INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE FÍSICA Y MATEMÁTICAS

Criptografía visual

TESIS

que, para obtener el título de

Ingeniero Matemático,

presenta

Luis Omar Barbosa García

Asesor:
Dr. Luis Carlos Coronado García

México, D.F., 2 de julio de 2008

Dedicatoria...

El presente trabajo está dedicado a mis padres, Jesús y María Elena,
quienes me dieron la vida y gracias a ellos el día de hoy
concluyo una parte de mi formación académica.
Gracias...

Reconocimiento

Agradezco a mis padres por darme una carrera profesional.

Agradezco a mis hermanos Jair, Daniel y Melanie por compartir siempre esos buenos y malos momentos.

A mis amigos de la ESFM, Brody, Martínez, Fred, Tizoc, Antonio, Oscar, Uriel, Paloma, Gabriela, Gustavo, Rogelio, Christian, Edgar, Jorge, Axinia, Luis, Paulo, y otros tantos que no aparecen pero hicieron que mi estancia en la escuela fuera más agradable y llena de sentimientos encontrados.

A los profesores que me impartieron catedra a lo largo de la ingeniería.

Agradezco a mis asesores de Tesis, el Dr. Carlos Coronado y a su esposa la M. en C. Veronica Bolaños, quienes me tuvieron paciencia y apoyado a lo largo de este tiempo para llevar acabo este trabajo.

Introducción

La presente tesis tiene como objetivo dar a conocer lo que es la criptografía, particularmente la **criptografía visual** y sus diferentes aplicaciones, puede considerarse como una extensión nueva considerando que la criptografía tiene muchísimos años de vida.

Cada vez es más común escuchar la palabra criptografía debido al crecimiento tecnológico que han tenido los recursos a través de la internet tales como transacciones bancarias, acceso a contenidos privados (correo electrónico, cuentas bancarias, etc.), en el sentido de que los datos confidenciales no se quiere que sean vistos o modificados en la red, esto a su vez trae consigo necesidades para el uso como la confiabilidad, eficacia y seguridad, y la criptografía nos proporciona las técnicas para satisfacer dichas necesidades.

La tesis consiste de 6 capítulos: introducción a la criptografía; identificación y confirmación de usuario; imágenes como auxiliares; interfaz para la generación de imágenes compartidas; aplicaciones prácticas; conclusiones.

En el primer capítulo como su nombre lo dice es la introducción a la criptografía de forma general, objetivos, nociones de seguridad, identificación, secretos compartidos y complejidad de algoritmos; en el segundo se detalla los métodos para la identificación y/o confirmación: frases de acceso; firma digital; continuamos con secretos compartidos; conocimiento cero.

Es importante mencionar que la criptografía visual emplea imágenes para ocultar la información contenida en éstas, asimismo en el capítulo 3 se presenta la confirmación, compartición de un secreto (2 ó más personas), identificación de un usuario todo esto usando imágenes solamente.

Posteriormente en el capítulo 4 podemos encontrar un sencillo tutorial para el

manejo de una implementación desarrollada en lenguaje java crea por nosotros, llamada “Cifrador de imágenes CV” o “CICV”, y se incluye en un disco anexo junto con la máquina virtual de java. En el siguiente capítulo se presentan escenarios de uso de la criptografía visual (autenticación, confirmación e identificación) y finalizamos el trabajo con las conclusiones en el capítulo 6.

Índice general

Índice de figuras

Capítulo 1

Introducción a la criptografía

A través de la historia el hombre se ha interesado por mantener en secreto la información cuando trata de comunicarse, es decir, no se quiere que ésta sea interceptada por extraños en el transcurso del envío y además leída o modificada.

Así es como el hombre ha diseñado métodos o técnicas para ocultar la información con el propósito de mantenerla segura. A lo largo del tiempo se fueron desarrollando y mejorando cada vez más estás técnicas para ocultar (cifrar) la información, un método antiguo y muy conocido es el *Cifrado César* usado por los romanos en la antigüedad, el cual consistía en reemplazar cada letra del abecedario por la que se encuentra tres posiciones hacia la derecha de ésta, hablando formalmente los romanos usaban la operación suma modúlo N, donde N es el número de letras del abecedario. Entonces ellos podían cifrar un texto usando la fórmula $C \equiv P + 3 \mod 26$, esto significa que puede reemplazarse cada letra del texto original, por la letra que se encuentra 3 posiciones después de ésta, por ejemplo podemos cambiar $X \mapsto A$, $Y \mapsto B$ y $Z \mapsto C$.

A esta ciencia o arte se le conoce como **criptografía** esta palabra proviene del griego *"kryptós"*, ocultar y *"graphos"*, escribir; que proporciona las técnicas y métodos matemáticos para mantener segura la información de entidades que no deben tener acceso a ésta.

Actualmente ha habido un gran avance en la ciencia de la computación y en otras áreas, y esto a su vez trae consigo mayores necesidades de mantener segura la información, ya no sólo cuando tratamos de comunicarnos sino también en nuestra vida cotidiana; por ejemplo, al realizar una operación bancaria por internet; tal vez nunca nos hemos preguntado sí es un medio "seguro" para realizarla, el simple hecho de proporcionar datos personales, ya que si estos son vistos por entidades ajenas

(adversarios), éstos a su vez podrían usar dicha información sin darnos cuenta, y considerando en el peor de los escenarios, perjudicarnos. Así es como la criptografía nos proporciona las herramientas para resolver este tipo de problemas y algunos otros.

Antes de continuar, es necesario introducir algunos conceptos.

- $\mathcal{A}$ denota un conjunto finito de símbolos llamado *alfabeto*, por ejemplo éste puede ser el abecedario que consta de 26 símbolos y una cadena es secuencia de símbolos $s = s_1, s_2, \ldots, s_n$; y en este caso los s_i pueden ser números binarios que forman una cadena de tamaño n, en cuyo caso $\mathcal{A} = \{0, 1\}$.
- $\mathcal{M}$ es un conjunto finito conocido como espacio de mensajes. $\mathcal{M}$ consiste de cadenas de símbolos o caracteres de un alfabeto, $\mathcal{A}$. También es conocido como **texto llano** o información original.
- $\mathcal{C}$ es un conjunto finito, que se le conoce como espacio de texto cifrado. $\mathcal{C}$ consiste de cadenas y símbolos de un alfabeto, éste no necesariamente es el mismo que el de $\mathcal{M}$.
- $\mathcal{K}$ es un espacio de claves. Un elemento k de este conjunto es conocido como clave, donde esta información es una secuencia de símbolos (números o letras).
- Cada elemento $e \in \mathcal{K}$ determina una biyección de $\mathcal{M}$ hacia $\mathcal{C}$, denotado por E_e, es decir, $E_e : \mathcal{M} \to \mathcal{C}$, es conocido como transformación de cifrado.
- Analogamente para cada $d \in \mathcal{K}$, D_d denota también una biyección de $\mathcal{C}$ hacia $\mathcal{M}$, $D_d : \mathcal{C} \to \mathcal{M}$. D_d es una transformación de descifrado.

Un esquema de cifrado consiste de lo siguiente: transformaciones de cifrado, $\{E_e : e \in \mathcal{K}\}$; de descifrado, $\{D_d : d \in \mathcal{K}\}$, con la propiedad de que para cada $e \in \mathcal{K}$ hay únicamente una clave $d \in \mathcal{K}$ tal que $D_d = E_e^{-1}$, esto es $D_d(E_e(m)) = m$ para todo $m \in \mathcal{M}$. Así es como un esquema de cifrado nos permite esconder la información original de los adversarios que no deben tener acceso a ésta.

¿Por qué se usan claves?, ¿qué significado tienen?, como se menciono hay dos claves e y d, e es necesaria para cifrar el mensaje m usando la transformación E y d para descifrar c usando la transformación correspondiente D, es decir, se usa para poder recuperar el mensaje cifrado. Qué pasa cuando $e = d$, en este caso se usa la misma clave para cifrar y descifrar los mensajes, los esquemas criptográficos que se caracterizan por usar sólo una clave son conocidos como **simétricos** o como **criptografía simétrica**; supongamos que dos entidades, Alan (A) y Bety (B)

desean comunicarse de forma "segura", entonces lo que necesitan hacer es ponerse de acuerdo sobre que método simétrico y qué clave van a usar; y así evitar que los adversarios, digamos una tercera entidad, Claudia (C) obtenga alguna información que sólo le corresponde a A y a B. Por ejemplo, escogen una clave e_1, y ahora sí Alan usa e_1 para cifrar el mensaje y después lo puede enviar a Bety, una vez que lo recibe ella lo descifra usando e_1 también. La seguridad de estos esquemas está en la clave, por eso es muy importante que sea muy difícil adivinar qué clave es.

Veamos el otro caso, cuando $e \neq d$ los esquemas criptográficos que usan dos claves, la clave e para cifrar, mejor conocida como *clave pública* y d para descifrar llamada *clave privada*, a estos esquemas se les conocen como **asimétricos** o **criptografía asimétrica**. La criptografía asimétrica surge con el fin de evitar el problema del intercambio de claves de los esquemas simétricos; sí por ejemplo Claudia llegase a conocer la clave que usa Alan y Bety ya no sería seguro el esquema que usan; entonces tendrían que cambiar hacia uno asimétrico. Ahora Alan usa la clave pública de Bety para cifrar el mensaje y después enviarselo, aquí ya no es será necesario que ellos se pongan de acuerdo sobre la clave a usar. Y Bety con su clave privada podrá recuperar el mensaje original; la claves usadas en este esquema le pertenecen solamente a Bety. La seguridad de éste radica en que ella mantenga la clave privada fuera del alcance de Claudia.

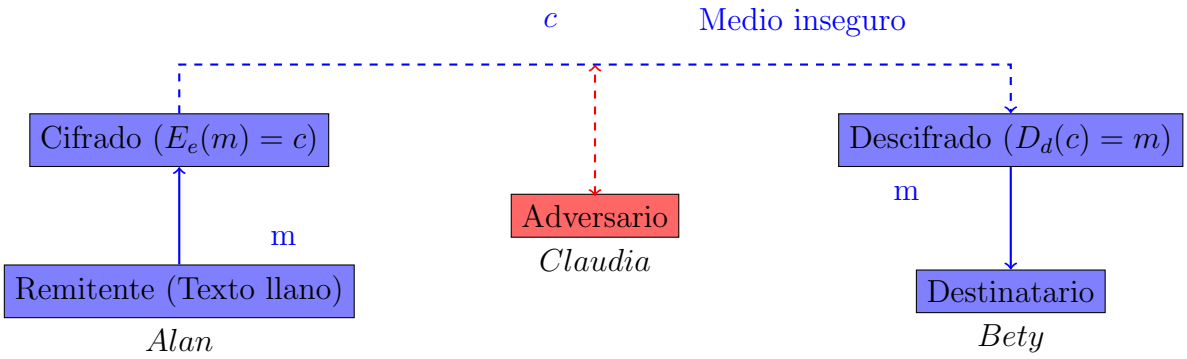

Figura 1.1: Esquema de la comunicación entre dos entidades con texto cifrado.

1.1. Objetivos de la criptografía

Al hablar de criptografía, el concepto de *información* toma mayor relevancia, ya que lo que pretendemos al usar algún protocolo criptográfico es procurar que la información se mantega segura ante entidades desconocidas, o simplemente fuera del alcance de terceros.

La seguridad de la información se manifiesta por sí misma de varias maneras según la situación y/o necesidad. Así es como en los esquemas de comunicación existen amenazas tales como la interrupción, intercepción, generación o modificación de la información que se transmite o contra la identidad de los participantes.

Para contrarrestar estas amenazas la criptografía se han planteado algunos objetivos que se presentan en el cuadro 1.1. Los más relevantes se presentan a continuación:

- Confidencialidad. La información sólo puede ser leída por las partes autorizadas, es decir, que tengan el derecho a usarla. Por ejemplo, cuando recibimos un correo eléctronico, es muy díficil saber si este ha sido visto por otras entidades, ya que no se tiene control del medio de comunicación.
- Integridad de los datos. Se refiere a que la información no haya sido alterada en el transcurso del envío. Regresando al ejemplo anterior, supongamos que el contenido del mensaje ha sido alterado, y nosotros confiamos en el contenido de éste, y entonces realizamos alguna acción con dicha información, y después nos damos cuenta de que ésta es falsa.
- Autenticación. Verificar que el mensaje que haya sido enviado es de quién dice ser. Es muy común recibir correos electrónicos que aparenten ser de nuestros contactos, y la verdad no lo son. Estos correos tienen el propósito de obtener cierta información confindencial; nuestros gustos, hábitos de consumo por ejemplo, con el fin de vendernos algo y en el peor de los casos perjudicarnos (*phishing*) por eso es muy importante tener la certeza de saber quién es el remitente.
- No repudio del origen de los datos. Nos permite asegurar que cualquier parte que envía o recibe información no pueda rechazar ante terceros que la envió o recibió.

Confidencialidad	Mantener la información en secreto y solo autorizar a ciertas entidades a verla
Integridad de datos	Asegurar que la información no ha sido alterada por entidades no autorizadas
Autenticación	Corroborar la identidad de una entidad
Autenticación de mensajes	Identificación del origen de los datos
Firmas	Es un mecanismo para ligar información a una entidad
Autorización	Permitir a una entidad realizar algo sobre información o recursos
Validación	Proporcionar periodos de autorización para utilizar o manipular información o recursos
Control de Acceso	Dar privilegios de acceso a recursos restringidos a ciertas entidades
Certificación	Endoso de información acerca de algo o alguien por una entidad en la cual se confía
Estampas de Tiempo	Registro de la fecha de creación o de la existencia de la información
Atestiguar	Verificar la creación o la existencia de la información por una entidad que no es la creadora de dicha información
Recibo	Reconocimiento de que se ha recibido la información
Confirmación	Reconocimiento de que los servicios se han proporcionado
Propiedad	Proveer a una entidad el derecho legal de utilizar o de transferir un recurso a otras entidades
Anonimato	Encubrir la identidad de una entidad implicada en algunos procesos
No repudio	Prevención de la negación de una entidad sobre acciones realizadas
Revocación	Anular la certificación o la autorización a una entidad o documento

Cuadro 1.1: Objetivos de la Criptografía.

1.2. Nociones de seguridad criptográfica

¿Qué entendemos por seguridad?, según [Esp03], es cualidad de lo seguro; pero ¿qué entendemos realmente por seguridad criptográfica?, digamos que nos indica que un texto cifrado, una clave privada y el acceso hacia algún recurso esté libre de peligro, daño o riesgo, es decir que este fuera del alcance de los adversarios y se logren los objetivos planteados en la sección anterior.

El auge de las tecnologías de la comunicación y, por ejemplo las relacionadas a internet, en nuestros días se ha desarrollado un nuevo concepto de trabajo, creando con ello herramientas que facilitan trabajar a través de la red. Por eso se hace cada vez más necesario, sobre todo en algunas operaciones (tales como transferencias bancarias, consulta de expedientes, todas las que se refieran a circulación de datos confidenciales por su naturaleza), garantizar al usuario una seguridad razonable. Uno podría pensar que al usar un esquema criptográfico para comunicarse o para realizar cualquier otra acción lo está haciendo de una forma segura, pero ningún sistema o esquema es 100 % seguro como lo veremos a continuación.

La seguridad de algunos esquemas criptográficos se basan en el funcionamiento del algoritmo que usan, es decir, su seguridad radica en guardar en secreto la forma en que éste funciona, a estos algoritmos se les conoce como *algoritmos restringidos*, pero sin mucho uso ya éstos no pueden ser estandarizados, ya que cada usuario debería tener su propio algoritmo.

Hay otros algoritmos en donde la seguridad radica en el uso de claves, esto quiere decir que el algoritmo puede ser público y analizado. No importa si un adversario conoce como funciona éste, si no conoce la clave particular no podrá tener acceso hacia algún recurso o vulnerar su seguridad.

Se dice que un criptosistema tiene seguridad incondicional cuando, sin importar el poder de cómputo, este no puede ser roto. Se dice que tiene seguridad computacional cuando con recursos limitados de cómputo el criptosistema no puede ser roto.

1.2.1. Ataques sobre los esquemas de cifrado

El propósito de éstas acciones es vulnerar los textos cifrados y recuperar el texto original y en el peor de los casos deducir la clave de descifrado. Veamos cuales son estos ataques, y que se le permite hacer al adversario.

- **Ataque sólo sobre la clave**, este ataque consiste en que el adversario conoce únicamente la clave pública de $\mathcal{E}$
- **Ataque con mensajes**, aquí se permite que el adversario examine algunos textos cifrados con sus mensajes, ya sean sólo conocidos o escogidos. Hay diferentes tipos de ataques con mensaje:

  1. *Ataque sobre el texto cifrado*, es donde el adversario trata de deducir la clave de descifrado o el texto original con sólo observar el texto cifrado, cualquier esquema vulnerado con este ataque es considerado completamente inseguro.
  2. *Ataques con mensajes conocidos*, el adversario tiene acceso a los textos cifrados de un conjunto de mensajes $m_1 \dots m_t$ los cuales conoce, más no los escoge.
  3. *Ataque genérico con mensajes escogidos*, al adversario se le permite escoger textos cifrados válidos $\mathcal{E}$ de una lista de mensajes $m_1 \dots m_t$ antes que él intente romper el esquema de texto cifrado.
  4. *Ataque dirigido con mensajes escogidos*, es parecido al ataque genérico con mensajes escogidos, excepto que la lista de mensajes cifrados puede crearse después de conocer la clave pública de $\mathcal{E}$ pero antes de producirse cualquier texto cifrado. Este ataque es dirigido en contra de un usuario $\mathcal{E}$ en particular.
  5. *Ataque adaptativo con mensajes escogidos*, al adversario se le permite utilizar al usuario $\mathcal{E}$ como un óraculo; no sólo se puede solicitar textos cifrados de $\mathcal{E}$ de mensajes que dependan de la clave pública de $\mathcal{E}$, sino también de mensajes que adicionalmente dependan de textos cifrados previamente obtenidos.

Llamaremos **ruptura total** a la obtención de información secreta de $\mathcal{E}$ y **falsificación existencial**, a falsificar una firma de al menos un mensaje. El adversario no tiene control sobre el mensaje de la firma que obtiene.

1.3. Identificación entre partes

En esta sección consideraremos técnicas que permiten a una entidad conocida como *verificador* asegurarse de que la identidad de otro conocido como *demandante* es cierta, y de este modo prevenir la suplantación. El propósito de la identificación entre partes en esta tesis será asegurarnos de que no compartiremos o distribuiremos información condifencial con entidades ajenas; esto es que alguien que pretende imitar a una entidad no logre obtener información alguna.

La identificación entre partes es el proceso a través del cual una de las partes, el verificador, se asegura de que la identidad (corroborando cierta evidencia) de una segunda parte, el demandante, implicada en el protocolo es cierta. Para establecer un protocolo de identificación, generalmente implica a las siguientes entidades el demandante A y el verificador B, el verificador cuenta de antemano con la supuesta identidad de A, es decir, proporciona la autenticación de entidad.

Así como hay métodos para mantener oculta (cifrado) la información, también existen técnicas que permiten al verificador tener la certeza de que la identidad del demandante es quien dice ser y con ello se previene de imitaciones. En este capítulo sólo se mencionan algunas técnicas a usar como son *esquemas de frases de acceso*, *protocolos de desafío-respuesta* y *conocimiento cero*, que éstas son realizadas por el verificador con el objetivo de revisar la veracidad de cierta información (posiblemente en respuesta de un mensaje previo enviado por él) en el cual demuestra que el demandante posee información genuina.

Objetivos de los protocolos de identificación

Desde el punto de vista del verificador, el resultado de un protocolo de autenticación puede ser cualquiera, es decir, aceptar la identidad del demandante como auténtica o rechazarla. Asimismo se plantea los siguientes objetivos:

- En el caso de entidades honestas A y B, A es capaz de autenticarse con éxito por sí misma con B, es decir B completará el protocolo aceptando la identidad de A.
- B no puede reutilizar un intercambio de identificación con A con éxito imitando a A una tercera parte C.
- *Imitación.* La probabilidad de que cualquier entidad C distinta de A lleve a cabo

el protocolo y tome el papel de A es insignificante, porque B puede completar y aceptar la identidad de A.

Las técnicas de autenticación se pueden dividir en tres categorías, dependiendo de la seguridad en la cual se basa su esquema es la siguiente:

1. Algo conocido, el ejemplo más común es el número de identificación personal (NIP) y las claves secretas o privadas cuyo conocimiento es demostrado en los protocolos de desafío y respuesta.

2. Posesión de algo, un accesorio físico, como lo es el chip de una tarjeta de crédito que nos sirve para confirmar alguna operación, con información contenida en este.

3. Algo inherente, esta categoria incluye métodos los cuales hacen uso de las características físicas del ser humano, tales como firma autógrafa, voz y huella digital, estas son conocidas como técnicas no criptográficas (biometría).

1.4. Secretos compartidos

En esta sección hablaremos brevemente lo que es la técnica de secretos compartidos, como funciona y para que nos sirve. El propósito de usar esta técnica es que considero que un secreto (información confidencial) está mejor guardado si se distribuye entre varias personas, y ésta nos ayuda a llevar a cabo dicha tarea.

En un banco, como sabemos hay una bóveda la cual debe ser abierta cada día. Los directivos del banco emplean a tres cajeros para hacerlo, pero a ellos no les agrada la idea de que alguno de los cajeros tenga la combinación, por lo tanto a los directivos les gustaría diseñar un sistema a través del cual dos de los tres cajeros puedan tener acceso a la bovéda, de tal manera que no lo puedan hacer individualmente.

Para poder realizar este proceso, nos guiaremos en un método llamado esquema umbral (secretos compartidos), este esquema requiere una cantidad mínima de componentes para llevarse a cabo; entonces el esquema umbral será la pareja ordena $(t, w), t \leq w$, donde $t, w \in \mathbb{N}$; es un método para compartir, digamos cierta información K entre un conjunto de participantes P de tamaño w, de tal manera que cualquier grupo de t participantes puede obtener K, pero ningún grupo menor a t

pueda hacerlo.

La compartición de la información K es llevada a cabo por una entidad conocida como *director*. A lo largo de la tesis lo denotaremos por D y asumimos que $D \notin P$, es decir, el director no participa. Cuando D desea compartir la información K entre los participantes en P, él da a cada participante un pedazo de información llamada *parte*. La manera en que éstas se distribuyen debe ser secreto dado ningún participante debe saber la parte que le fue dada a otro.

Podemos decir que está resuelto el problema de los directivos con ayuda de este método para compartir la combinación de la bóveda. El ejemplo anterior es un esquema umbral $(2, 3)$. Recordando un artículo publicado en la revista *Time Magazine*[1], acerca del control de armamento nuclear ruso que involucra un esquema "2 de 3" también, de acceso muy similar al previo. Donde las tres partes involucradas son el Presidente, el Ministro de Defensa y el Ministerio de Defensa.

1.5. Complejidad de Algoritmos

Hasta esta parte hemos usado en varias ocasiones la palabra *algoritmo* sin dar detalles de lo que es, entonces un algoritmo es una serie de instrucciones finita y ordenada para realizar una tarea, así un algoritmo debe proporcionar el mismo resultado si repetimos el valor (o valores) de entrada. El análisis de un algoritmo consiste en estimar la cantidad de recursos (tiempo, memoria, comunicación) que éste requiere para cualquier tamaño de entrada. En ese sentido se considera el tiempo de ejecución de un algoritmo como una función f, del número de operaciones elementales que realiza éste para una entrada de tamaño n, $f(n) : \mathbb{N} \longrightarrow \mathbb{R}^+$.

Cabe mencionar que los algoritmos se clasifican de la siguiente manera:

- **Límite superior asintótico:** Sean $f, g : \mathbb{N} \to \mathbb{R}^+$, se dice $f(n) = O(g(n))$ si existe una constante positiva c y un número entero positivo n_0 tales que $0 \leq f(n) \leq cg(n)\ \forall n \geq n_0$.

  De manera informal se dice que f no crece más rápido que g, n_0 nos indica un número muy grande, y que solamente toma en cuenta como son f y g para

[1] Time Magazine, Mayo 4, 1992, pag.13

valores de n mayores que n_0.

- **Límite inferior asintótico:** Se dice $f(n) = \Omega(g(n))$ si existe una constante positiva c y un número entero positivo n_0 tales que $0 \leq cg(n) \leq f(n)\ \forall n \geq n_0$. Si tenemos $f(n) = 0{,}5n^3 = \Omega(n^2)$, entonces para $n > n_0 = 8$ y $c = 4$ se cumple.

- **Límite exacto asintótico:** $f(n) = \Theta(g(n))$ si existen dos constantes positivas c_1, c_2 y un número positivo n_0 tales que $c_1 g(n) \leq f(n) \leq c_2 g(n)\ \forall n \geq n_0$. Es decir, $f = O(g)$ y $f = \Omega(g)$, se dice en este caso que f es Θ de g. Veamos un ejemplo:

  $f(n) = \frac{n^2}{2} - 3n = \Theta(n^2)$ si existen c_1 y c_2

  $$c_1 n^2 \leqslant \frac{n^2}{2} - 3n \leqslant c_2 n^2,$$

  $$c_1 \leqslant \frac{1}{2} - \frac{3}{n} \leqslant c_2.$$

  Lo cual es válido para $n > n_0 = 7$, si $c_1 = \frac{1}{4}$ y $c_2 = \frac{1}{2}$.

- **Notación** o: Si $f(n)$ y $g(n)$ son dos funciones positivas para $n \geq n_0$ y sí

  $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$

  En este caso "$f = o(g)$" , significa que $f(n)$ es mucho más pequeña que $g(n)$ cuando n es muy grande. Lo que nos esta diciendo es que f crece mucho más lento que g ($f << g$).

  Si

  $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = k(cte),$$

  entonces tenemos $f = O(g(n))$, por lo que o nos proporciona información más precisa que O. Por ejemplo $2n = o(n^2)$.

Proposición Sean f, g: $\mathbb{N}_0 \to \mathbb{R}^+$.

- Si f es un polinomio de grado d en n, entonces $f(n) = O(n^d)$. Por ejemplo, $f(n) = n^2 + n = O(n^2)$.

- $O(kf(n)) = O(f(n))$.
- Si $g(n) \leq f(n)$, entonces $O(f(n)) + O(g(n)) = O(f(n) + g(n)) = O(f(n))$.
- $f(n)O(g(n)) = O(f(n)g(n))$.
- $O(\lg(f(n))) = O(\lg(n))$, si $f(n)$ es un polinomio en n.
- $\lg(n) = O(n^{\epsilon})$ $\forall \epsilon > 0$, $\epsilon \in \mathbb{R}^{+}$.

Una vez que contamos con algunos de los conceptos acerca de la complejidad de algoritmos veamos como nos podemos apoyar en éstos con un ejemplo que se verá más adelante. Como se ha mencionado podemos medir la cantidad de operaciones que realiza un algoritmo en su ejecución, por decirlo de alguna manera; ya que la forma en que podemos hacer esto es encontrando una función f, positiva que nos indique aproximadamente el número de operaciones que realiza éste para llevarse a cabo con cierto valor de entrada y así clasificarlo.

Consideremos el siguiente problema, ordenar una cantidad n de números. Para poder hacer esto contamos de antemano con 2 algoritmos que fueron diseñados para realizar dicha tarea, estos algoritmos son: *Quicksort* y *Bubblesort*. Ahora la pregunta sería: qué algoritmo vamos a usar o por qué usar uno en lugar del otro, entre otras preguntas que nos podemos plantear. Para responder estas preguntas hay que tener en cuenta la complejidad de cada algoritmo, es decir, el tiempo de ejecución de cada uno.

El algoritmo Quicksort nos proporciona una técnica para ordenar una secuencia de números, basado en el enfoque "divide y vencerás", es decir, podemos ver que éste consiste de dos partes:

- Dividir. Dada una cantidad n de elementos, consideramos a éstos como un conjunto A, de tamaño n. Dicho conjunto se dividirá en dos subconjuntos $A_1 = [p, \ldots, q]$ y $A_2 = [q + 1, \ldots, n]$ donde todo elemento de A_1 será mayor a todo elemento de A_2.
- Vencer. Una vez que ya está dividido A en dos partes, A_1 y A_2, estós subconjuntos son ordenados recursivamente.

Algoritmo Quicksort

```
Quicksort(A,p,n)
 si p < r entonces
 q= particion (A,p,n)
Quicksort(A,p,q)
Quicksort(A,q+1,n)
```

Donde el algoritmo para la partición es el siguiente:

Algoritmo Partición

```
Particion(A,p,n)
x=A[p]
i=p-1, j=r+1
mientras sea verdadero
repetir j=j+1 until A[j] <= x
repetir i=i+1 until A[j] >= x
si i < j entonces
cambiar (A[i],A[j])
devolver j
```

La forma en que este algoritmo funciona es la siguiente: es escoger de manera pseudoaleatoria un elemento x, del conjunto A, que por ejemplo pudiese ser un elemento del intervalo A[p] a A[n], dicho elemento es conocido como pivote y éste se toma como referencia para dividir al conjunto en dos subconjuntos, que no necesariamente deben tener el mismo número de elementos, ahora se ordenan todos los elementos menores a x a un lado del pivote, dejando a su lado izquierdo éstos y todos los elementos mayores o iguales a x se pasarían al lado derecho; en está primera parte del proceso x pasa a ocupar exactamente la posición que le correspondería en el conjunto ordenado.

Este algortimo es recursivo y en ese sentido se tiene que escoger varias veces elementos que tomarán el papel de pivote, es decir, después de la primera elección del pivote y el reordenamiento correspondiente tenemos dos subconjuntos. Ahora se vuelve a escoger un elemento x_1 ($x_1 \neq x$), que será el nuevo pivote de alguno de los subconjuntos y repetir el reordenamiento de los elementos menores a x_1 de un lado y

los mayores o iguales a esté nuevo pivote en el otro lado. Entonces este procedimiento se realiza varias veces mientras haya más de un número por cada subconjunto que vaya dejando el pivote correspondiente. Una vez terminado dicho proceso el conjunto estará ordenado.

Veamos ahora la complejidad de este algoritmo; supongamos que el número de elementos es potencia de 2, $n = 2^k$, entonces $log_2(n) = k$, donde k es el número de divisiones que realizará el algoritmo.

En la primera fase por decirlo de alguna manera el algoritmo habrá hecho n comparaciones, en la segunda el algoritmo creará dos subconjuntos aproximadamente de tamaño $\frac{n}{2}$, así que el número total de comparaciones hechas en estos dos subconjuntos es: $2(\frac{n}{2}) = n$. En la tercera fase el algoritmo tendrá 4 subconjuntos más por lo tanto el número de comparaciones en está fase es $4(\frac{n}{4}) = n$. Por consiguiente, el número de comparaciones que realiza el algoritmo es $n + n + \cdots + n = kn$, donde $k = log_2(n)$, entonces este algoritmo tiene una complejidad $O(nlog_2(n))$. Para más detalles ver [Vel98].

Ahora veamos como funciona el algoritmo **Bubblesort** (burbuja), a diferencia del anterior el algoritmo de la burbuja funciona comparando cada elemento del conjunto que va a ser ordenado con el siguiente, intercambiándolos de posición si es que se encuentran en el orden equivocado. El nombre de este algoritmo se debe a la forma con la que son ordenados los elementos durante los intercambios porque parece que fuesen burbujas subiendo; que también es conocido como **método del intercambio directo**.

Algoritmo Bubblesort

```
Para i=1 hasta n-1 hacer
Para j=n hasta i+1 hacer
si A[j]<A[j-1] entonces
temp=A[j-1]
A[j-1]=A[j]
A[j]=temp
termina
termina
```

Al igual que en el algoritmo anterior se necesita un arreglo A de n números como entrada, dichos elementos serán ordenados en forma ascendente. Empieza comparan-

do el último elemento del arreglo $A[n]$, con el anterior, $A[n-1]$. Si $A[n] < A[n-1]$, se intercambian los valores guardados en $A[n-1]$ y $A[n]$, y en caso contrario tendremos $A[n-1] < A[n]$. Después comparamos $A[n-1]$ con su predecesor inmediato, $A[n-2]$; si $A[n-1] < A[n-2]$, los intercambiamos y se continua con el proceso. Después de haber hecho $n-1$ comparaciones de este tipo, el número más pequeño de la lista está en la posición $A[1]$. Después repetimos este proceso para los $[n-1]$ números guardados en el arreglo $A[2], A[3], \ldots, A[n]$; así cada vez que se realiza este proceso, el elemento más pequeño de la sublista restante "sube" (como una burbuja) hasta el frente de esa sublista.

Así que para poder determinar la complejidad de este algortimo, en una entrada (arreglo) de tamaño $n \geq 1$, contamos el total de comparaciones realizadas para ordenar los n números dados en forma ascendente. Ahora contemos el número de comparaciones; en caso de que el arreglo tenga un sólo elemento el algoritmo no realiza ninguna comparación. Si a_n, denota el número de comparaciones necesarias para ordenar n números, entonces podemos usar la siguiente relación de recurrencia:

$$a_n = a_{n-1} + (n-1),$$

para $n \geq 2$ y $a_1 = 0$.

$$\begin{array}{lll} a_1 = & 0 & \\ a_2 = & a_1 + (2-1) & = 1 \\ a_3 = & a_2 + (3-1) & = 2 \\ a_4 = & a_3 + (4-1) & = 3 \\ \ldots & \ldots & \ldots \end{array}$$

Después la suma de lo anterior es:

$$a_n = 1 + 2 + \cdots + (n-1) = \frac{n(n-1)}{2} = \frac{n^2 - n}{2} = f(n).$$

Por lo tanto el tiempo de ejecución de este algoritmo es $f \in O(n^2)$, es decir, que este algoritmo requiere $O(n^2)$ comparaciones para llevarse a cabo.

Ya que tenemos las complejidades de ambos algoritmos, ahora sí podemos responder la pregunta: ¿Qué algoritmo debemos usar?, para esto es necesario comparar dichas complejidades, que ambas se encuentran en la categoría $O(.)$, de los límites superiores asintóticos, es decir, $nlog_2(n) \leq n^2$ basta con ver que para todo $n > 0$, n^2 siempre es mayor que $nlog_2(n)$, entonces

$$O(nlog_2(n)) + O(n^2) = O(nlog_2(n) + n^2) = O(n^2).$$

Por lo tanto decidimos usar el algoritmo Quicksort para ordenar una serie de números porque es el tiene un menor tiempo de ejecución.

Definición 1.1. *La ejecución de un algoritmo se dice que es de* ***Tiempo polinomial****, si existe $n \in \mathbb{N}$ tal que el número de operaciones bit requerido para llevarse a cabo con una entrada de longitud de a lo más k es $O(n^k)$.*

Definición 1.2. *Se dice que un algoritmo es de* ***Tiempo exponencial*** *si existe $c \in \mathbb{R}^+$ tal que el número de operaciones bit requerido para ejecutar el algoritmo en una entrada de longitud a lo más k es $O(e^{ck})$.*

Problemas P, NP y NP Completo

Se considera el término "problema" como la descripción general de una tarea, y el término "instancia" de un problema como un caso particular de la tarea.

Al resolver un problema obtenemos una respuesta, es decir, dicho resultado lo podemos ver en dos clases, digamos, una en donde la respuesta puede ser **"SI"** o **"NO"**, los problemas que nos arrojen ésta clase de resultados son conocidos como **Problemas de decisión**, por ejemplo un problema de decisión es la pregunta: ¿Es un número entero dado primo? y en la otra están los resultados que van más allá de un si o un no, es decir, el resultado de una instancia es más específico, como lo sería el determinar si el **17** es un número primo, problemas (instancias) cuya respuestas sean de este tipo los llamaremos **Problemas de búsqueda**.

Clase P

Definición 1.3. *Un problema de decisión $\mathcal{P}$ está en la clase P de problemas que pueden ser resueltos en un tiempo polinomial, si existe un polinomio $p(n)$ y un algoritmo tal que si una instancia de $\mathcal{P}$ tiene una entrada de longitud de al menos n, entonces el algoritmo responde correctamente en un tiempo de al menos $p(n)$.*

Algunos problemas que podemos encontrar en está clase son algunos programas lineales, el algoritmo para encontrar el máximo común divisor de dos números, ordenamiento y multiplicación de matrices.

Clase NP

Definición 1.4. *Un problema de decisión $\mathcal{P}$ está en la clase NP si dada una instancia de $\mathcal{P}$, una persona con poder de cómputo ilimitado no solo puede resolver la pregunta sino en el caso de que sea un "si", ésta puede dar evidencia a otra persona que podría usarla para verificar la correctitud de la respuesta en tiempo polinomial.*

Estos problemas "intratables" pueden caracterizarse por el curioso hecho de que puede aplicarse un algoritmo polinómico para comprobar si una posible solución es válida o no. Un ejemplo de un problema NP es de las *torres de Hanói* en el que se tiene n anillos y tres postes A, B y C. Se requiere pasar los n anillos del poste A al B usando el poste C como auxiliar, siguiendo las siguientes reglas:

- Los anillos se mueven uno a la vez.
- Nunca puede estar un anillo encima de uno más pequeño.

Algoritmo de las torres de Hanói (n,A,B,C)

```
Si n=1 mover anillo de A a B
Ademas
hanoi(n-1,A,C,B)
mover anillo de A a B
hanoi(n-1,C,B,A)
```

Este algoritmo recursivo tiene una complejidad de $O(2^n)$ porque al pasar los n discos de A a B, se requieren $2^n - 1$ movimientos, entonces la función que representa su complejidad es $f(n) = 2^n - 1$ y por lo tanto $f \in O(2^n)$.

Reducción de un problema a otro

Definición 1.5. *Sean $\mathcal{P}_1$ y $\mathcal{P}_2$ dos problemas de decisión. Se dice que $\mathcal{P}_1$ se reduce a $\mathcal{P}_2$, si existe un algoritmo que es de tiempo polinomial como una función de la entrada de $\mathcal{P}_1$ y que dada cualquier instancia P_1 de $\mathcal{P}_1$, se construye una instancia P_2 de $\mathcal{P}_2$ tal que la respuesta para P_1 es la misma respuesta para P_2.*

Supongamos que tenemos un algoritmo eficiente para $\mathcal{P}_2$. Si $\mathcal{P}_1$ se reduce a $\mathcal{P}_2$, entonces podemos utilizar el algoritmo de $\mathcal{P}_2$ para resolver a $\mathcal{P}_1$ también. Dado una instancia p_1 de $\mathcal{P}_1$, en tiempo polinomial se encuentra una instacia correspondiente p_2 de $\mathcal{P}_2$, y usando la definición 1.5. Entonces si se aplica el de $\mathcal{P}_2$ a p_2, la respuesta obtenida será también la respuesta a la pregunta $\mathcal{P}_1$, es decir, teniendo un algoritmo para $\mathcal{P}_2$ nos da un algoritmo para $\mathcal{P}_1$, y si el algoritmo de $\mathcal{P}_2$ es de tiempo polinomial, lo será para $\mathcal{P}_1$ también.

Clase NP-Completo

Definición 1.6. *Un problema de decisión $\mathcal{P}$ esta en NP, se dice ser* ***NP-Completo*** *si cualquier otro problema $\mathcal{Q} \in NP$ puede reducirse a $\mathcal{P}$ en tiempo polinomial.*

Si tuvieramos un algoritmo de tiempo polinomial para un problema $\mathcal{P}$ NP-Completo, entonces también tendríamos algoritmos de tiempo polinomial para los demás problemas $\mathcal{Q} \in NP$. Esto significaría que P=NP, y que la conjetura de que P≠NP sería falsa, por esta razón no es probable que alguien de un algoritmo de tiempo polinomial para cualquier problema NP-Completo. En ese sentido los problemas NP-Completo son los más difíciles de la clase NP.

Conclusiones

En resumen, es importante distinguir entre las diferentes clases de problemas que hay, es decir, un problema fácil de uno difícil, en el sentido de que uno "fácil" (instancia), puede ser resuelto en un tiempo razonable (polinómico), considerando el tamaño de entrada, por ejemplo la multiplicación de matrices, en este caso una instancia es la multiplicación de matrices de tamaño 3×3 cuyo tiempo de ejecución es menor en comparación a la ejecución del mismo algoritmo pero con una entrada de tamaño 10×10. Sin embargo, este problema se considera fácil porque se conoce un algoritmo que resuelve las instancias de éste, y por lo tanto está en la clase P.

Consideraremos un problema difícil aquel que no puede ser resuelto en tiempo polinomial debido a que no se conoce un algoritmo eficiente y en algunos casos el que se conoce para resolver una instancia de éste es ineficiente y por tal motivo se encuentran en la clase NP, podemos ver esto observando el algoritmo recursivo de las torres de Hanoi, cuya ejecución es en tiempo exponencial; primero considerando que tenemos 3 discos, cuya ejecución se lleva a cabo en 1min 10s, ahora si pensamos en 64 discos, el tiempo de ejecución es aproximadamente de $2^{64} - 1 \approx 5 \times 10^{12}$ años, que esto no puede ser considerado como un tiempo razonable, aunque es importante notar que 64 discos no es una gran número como para pensarse que tarde tanto.

Capítulo 2

Identificación y/o confirmación de usuario

Los problemas de identificación y de autenticación son cada vez más comunes dada la cantidad de servicios que se ofrecen a través de internet y de forma física en donde éstos requieren cierta información para tener acceso hacia algún recurso; por ejemplo lo que sería la autenticación e identificación del propietario de una tarjeta de crédito, dichos problemas son tratados con técnicas criptográficas que en este capítulo veremos algunas de ellas: los protocolos de *desafío-respuesta*, las *frases de acceso*, *firma digital*, *secretos compartidos* (continuando el tema después de la introducción en 1.4) y por último *conocimiento cero*. El usar estas técnicas en la tesis nos ayudará a tener la certeza de saber con quién estamos intercambiando información confidencial, y así evitar posibles suplantaciones o engaños por parte de los adversarios. El objetivo de este capítulo es dar a conocer el uso de las técnicas que se implementaran.

Regularmente la tarjeta bancaria es utilizada en los cajeros automáticos con el propósito de disponer de cierta cantidad de dinero. Que para poder llevar a cabo dicha operación bancaria requiere de dos partes que son indispensables que son la tarjeta bancaria y el número de identificación personal (NIP); dicho protocolo de autenticación necesita de la tarjeta como medio de acceso (log in) hacia la cuenta bancaria y el otro elemento necesario es el NIP que sirve para completar el proceso de autenticación; ya que si el NIP es erróneo o tecleado incorrectamente el sistema del cajero automático nos negará el acceso y por lo tanto no podremos disponer del efectivo. Es así como la criptografía nos ayuda a dar certidumbre a los usuarios de los servicios electrónicos entre otros.

2.1. Frases de acceso

Una de las medidas de seguridad para evitar el acceso de personas malintencionadas a un sistema o lugar físico es autenticar a los usuarios que desean ingresar a dichos espacios, como lo mencionamos en la sección 1.3, hay tres tipos de autenticación, digamos, uno, por lo que se tiene; dos, por lo que se sabe; y tres, por lo que se es.

En esta sección veremos las frases de acceso, que es una técnica que como su nombre lo dice permite el acceso al usuario por lo que se sabe. Actualmente hacemos uso de ésta técnica de autenticación mediante las contraseñas (passwords) para ingresar hacia algún recurso. Como sabemos la contraseña consta de simbolos, que pueden ser números y letras que regularmente son cadenas cortas por ejemplo de 6 ó 8 símbolos, pueden ser más. Sin embargo una frase de acceso es mucho más larga que una contraseña, porque en algunas ocasiones está formada por varias palabras, ya que esto es lo que proporciona una mayor seguridad, es decir, es más difícil adivinar una frase que una contraseña, por la cantidad de operaciones que uno debe realizar para poder hacerlo.

Dada la cantidad de servicios que existen en la red en la mayoría de éstos se tiene la necesidad de convertirse en usuario de uno que otro servicio, y esto a su vez nos lleva a crear una cuenta para cada uno de ellos, lo que implica crear contraseñas para hacer uso de los recursos que ofrecen. Ahora imaginemos la cantidad de contraseñas que deberíamos crear; además de que es demasiada información que alguien tendría que guardar, en ocasiones es muy complicado recordar la información confidencial de cada servicio. Todo esto ha desarrollado el concepto de contraseña a frase de acceso; aunque al final del día tienen el mismo propósito, que es autenticar a una entidad.

Antes de continuar es necesario introducir la notación y conceptos preliminares.

Se escribe $M \xleftarrow{\$} \mathcal{S}$ para el experimento de escoger un elemento de una distribución $\mathcal{S}$ y se le llama M. Cuando $\mathcal{S}$ es un conjunto finito dado de una distribución uniforme. La concatenación de cadenas M y M' es denotado por $M\|M'$ o MM'. Si $M = M_1 \dots M_m \in \{0,1\}^m$ es una cadena de m-bits y $1 \leq a \leq b \leq m$, se escribe como $M[a,b]$ para $M_a \dots M_b$. El complemento de la cadena de bits M, se denota por $\overline{M}$.Una cadena vacía se denota por ϵ y por último cuando a sea un número se denotará $\langle a \rangle_r$ para su representación en forma de cadena de r-bits de longitud.

Una función hash o de resumen $H : \mathcal{M} \to \mathcal{Y}$, donde $\mathcal{Y}$ es un conjunto finito no vacío. Además $\mathcal{M}$ y $\mathcal{Y}$ son conjuntos de cadenas tales que $\mathcal{Y} = \{0,1\}^n$ y $\mathcal{M} = \{0,1\}^m$ para algún $n >0$, el número n es llamado la longitud hash de H, con $m > n$.

Definición 2.1. ***Función Resistente a Preimágenes***

Sea $H : \mathcal{M} \to \mathcal{Y}$ una función Hash y sea m un número tal que $\{0,1\}^m \subseteq \mathcal{M}$. Y sea A un adversario, entonces se define lo siguiente:

$$Avd_H^{Pre[m]}(A) = P[M \leftarrow \{0,1\}^m; Y \leftarrow H(M); M' \leftarrow A(Y) : H(M') = Y] < \epsilon$$

en un tiempo t.

Se considera que la ventaja del adversario A, es la probabilidad que dado M en el dominio de H, A pueda encontrar un $M' \in \mathcal{M}$ en tiempo de a lo más t, tal que $H(M) = H(M')$, es decir que la imagen sea la misma para ambas preimágenes. Entonces se espera que ϵ sea muy pequeño.

Definición 2.2. ***Resistente a Colisiones***

Sea $H : \mathcal{M} \to \mathcal{Y}$ una función Hash y sea A un adversario, entonces se define lo siguiente:

$$Avd_H^{Coll} = P[(M, M') \leftarrow A : (M \neq M') \wedge (H(M) = H(M'))] < \epsilon$$

en un tiempo t.

Nuevamente la ventaja del adversario A, es la probabilidad que éste pueda encontrar dos elementos M y M' distintos que tengan la misma imagen, $H(M) = H(M')$; dándole a éste un tiempo t para poder hacerlo.

Un sistema de frases de acceso funciona de la siguiente manera. Sea $f(x)$ una función de un sólo sentido, tal que $x \mapsto y = f(x)$, que dicha función es fácil de calcular pero computacionalmente imposible de invertir. Las frases de acceso de los usuarios son valores x, en el dominio de $f(x)$. Para mantener la lista de frases de acceso fuera de las manos de los intrusos (*hackers*), la computadora no almacena los passwords x. Mejor dicho, bajo cada nombre del usuario se almacena el valor $f(x)$, esto se obtiene de la aplicación de la función f a su frase de acceso x. Cada vez que alguien desea identificarse, teclea su frase de acceso x, entonces la computadora

calcula $f(x)$, si coincide con la $f(x)$ almacenada se le otorga el acceso, y después borra cualquier registro de x.

2.2. Firma digital

Como sabemos la firma es algo físico que se adjunta a un documento para ligar el contenido de éste con la persona que lo firmo. La firma se usa en muchas situaciones cotidianas como firmar cartas, un cheque o un contrato, etc. Y también existe la versión digital, la **firma digital** es un método criptográfico que nos permite garantizar la integridad de un documento, es decir, queremos que el documento ha ser firmado digitalmente se pueda ligar a una entidad mediante el uso de éste método. Sin embargo la firma digital no es algo físico que se pueda adjuntarse al mensaje. Así la firma digital debe tener propiedades análogas a la firma autógrafa tales como la verificación ante terceros, no repudación de documentos y autenticación de identidad entre otros. Pero cómo podemos verificar una firma digital; a diferencia de la autógrafa la cual podemos verificar con sólo comparar dos firmas, por ejemplo cuando realizamos una compra con la tarjeta de crédito, el vendedor nos pide que firmemos un comprobante donde estamos aceptando el cargo por dicha compra, y él compara la firma del comprobante con la firma que está detrás de la tarjeta. Que por supuesto este no es un método muy seguro, porque es relativamente fácil falsificar la firma de alguien. Y en el otro caso, la firma digital puede ser verificada haciendo uso de un algoritmo público de verificación, entonces cualquier persona puede verificar la firma digital. El objetivo que se pretende alcanzar con el uso de este esquema es prevenir posibles suplantaciones al momento de autenticar a una entidad al compartir cierta información.

Además la firma digital cumple lo siguiente:

- Debe verificar al autor que creó la firma.
- Autenticar el contenido de la información que va a ser firmada.
- Debe ser verificada por terceras personas para resolver posibles conflictos.

Un esquema de firma digital (figura 2.1) consiste de tres procedimientos, *generación de claves*, *firmado* y *verificación*. Antes de detallar éstos es necesario introducir la notación correspondiente.

$\mathcal{M}$ es un conjunto de elementos llamado espacio de *mensajes* (texto llano).

$\mathcal{M}_s$ es un conjunto de elementos llamado espacio de *mensajes firmados*.

$\mathcal{S}$ es un conjunto de elementos conocido como espacio de *firmas*.

Esquema de firma digital

- **Generación de claves (Gen).** En este primer paso se crean dos claves la pública (PK) y la privada (SK) de manera segura, que podemos decir que no hay una forma general para generar éstas sino que depende del esquema de firma que uno desee emplear.

$$\{PK, SK\} \leftarrow Gen(1^n).$$

- **Firmado (Sig).** Es un algoritmo que necesita de un mensaje $m \in \mathcal{M}$ el cual se va a firmar y de la clave secreta (SK) para poder obtener la firma σ.

$$\sigma \leftarrow Sig(m, SK).$$

- **Verificación (Ver).** Una vez que tenemos la firma σ, la usamos con la clave pública PK y el mensaje m para confirmar si la firma es válida, es decir, 0 cuando la firma sea falsa y 1 en caso de que sea válida.

$$\{0, 1\} \leftarrow Ver(m, \sigma, PK).$$

Podemos ver los esquemas de firmado en dos categorías:

1. **Esquemas de firma digital con apéndice**, este esquema requiere del mensaje original como entrada para el procedimiento de verificación. Entre los esquemas que usan el mensaje para verificar la firma son ElGamal y DSA entre otros.

   Esquema de firma ElGamal

   Es un esquema basado en la complejidad del cálculo del logaritmo discreto. Este esquema fue descrito por Taher ElGamal en 1984. Éste a su vez permite que el verificador pueda confirmar la autenticidad de un mensaje m enviado por el emisor sobre un canal inseguro.

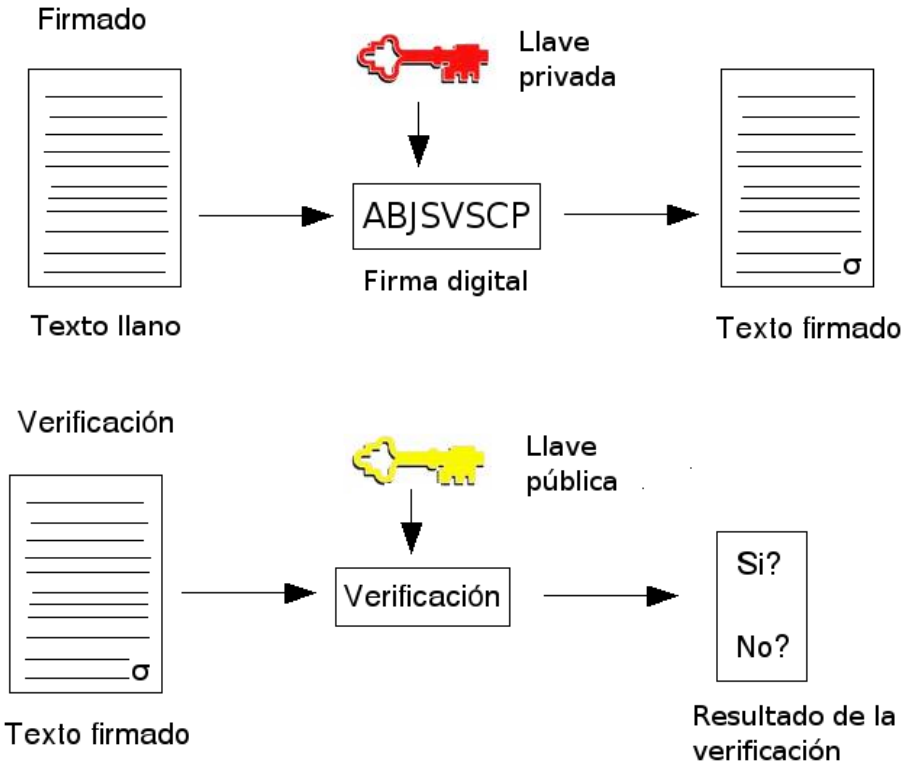

Figura 2.1: Esquema de firma digital.

El procedimiento de generación de claves consiste en tomar un número primo p muy grande, con un generador de números pseudoaleatorio escogemos g en el grupo multiplicativo $\mathbb{Z}_p^*$, y escoger aleatoriamente un número x que está entre $1 < x < p-1$. Después se calcula $y = g^x(\mod p)$. La clave pública será la terna (p, g, y) y la privada será x.

Para firmar el mensaje m, el firmante realiza lo siguiente: se escoge un número aleatoriamente k tal que cumpla $0 < k < p-1$ y $\text{mcd}(k, p-1) = 1$. Se calcula $r \equiv g^k(\mod p)$ y $s \equiv (H(m) - xr)k^{-1}(\mod p-1)$, donde H es una función de resumen. Si s es cero se reinicia el proceso. El par (r, s) es la firma del mensaje m.

El procedimiento de verificación requiere del par (r, s) y del mensaje m que proviene de un firmante con clave pública (p, g, y), si se cumplen las siguientes condiciones el verificador acepta el mensaje:

- $0 < r < p$
- $0 < s < p-1$
- $g^{H(m)} \equiv y^r r^s(\mod p)$

2. **Esquemas de firma digital con recuperación del mensaje**, en este caso no se necesita el mensaje original como entrada del procedimiento de verication, es decir, como su nombre lo dice se recupera el mensaje de la misma firma.

Esquema de firma digital RSA

Es un esquema criptográfico asimétrico, porque su uso radica en una clave pública, asimismo puede ser usado para cifrar, por ejemplo algún mensaje y en nuestro caso para autenticar a una entidad mediante su firma digital. Dicho esquema fue desarrollado por Ron Rivest, Asi Shamir y Leonard Adleman en 1977, y el nombre del esquema es tomado de las iniciales de sus apellidos.

La seguridad del esquema RSA está en la dificultad de factorizar números demasiado grandes (números de por lo menos 200 dígitos o más), en factores primos; veamos porque se menciona que es difícil factorizar un número en primos, por ejemplo tomamos el número 486, vemos que es par entonces dividimos por 2 y resulta 243. Este ya no es par, identificamos que es múltiplo de 3 y el resultado es 81 que este se puede factorizar como 3^4; ahora podemos ver a $486 = 2 \times 3^5$

y listo, pero ¿Qué pasa cuando el número es 713? que no es demasiado grande en comparación al anterior, en este caso dicho número no es múltiplo de 2, ni de 3, ni de 5, ni de 7, ni de 11, ni de 13, ni de 17, ni de 19, sino el 23 y dividir 713 por éste resulta 31, entonces ya tenemos factorizado el 713, que esto nos llevo más tiempo para factorizarlo por el hecho de ir probando cada número primo. Ahora imaginemos cuánto tiempo llevaría factorizar un número de 200 dígitos o más, que el intentarlo ya no sería viable por la cantidad enorme de años que nos llevaría hacerlo y eso si lo logramos.

Veamos en que consiste este esquema:

Generación de claves

En este procedimiento se escogen aleatoriamente dos números primos demasiado grandes, digamos p y q; donde estos son distintos. Con estos formamos n como el producto de p y q ($n = pq$); después calculamos $\phi(n)$[1] $= (p-1)(q-1)$, ahora escogemos nuevamente al azar un número entre 1 y $\phi(n)$, que llamaremos e, tal que el máximo común divisor entre e y $\phi(n)$ sea 1. Después encontramos un número $d \in (1, \phi)$, tal que $ed \equiv 1(\mod \phi)$, una vez que hemos realizado este cálculo ya tenemos las claves, donde la pública es la pareja (n, e) y la privada es d.

Firmado

En esta parte del proceso, deseamos firmar un documento m, antes de firmarlo necesitamos resumirlo mediante una función hash H, es decir, $m_1 = H(m)$ y entonces el resumen m_1, es el que se va a firmar de la siguiente manera:

$$s = m_1^d \mod n,$$

donde s, es la firma que obtenemos.

Verificación

Para llevar a cabo la verificación de la firma s del mensaje m, se calcula lo siguiente:

$$m_1 = s^e \mod n,$$

como $ed \equiv 1 \mod \phi$, $s^e \equiv m_1^{ed} \equiv m_1 \mod n$.

Consideremos el siguiente ejemplo, aunque es sencillo nos sirve para entender mejor este esquema. Los parámetros son $p = 3$, $q = 5$, $n = 15$, $\phi(15) = 8$.

[1]Función $\phi(n)$ de Euler, ésta devuelve la cantidad de números que son primos relativos con n.

Ahora escogemos $e = 3$, entonces ahora buscamos d, tal que $ed \equiv \mod n$ se cumpla y así obtenemos $d = 3$, ya que $9 \equiv 1 \mod 8 = 1$.

Firmado, como ya se menciono en esta etapa se firma el mensaje ($m = 2$, en este sencillo ejemplo) de la siguiente manera $s = md \mod n$ con lo que se obtiene que $s = 8$ así es como el mensaje firmado es la pareja (m, s)=$(2, 8)$. La parte de la verificación de la firma se hace comprobando la siguiente congruencia $se \mod n$, $8^3 \mod 15 = 2$, donde se puede apreciar que también obtenemos un 2 como en la fórmula anterior.

2.3. Secretos compartidos

Veamos el esquema umbral (t, w) inventado por *Shamir* en el año 1979. El director D escoge w elementos de $\mathbb{Z}_p$ distintos de cero, denotado por x_i, $1 \leq i \leq w$, (donde $p \geq w+1$ y es primo). Para cada $1 \leq i \leq w$, D da el valor de x_i a cada participante, P_i. Estos valores x_i son públicos. Supongamos que D quiere compartir una clave $K \in \mathbb{Z}_p$. D secretamente escoge (aleatoriamente) $t-1$ elementos de $\mathbb{Z}_p$, $a_1, a_2, \ldots, a_{t-1}$. Ahora para cada $1 \leq i \leq w$, D calcula $y_i = a(x_i)$, donde

$$a(x) = K + \sum_{j=1}^{t-1} a_j x^j \mod p. \tag{2.1}$$

Después de calcular los y_i, D da a cada P_i el valor de la parte y_i, para $1 \leq i \leq w$.

Lo que esta haciendo el director es construir un polinomio aleatorio $a(x)$ de grado al menos $t-1$ en el cual la clave, K es el término constante. Y cada participante P_i obtiene un punto (x_i, y_i) de este polinomio.

Ahora los participantes $P_{i_1}, \ldots, P_{i_t}$ quieren recuperar la clave K, ellos saben que

$$y_{i_j} = a(x_{i_j}), \tag{2.2}$$

para $1 \leq i \leq t$, donde $a(x) \in \mathbb{Z}_p[x]$ es el (secreto) polinomio escogido por D porque $a(x)$ es al menos de grado $t-1$, $a(x)$ lo podemos escribir como

$$a(x) = a_0 + a_1 x_1 + \ldots + a_{t-1} x^{t-1}, \tag{2.3}$$

donde los coeficientes $a_0, \ldots, a_{t-1}$ son elementos desconocidos de $\mathbb{Z}_p$ y $a_0 = K$ que es la clave. Dado que contamos con los puntos (x_i, y_i) podemos encontrar los

coeficientes después de sustituir los puntos en la ecuación anterior; obtendremos t ecuaciones lineales, y si éstas son linealmente independientes, entonces tendremos solución única y descubriremos el valor de a_0, que es la clave que buscamos.

Sin pérdida de generalidad, veamos que ocurre con un ejemplo. Supongamos que $p = 17$, $t = 3$ y $w = 5$ y que las coordenadas públicas son x_i, $1 \leq i \leq t$. Y se escoge al grupo P_1, P_3, P_5 con sus partes del secreto, las cuales son 8, 10 y 11 respectivamente. Escribiendo el polinomio $a(x)$ como

$$a(x) = a_0 + a_1 x + a_2 x^2,$$

después de calcular $a(1), a(3)$ y $a(5)$, obtenemos las siguientes tres ecuaciones lineales en $\mathbb{Z}_{17}$:

$$\begin{array}{llll} a_0+ & a_1 & + a_2 & =8 \\ a_0+ & 3a_1 & + 9a_2 & =10 \\ a_0+ & 5a_1 & +25a_2 & =11. \end{array}$$

Este sistema de ecuaciones tiene solución única en $\mathbb{Z}_{17}$ y es $a_0 = 13$, $a_1 = 10$ y $a_2 = 2$, entonces la clave es $K = 13$.

En el siguiente capítulo veremos más a detalle los esquemas umbrales para compartir un secreto a diferencia del ejemplo de arriba lo que compartiremos será un imagen y no un número; ya que el propósito de la tesis es compartir un número o clave, que esté contenida en una imagen.

2.4. Conocimiento cero

En esta sección veremos lo que significa conocimiento cero. La implementación de este protocolo de identificación, es para convencer a alguien de que tenemos cierta información sin necesidad de darla a conocer.

"El rey se sentó en su trono, visiblemente preocupado y deseando no haber realizado nunca aquella apuesta. En un momento de debilidad, después de haber bebido más de la cuenta, su desmedida afición por el juego le había llevado a apostar con el monarca vecino que era capaz de demostrar cierto complicado teorema en menos de una semana. El mayor problema no era el hecho de que careciera de dicha demostración, puesto que en su apuesta no se decía que no pudiera recibir ayuda de los matemáticos de su país, sino que al proporcionársela al rey vecino, éste podría emplearla para construir cierto tipo de máquinas, que posteriormente utilizaría para declararle la guerra. Cuando el matemático más sabio del reino escuchó la historia se limitó a sonreir, con un brillo enigmático en la mirada... -Majestad -dijo-, podemos ganar la apuesta sin revelar la demostración. [2]"

Aunque nos parezca un poco loco o extraño, el matemático de esta pequeña historia tiene razón. Existe un mecanismo que nos permite demostrar que tenemos cierto conocimiento, sin necesidad de revelarlo, a esto se le llama Conocimiento-Cero.

Otro ejemplo de conocimiento cero, lo tenemos en el siguiente problema.

Sabemos que para resolver un polinomio es suficiente con encontrar sus raices. En la edad media no se conocía la solución del polinomio de tercer grado. La solución fue dada por Tartaglias. Tartaglias pretendiese convencer a los científicos de su época de que encontró la solución al problema sin dar a conocer la misma, lo que tenía que hacer Tartaglias era solicitar un polinomio de grado 3, del cual estos conociesen las raices, y el verificando las raices que ellos ya conocian. Repitiendo suficientes veces el proceso las personas quedaban convencidas de que encontró la solución sin necesidad de proporcionar la fórmula que encontró.

Así pues, se tiene que el conocimiento cero nos ayuda a identificar personas o información sin necesidad de dar a conocer la información. Es una prueba entre dos partes, llamados verificador y probador, en el cual el probador desea convencer al verificador de la validez de una aseveración dada. La interacción entre las partes

[2]Manuel Lucena, Profesor del Departamento de Informática de la Universidad de Jaén, wwwdi.ujaen.es/~mlucena/wiki/pmwiki.php

debería permitir al probador convencer al verificador de la validez de cualquier afirmación. Mientras que ninguna estrategia del probador puede engañar al verificador de aceptar una aseveración falsa.

Una Prueba de Conocimiento-Cero debe satisfacer las siguientes condiciones:

- Completo: Si la declaración es verdadera, el verificador honesto será convencido de este hecho por el probador honesto.
- Validez: Si la declaración es falsa ningún probador que mienta puede convencer al verificador honesto de que sea verdad, excepto con una probabilidad pequeña.
- Conocimiento-Cero: Si la declaración es verdadera, ningún verificador que engaña aprende cualquier cosa con excepción de este hecho.

Cuando se usan los sistemas de conocimiento-cero se desea convencer a otros de que conocemos cierto secreto, pero sin revelarlo.

Tales sistemas son idealmente adecuados como sistemas de identificación, en particular para la verificación de identidades de personas. Una persona A se puede identificar con otra persona B a través de la comprobación de cierto secreto. Se requiere que el sistema satisfaga lo siguiente:

- B no debe conocer de antemano el secreto de A, y
- durante el proceso no debe adivinar el secreto.

En este caso B no tendra la posibilidad de identificarse ante terceras personas como A.

El procedimiento de conocimiento-cero satisface esta condición de forma óptima: B puede convencerse, con el requerido nivel de seguridad, de la identidad de A, sin que en el camino obtenga *alguna* información; en particular no descubre el secreto. A esto se le conoce como **Propiedad del Conocimiento-cero**. Los protocolos de conocimiento-cero son, en la práctica extremadamente importantes (control de acceso electrónico), como teóricamente interesantes, ya que en ellos se utilizan métodos matemáticos no triviales y teorias complejas.

El algorimto Fiat-Shamir

El más conocido, y en la practica el más importante procedimiento de Conocimiento-cero es el algoritmo de Fiat-shamir. La seguridad de este algoritmo se basa en el hecho de que el la práctica es casi imposible calcular raices cuadradas en $\mathbb{Z}_n^*$.

El algoritmo de Fial-Shamir consta -como la mayoría de los algoritmos criptográficos- de dos fases, la fase de generación de clave y la fase de aplicación.

En la fase de *generación* A genera 2 números primos grandes p y q y forma el producto $n = pq$. El número n es público, mientras que p y q sólo los puede conocer A. Después A elige un número s y construye $v := s^2 \mod n$. El número s es el secreto individual de A, mientras que con la ayuda de v (la marca de identificación) se puede verificar si una persona lo conoce o no. Esto significa en particular que s debe permanecer en secreto, mientras que v debe ser público.

Uno se puede imaginar que n es una constante del sistema y que los números s y v debe ser dados, para cada parte, por una central.

En la fase de *aplicación* A debe convencer a B de que conoce el secreto s. Para ello A y B siguen el siguiente protocolo:

- A elige aleatoriamente un elemento r de $\mathbb{Z}_n^*$ y lo eleva al cuadrado módulo n: $x := r^2 \mod n$. Después A manda a B el valor x.
- B elige aleatoriamente un bit b y lo manda a A.
- Si $b = 0$ A manda el valor $y := r$ a B.
  Si $b = 1$ A manda el valor $y := rs \mod n$ a B.
- B verifica esta respuesta. En caso de que $b = 0$ comprueba que $y^2 \mod n = x$.
  En el caso $b = 1$ prueba el si la igualdad $y^2 \mod n = xv \mod n$.

También este protocolo cumple con las condiciones para los procedimientos de conocimiento-cero:

Operabilidad Cuando A conoce el secreto s podrá convencer a B, ya que en $\mathbb{Z}_n^*$ se cumple:

$$y^2 \equiv (rs)^2 \equiv r^2 s^{2b} \equiv r^2 v^b \equiv xv^b \mod n.$$

Corrección Un impostor, A', puede contestar a lo más una de las dos preguntas, $b = 0$ o $b = 1$.

Si se pudiese contestar ambas preguntas (con y_0 o con y_1), entonces se poseerá una de las raíces de v: de $y_0^2 = x$ y de $y_1^2 = xv$ se sigue $(\frac{y_1}{y_0})^2 = v$, y con ello es $(\frac{y_1}{y_0})$ una raíz cuadrada de v módulo n. Se puede entonces engañar en una ronda, adivinando la respuesta correcta con la probabilidad de $\frac{1}{2}$.

Por lo que A' puede adivinar la respuesta de la primera ronda con probabilidad de $\frac{1}{2}$ (echar un volado para ver si b es uno o cero). Cuando suponga que B la pregunta b hará podrá preparar la respuesta como: Cuando se tenga $x := r^2 v^{-b} \mod n$ y $y = r$, así B durante la verificación no podrá obtener ninguna irregularidad. Como arriba A' en una ronda t podrá convencer de que es A sólo con probabilidad $(\frac{1}{2})^t$.

Conocimiento-Cero: Un simulador M puede realizar un dialogo con B de la siguiente manera:

- M elije aleatoriamente un bit c y un número r; después calcula $x := r^2 v^{-c} \mod n$ y manda x a B.
- B responde con un bit b.
- si $b = c$, entonces M manda el mensaje $y = r$ a B. En este caso si la verificación es exitosa se tendrá:

$$xv^b \equiv r^2 v^{-c} v^b \equiv r^2 \equiv y^2 \mod n.$$

La terna (x, b, y) representa un paso de está simulación de plática.

Si b es distinto de c, entonces todos los mensajes enviados serán borrados y la simulación en está ronda debe empezar de nuevo.

Tanto en el dialogo original, como en el previo aparecen el mismo número de (aleatorios) ternas (x, b, y) para las cuales la igualdad $xv^b = y^2$ es válida. Ambos dialogos no pueden ser diferenciados por un observador. El esquema de *Fiat-Shamir* es especialmente bueno para la comprobación de identidades de personas correspondientemente.

Conclusiones

Las técnicas presentadas a lo largo de este capítulo nos proporcionan algunas soluciones al problema de la identificación de una entidad conocida como demandante, la ejecución de las mismas son llevadas a cabo por otra entidad llamada verificador, el caso más común de identificación por medio de frases de acceso es el ingresar a la cuenta de correo electrónico. En este caso el verificador cuenta de antemano con la supuesta identidad del demandante, es decir, tiene el registro del demandante (usuario y frase de acceso); después de que se ha ingresado el usuario es necesario ingresar la frase (contraseña) también. El verificador se encarga de confirmar que realmente sea la frase que permita el acceso al recurso (correos); en el caso de que ésta fuese incorrecta se le niega.

También es importante mencionar que la seguridad de los procotolos va a depender de la técnica que se emplee, y en ese sentido queremos utilizar la técnica de los secretos compartidos adecuando al uso de imágenes como medio de identificación (ver 3.1).

Capítulo 3

Imágenes como auxiliares

Como es bien sabido una imagen es la representación visual de un objeto mediante técnicas de diseño, pintura, fotografía, video entre otras. Y en ese sentido queremos apoyarnos de una imagen para representar una clave de acceso (números y letras) para éste fin. Entre los objetivos de la criptografía, podemos considerar el intercambio seguro de mensajes como uno, de tal manera que ninguna persona pueda tener acceso al contenido de éste; ahora qué pasa cuando dicho mensaje consiste de imágenes, cómo podemos cifrar las imágenes para que dicho mensaje sea ilegible ante entidades que no debiesen tener acceso a éste. Dicho problema fue tratado y resuelto por Moni Noar y Adi Shamir en [NS94], la presente tesis toma como referencia su trabajo.

La información contenida en los mensajes en este caso ya no consiste de letras y números (símbolos) sino de imágenes, en el presente trabajo sólo consideraremos imágenes monocromáticas, es decir, en color blanco y negro. Una imagen está formada por píxeles, un píxel es una unidad de una imagen digital que puede representarse por medio de bits, que contiene información referente al color. La técnica que nos ayuda a cifrar una imagen se le conocen como **Criptografía Visual, (CV)**, lo curioso e interesante de ésta técnica es que no se necesita ningún conocimiento criptográfico para poder recuperar la información original. Tal vez uno se pregunte cómo es posible esto, ya que la mayoría de los esquemas criptográficos se basan en algoritmos matemáticos muy avanzados que implican cálculos complejos que solamente pueden ser llevados a cabo por las computadoras. Sin embargo esto sí es posible con ayuda de la CV.

Para poder llevar a cabo el cifrado de imágenes nos apoyaremos en una modificación de los esquemas umbrales (2.3), llamados esquemas visuales umbrales.

3.1. Secretos compartidos con imágenes

Un esquema visual umbral (t,w), es una extensión de un esquema umbral (t,w) (2.3), donde $t < w$ y ambos son enteros, en el sentido de que el secreto es una imagen y no un número como tal. Donde el propósito es el mismo, compartir w pedazos de la imagen original entre el mismo número de participantes de tal manera que para poder recuperar el secreto se necesitan al menos t de sus pedazos. Así es como pasamos de cifrar cadenas de símbolos a imágenes en el texto llano.

La técnica de los secretos compartidos con imágenes considera que los píxeles pueden ser manipulados separadamente, y en eso consiste el procedimiento de cifrado como veremos a continuación.

Procedimientos de Cifrado y Descifrado

El cifrado de una imagen consiste en descomponerla, es decir, los píxeles que forman ésta ya sean blancos o negros deben ser codificados de tal manera que cada uno de ellos aparezca modificado w veces diferentes del original, lo que se pretende con esto es ocultar el color de los mismos. Una imagen tiene un cierto tamaño, por ejemplo n píxeles de ancho por m de alto, entonces tendremos que manipular $n \times m$ píxeles.

Considerando que una imagen es a la vez un arreglo de píxeles blancos y negros, como ya mencionamos habrá w codificaciones para cada píxel, a dicha interpretación la llamaremos **subpíxel** o también conocidos como **parte**, que también será una matriz de píxeles blancos y negros cuyo tamaño estará en función de la cantidad mínima requerida de participantes que deseen recuperar el secreto, en general de tamaño $t \times w$. Es importante mencionar que habrá dos tipos de subpíxeles los que definen a un píxel blanco y a uno negro. Llamaremos **sombra** a una codificación de todos los píxeles de la imagen original, así que el resultado del procedimiento de cifrado es que contaremos con w sombras a compartir, en donde el tamaño de éstas es superior al de la imagen original.

Podemos ver que una sombra es una colección de $n \times m$ subpíxeles, esta colección la podemos escribir de manera formal como una matriz boolena $S = [s_{ij}]$ de tamaño $n \times m$, en donde $s_{ij} = 1$, si el j-ésimo píxel en la i-ésima parte es negro, y $s_{ij} = 0$, si el píxel es blanco. Cuando las sombras $i_1, i_2, \ldots, i_r$, se encuentran impresas en papel y son sobrepuestas encima una sobre la otra de tal manera que los subpíxeles queden totalmente alineados se logrará ver la imagen original; esto es gracias a la contri-

bución que cada subpíxel aporta para la recuperación del color del píxel original, esto es lo que conocemos descifrado. Es curioso el procedimiento de descifrado, ya que podemos recuperar la información original sin la necesidad de realizar cálculos complicados.

Es importante mencionar que hay una pérdida de definición de la imagen recuperada en comparación con la original debido a la codificación de los píxeles, y en ese sentido la capacidad visual de los participantes es importante. Ya que la imagen recuperada aparecerá en tonos grises y corresponderá al ojo humano distinguir entre el color blanco del negro. La intensidad del tonos gris de cada subpíxel está asociado con el peso de Hamming $H(V)$, y será interpretado por la vista como negro si $H(V) \geq d$ y blanco si $H(V) < d - \alpha m$, para un umbral fijo $1 \leq d \leq m$ y $\alpha > 0$.

Esto nos da como resultado que durante el procedimiento de cifrado a cada píxel se le vaya agregando ruido con el propósito de que no se distinga su color, y una vez que se sobreponen (traslapan) las sombras poder recuperarlos aunque dicho ruido no es eliminado completamente. Adi y Shamir en [NS94] propusieron la siguiente solución al problema de los secretos compartidos con imágenes:

Definición 3.1. ***Esquema umbral de secretos compartidos con imágenes***

Una solución al esquema visual de secretos compartidos t de w, (en la presente tesis tomaremos $t = w = 2$) consiste de dos colecciones de matrices boolenas C_0 y C_1 de tamaño $n \times m$. Para compartir un píxel blanco el director D, escoge una de las matrices en C_0 y para compartir el píxel negro, D escoge aleatoriamente una de las matrices en C_1. La matriz escogida define el color de los m subpixeles de cada una de las sombras. La solución se considera válida si se cumplen las siguientes tres condiciones:

1. *Para cualquier S en C_0, el "or" V de cualquier k de n filas satisface $H(V) \leq d - \alpha m$.*
2. *Para cualquier S en C_1, el "or" V de cualquier k de n filas satisface $H(V) \geq d$.*
3. *Para cualquier subconjunto $\{i_1, i_2, \ldots, i_q\}$ de $\{1, 2, \ldots, m\}$, con $q < k$, las dos matrices de tamaño $q \times m$, D_t para $t \in \{0, 1\}$ se obtienen por la restricción de cada matriz $n \times m$ en C_t (donde t=0,1) de las filas $\{i_1, i_2, \ldots, i_q\}$ son indistinguibles en el sentido de que las matrices son las mismas con las mismas frecuencias.*

La tercera condición nos indica que si alguien llegase a inspeccionar menos de t partes, sin importar la capacidad de análisis que éste tenga no obtendrá ventaja alguna en decidir si el píxel compartido fue blanco o negro. Existe una función f tal que la fusión de partes de las $q < k$ transparencias consiste de todos los $V's$ con $H(V) = f(q)$ con una distribución uniforme de probabilidad, a pesar de si las matrices fueron tomadas de C_0 o C_1. Tal esquema es llamado uniforme. Las dos primeras condiciones son llamadas **contraste** y la tercera como **seguridad**.

En este esquema es importante considerar los siguiente parámetros:

- m, el número de píxeles en cada parte. Esto representa la pérdida de resolución de la imagen original. Deseamos que m sea lo más pequeña posible, en nuestro caso $m = 2$.
- α, la diferencia relativa en el peso de la combinación de las partes que viene de un píxel blanco a uno negro en la imagen original. Que representa la pérdida en el contraste, en este caso deseamos que α sea lo más grande posible.
- r, el tamaño de las colecciones de C_0 y C_1 (estas no necesitan ser del mismo tamaño, pero en todo nuestro trabajo serán del mismo tamaño). $log\ r$ representa el número de bits aleatorios necesarios para generar las partes y esto no afecta la calidad de la imagen.

Las diferentes codificaciones de subpíxeles blancos se obtienen permutando las columnas de C_0.

$$C_0 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix}$$

Figura 3.1: Representación matricial de un subpíxel blanco.

Las diferentes codificaciones de subpíxeles negros se obtienen permutando las columnas de C_1 (3.2).

Dada una imagen S, que la denotamos por S refiriéndonos al secreto contenido en esta, que puede ser una clave. Nos corresponde representar S como una matriz

$$C_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Figura 3.2: Representación matricial de un subpíxel negro.

booleana S' de tamaño $n \times m$. Considerando que en cada entrada de S' hay 1 ó 0, estos valores indican el color del píxel a cifrar. Ahora necesitamos transformar cada entrada (i, j) de S', en una submatriz de tamaño 2×2, en donde esta será la representación codificada del píxel original (subpíxel) (i, j) de S.

En el esquema de secretos compartidos con imágenes (2,2) que usaremos, cada píxel es cifrado dando como resultado 2 partes que lo representan, es decir, el resultado de este procedimiento será la permutación de las columnas de la matriz C_0 en dos ocasiones en el caso de un píxel blanco y analogamente será el caso para un píxel negro con C_1. En donde las matrices C_0 y C_1 son de tamaño 2×2.

En las figuras 3.3 y 3.4, podemos ver las distintas codificaciones para un píxel blanco y uno negro. Además en éstas hay 6 pares de subpíxeles para codificar un píxel blanco y otros 6 pares para la codificación de uno negro, en donde estos 12 pares son las representaciones de los subpíxeles de las matrices C_0 y C_1 de las figuras 3.1 y 3.2 respectivamente, sólo que éstas tienen menor tamaño. Veremos que las sombras duplicaran en tamaño a la imagen original.

Las codificaciones que se muestran son las que emplearemos para el procedimiento de cifrado. Cuando un adversario pretende adivinar la codificación de un píxel negro en un primer intento, la probalidad de que éste acierte será de $\frac{1}{6}$, y en el caso de que intente adivinar la codificación por lo menos k veces, la probabilidad de éxito será de $(\frac{1}{6})^k$.

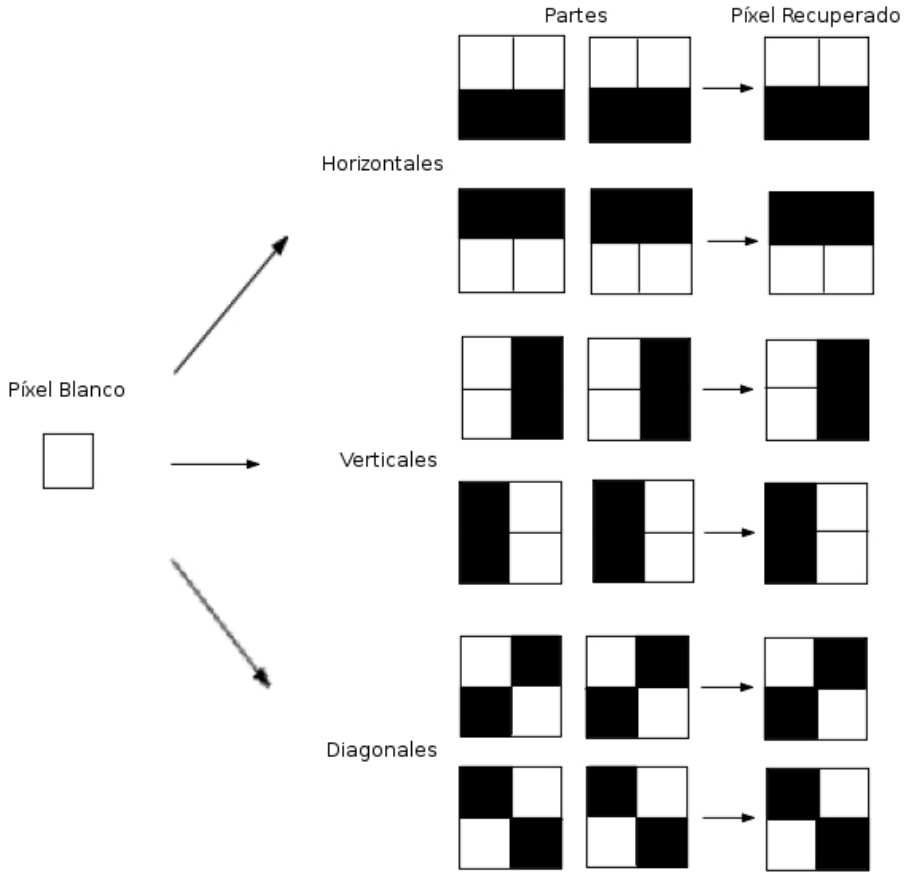

Figura 3.3: Codificación de un píxel blanco.

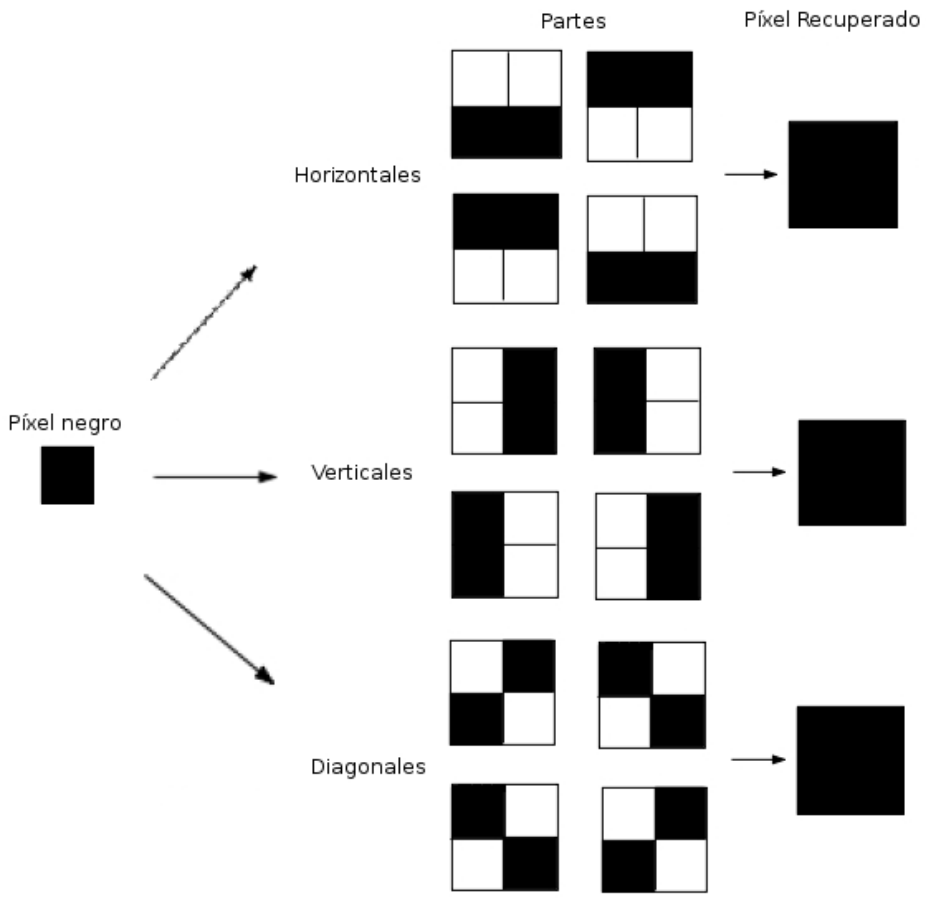

Figura 3.4: Codificación de un píxel negro.

3.2. Identificación de usuario con imágenes

Saber con quién tenemos algún tipo de comunicación, principalmente por medio de la internet, un medio de masivo de comunicaión, por decirlo de alguna manera manipulable en el sentido de alguien puede proporcionar información referente a su persona que no necesariamente sea cierta y este caso depende de uno identificar al tercero, y hacerlo gráficamente (imágenes) es interesante como veremos a continuación.

La identificación visual humana es sin duda una forma diferente de llevarse a cabo, debido a que ahora utilizamos una imágen para la identificación, cuyo contenido puede ser un NIP (Número de identificación personal) o simplemente un dibujo. No sólo por internet tenemos la necesidad de identificarnos ante un tercero, por ejemplo cuando se acude al banco a disponer de efectivo, uno podría pensar que con tantos avances técnologicos es más sencillo llevar a cabo dicha tarea. Sin embargo esto no es así debido a que en algunos casos el verificador es un persona como se menciono y no un dispositivo, la cual no tiene la capacidad necesaria para realizar cálculos complicados o almacenar mucha información.

Así es como la identificación visual tiene el mismo objetivo, permitir a una entidad, A, probar su identidad ante un verificador, B, como se ha dicho sin la necesidad de consultar al dispositivo computacional. Considerando la amenaza de un adversario C, cuyo objetivo es convercer a B que éste es A. No hay razón alguna para construir protocolos de identificación visual que permitan sólo una identificación segura ya que esto se puede llevar a cabo suministrando a una entidad una simple contraseña, y usarla las veces que desee identificarse. En ese sentido sólo consideraremos protocolos que permitan la identificación varias veces, es decir, protocolos en los cuales con sólo una sombra una entidad se pueda identificar en varias ocasiones. Los protocolos de desafío-respuesta son del tipo de protocolos en los cuales el verificador envía un desafío al usuario, quien debe responder en base a la información secreta que posee.

En [NP97], proponen la siguiente definición de protocolo visual:

Definición 3.2. ***Protocolo de identificación visual***

Se define el protocolo para la i-ésima identificación de A con B.

- *B envía un desafío c_i a A, el cual es está en función del secreto r.*

- *Una vez recibido c_i A responde con a_i, que es función de c_i y de su información secreta T_r y A_r, que ésta es enviada de regreso a B.*
- *B decide si la otra entidad es realmente A en base a los mensajes c_i, a_i y el secreto r. Entonces al verificador le corresponde **aceptar** o **rechazar** dicha información.*

El adversario C podría tratar de fingir ser A. En este caso él incluso podría tratar de preguntar a A reclamando ser B y pedir que se identifique. Entonces C inicia el protocolo de identificación con el verificador B y envía la respuesta en la cual esperaría convencer a B de que es la otra entidad, A.

En la capítulo 5 daremos un ejemplo de cómo implemnentar un esquema de secretos compartidos con imágenes $(2, 2)$ para la identificación visual.

3.3. Confirmación de operación con imágenes

Recordando brevemente cuál es el objetivo de la autenticación que es verificar que el contenido de un mensaje enviado por algún medio del que no se tiene control no haya sido alterado por los adversarios, y en ese sentido podemos darle un enfoque hacia la confirmación de operaciones apoyandonos de imágenes.

Noar y Pinkas ([NP97]) proponen la siguiente definición para un escenario de autenticación visual:

Definición 3.3. ***Escenario de autenticación visual***

Hay tres entidades A (Alan), B (Bety) y C (Claudia). A es una persona y tiene capacidades visuales humanas. Para cada protocolo las capacidades que se requieren de A deben ser declaradas, estas capacidades deben incluir la habilidad de identificar una imagen resultado de la composición de dos sombras de un esquema de secretos compartidos $(2, 2)$. Otras capacidades pueden ser la habilidad de verificar si cierta área es de color negro, la habilidad de comprobar si dos imágenes son similares, etc. Existe un parámetro seguridad, n, tal que la capacidad de almacenamiento y cálculo de B y C son polinomiales en n.

En la fase de inicialización Bety produce una cadena aleatoria r, y crea una sombra S_r y alguna información adicional A_r que están en función de r. Su tamaño es polinomial en el parámetro de seguridad n. B envía S_r y A_r a través de un canal privado al cual Claudia (adversario) no tiene acceso (ésta es la única ocasión que un canal privado es usado). También B envía un conjunto de instrucciones que A debe llevar a cabo en el protocolo, por ejemplo comprobar en cierto punto del tiempo si cierta área es negra, comparar dos áreas, etc. Las instrucciones son públicas y C podría conocerlas, pero es incapaz de cambiarlas.

Después de la fase de inicialiazación toda la comunicación es hecha a través de un canal controlado por C, quien podría cambiar los mensajes transmitidos.

Es difícil analizar procesos que implican a personas porque no hay un modelo matemático fácil que describa el comportamiento humano. Para demostrar la seguridad de dichos protocolos la parte humana en el protocolo debe estar explicitamente definida. Entonces es posible aislar las capacidades requeridas del participante humano (la habilidad de verificar que cierta imagen es totalmente negra). La seguridad del protocolo debe reducirse a la supocisión de que una persona "normal" tiene éstas capacidades. Esta supocisión puede ser verificada a través de pruebas empíricas.

Aunque la restricción que se hace sobre la capacidad de almacenamiento y cálculo de C, es decir, polinomial en n del parámetro de seguridad, esto no pretende ser una limitante, ya que los esquemas que se proponen son seguros contra un adversario con capacidad de almacenamiento y cálculo ilimitado, sin embargo no ocurre lo mismo para B en donde sus capacidades son lineales con respecto al tamaño del mensaje.

Definición 3.4. ***Protocolo de autenticación visual***

B desea comunicarle una parte m de información a A, cuyo contenido es conocido por C, entonces

- *B manda un mensaje m_1 a A, el cual está en función de m y r.*
- *C puede cambiar el mensaje m_1 antes de que lo reciba A.*
- *Una vez que A recibe el mensaje m'_1, corresponde a éste rechazarlo o aceptarlo, m' como una función de m'_1 y su información secreta S_r y A_r. Cuando A acepta m'_1, él también devuelve m_1, además cree que es la información enviada hacia él proviene de B.*

A continuación se presenta los requisitos de seguridad para un sistema de autenticación visual. La primera defición asegura que el adversario no puede convencer al receptor humano de recibir cualquier mensaje diferente del mensaje original. La segunda definición sólo asegura que para cualquier mensaje m' determindo a priori el adversario no pueda convencer al receptor de creer que el mensaje recibido fue m'.

Definición 3.5. ***Sistema de autenticación visual***

Se asume que A tiene las capacidades requeridas para el protocolo, él actua de acuerdo a las instrucciones dadas en el protocolo, y que el sistema de autenticación visual tiene la propiedad de que cuando C es honesta entonces A siempre $\{ACEPTA, m\}$*. El sistema es*

- ***(1-p)-auténtico*** *si para cualquier mensaje m enviado de B a A la probabilidad de que A* $\{acepte, m'\}$ *es al menos p, donde* $m' \neq m$*.*
- ***(1-p)-Una transformación segura*** *que también llamaremos (1-p)-uts, si para cualquier mensaje m enviado de B a A y cualquier* $m' \neq m$ *(él cual fue determinado a priori) la probabilidad de que A* $\{acepte, m'\}$ *es de al menos p.*

El sistema de autenticación visual (1-p)-uts obviamente es menos seguro que el (1-ps)-auténtico, sólo si se garantiza que es difícil cambiar el mensaje enviado por uno específico, el cual fue determinado antes de que la comunicación comience.

Se puede implementar un esquema de secretos compartidos con imágenes $(2, 2)$, en el sistema de control escolar con el propósito de autenticar documentos que hayan sido impresos desde el portal de internet, en donde una de las necesidades de los estudiantes es contar con documentos que avalen su estatus académico como lo son constancias de estudio, boletas de calificaciones, etc. y en ese sentido tomando en cuenta la infraestructura del Sistema Institucional de Gestión y Unificación Escolar (SIGUE), dado que cada alumno es un usuario del SIGUE por sí mismo, ya que el número de boleta asignado a cada uno de éstos es su correspondiente nombre de usuario de dicho sistema que también cuenta con su NIP.

El objetivo de está implementación es tratar de reducir el tiempo que toma en llevarse a cabo algún trámite administrativo ya que el mismo es engorroso e implica mucho papeleo. Suponiendo que toda la información académica (calificaciones parciales, globales, y semestres cursados, ect.) de la comunidad estudiantil este actualizada

en el SIGUE, el escenario sería el siguiente en el caso de que algún estudiante necesítase un boleta de calificaciones global con el objetivo de reinscribirse al siguiente semestre, es decir, que haya aprobado todas sus materias previas a las que desea cursar, este con sólo ingresar a su cuenta del SIGUE y desde ahí solicitar el documento e imprimirlo sin la necesidad acudir a control escolar a solicitarla. A dicho documento se le adjunta una imagen cifrada (sombra, S_1), que bien podría ser el hash del mensaje, y con su boleta impresa acudir a control escolar solicitando su reinscripción en donde el personal de éste juega el papel del verificador, es decir, el personal de control cuenta con la otra parte de la imagen cifrada (sombra, S_2), que llamaremos **llave**, ésta está impresa en papel transparente, y aceptará su boleta como auténtica una vez que aparezca el valor del hash al sobreponer su llave sobre la imagen adjunta a la boleta ($S_1 + S_2$).

Este protocolo reduciría de una manera significativa el tiempo que requiere el procedimiento de reinscripciones ya que para autenticar las boletas o mejor dicho corroborar la condición académica de los estudiantes ya no sería necesario revisar nuevamente el sistema y así con esto también prevenir modificaciones en las calificaciones.

Capítulo 4

Interfaz para la generación de imágenes compartidas

Para llevar a cabo el cifrado de imágenes usaremos una implementación desarrollada en lenguaje java, que llamaremos **"Compartiendo imágenes CV"**, (CICV); el hecho de usar este lenguaje es debido a que es portable, en el sentido de que no es necesario cargar con librerias adicionales para el adecuado funcionamiento de las implementaciones creadas en java, entre otras. La implementación CICV puede ser ejecutada en cualquier máquina siempre y cuando tenga instalada la **Máquina Virtual Java (MVJ)**[1].

El entorno de desarrollo utilizado fue Netbeans versión 5.5.1 [2], por ser un entorno amigable y gráfico. Este entorno permite crear el archivo ejecutable correspondiente a CICV, cuya extensión es **.jar**, éste archivo es el que debemos transportar de una máquina a otra cuando deseemos usarlo.

Hay varios tipos de formatos de imágenes debido a esto nuestra implementación funciona solamente con imágenes en formato **.gif** (Graphics interchange format), ya que es un formato sin pérdida de calidad para imágenes con hasta 256 colores, con una paleta de colores restringida a este número de colores, y fácil de usar a través de la web. También es importante considerar el tamaño de las imágenes a trabajar. Por está razón ponemos límite al tamaño de éstas, digamos de 32×64 píxeles de alto

[1] Es un programa ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (Java bytecode), el cual es generado por su compilador java.

[2] www.netbeans.org

por ancho respectivamente. Así en este capítulo describiremos el uso de CICV y los métodos empleados.

4.1. Descripción de la implementación

En la figura 4.1 se muestra una ventana que podemos decir que es la base de la implementación, ésta contiene 2 menús: **Archivo**; **Acerca de...**.

Figura 4.1: Interfaz "CICV".

Menús

Archivo, este menú tiene 2 submenús, **Abrir...**, y **Cerrar**, ver la figura 4.2. El submenú Abrir, cuenta con 2 submenús, **Imagen original...**, dando click en este nos despliega un cuadro de dialogo (figura 4.3) con la ubicación de los directorios en donde puede ubicarse la imagen que se desea cifrar, además de mostrar un panel llamado PCifrado, el cual cuenta con tres paneles; un botón para cifrar y checkboxes para imprimir, guardar, y visualizar las imágenes una vez ya cifradas, ver figura 4.4.

Una vez que se tenga selecionada la imagen original a cifrar esta será cargada y posteriormente pintada en el panel ubicado en la parte superior izquierda de la

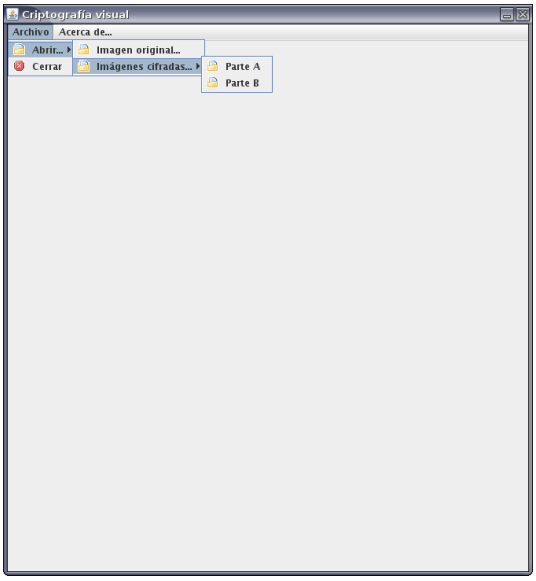

Figura 4.2: Menú Archivo y submenús Abrir, y Cerrar.

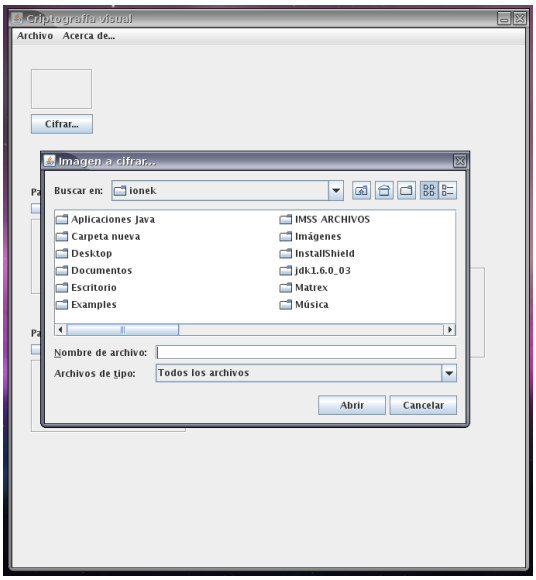

Figura 4.3: Cuadro de dialogo para seleccionar la imagen a cifrar.

ventana como se muestra en la figura 4.5. Cuando la imagen se encuentra cargada la implementación tiene su interpretación interna de ésta de cual nosotros obtenemos los píxeles en un arreglo de tipo entero cuyo tamaño es el producto del ancho por el alto de la imagen original; cada uno de valores almacenados contiene información del color de cada píxel.

El siguiente paso sería cifrar la imagen original, y podemos llevar a cabo esto presionando el boton -Cifrar...-, mandando la instrucción de cifrar la "imagen" (ver 3.1), primero identificando el color de cada píxel, es decir, ya sea color blanco o negro mediante distintas interpretaciones definidas por arreglos de tipo booleano de tamaño 2×2.

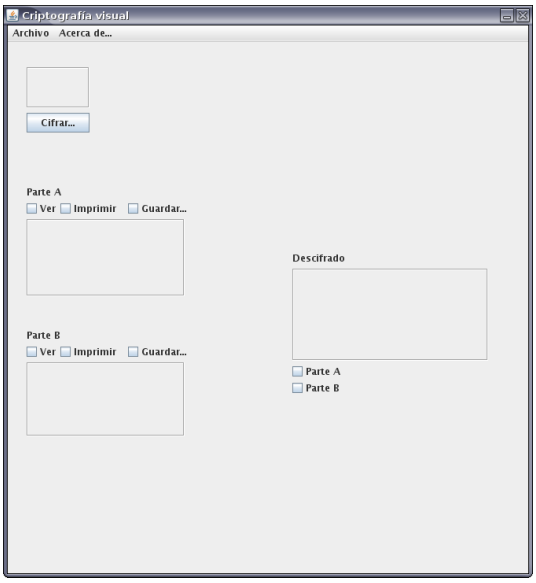

Figura 4.4: Panel PCifrado.

El resultado del procedimiento de cifrado lo podemos ver en los paneles A, B y Descifrado, seleccionando cualquiera de los checkboxes: **Ver**, visualiza las partes cifradas A y B, en sus respectivos paneles; **Imprimir**, manda la instrucción a la impresora que procesa a escribir las partes A y B en una hoja de papel cada una; **Guardar**, despliega un cuadro de dialogo con el propósito de guardar las partes A y B en formato .gif, ver 4.6.

En el panel Descifrado se muestra la sobreposición de las partes A y B cuando se

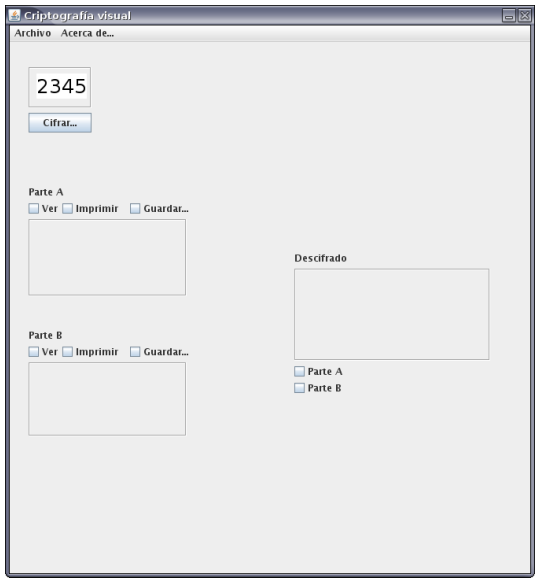

Figura 4.5: Imagen cargada.

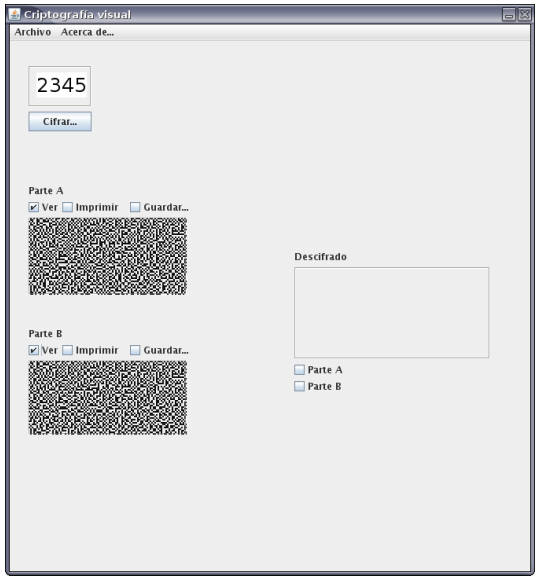

Figura 4.6: Checkbox: Ver, Partes A y B.

selecciona ver ambas partes el resultado está en la figura 4.7.

Figura 4.7: Sobreposición de las partes A y B.

El submenú Imágenes cifradas, despliega un nuevo panel llamado PDescifrado, después de dar click en éste, el propósito de mostrar un panel adicional es confirmar que se hayan cifrado y guardado adecuadamente las imágenes (partes A y B), ver 4.8. Los componentes del PDescifrado son dos cuadros de dialogos, que muestran la ubicación de las imágenes cifradas; dos paneles A y B donde se visualizan las imágenes; y tres botones, Parte A y B pintan las imágenes cifradas, el tercer botón, Guardar, con la instrucción de mandar a escribir en archivo la sobreposición de las partes A y B, figura 4.9.

Menú Acerca de..., éste da a conocer información del tesista en una ventana emergente, ver las figuras 4.10 y 4.11.

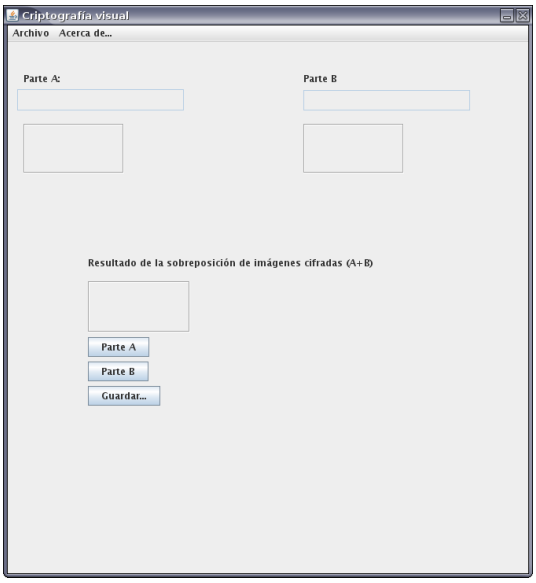

Figura 4.8: Panel de verificación de imágenes cifradas.

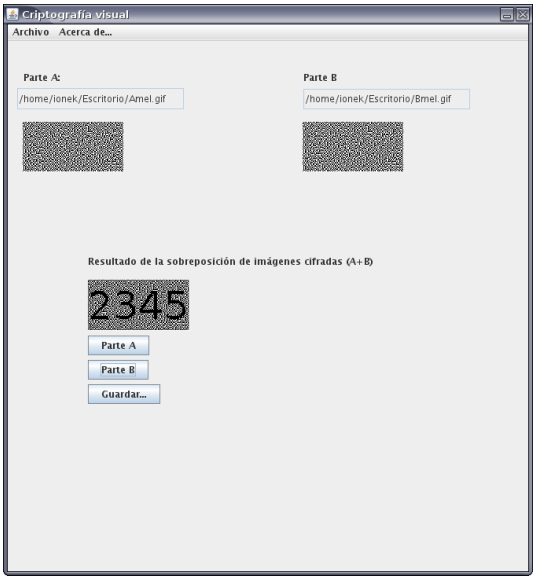

Figura 4.9: Confirmación del procedimiento de cifrado de imágenes.

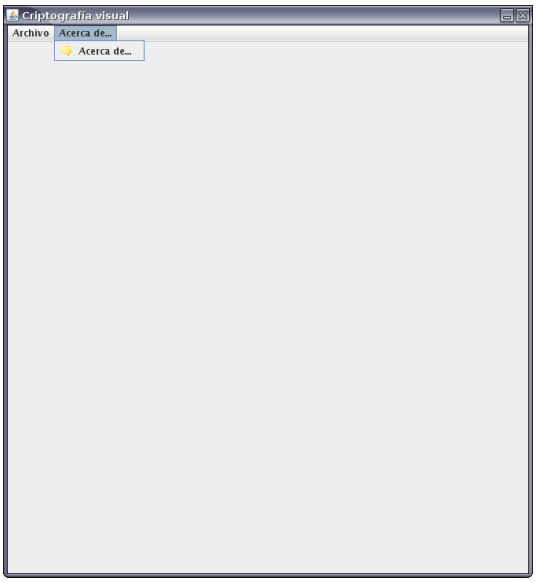

Figura 4.10: Menú “Acerca de...”.

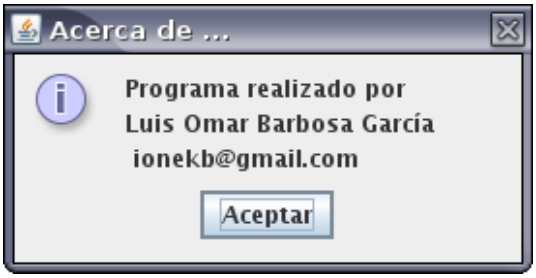

Figura 4.11: Ventana emergente del menú Acerca de...

4.2. Descripción de métodos empleados en CICV

Además de describir algunos de los métodos empleados invito a los interesados ver la documentación anexa del código fuente de este sencillo programa con el propósito de continuar con este trabajo.

Métodos de escritura

Los métodos de escritura que utilizamos son: **GIFEncoder**; **Imprimir**; **Pintarpartes**.

GIFEncoder, nos hemos apoyado del trabajo del Sr. Adam Doppelt que éste a su vez se basó en el código gifsave.c escrito y liberado por el Sr. Sverre H. Huseby, para poder guardar las imágenes resultantes (cifradas) en archivos con extensión .gif. El cual requiere como entrada tres arreglos de tipo byte que devuelve un archivo (imagen).

Imprimir, en este método nos apoyamos de los métodos estándar que tiene java ya definidos, haciendo unas pocas modificaciones, se necesita de un arreglo de tipo booleano y de un entero y el resultado es mandar intrucciones a la impresora predetermina de escribir una imagen en papel.

Pintarpartes, nuevamente haces uso de los métodos que tiene java ya definidos para poder pintar las imágenes resultantes del procedimiento de cifrado en los paneles de la implementación.

Métodos de lectura

La lectura de imágenes se hizo utilizando métodos de las librerias de java, como awt, image, y Toolkit entre otras. Haciendo las adecuaciones a estos métodos para hacer posible la lectura de los píxeles de las imagenes y obtener el resultado deseado.

Capítulo 5

Aplicaciones prácticas

Alguien podría preguntarse por qué usar esquemas criptográficos que emplean imágenes, como lo hemos venido mencionando hay ocasiones en las cuales uno no cuenta con una computadora al momento de revelar un secreto o clave de un esquema criptográfico ya que en la mayoría de las veces se necesita una para poder hacerlo, y en ese sentido tener algo a la mano como lo es un papel para descubrirlo es mucho más práctico y sencillo.

5.1. Identificación

Una implementación del esquema visual de identificación $(2, 2)$, pudiese ser el acceso al sistema informático de la Escuela Superior de Física y Matemáticas, (ESFM), mejor conocido como SIGUE, el escenario es el siguiente:

Actualmente el SIGUE funciona de la siguiente manera: el acceso (log in), es mediante una frase de acceso (contraseña) la cual está asociada al número de boleta de un alumno que es el usuario registrado en el SIGUE. Los alumnos tienen facilidad de ocupar equipos de cómputo para revisar su información académica, sin embargo debido al protocolo de acceso alguien podría llegar ver la contraseña de otro, y haciendo mal uso de ésta puede accesar a la cuenta del tercero. Se propone usar una contraseña de un sólo acceso:

1. El SIGUE cuenta de antemano con la identidad de todos los alumnos inscritos en el semestre, a los cuales se les dado un número de boleta al momento de que se inscriben por primera vez, éste es el nombre de usuario registrado en el sistema. Al emplear un esquema de secretos compartidos con imágenes $(2, 2)$,

se crean dos sombras $S_{e,i}$ y $S_{s,i}$, el subíndice e y s se refieren a la sombra que le corresponde al estudiante y al sistema (SIGUE) respectivamente, y la i indicará el alumno. $S_{e,i}$ es enviada al correo electrónico de cada estudiante registrado en el sistema, además se les pide que impriman su $S_{e,i}$ en un papel transparente.

2. Cuando un alumno pretende acceder al SIGUE, éste ingresa su ńumero de boleta, y a continuación le despliega una imagen ($S_{s,i}$) en pantalla sobre la cual el alumno debe sobreponer su sombra, $S_{e,i}$ y el resultado será la visualización de un NIP, que deberá ingresar para entrar a su cuenta. El NIP visualiazado en pantalla sólo sirve una vez.

   De alguna manera el SIGUE tendrá que crear un nuevo NIP, el cual será el reemplazante del anterior. El SIGUE sabe de pies a cabeza el contenido de cada sombra (píxeles) asociada al alumno, y en ese sentido crear una nueva sombra del NIP reemplanzante no será problema, ya que cambiará los píxeles negros por blancos de $S_{s,i}$ que fuesen necesarios en base a la posición de los píxeles de $S_{e,i}$ (que ésta sombra nunca cambiará) todo esto con el fin de construir la codificación del nuevo NIP. Es importante mencionar que la creación de nuevos NIP's será constante, debido a la cantidad de veces que el alumno desee ingresar a su cuenta.

5.2. Confirmación de operación

Pasemos a otro escenario, el caso de los cajeros automáticos (ATM) cuyo funcionamiento es sencillo y confiable; recordemos brevemente cómo funciona: Las tarjetas bancarias en algunos casos cuentan con chips, disposito que almacena información de las tracciones realizadas por el propietario, y del usuario mismo. En ese sentido las ATM's saben quien es cuando leen la información contenida en el chip, y lo siguiente que hacen es confirmar si el usuario que ingreso la tarjeta en la ATM es realmente quién dice ser mediante su NIP. Una vez validado al usuario se le permite realizar varias operaciones, consulta de saldos y retiros de efectivo, en está última operación podemos hacer uso de imágenes para confirmar el retiro de efectivo.

Veamos cómo sería la implementación, supongamos que el usuario ya se dispone a retirar efectivo pero antes de poder hacer esto la ATM le pide otro número que confirme está operación, desplegando una imagen B ilegible en la pantalla (5.1), y el usuario con su imagen A (5.2), también ilegible (transparencia, previamente dada

por el banco) y al sobreponer la imagen A sobre B, de como resultado es un número oculto (5.3), que tecleado en la máquina permita ahora la disposición del efectivo.

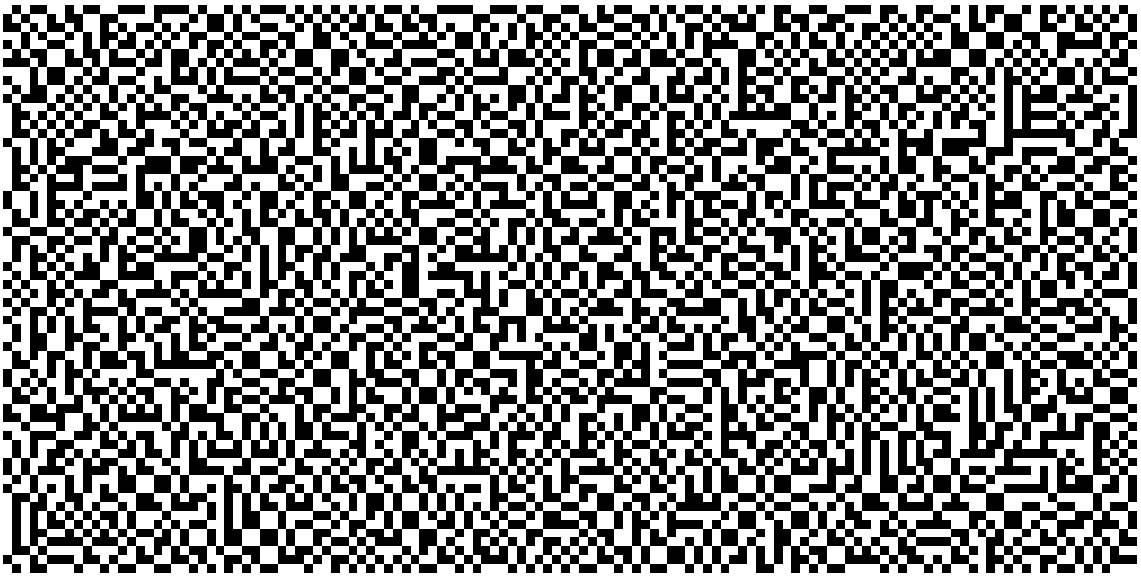

Figura 5.1: Parte B, desplegada en la pantalla de ATM

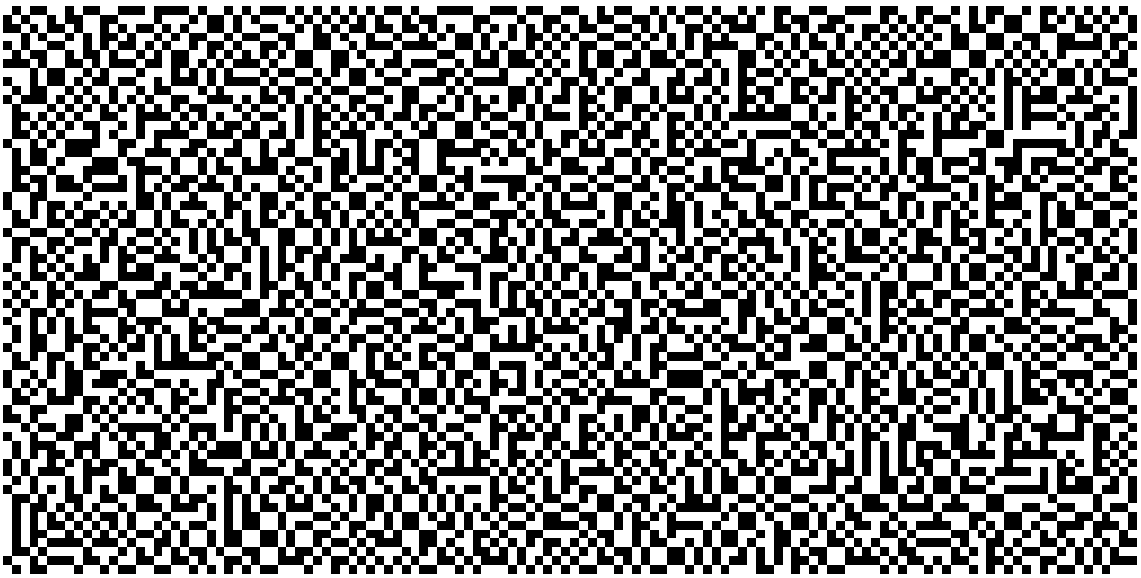

Figura 5.2: Parte A

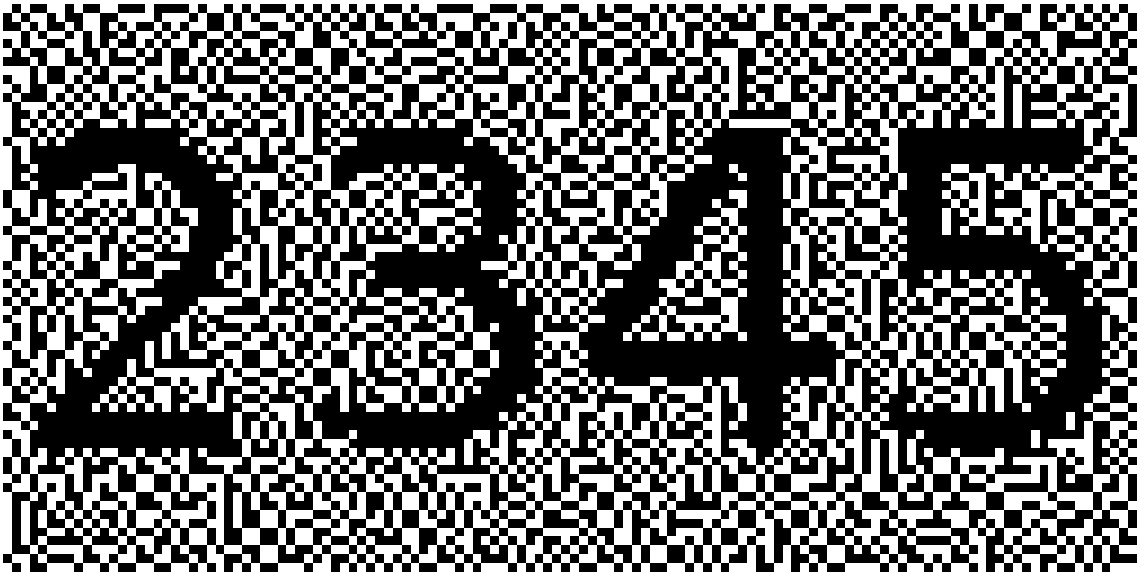

Figura 5.3: Sobreposición de las partes A y B

Escenario de identificación

Consideremos a la empresa *XTickets*, dedicada a la venta de boletos para diferentes eventos culturales, de entretenimineto, y deportivos entre otros; la venta de éstos es a través de internet. La gerencia está mejorando sus prácticas comerciales, se enfoncará de momento en clientes frecuentes. Por lo tanto les gustaría implementar algún procedimiento para diferenciar un cliente frecuente de uno casual.

El procedimiento de compra de boletos es el siguiente:

1. Creación de una cuenta en el portal de la empresa, registro de datos personales, correo electrónico entre otros, dando como resultado la creación de un usuario con su respectiva contraseña que serán usados cada vez que se desee adquirir boletos.

2. Confirmación por parte de la empresa de los datos proporcionados, una vez que sean validados por el usuario, el sistema le asigna un número al registro para su control y monitoreo. Si ya llegase haber algún error en la información ingresa se le pide al usuario que nuevamente ingrese la información correcta, y se repite el paso 2.

3. Ahora que el cliente cuenta con su usuario y contraseña, éste ingresa al portal para adquirir sus boletos donde se muestra una lista con la información de eventos, elige el de su preferencia y compra el boleto.

Se considerará un cliente frecuente aquel que haya realizado por lo menos 5 compras en los dos meses anteriores al inicio de cada mes; dado que contamos previamente con el registro del cliente que ha comprado algún boleto, es más sencillo ubicar aquellos que que hayan hecho la cantidad mínima de compras. Ya que se tienen identificados los clientes frecuentes se les hace llegar vía correo electrónico información referente a próximos eventos, y promociones (descuentos, boletos de cortesía, etc.).

Sin embargo cada vez es más común encontrarnos con correos basura, mejor conocidos como spam, que regularmente contienen información publicitaria. Ya que el objetivo de *XTickets* es vender más boletos y el medio electrónico parece ser lo adecuado para comercializar sus productos. Está primera etapa de comercialiazación sólo se enfoncaría en clientes frecuentes con promociones. *XTickets* quiere evitar que los correos promocionales no sean vistos como spams por parte de sus clientes frecuentes, que muchas veces no son leidos.

Otra forma de dar a conocer las promociones mensuales es a través de su portal identificando a los clientes frecuentes. Para llevar a cabo dicha tarea nos proponemos hacer uso de la criptografía visual, utilizando un esquema $(2, 2)$ de secretos compartidos con imágenes, implementado en CICV.

A continuación se presenta el escenario:

1. A cada cliente frecuente se le asigna una sombra, $S1_i$, la i indica el número de clientes frecuentes que hay por mes. La sombra es una parte del procedimiento de cifrado en CICV, cuyo contenido puede ser un código de cuatro caracteres, las tres primeras letras de cada mes. Dicha imagen es enviada por correo electrónico y se le pide al usuario que imprima ésta tal y cual como se le envía en una hoja transparente (mica). Es importante mencionar que las $S1_i$ son iguales.
2. Al momento de que algún cliente se disponga a realizar una compra, en el portal $XTickets$ aperecerá una imagen ilegible (sombra $S2$), en caso de que la persona sea un cliente frecuente y cuente con su sombra puede sobreponer su mica en la pantalla de su equipo de cómputo dando como resultado el código de acceso a la página de promociones.

De esta manera el cliente tiene la oportunidad de participar de una forma más activa a la hora de identificarse, sin la necesidad de recordar una contraseña.

Escenario de autenticación visual

Otro problema presente es la falsificación de boletos impresos en papel especial y electrónicos. La autenticación de los boletos va estar en función de la infraestructura con que cuente el sitio donde se lleve a cabo el evento. En ese sentido los boletos impresos en papel especial se encuentrán foliados, además de contar con un código de barras. El personal que verifica la autenticidad del boleto mediante un escáner. Éste procedimiento puede llegar a ser costoso por la renta de los equipos, y en ese sentido podemos apoyarnos nuevamente en la CV para reducir estos costos, ya que la autenticación no requiere de ningún dispositivo adicional a una mica de acetato.

También es un esquema $(2, 2)$ de secretos compartidos con imágenes, este caso el personal del inmueble donde se lleve a cabo el evento tomará el papel del verificador,

y los asistentes serán los demandantes. El escenario es el siguiente:

Cifrado

En esta parte se generan las sombras $S1$ y $S2$ de una imagen, S, cifrada en CICV. La $S1$ le corresponde al verificador y $S2$ al demandante. El contenido de S puede ser alguna clave, por mencionar una, la fecha del evento. La sombra $S2$ acompaña a los boletos de papel especial o electrónicos.

Autenticación

El personal autorizado para permitir el acceso al evento cuenta con su sombra $S1$, y para permitir el acceso el encargo debe sobreponer la mica sobre el boleto de tal manera que queden alineadas las imágenes, dando como resultado la visualiazación de la fecha del evento, si aparece ésta es un boleto auténtico y se le permite el acceso a la persona en caso contrario se le niega. Podemos pensar en un ahorro significativo de dinero al no usar dispositivos en caso de un evento masivo por ejemplo.

Capítulo 6

Conclusiones

A través de los 5 capítulos de esta tesis, presentamos sólo una pequeña parte de la interesante y cautivadora ciencia que es la criptografía, y una parte de ella, la gráfica (criptografía visual).

Que podemos concluir del trabajo, del capítulo uno la importancia que ha tenido y tendrá esta ciencia conforme avance el tiempo, en el uso comercial, educativo y sin duda en el militar, cuyos objetivos no son tan diferentes a los de la criptografía, que bien podríamos resumir estos en una sóla palabra, seguridad. En el segundo, platicamos de las técnicas de autenticación e identificación, estás son: firma digital, frases de acceso, y secretos compartidos. Que si bien fueron desarrolladas ya hace varios años se siguen empleando por el hecho de que son confiables aun. Nos permiten tener la certeza de saber con quién estamos tratando, es decir, comunicación a través de la red (internet).

En el tercero nos enfocamos a la criptografía visual, en donde trabajamos con imágenes que para nosotros toman el papel de mensajes. La manera en que son cifrados (secretos compartidos con imágenes) es muy diferente a los esquemas de cifrado (RSA, ELGamal) que implican demasiados cálculos, además de ser complicados que sólo pueden ser llevados a cabo por una computadora. Y qué decir del procedimiento de descifrado visual, con sólo juntar las sombras (traslape de hojas de papel por ejemplo) podemos recuperar el secreto. Considero que hay muchas áreas para la aplicación de los esquemas visuales, obviamente en computación, financiera, militar, incluso legal, una de las razones es que no necesitamos de algún dispositivo computacional para recuperar el secreto, llendonos a la parte legal, papelito habla.

Para llevar a cabo el cifrado de imágenes desarrollamos una implentación CICV, en lenguaje java, muy sencilla y funcional sin fines de lucro sólo educativo, ya que el CICV sólo diseñada para crear 2 sombras resultado de un esquema de secretos compartidos con imágenes $(2, 2)$, quedando pendiente el esquema (t, w) para algún despistado que se encuentre con esto. El tutorial se presenta en el capítulo 4, y en 5 presentamos escenarios en los que puede implementar un esquema visual $(2, 2)$ de secreto compartidos con imágenes para la autenticación e identificación visual.

Bibliografía

[Bru96] Schneier Bruce. *Applied Cryptography.* Editorial John Wiley & Sons, 1996.

[BSW06a] A. Beutelspacher, J. Schwenk, and K. Wolfenstetter. *Moderne Verfahren der Kryptographie, Von RSA zu Zero-Knowledge.* Mathematik. Vieweg, 2006.

[BSW06b] Albrecht Beutelspacher, Jorg Shwenk, and Klaus-Dieter Wolfenstetter. *Modern Verfahren der Kryptographie.* Vieweg, 6 edition, 2006.

[Bur05] Mark Burnett. *Perfect Passwords.* Syngress, first edition, November 2005.

[Cha85] De Chaum. Manufacturing security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28:1030–1044, 1985.

[Coh96] Henri Cohen. *A Course in Computational Algebraic Number Theory.* Graduate Texts in Mathematics. Springer-Verlag, 1996.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–65, November 1976.

[DM83] R. DeMillo and M. Merritt. Protocols and data security. *IEEE Computer*, pages 16(2):39–54, 1983.

[Eck02] Bruce Eckel. *Piensa en Java.* Pearson educación, S.A., 2002.

[Enc00] Luis Hernández Encinas. Esquemas criptógraficos visuales. *Agorasic*, Febrero 2000.

[Eng03] Michael Englbercht. *Entwicklung sicherer Software.* Spektrum, 2003.

[Esp03] Real Academia Española. *Diccionario de la Lengua Española*, volume 22. ESPALSA-CALPE, 2003.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman and company, 1979.

[GMR88] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *Siam Journal on Computing*, pages 281–308, 1988.

[Gol99] O. Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*. Springer, 1999.

[Gue92] Fausto Guerra. Criptografía visual. *Laberintos e infinitos*, (3):8–12, 1992.

[Klo97] Neal Kloblitz. *Algebraic Aspects of Cryptography*, volume 3 of *Algorithms and Computation in Mathematics*. Springer, second edition, September 1997.

[Knu69a] Donald E. Knuth. *The art of computer programming*, volume 1. Addison-Wesley, 1969.

[Knu69b] Donald E. Knuth. *The art of computer programming*, volume 2. Addison-Wesley, 1969.

[Kur06] Budi Kurniawan. *A Beginner's Tutorial Java 5*. BrainySoftware, 2006.

[LV01] A.K. Lenstra and E.R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology*, 14(4):255–293, 2001.

[MvOV96] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Criptography*. CRC Press, 1996.

[Nef04a] C. Andrew Neff. Practical high certainty intent verification for encrypted votes. *http://www.votehere. net/vhti/documentation*, 2004.

[Nef04b] C. Andrew Neff. Verifiable mixing (shuffling) of el-gamal pairs. *http://www.votehere.net/vhti/ documentation*, 2004.

[NK00] Pat Niemeryer and Jonathan Knudsen. *Learning Java*. O'Reilly, 2000.

[NP97] Moni Naor and Benny Pinkas. Visual autentication and identification. 1997.

[NS94] Moni Noar and Adi Shamir. Visual cryptography. 1994.

[NY89] Moni Naor and Moti Yung. Universal one-way hash function and their cryptographic applications. 1989. Revised March 13, 1995.

[Pre99] Bart Preneel. The state of cryptographic hash functions. volume 1561 of *Lecture Notes in Computer Science*, pages 158–182. Springer-Verlag, 1999.

[RS04a] P. Rogaway and T. Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal Roy and Willi Meier, editors, *Lecture Notes in Computer Science*, volume 3017, pages 371 – 388. Springer-Verlag Heidelberg, 2004.

[RS04b] P. Rogaway and T. Shrimpton. Cryptographic hash-function basis: Definitions, implications, and separations for preimage resistance, second-preimage resistance and collision resistance. *Fast Software Encryption (FSE 2004), Lecture Notes in Computer Science, Springer-Verlag*, page 20, Febraury 2004.

[RSA78] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[Sch90] C.P. Schnorr. Efficient identification and signature schemes from smart cards. In *Crypto '89*, volume 435, pages 239 – 251. Springer LNCS, 1990.

[Sch07] Klaus Schmeh. *Kriptografie*. dpunk.verlab, 2007.

[SY91] Alfredo De Santis and Moti Yung. On the design of provably-secure cryptographic hash functions. In *Advances in Cryptography - EUROCRYPT'90*, volume 473, pages 412–431. SpringerVerlag, 1991.

[Vel98] Jesús Sánchez Velázquez. *Introducción al Análisis de Algoritmos*. Trillas, Junio 1998.

[Voß06] Herbert Voß. *Kryptografie mit Java, Grundlage und Einfürung zur kryptograpfischen Programmierung mit*. FRANZIS professional series, 2006.

[Wol07] Gottfried Wolmeringer. *Java 6 lernen mit Eclipse*. Galileo Computing, 2007.

[yPJD04] Harvey M. Deitel y Paul J. Deitel. *Cómo programar en Java*. Pearson Educación, México, 2004.