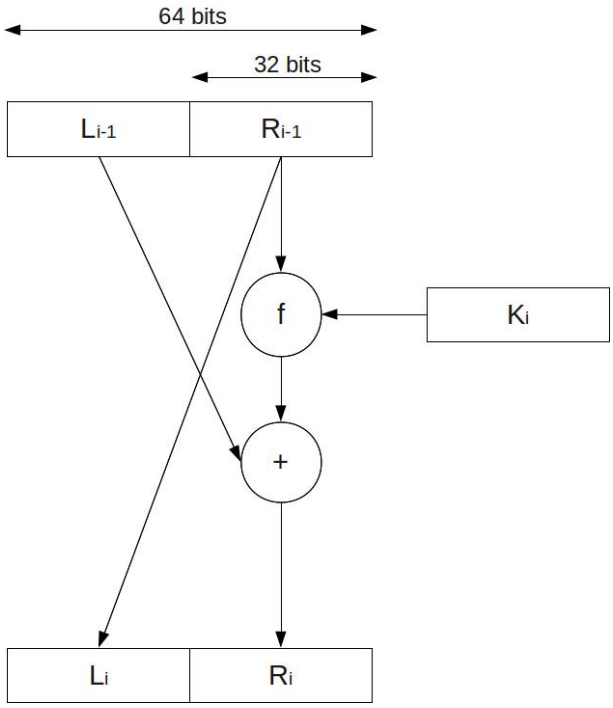



Figura 2.1: Esquema correspondiente a una iteración del paso 6 en el algoritmo de cifrado DES [9].

El paso 6 es un ciclo ejecutado 16 veces, el cual puede ser un poco confuso porque no hemos definido algunas cosas. Cada una de las $K_1, K_2, \ldots, K_{16}$ son cadenas de 48 bits calculadas a partir de $K$ y más adelante detallaremos la forma, el símbolo $\oplus$ representa la operación sobre bits OR exculsivo (XOR).

La función $f$ toma como argumentos una cadena de 32 bits y otra de 48 bits, supongamos que $A$ y $J$ son las respectivas cadenas, El resultado tendrá 48 bits.

- Primero hay que expandir $A$ a una cadena de 48 bits con una función E que hace una permutación con una cadena de 48 bits formada con los 32 bits originales en la cadena $A$ y 16 repetidos.
- Después se calcula $B = B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8 = E(A) \oplus J$, donde $B_i$ consta de 6 bits para cada $i = 1, \ldots, 8$.
- El siguiente paso son 16 cajas $S$, cada una de éstas es una matriz de tamaño $4 \times 16$, con entradas entre 0 y 15, para cada $B_i = b_1b_2b_3b_4b_5b_6$ los digitos binarios determinan las coordenadas dentro de la matriz como sigue: los bits $b_1b_6$ se transforman a su representación decimal y se les suma uno, este número será el número de fila en la caja, lo mismo hacemos con los bits centrales, es decir la columna está dada por la representación decimal de $b_2b_3b_4b_5$ más uno, una vez que sabemos la fila y columna de la matriz, la entrada es un número entre 0 y 15 que cuando lo transformamos de nuevo a su representación binaria consta de 4 bits, al final de este paso tendremos que $C_i = S_i(B_i)$ donde $C_i = c_1c_2c_3c_4$.
- Por último obtendremos el resultado $f(A, J) = P(C)$ donde

  $$C = C_1C_2C_3C_4C_5C_6C_7C_8$$

  y $P$ es una permutación fija de los 32 bits de $C$.

  Las claves $K_1, \ldots, K_{16}$ son calculadas convirtiendo primero la clave $K$ de 64 bits a una cadena de 56 bits borrando los bits en posiciones un multiplo de 8 y continuando con un ciclo de permutaciones que son fijas. Todas las permutaciones se pueden consultar en [17] [9] [11][12].

### 2.5.1. Triple-DES

Cuando se descubrió que DES podia ser atacado por fuerza bruta en un tiempo razonablemente efectivo se ideó una manera de aumentar la seguridad de éste sin modificar en gran cosa el algoritmo. La variante propuesta fue Triple-DES que consiste en realizar tres rondas del DES sencillo ya sea ocupando dos o tres claves distintas, la seguridad se aumenta en el doble (claves de 112 bits) pero en contra parte se necesitan realizar tres veces más operaciones aumentando costos de cómputo.

En la versión más simple usando dos claves, el texto plano lo ciframos con una clave, luego hacemos como si este fuese el texto cifrado para el proceso con la segunda clave y los desciframos, por último volvemos a cifrar el texto que nos resulte, es decir supongamos que $\mathsf{x}$ es el texto plano que $k_1, k_2$ son las claves y que $\mathsf{e}$,$\mathsf{d}$ son las respectivas funciones de cifrado y descifrado, entonces el texto cifrado $\mathsf{y}$ esta dado por

$$\mathsf{y} = \mathsf{e}_{k_1}(\mathsf{d}_{k_2}(\mathsf{e}_{k_1}(\mathsf{x})))$$

para poder descifrar al igual que en el DES tenemos que revertir el proceso pero en esta ocasión tres veces.

$$\mathsf{x} = \mathsf{d}_{k_1}(\mathsf{e}_{k_2}(\mathsf{d}_{k_1}(\mathsf{y})))$$

En la versión usando tres claves lo único que hay que cambiar es que en la última ronda ciframos o desciframos con la tercer clave según el caso.

$$\mathsf{y} = \mathsf{e}_{k_3}(\mathsf{d}_{k_2}(\mathsf{e}_{k_1}(\mathsf{x})))$$

y para el descifrado

$$\mathsf{x} = \mathsf{d}_{k_3}(\mathsf{e}_{k_2}(\mathsf{d}_{k_1}(\mathsf{y}))).$$

## 2.6. Imagen digital como mapa de bits

Aplicaremos un cifrado Triple-DES a imágenes digitales que están descritas por un modelo sencillo de representación y almacenamiento digital llamado mapa de bits, estas imágenes están formadas por pixeles[4] los cuales son pequeñas

[4]La palabra *pixel* tiene su raiz en una contracción inglesa de *picture element* por lo tanto no es una unidad de medida, sino que se trata en realidad de un elemento de la imagen como viene a indicar su origen.

celdas de color que al observarse en conjunto proporcionan una ilusión óptica interpretada por el cerebro humano com una imagen, la idea básica de este tipo de esquema es pensar en la imagen como una matriz de tamaño arbitrario con entradas cuyo valor está dentro de un rango discreto de número reales, cada pixiel se hace corresponder con una entrada de la matriz que la describe, por lo tanto computacionalmente hablando cuantas más entradas tenga la matriz y más grande sea el intervalo donde puedan ser tomados los valores de las entradas mejor será la calidad de la imagen. En la figuras 2.2 y 2.3 mostramos distintas imágenes en mapa de bits.

Figura 2.2: Esta es una imagen cuyas dimenciones en pixeles son $512 \times 512$, se asigna sólo un bit para almacenar cada pixel por lo que el valor es cero o uno (negro o blanco).

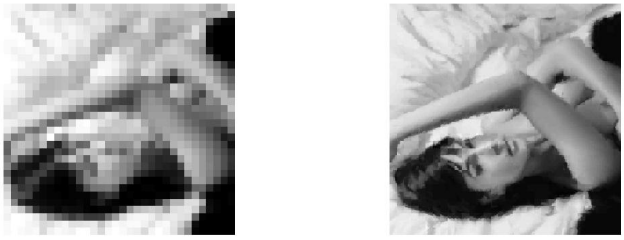


Figura 2.3: El número de pixeles de la representación de la imagen a la izquierda es de $128 \times 128$, mientras que para la imagen de la derecha es de $256 \times 256$.

En el caso de las imágenes a color bajo este formato, se superponen tres matrices donde cada una representa uno de los tres canales de color con que despliegan imágenes en formato RGB *(Red, Green and Blue)* en los monitores, de esta forma cada pixel posee tres valores, uno para cada componente de color en que se integran las imágenes. en la figura 2.4 se muestra una imagen con sus respectivos canales de color.

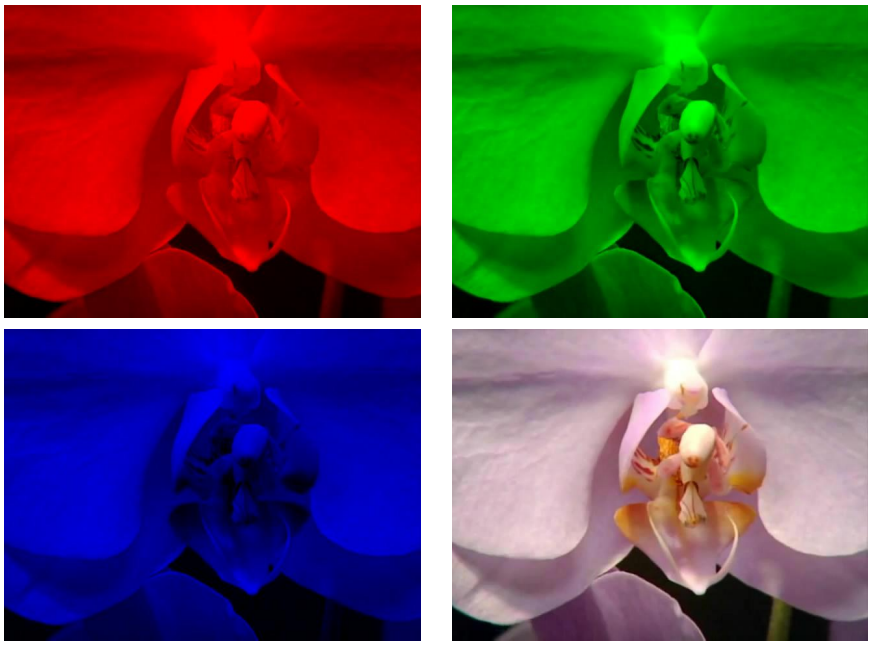


Figura 2.4: Los tres canales de color superpuestos dan como resultado la imagen colocada a la derecha y abajo en el arreglo de las cuatro imágenes.

El formato general del archivo, se integra por un encabezado de información de archivo donde se especifican cosas como el tipo, tamaño y esquema del archivo; una cabecera de información de la imagen que contiene datos como las dimensiones de la imagen, el tipo de compresión y el formato de color; después viene la tabla de color que detalla cada uno de los colores que la imagen puede contener y por último el vector de valores de cada uno de los pixeles en la imagen.

# Capítulo 3

# Implementación

Este capítulo es la parte central de la tesis ya que con la ayuda de la teoría matemática mencionada daremos solución a la motivación del trabajo cumpliendo con los objetivos.

## 3.1. Implementación de ElGamal

A lo largo de esta sección hablaremos de la implementación del criptosistema de clave pública ElGamal y mencionaremos los algoritmos que empleamos en cada una de las partes en que dividimos la implementación de éste. Para nuestro proposito necesitamos operar con números grandes que no pertenecen al rango que manejan las bibliotecas estandar del lenguaje c++, además algunas operaciones que usaremos como parte de los algoritmos no están definidas como tal en ellas, por estas razones ocuparemos *The GNU Multiple Precision Aritmetic Library 5.0.2 (GMP 5.0.2)*[1] la cual es una biblioteca elaborada en lenguaje c que posee una amplia variedad de funciones aritméticas de presición arbitraria, es decir sin limitaciones prácticas salvo la capacidad física de memoria del hardware donde se programa, *GMP* opera sobre números enteros, racionales y de punto flotante con algoritmos que sino en todos los casos son los óptimos, son muy eficientes.

[1] Ver terminos de la licencia GNU en [21]

### 3.1.1. Números primos

Lo primero que tenemos que hacer es encontrar un número primo $\gamma$ lo suficientemente grande para que a partir de él construyamos el grupo $((\mathbb{Z}/\mathbb{Z}_\gamma)^*, \cdot)$ en donde el problema del logaritmo discreto se vuelva computacionalmente intratable [14].

Los números primos son un subconjunto infinito de números enteros los cuales cumplen la propiedad descrita en la **Definición 2.1.8** éstos han sido objeto de estudio para muchos matemáticos a través del tiempo. Los números primos no tienen un patrón de aparición en la secuencia de los números naturales sin embargo se han hecho estudios para poder determinar como es su distribución dentro de ellos, Johann Carl Friedrich Gauss (1777-1855) y Adrien Marie Legendre (1752-1833) fueron los iniciadores de estos estudios, y conjeturaron de forma independiente que si se define a $\pi(n)$ como la cantidad de números primos menores o iguales que $n$, se tiene que cuando $n$ tiende a infinito entonces $\pi(n)$ tiende asintóticamente a $ln\, n$ [4], o en otras palabras para valores de $n$ suficientemente grandes se cumple que

$$\pi(n) \approx \frac{n}{ln\, n}.$$

En general el problema de determinar si un número entero mayor que uno es o no primo es de bastante relevancia y para ello existen algunas pruebas de primalidad las cuales pueden ser determinísticas o probabilísticas, las primeras dado un número verifican las hipótesis de un teorema matemático respondiendo de forma exacta a la pregunta de si un número es primo o no, las segundas que son probabilísticas como el nombre lo indica sólo dan una probabilidad de que un determinado número sea primo. Las pruebas de primalidad determinísticas son difíciles de implementar en una rutina de cómputo, no así las probabilísticas que pueden equivocarse sobre la afirmación de que un número es primo con un margen de error que podemos hacer tender a cero cuando la prueba se realiza en varias ocaciones.

### 3.1.2. Prueba Miller-Rabin

En nuestro caso usaremos la prueba de primalidad probabilística Miller-Rabin una de las más efectivas que existen[21]. Presentaremos el algoritmo en pseudo-

código, pero en la implementación de nuestra rutina de cómputo no hubo necesidad de programarlo pues la biblioteca *GMP* incorpora la función de prueba de primalidad que corresponde a la prueba probabilística Miller-Rabin, dado un número impar $n$ la prueba responde si el número es compuesto o si es primo con una probabilidad aproximada de 0.75.

1. Inicio
2. Leer $n$
3. Escribir $n - 1 = 2^k m$, donde $m$ es impar.
4. Generar un número aleatorio $a$ tal que $1 \leq a \leq n - 1$.
5. Hacer $b = a^m \bmod n$.
6. Si $b \equiv 1 \bmod n$, entonces $n$ es primo; Terminar.
7. Para $i = 0, \ldots, k - 1$ hacer
   - Si $b \equiv -1 \bmod n$, entonces $n$ es primo; Terminar.
   - Caso contrario hacer $b = b^2 \bmod n$.
8. Responder $n$ es compuesto.
9. Fin.

### 3.1.3. Propuesta para calcular un elemento primitivo

Lo siguiente que tenemos que hacer para implementar el criptosistema es saber quien es un elemento primitivo del grupo multiplicativo $((\mathbb{Z}/\mathbb{Z}_\gamma)^*, \cdot)$ estos elementos son especiales en el grupo pues como hemos visto todos los demás elementos son una potencia de éste. El **Teorema 2.4.1** nos dice las condiciones suficientes y necesarias que un elemento del grupo debe cumplir para ser considerado un generador del grupo, pero podría ser poco eficiente cuando se ponga en práctica en un programa de cómputo si se trabaja con un número primo grande como es este caso, se sugiere con objeto de que el criptosistema sea seguro que el número primo $\gamma$ a partir del cual formaremos el grupo $((\mathbb{Z}/\mathbb{Z}_\gamma)^*, \cdot)$ sea del orden de $10^{300}$,

por tanto y según lo afirma el teorema habría que comprobar para $\alpha \in (\mathbb{Z}/\mathbb{Z}_\gamma)^*$ que

$$\alpha^{(\gamma-1)/q} \not\equiv 1 \bmod \gamma,$$

para todos los $q$ divisores primos de $(\gamma - 1)$, lo cual representa un reto en cuanto a cálculos para una computadora comenzando con el hecho de hallar a cada uno de estos primos divisores de $(\gamma - 1)$.

Para evitar el problema que nos presenta el teorema haremos la siguiente propuesta la cual es una de las aportaciones de la tesis ya que no se encontró en ningún texto referente al tema la forma en que esto se pudiera realizar. Generaremos dos números primos $p$,$q$ del orden de $10^{150}$, luego a partir de ellos generaremos uno más de la forma

$$\gamma := 2npq + 1.$$

Debe existir un número primo de esta forma pues $\text{mcd}(2pq, 1) = 1$ y según el **Teorema 2.1.2 (Teorema de Dirichlet)** la progresión $2pq(n) + 1$ tiene una infinidad de números primos. Por otra parte al multiplicar por dos cualquier número el resultado es un número par y sumarle uno a un número par siempre lo convierte en un número impar, escoger de esta forma al número $\gamma$ tiene sentido pues el único número primo par es 2. Hagamos un ejemplo de lo anterior a pequeña escala.

Ejemplo. Tomemos $p = 7$ y $q = 11$ dos números primos pequeños y a partir de ellos veamos para que valor de $n$ encontramos un número primo de la forma $2n(7)(11) + 1$.

$$\begin{aligned} n = 1; &\qquad \gamma := 2npq + 1 = 2(1)(7)(11) + 1 = 155 \\ n = 2; &\qquad \gamma := 2npq + 1 = 2(2)(7)(11) + 1 = 309 \\ n = 3; &\qquad \gamma := 2npq + 1 = 2(3)(7)(11) + 1 = 463. \end{aligned}$$

En este pequeño ejercicio cuando $n$ toma los valores de 1 ó 2 se encuentra que $2npq + 1$ resulta ser un número compuesto de hecho multiplo de cinco en el caso $n = 1$, y de tres en el caso de $n = 2$, pero cuando $n = 3$, se tiene que $2npq + 1$ es igual a 463 un número primo.

Otra gran ventaja que tendremos al fabricar nuestro número primo con la forma que mencionamos es que el criptosistema de ElGamal se vuelve fuerte contra un ataque Pohling - Hellman [9], donde se requiere tener la factorización en potencias de primos de $\gamma - 1$ pero la manera en que está construido $\gamma - 1$ es como aplicar el famoso RSA lo cual es como un doble candado, a diferencia de un intruso que quisiera violar nuestro criptosistema nosotros conoceremos inmediatamente los factores primos de $\gamma - 1 = 2npq$ que corresponden a $2, p, q$, y los divisores primos de $n$, factorizar a este último no es complicado pues generalmente no es tan grande.

Una vez conociendo todos los factores primos de $\gamma - 1$, aplicar el **Teorema 2.4.1** para encontrar un elemento primitivo del grupo $((\mathbb{Z}/\mathbb{Z}_\gamma)^*, \cdot)$ se reduce a realizar los calculos de exponenciación y de reducción módulo $\gamma$ que el teorema establece.

### 3.1.4. Programa de cómputo

En este punto ya podemos generar un número primo $\gamma$ adecuado para nuestro criptosistema y también sabemos como hallar un elemento primitivo del grupo $((\mathbb{Z}/\mathbb{Z}_\gamma)^*, \cdot)$ los dos ingerdientes pricipales para nuestra implementación la cual haremos en tres rutinas distintas: una que generará la claves, una para cifrar y otra más para descifrar.

#### Generación de claves

Volvamos al ejemplo planteado con Alicia y Bruno, supongamos que Alicia quiere comunicarse con Bruno, éste genera las claves y con ello se generaran los números $\alpha, a, \beta, \gamma$ descritos en la **Definición 2.4.1**, los números $\alpha, \beta, \gamma$ son los que se necesitan para cifrar un mensaje y además son públicos, pueden ser del conocimiento de cualquier persona pero de particular interes para Bruno es que los sepa Alicia, él será el único que pueda descifrar los mensajes cifrados con la claves públicas a tavés de su clave privada $a$.

Algoritmo de generación de claves.

1. Inicio.
2. Generar dos números primos aleatorios p,q del orden de $10^{150}$.
3. Para $n = 1, 2, 3, \ldots$ hacer el ciclo:
   - Calcular $2npq + 1$.
   - Si $2npq + 1$ es primo, entoces terminar ciclo.
4. hacer $\gamma = 2npq + 1$.
5. Factorizar n.
6. Almacenar los factores primos de n.
7. Hacer el ciclo:
   - Generar un número aleatorio $0 < \alpha < \gamma$ con almenos un factor primo grande.
   - Si $\alpha^{(\gamma-1)/p} \equiv 1 \bmod n$, comenzar el ciclo de nuevo.
   - Si $\alpha^{(\gamma-1)/q} \equiv 1 \bmod n$, comenzar el ciclo de nuevo.
   - para cada factor primo r de n hacer el ciclo:
     - Si $\alpha^{(\gamma-1)/r} \equiv 1 \bmod n$, comenzar el ciclo de nuevo.
   - Terminar ciclo.
8. Generar un número aleatorio a.
9. Calcular $\beta = \alpha^a \bmod n$.
10. Imprimir los valores $\alpha, a, \beta, \gamma$.
11. Fin.

**Cifrado**

Para cifrar los textos planos debemos conocer sólo las claves públicas, generar un número aleatorio $0 < k < \gamma$ y realizar los cálculos

$$y_1 = \alpha^k \bmod \gamma$$

$$y_2 = x\beta^k \bmod \gamma,$$

los números $y_1$ y $y_2$ son el texto cifrado. Escribiremos la sucesión completa de pasos a seguir para cifrar.

Algoritmo de Cifrado.

1. Inicio.
2. Conseguir el texto plano $x$.
3. Conseguir las claves públicas $\alpha, \beta, \gamma$.
4. Generar un número aleatorio $0 < k < \gamma$.
5. Calcular $y_1 = \alpha^k \bmod \gamma$.
6. Calcular $y_2 = x\beta^k \bmod \gamma$.
7. Imprimir el texto cifrado $(y_1, y_2)$.
8. Fin.

Notemos que si tomamos un texto plano y lo ciframos dos veces lo más probable es que los textos cifrados no coinsidan ya que éstos dependen directamente del número aleatorio $k$ lo que hace fuerte al criptosistema de ElGamal.

**Descifrado**

Para poder descifrar un texto cifrado se necesitan la clave privada y las claves públicas, necesitaremos calcular el inverso módulo $\gamma$ de $y_1^a$ para saber quien es este número tenemos que aplicar un algoritmo que es un tanto complejo, pero la biblioteca *GMP- 5.0.2* cálcula tal inverso si éste existe con la función *invert*.

Algoritmo de Descifrado.

1. Inicio.
2. Conseguir el texto cifrado $(y_1, y_2)$.
3. Conseguir la clave privada $a$ y las claves públicas $\alpha, \beta, \gamma$.
4. Calcular $(y_1^a)^{-1} \bmod \gamma$.
5. Calcular $x = y_2((y_1^a)^{-1}) \bmod \gamma$.
6. Imprimir el texto plano $x$.
7. Fin.

Para el descifrado son necesarias la clave privada $a$ y las claves públicas $\alpha, \beta, \gamma$ y resolver los cálculos

$$\begin{aligned} x &= y_2(y_1^a)^{-1} \bmod \gamma \\ &= x\beta^k((\alpha^k)^a)^{-1} \bmod \gamma \\ &= x\beta^k((\alpha^{ka})^{-1} \bmod \gamma \\ &= x\beta^k((\alpha^a)^k)^{-1} \bmod \gamma \\ &= x\beta^k((\beta)^k)^{-1} \bmod \gamma \end{aligned}$$

## 3.2. Cifrado de imágenes

Las imágenes que cifraremos estan en formato .bmp que corresponde a mapas de bits y como vimos esto no es más que una matriz en donde el tamaño de la misma corresponde al número de pixeles que conforman la imagen y el rango de valores al que pertenece cada entrada de la matriz es el umbral de color de la imagen, supongamos que tenemos una imagen de $512 \times 512$ pixeles y si que cada uno de estos pixeles tiene un valor dentro de una escala de grises entre 0 y 255 sólo necesitaremos una cadena de 8 bits para representar tal valor en una computadora.

La implementación del cifrado de imágenes la haremos con la siguiente receta:

1. Inicio.
2. Leer la imagen.
3. Convertir el mapa de bit en una matriz.
4. Transformar cada entrada de la matriz a su representación binaria.
5. Agrupar en cadenas de 64 bits de izquierda a derecha y de arriba a abajo las representaciones binarias de los pixeles vecinos.
6. A cada grupo de 64 bits cifrarlo con Triple-DES.
7. Volver a agrupar en cadenas de 8 bits.
8. Convertir cada cadena de 8 bits en un número decimal entero sin signo y formar una matriz
9. Construir un mapa de bits con base a la matriz formada.
10. Desplegar imagen.
11. Fin.

En cuanto al descifrado de la imagen el algoritmo es prácticamente el mismo salvo en el paso 6, en donde esta vez a cada grupo de 64 bit lo descifraremos usando Triple-DES.

Para más detalles de la implementación de Triple-DES se recomienda ver [20] el cual es un trabajo dedicado a la programación de un sistema en lenguaje c++ que ejecute ambos algoritmos (cifrado y descifrado).

# Capítulo 4

# Resultados

En este punto dentro del trabajo es hora de ver el desempeño de nuestra implementación del criptosistema híbrido basado en ElGamal y Triple-DES propuesto para cifrar imágenes y transmitirlas de forma segura por medios digitales.

Realizaremos un ejemplo completo de como funcionan los procesos de cifrado y descifrado. Generaremos primero nuestras claves para ElGamal ya que en tiempos posterior estas pueden ser ocupadas para que nos envien cualquier imagen. Al ejecutar el programa nos encontraremos con el siguiente diálogo de pantalla.

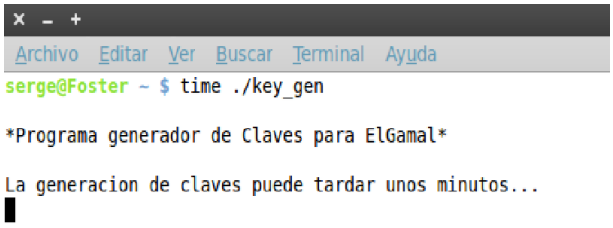



Figura 4.1: Programa de generación de claves.

En este caso el programa se corrió en una máquina de procesador intel core 2 Duo y 2 GB de memoria RAM con sistema operativo base linux (Kernel 2.6.38) y después de 50.931 segundos el resultado aparece dando los valores para las claves públicas y privada.

$\gamma$ = 2npq+1 = 14069322016356400095905982417322691184399895021674914
569735910541846138946264971341262482929487785984462379667370719 23
236568260736241523400161833251597404169998264918407141436379280 46
379207712003228492121697190698612339549097076094680501530836119 46
497561782445800038489157860603082561075669386324129404 7

donde

$2n = 242 = (2)(11^2)$

p = 936507182978869145991330955634918625904060108129554974522802 93
792056607247091796479451110527458595551045616874539889336453195 028
3926312791015049003309

q = 620792828278565926051432973242238288549222205330866472386538 14
069453544009557961050224219620761611370336537472759803998749359 673
6133365858073993148707,

$\alpha$ = 294703980101393555159096563709327278503384394336380380748702 62
570529149140698099887068312055330120398243235793706586307201466 317
058782708678398061375564545596120828024939343217202395333006224 559
231648

$\beta$ = 859438104591488815598950509641947663013478855896803214685277 841
554942498766233696878516660892085715256125430933687142837547408 747
983532182582924370749535585624017061642351521360805103190835385 228
685960292391057931732647698626023451015679503214498519548060250 479
986634445614987243304807492704972120233 44.

Y la clave privada a tal que $\beta = \alpha^{a} \bmod n$

a = 988317833511805243976333812157696358499303603593842521221665 137
609201052126219776171802354167592298880123242924357861057630092 863

219506056647199422686120148196506779215327985214996950289338111548
78709.

El programa de generación de claves no es interactivo con el usuario, las claves son generadas aleatoria y automáticamente por el programa lo que proporciona más seguridad. Los tiempos de ejecución son muy variados gracias a la generación aleatoria de números grandes y lo tardado que puede resultar encontrar primos a partir de ellos, también debemos considerar los tiempos al efectuar los cálculos. En el peor de los casos cuando el programa fue corrido 200 veces el tiempo el tiempo de ejecución fue de 5 minutos y 47.7 segundos y el mejor tiempo registrado fue de 0.45 segundos, un rango muy aceptable en comparación del tiempo que le toma al sofware libre GPG hacer la misma generación de claves. Recordemos que las claves las podemos generar una sola vez y ocuparlas indefinidamente, pero para mayor seguridad recomendamos que se cambien cada vez que se quiera transmitir información o al menos periodicamente.

Una vez conseguidas la claves se publícan los números $\gamma, \alpha, \beta$ y lo siguiente que hará la persona que quiera enviarnos una imagen es cifrarla con Triple-DES, para esto ocuparemos el programa diseñado en la tesis [20]. Quien va cifrar la imagen selecciona dos claves $\mathsf{k}_1, \mathsf{k}_2$ de 64 bits que serán números menores o iguales que $2^{64} - 1$, y que en base decimal son del orden de $10^{20}$ por tanto pertenecen a $(\mathbb{Z}/\mathbb{Z}_\gamma)^*$ por que $\gamma$ es un número del orden de $10^{300}$. Supongamos para nuestro ejemplo que esta vez se tratan de las claves $k_1 = 1334567896700771$, $k_2 = 1334567891291072$ y la imagen de la figura 4.2.

Figura 4.2: Imagen considerada como texto plano.

El programa para cifrar imágenes ofrece las opciones de *Cifrar, Descifrar, Guardar, Salir* adicionalmente también de cifrar con una permutación inicial variable en el proceso de DES, para cifrar oprimimos el boton del mismo nombre y navegando entre carpetas elegimos el archivo que corresponde a la imagen deseada, el programa al final de 70 segundos, un poco más lento que lo que se documenta en [28] genera la imagen cifrada y muestra el histograma correspondiente, la imagen puede ser almacenada si oprimimos el boton guardar y seleccionamos un destino para ello.

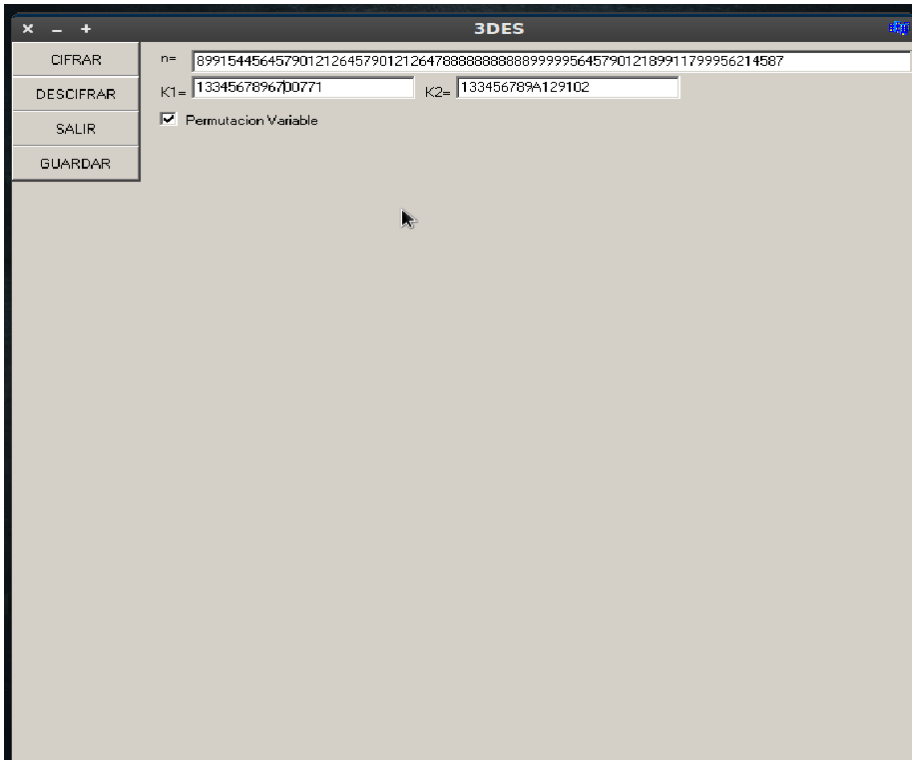



Figura 4.3: Programa para cifrado de imágenes.

La imagen cifrada por el programa apartir de la imagen de la figura 4.2 es la que se muestra en la figura 4.4.

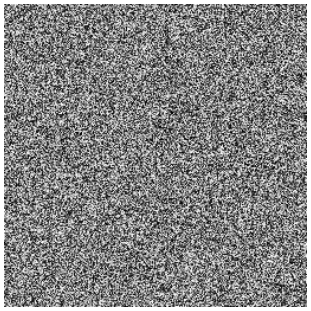


Figura 4.4: Imagen cifrada.

La calidad en la difusión de los piexeles según los histogramas que muestra el programa son parecidos a los que se registran en [28], el histograma es casi constante lo que asegura poca relación entre pixeles.

La instrucción en turno es cifrar las claves $k_1$ y $k_2$ mediante ElGamal, el programa de cifrado de ElGamal nos pedirá las claves públicas y el texto plano, podemos hacer el cifrado de las claves por separado o juntas, para este ejemplo cifraremos las claves de forma separada, la pantalla mostrada en la figura 4.5 es la que corresponde al programa para cifrar con ElGamal.

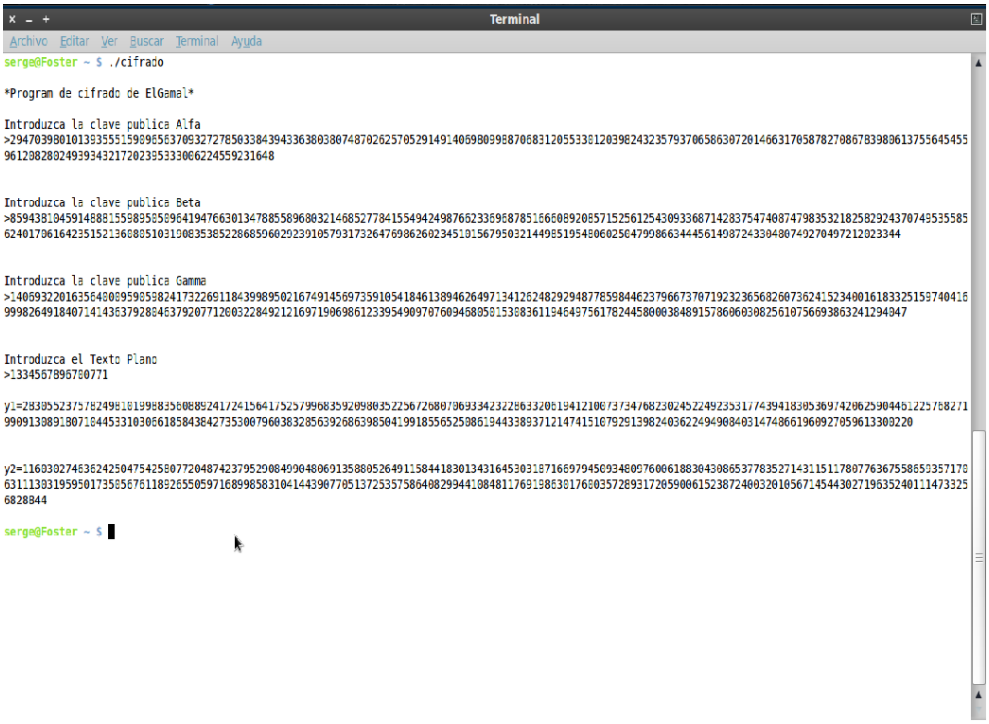



Figura 4.5: Programa de cifrado ElGamal.

los textos cifrados se obtienen en menos de 1 segundo y para nuestras claves antes generadas son los siguientes.

$y_1$ = 28305523757824981019988356088924172415641752579968359209803522567268070693342322863320619412100737347682302452249235317743941830536974206259044612257682719909130891807104453310306618584384273530079603832856392686398504199185565250861944338371214741516792913982403622494908403147486619609270596133002 20,

$y_2$ = 11603027463624250475425807720487423795290849904806913588052649115844183013431645303187166979450934809760061883043086537783527143115117807763675585503571786311330195950173505676118926550597168998583104144390770513725357586408299441084811769198638176803572893172059006152387240032010567145443027196352401114733256828844

30176003572893172059006152387240032010567145443027196352401114
733256828844

A continuación quien va a recibir la imagen hace el descifrado de las claves con el programa adecuado que pedirá el texto cifrado $y_1, y_2$, la clave pública $\gamma$ y la clave privada a que sólo debe conocer el receptor de la imagen.

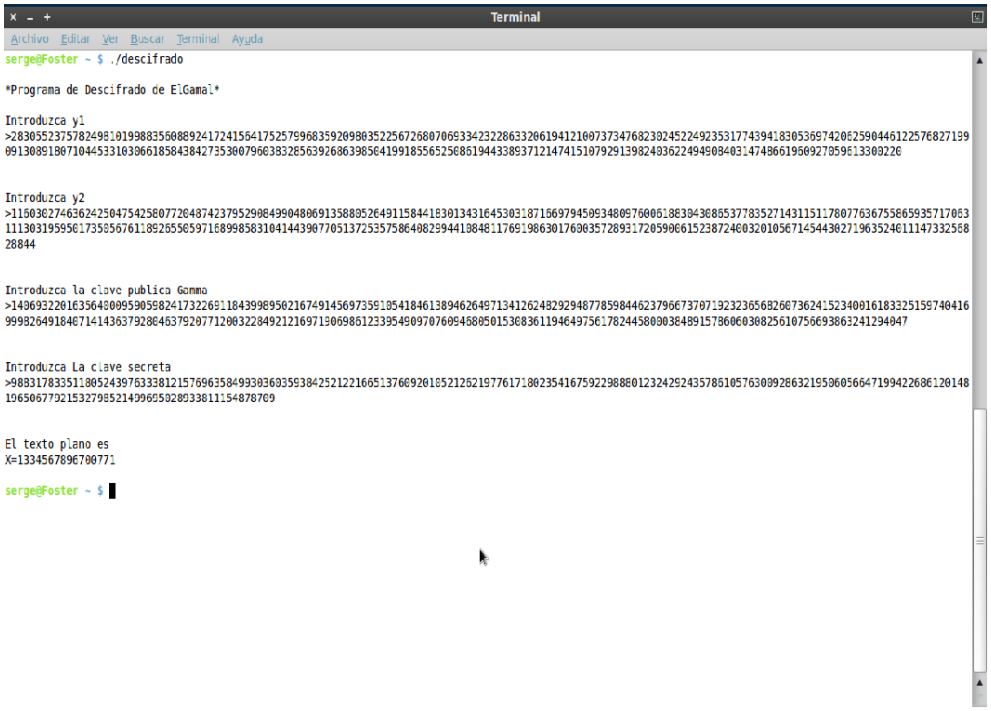



Figura 4.6: Programa de descifrado ElGamal.

cuando se han descifrado ambas claves, lo único que resta por hacer es descifrar las claves, así que ejecutamos el programa para imágenes y veremos de nuevo la pantalla que muestra la figura 4.3, colocamos las claves en las cajas de texto contiguas a las etiquetas $k_1, k_2$ y oprimimos el botón *Descifrar* seleccionamos el archivo de la imagen cifrada y tendremos de nueva cuenta la imagen original, cumpliendo con nuestro objetivo.

# Capítulo 5

# Conclusiones

A lo largo de este trabajo hemos podido adentrarnos en el mundo de la criptografía y estudiar fundamentos teóricos y modos de operación de ElGamal y Triple-DES, dandonos cuenta de que acoplarlos en un criptosistema híbrido no resulta en una mala idea sino todo lo contrario, se logra un criptosistema eficiente y seguro. Triple-DES es capaz de cifrar textos largos en tiempos cortos mientras ElGamal siendo un poco más lento en su generación de claves pero rápido en sus procesos de cifrado y descifrado resuelve el problema de propagación de las claves para Triple-DES.

Más especificamente podemos decir que la implementación de un criptosistema híbrido para cifrar imágenes en formato bmp con Triple-DES y luego cifrar sus claves con ElGamal para propagarlas de forma segura es un método bastante funcional y seguro para transmitir imágenes digitales por canales inseguros como lo puede ser el internet, además de no ser tan complicado de progamar en lenguaje c++ con ayuda de la biblioteca GMP.

El programa de cómputo que lleva a cabo la generación de claves de ElGamal cumple con niveles de seguridad altos al trabajar con claves de $2^{1024}$ bits o más, adherido a esto su despeño está dentro de los tiempos esperados pues compite con los softwares comerciales y de licencia GNU que existen. Por otra parte el programa para cifrar la imágenes de Triple-DES mostró resultados de seguridad buenos pues la correlación de los pixeles en la imagen cifrada es casi nula y los tiempos de ejecución son del orden de segundos.

## 5.1. Trabajo a futuro

Hay mucho por estudiar aún en la criptografía y el cifrado de imágenes, la mayoría de estos estudios con el fin de que se diseñen nuevos criptosistemas o bien mejorar los ya existentes haciendolos cada vez más díficiles de vulnerar y más rápidos en sus tiempos de operación.

Las siguientes propuestas son interesantes.

- Mantener vigente ElGamal acorde al orden del número primo más grande encontrado.
- Diseñar un criptosistema híbrido para cifrar imágenes basado en ElGamal y AES.
- Diseñar un criptosistema híbrido para cifrar imágenes basado en un algoritmo de curva elíptica y triple-DES.
- Diseñar un algoritmo para comprimir y cifrar imágenes digitales.

Sería muy productivo en cada una de las propuestas poder paralelizar los algoritmos e intentar tomar caminos hacia criptografía cuántica.

# Referencias


[1] I.N. Herstein, ***Álgebra moderna: grupos, anillos, campos, teoría de Galois***. 2a Edición, Trillas, 1990.

[2] J.A. Beachy, W.D. ***Abstrac Algebra*** Second Edition, Blair Waveland Press, Inc, 1996.

[3] F. A. González de la Hoz, ***Teoría de números entrega 2: congruencias***, disponible en http://www.opencontent.org/openpub/).

[4] F. A. González de la Hoz, ***Teoría de números entrega 3: La funcion Zeta de Riemann Teorema de los números primos***, disponible en http://www.opencontent.org/openpub/).

[5] G.H. Hardy, E.M. Wright ***An introdution to the theory of numbers*** Fourth edition, Oxford UniversityPress, Ely House, London W, 1968.

[6] M. Habib, C. McDiarmind J. Ramirez-Alfonsin, B. Reed ***Probalistic methots for algorithmic discrete mathematics*** Springer Verlan, 2003.

[7] D.M. Burton ***Elementary number theory*** Sixth edition, MacGrawHill, 2007.

[8] R.P. Grimaldi, ***Matemáticas discretas***, Tercera edición, Pearson, 1998.

[9] D.R. Stinson, ***Cryptography: Theory and practice***, CRC Press, 1995.

[10] N. Smart, ***Crytography: An introduction*** Tercera edición, Cambrige University Press, 2001.

[11] M. Stamp, ***Information security: principle and practice***, Wiley-Interscience, 2005.

[12] A. Menezes, P. van Oorschot, S. Vanstone, ***Handbook of applied cryptography***, CRC Press, 1997

[13] D. Hankerson A. Menezes S. Vanstone ***Guide to eliptic curve cryptography***, Springer, 2004.

[14] Koblitz ***Algebraic aspects of cryptography.*** Springer, 1998.

[15] W. Diffie, M. Hellman, ***New Directions in cryptography***,IEEE Transaction on information theory, IT-22 (1976), 472-492.

[16] T. ElGamal, ***A public key cryptosystem based and a signature scheme based on discret logarithms***, Hewlett-Packard Labs 1501 Page Mill Rd Palo Alto CA 94301

[17] FIPS PUB 46-3 FEDERAL INFORMATION PROCESSING STANDARS PUBLICATION, Reaffirmed 1999 october 25.

[18] M. Salleh, S. Ibrahim, I. F. Isnin, ***Image encryption based in chaotic mapping***, Journal Teknology 39(D) Dis (2003), 1-12.

[19] B. Jahn, ***Digital image processing*** Fifth edition, Springer, 2002.

[20] E. López González, ***Esquema de encriptación de imágenes usando 3-DES de permutación variable***, Tesis de maestría, CIDETEC-IPN, 2010.

[21] ***Mnual de referencia The GNU Multiple Precision Arithmetic Library***, Edition 5.0.2, 2011

[22] W. Zhang, Z. Zhu, H. Yu, ***An image encryption scheme based on chaotic maps***, International workshop on Chaos-Fractals theories and applidations, Northeastern University, Shenyang, Liaoning, China, 2009.

[23] J.Peng, S. Jin, X. Liao, ***A novel digital image encryption algorithm based on hyperchaos by controlling lorenz system***, Fifth International conference on natural computation, Chongquing University of science and technology, Departament of computer science and engeneering, China, 2009.

[24] L. Jinqiu, S Xica, ***Image encryption algorithm based on hyperchaotic system***, International workshop on Chaos-Fractals theories and applidations, Information and comucation engineering college, Harbin Engineering University, 2009.

[25] L. Zhang, J. Wu, N. Zhou, ***Image encryption with discrete fractional cosine transform and chaos.***, 50° International conference on information assurance and security, Departament of engineering, Nanchang University, China, 2009.

[26] V. Assche ***Quantum cryptography ans secret key distillation***, Cambrige University Press 2006.

[27] M. Baig ***Criptografía cuántica***, Facultad de ciencias, Universidad autónoma de Barcelona, 2007.

[28] K. Gupta, S. Silakari ***Efficient hybrid image cryptosystem using ECC and chaotic map***, International Journal of Computer Applications, Volumen 29, No. 3, 2011.