UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

EVALUACIÓN DEL DESEMPEÑO DE UN ALGORITMO PARA ESTEGANOGRAFÍA BASADO EN LA TÉCNICA DEL BIT MENOS SIGNIFICATIVO

T E S I S

PARA OBTENER EL GRADO DE

INGENIERO EN REDES

PRESENTA

PIN SU

DIRECTOR DE TESIS

DR. JAVIER VÁZQUEZ CASTILLO

UNIVERSIDAD DE QUINTANA ROO
AREA DE TITULACION

ASESORES

DR. JAIME SILVERIO ORTEGÓN AGUILAR

M.T.I VLADIMIR VENIAMIN CABAÑAS VICTORIA

M.S.I. RUBÉN ENRIQUE GONZÁLEZ ELIXAVIDE

M.T.I. MELISSA BLANQUETO ESTRADA

Universidad de Quintana Roo
DIVISIÓN DE CIENCIAS E INGENIERÍA

CHETUMAL, QUINTANA ROO, MÉXICO, JULIO DE 2019

UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

TRABAJO DE TESIS TITULADO

"EVALUACIÓN DEL DESEMPEÑO DE UN ALGORITMO PARA ESTEGANOGRAFÍA BASADO EN LA TÉCNICA DEL BIT MENOS SIGNIFICATIVO"

ELABORADO POR

PIN SU

BAJO SUPERVISIÓN DEL COMITÉ DE ASESORÍA Y APROBADO COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE

INGENIERO EN REDES

COMITÉ DE TESIS

DIRECTOR: _____
DR. JAVIER VÁZQUEZ CASTILLO

ASESOR: _____
DR. JAIME SILVERIO ORTEGÓN AGUILAR

ASESOR: _____
M.T.I. VLADIMIR VENIAMIN CABAÑAS VICTORIA

ASESOR SUPLENTE: _____
M.S.I. RUBÉN ENRIQUE GONZÁLEZ ELIXAVIDE

ASESORA SUPLENTE: _____
M.T.I. MELISSA BLANQUETO ESTRADA

UNIVERSIDAD DE QUINTANA ROO
AREA DE TITULACION

Universidad de Quintana Roo
DIVISIÓN DE CIENCIAS E INGENIERÍA

CHETUMAL, QUINTANA ROO, MÉXICO, JULIO DE 2019

Agradecimientos

A la primera persona que le quiero agradecer es mi asesor Javier Vázquez, que sin su ayuda y conocimientos no hubiese sido posible realizar este proyecto. A mis padres, por haberme proporcionado la mejor educación y lecciones de vida. Especialmente a mi madre Feng-Pi, por cada día hacerme ver la vida de una forma diferente, dándome la fuerza y confiar en mis decisiones. A mi padre Chien-Tao, por haberme enseñado con mucho esfuerzo, trabajo y constancia todo se consigue, y que en esta vida nadie regala nada. Agradezco de manera particular a mi abuelo, por haberme aportado todo desde pequeño. A mis amigos por estar siempre a mi lado y compartiendo sus experiencias conmigo. A todos mis familiares, por su apoyo. A todos aquellos que siguen cerca de mí y que le regalan a mi vida algo de ellos.

Dedicatoria

A mi familia, principalmente a mis padres por su apoyo incondicional brindado durante este proceso, que sin duda alguna no fue fácil no solo para mí, sino para ellos. Que, con su esfuerzo, juntos hemos logrado esta meta. A ustedes que día a día me demostraron que no existen barreras cuando quieres lograr algo, que las adversidades no son impedimentos, que cada logro es una gran contribución para crecer emocionalmente, laboralmente, pero sobre todo como ser humano. Que con su humildad y entrega han hecho de mí, una persona con valores, capaz de demostrar a los demás una gran esencia de humildad, con lo que sin duda alguna voy a ejercer esta profesión de todo corazón sin nunca olvidar de dónde vengo y hacia dónde quiero llegar.

Resumen

La seguridad de los datos en esta nueva era digital es indispensable para evitar que usuarios no deseados accedan a datos críticos personales. Por ello, en esta tesis se busca iniciar con trabajos dedicados a realizar criptografía y esteganografía. La criptografía es una de las técnicas populares utilizadas para la comunicación segura de datos confidenciales. Dado que la técnica de cifrado de datos produce un flujo de código sin sentido para la transmisión, puede atraer a un intruso para alterar el mensaje intencionalmente o para recuperar el mensaje mediante la explotación de varios ataques criptográficos en los datos cifrados.

En contraste, la esteganografía es otro mecanismo para proteger el secreto de los datos. No altera los datos para que no tenga sentido para el intruso. En este mecanismo, los datos secretos se incorporan en cualquier otro soporte o portada insospechada como medios de imagen, audio, video, etc. para formar un mensaje significativo.

Este trabajo tiene como objetivo general el de evaluar el desempeño de un algoritmo para esteganografía basado en la técnica de bit menos significativo (LSB por su sigla en inglés). Para lo anterior, dos algoritmos han sido implementados y fue medido su desempeño mediante las métricas definidas como: Índice de imagen de calidad universal (UIQI), peak signal-to-noise ratio (PSNR), mean square error (MSE), structural similarity index (SSIM), y coeficiente de correlación.

Contenido

Índice de Figuras

Capítulo 1 Introducción

En el mundo digital, uno de los problemas principales y esenciales es proteger el secreto de los datos confidenciales durante su transmisión a través de un canal público. En general, los datos digitales confidenciales se procesan previamente antes de su transmisión a través de un canal público. Esta operación de pre-procesamiento cambia el contenido de la información a otra forma, pero solo una persona autorizada puede ejecutar adecuadamente la operación reversible en los datos modificados para recuperar el contenido original. Se han ideado varias técnicas de protección de datos para proteger la confidencialidad de los datos digitales. La criptografía es una de las técnicas populares utilizadas para la comunicación segura de datos confidenciales. Dado que la técnica de cifrado de datos produce un flujo de código sin sentido para la transmisión, puede atraer a un intruso para alterar el mensaje intencionalmente o para recuperar el mensaje mediante la explotación de varios ataques criptográficos en los datos cifrados.

En contraste, la esteganografía es otro mecanismo para proteger el secreto de los datos. No altera los datos para que no tenga sentido para el intruso. En este mecanismo, los datos secretos se incorporan en cualquier otro soporte o portada insospechada como medios de imagen, audio, video, etc. para formar un mensaje significativo que se conoce como stego-media. La esteganografía conocida como el arte de ocultar mensajes sin ser detectados ha despertado gran interés por militares, o personas civiles con diferentes propósitos. La facilidad de ocultar información sin ser detectada en diferentes medios como imágenes digitales, archivos de audio o videos ha posibilitado que nuevos esquemas de envío de información sean posibles y sin que la seguridad de cierta información confidencial sea altamente comprometida. Este proyecto busca investigar y presentar las distintas técnicas de esteganografía reportadas en la bibliografía abierta, así como también, implementar algoritmos de esteganografía basados en la técnica del bit menos significativo (LSB por sus siglas en inglés), y cuyo desempeño será medido con base a las siguientes métricas definidas como: Índice de imagen de calidad universal (UIQI), peak signal-to-noise ratio (PSNR), mean square error (MSE), structural similarity index (SSIM), y coeficiente de correlación. Hoy en día, es común que este tipo de técnicas sean utilizadas en el comercio durante el proceso de firmas digitales, aplicaciones

militares, y en general para el ocultamiento de información sensible para un determinado usuario.

Entre las técnicas de esteganografía más utilizadas se encuentra la técnica de sustitución de bits, que es la de sustitución de los bits menos significativos de los cuales se compone un archivo con información. Por ejemplo, cualquier archivo multimedia (archivo con información de audio, imágenes, etc) contiene áreas de datos poco significativos, los cuales se pueden sustituir por otros datos, realizando cambios que son inapreciables visualmente (o auditivamente). Con lo anterior, se permite ocultar información dentro de un archivo portador, haciendo que el mismo parezca igual al original. Es difícil distinguir los medios de comunicación stego de los medios de cobertura originales por la percepción visual humana. Por lo tanto, en comparación con la criptografía, el proceso esteganográfico evita que un destinatario involuntario sospeche que se están transmitiendo datos secretos a través de un canal público a través de medios de cobertura significativos. Un sistema de seguridad basado en la esteganografía se utiliza en diversas aplicaciones, como comunicaciones militares, empresas comerciales, Internet de las cosas y multimedia [7–9]. Varios esquemas criptográficos y esteganográficos combinados [9,10] se encuentran en la literatura. Si bien el objetivo de los esquemas criptográficos y esteganográficos es garantizar la seguridad de los datos, el enfoque combinado de criptografía y esteganografía mejora aún más el sistema de seguridad con una mayor sobrecarga computacional. Por lo tanto, estos dos mecanismos de seguridad, es decir, la criptografía y la esteganografía, se explotan claramente en el campo de la seguridad de la información.

Los datos de imagen se usan frecuentemente en varias aplicaciones. En la literatura, se encuentran varios esquemas esteganográficos basados en imágenes para compartir datos digitales confidenciales de una manera segura. Entre ellos, el método de sustitución de bit menos significativo (LSB) [11] es uno de los métodos más utilizados debido a su sencillo proceso de incrustación y alta capacidad de ocultación. En el método LSB, 1 bit consiste en alterar el bit menos significativo de las muestras de las que se compone el portador (archivo con información), siendo esta la razón de la denominación de la técnica “Least Significant Bit o LSB”. La teoría nos dice que con la sustitución de los bits menos significativos no se pueden detectar los cambios de una manera sencilla. Una característica,

es que el método de sustitución LSB no incrementa el tamaño del archivo portador. También, es posible implementar técnicas LSB de más de un 1 bit; sin embargo, es el archivo con información a enviarse sufrirá un deterioro o distorsión impactando en la calidad de la información a trasmitir (E.d., tratándose de una imagen, ésta podrá cambiar su intensidad en los pixeles cuando se vea alterada).

1.1 Objetivos

1.1.1 Objetivo General

Evaluar el desempeño de un algoritmo para esteganografía basado en la técnica de bit menos significativo (LSB por su sigla en inglés).

1.1.2 Objetivos Particulares:

- Revisión de metodologías propuestas en la literatura para realizar esteganografía en imágenes a color.
- Definir la metodología a seguir en el proyecto de tesis para realizar esteganografía en imágenes a color.
- Proponer un algoritmo para realizar la esteganofía en las imágenes a color.
- Definir la métrica de desempeño del algoritmo propuesto.

1.2 Justificación

La seguridad de los datos en esta nueva era digital es indispensable para evitar que usuarios no deseados accedan a datos críticos personales. Es por eso por lo que este trabajo busca iniciar con trabajos dedicados a realizar criptografía y esteganografía, la cual ayuda a formar a estudiantes en el proceso digital de señales orientados a seguridad.

1.3 Antecedentes

La criptografía protege los datos al mezclarlos utilizando las técnicas y algoritmos disponibles. Así, los datos se modifican en el origen (transmisor de la información) y se convierten nuevamente en su formato original en el destino [1] (receptor de la información). La criptografía es la práctica y el estudio de técnicas para asegurar la comunicación y los datos en presencia de adversarios [6]. La criptografía es un método cuyo objetivo principal es cifrar y proteger un mensaje o archivo por medio de un algoritmo, usando una o más claves, sin ellas será realmente difícil obtener el archivo original. En nuestros tiempos, la protección de la información cada vez se vuelve una necesidad indispensable. Debido al gran crecimiento y auge de los sistemas informáticos, una gran parte de nuestra vida diaria se rige y ocupa información que se guarda en una computadora. Aún peor, el auge del Internet y de la banda ancha, pone a disposición de una gran cantidad de gente, equipos que contienen información delicada para muchos de nosotros, como direcciones, teléfonos e información financiera entre otras [2].

1.3.1 Seguridad

La necesidad de seguridad de la información ha cambiado en las últimas décadas. Antes del uso de las computadoras, la seguridad de la Información era proporcionada por medios físicos; por ejemplo, el uso de cajas fuertes y por medidas administrativas, como los procedimientos de clasificación de documentos. Con el uso de la computadora, y más aún con la llegada de Internet, fue indispensable el uso de herramientas automatizadas para la protección de archivos y otro tipo de información almacenada en la computadora, algunas de estas herramientas son los cortafuegos, los sistemas detectores de intrusos, el uso de sistemas criptográficos o estenográficos. Estas herramientas permiten proteger a la información, además de los sistemas informáticos que son los encargados de administrar la información. De la necesidad por proteger a la información y a los sistemas que la administran surge el término de Seguridad Informática. Tal como menciona [1] “el hecho de que actualmente los términos de seguridad, seguridad de la información y de seguridad informática han sido empleados de diversas maneras y se les han dado diversos significados de acuerdo con el contexto. Los siguientes párrafos son definiciones que tratan de ilustrar uno de los significados más comunes a cada término”.

a) Seguridad: De acuerdo con el diccionario de la Real Academia Española, seguridad es:
   - Cualidad de seguro.
   - Dicho de un mecanismo: Que asegura algún buen funcionamiento, precaviendo que este falle, se frustre o se violente.

b) Seguridad de la Información: Se puede hablar de la seguridad de la información como el conjunto de reglas, planes y acciones que permiten asegurar la información manteniendo las propiedades de confidencialidad, integridad y disponibilidad de la misma.
   - La confidencialidad es que la información sea accesible sólo para aquéllos que están autorizados.
   - La integridad es que la información sólo puede ser creada y modificada por quien esté autorizado a hacerlo.
   - La disponibilidad es que la información debe ser accesible para su consulta o modificación cuando se requiera.

c) Seguridad Informática: Conjunto de políticas y mecanismos que nos permiten garantizar la confidencialidad, la integridad y la disponibilidad de los recursos de un sistema (Por ejemplo: recursos de un sistema como memoria de procesamiento, espacio de almacenamiento en algún medio físico, tiempo de procesamiento, ancho de banda y por su puesto la información contenida en el sistema).

Una vez mencionadas las definiciones anteriores, para que exista seguridad ya sea de la información o informática hay que garantizar las propiedades de confidencialidad, integridad y disponibilidad. Así, es aquí donde se utiliza a la esteganografía, ya que mediante el uso correcto de sistemas esteganográficos se pretende garantizar las propiedades de confidencialidad e integridad.

En la Figura 1 se muestra el flujo que debe llevarse a cabo a través de una comunicación normal. En este caso, no existe ningún problema de seguridad informática. El mensaje que se envía se recibe sin alteración alguna.

Figura 1 Comunicación normal

El segundo caso, como se muestra en la Figura 2; uno de los problemas más grandes que hay, la interrupción de la transmisión del mensaje, que puede ser ocasionada por fallo del canal o de algún elemento del sistema de comunicación, ya sea de forma natural o intencional. Esto es traducido a un problema de disponibilidad.

Figura 2 Comunicación con interrupción

Como podemos ver en la Figura 3, la interceptación de los datos por un intruso (un intruso es un ente externo al sistema) es algo muy común dentro de las comunicaciones, ya que muchas de las transmisiones son enviadas mediante protocolos que son conocidos por todos y a los mensajes no se les hace ningún tratamiento especial; en otras palabras, viajan tal cual se generan. Lo único que se hace es escuchar todo lo que pasa por el canal sin alterar nada. Este es un problema de confidencialidad.

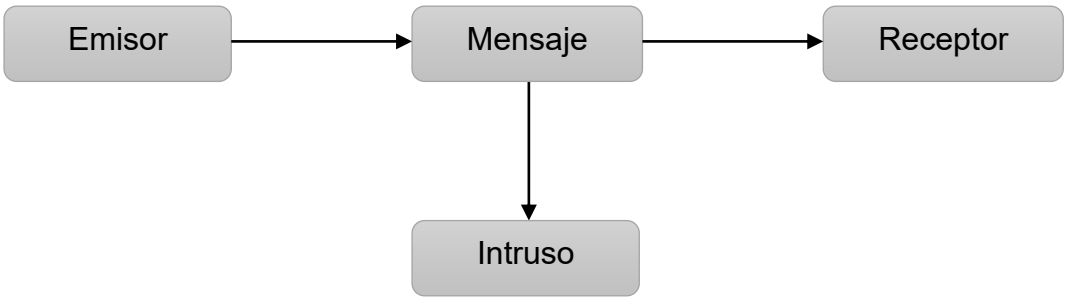


Figura 3 Comunicación con intercepción

Otro problema en la comunicación es el problema de la falsificación. Esto se produce cuando el intruso captura un mensaje, se adueña de él y de la identidad del emisor y genera un nuevo mensaje con la identidad del emisor. Este es un problema de integridad y confidencialidad (Ver Figura 4).

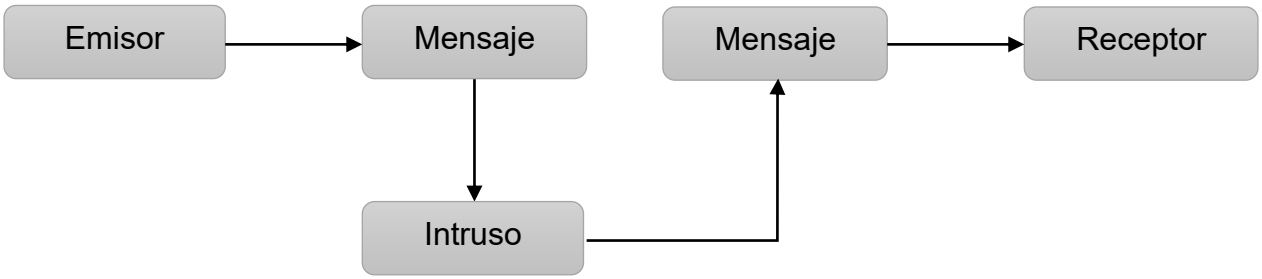


Figura 4 Comunicación con falsificación

Finalmente, como se muestra en la figura 5 la generación de mensajes se da cuando el intruso genera un mensaje engañando al receptor haciéndolo creer que es un emisor válido. Esto se traduce en un problema de integridad.

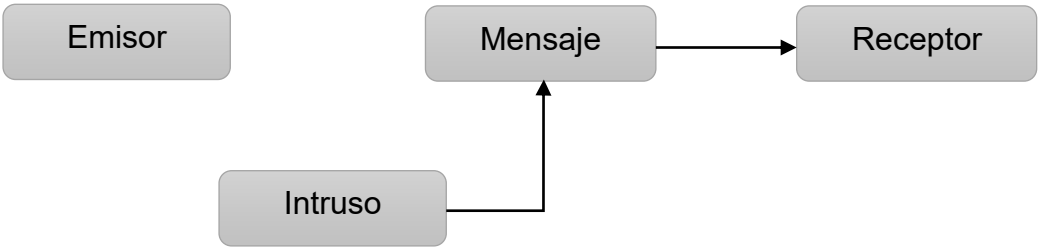


Figura 5 Generación de una comunicación apócrifa

Es muy fácil ver como una comunicación y un sistema informático son muy similares, ya que en un sistema informático se procesan, almacenan, envían y reciben datos. Ahora, si pudiéramos de alguna forma evitar los problemas de disponibilidad, integridad y confidencialidad, tendríamos un sistema seguro. Para lograr esto tendríamos que aislar al sistema de los intrusos y hacerlo anti-fallos lo cual es prácticamente imposible. Lo que se hace es crear mecanismos que garanticen en cierta medida las propiedades de disponibilidad, integridad y confidencialidad.

La disponibilidad generalmente se trata de solucionar con sistemas redundantes. La confidencialidad se puede lograr usando un mecanismo que, aunque sea robada la información, permita que no se pueda acceder a ésta o garantice de alguna forma que no se pueda llegar a ella, hasta que pierda su valor.

La integridad es más difícil de lograr y se hace con el uso de varios mecanismos que garantizan la identidad de un ente que está autorizado por el sistema para crear o hacer modificaciones a la información, de tal forma que se puede verificar posteriormente quién creó o modificó la información. Además, estos mecanismos permiten ver si la información ya creada ha sufrido o no alguna modificación no autorizada [2].

Los mecanismos para garantizar la integridad y la confidencialidad se pueden implementar con sistemas estenográficos, de ahí la importancia de la esteganografía en la seguridad de la información.

Por lo que surge la necesidad de tener bajo llave sus datos y para esto es necesario conocer de forma detallada el concepto de seguridad de información o seguridad informática partiendo de los 4 niveles básicos de seguridad los cuales se muestran en la Tabla 1.

Tabla 1 Niveles básicos de seguridad

Nivel	Especificación
Aplicación	➢ Es lo que ve el usuario. ➢ Es el nivel más complejo y el menos fiable. ➢ La mayor parte de los fraudes informáticos ocurren en este nivel.
Middleware	➢ Implicados en los sistemas de gestión de BD y la manipulación del software.
Sistema Operativo	➢ Se trata la gestión de ficheros y las comunicaciones.
Hardware	➢ Es el nivel menos complejo y más fiable. ➢ Características de seguridad en el CPU y en el hardware (ejemplo, para evitar desbordamientos de buffer o pila).

1.3.2 Criptografía

La criptografía es un método cuyo objetivo principal es cifrar y proteger un mensaje o archivo por medio de un algoritmo, usando una o más claves, sin ellas será realmente difícil obtener el archivo original. En nuestros tiempos, la protección de la información cada vez se vuelve una necesidad indispensable. Debido al gran crecimiento y auge de los sistemas informáticos, una gran parte de nuestra vida diaria se rige y ocupa información que se guarda en una computadora. Aún peor, el auge del Internet y de la banda ancha, pone a disposición de una gran cantidad de gente, equipos que contienen información delicada para muchos de nosotros, como direcciones, teléfonos e información financiera entre otras [12].

En la Internet es relativamente fácil realizar este tipo de engaño, pues uno nunca puede estar seguro de que el correo electrónico realmente proviene del remitente o si nos estamos comunicando realmente con nuestro banco o con otro sitio apócrifo que aparenta ser el banco, por ello es que el uso de algoritmos criptográficos es muy útil. Aun así, la criptografía por sí sola no resuelve todos los problemas, así que es necesario utilizar toda una infraestructura que le proporcione fortaleza, evitar el engaño y asegurar autenticidad e integridad.

Clasificación de la criptografía

La criptografía se puede clasificar históricamente en dos: La criptografía clásica y la criptografía moderna:

La criptografía clásica es aquella que se utilizó desde antes de la época actual hasta la mitad del siglo XX. También puede entenderse como la criptografía no computarizada o mejor dicho no digitalizada. Los métodos utilizados eran variados, algunos muy simples y otros muy complicados.

Se puede decir que la criptografía moderna se inició después de tres hechos: el primero fue la publicación de la Teoría de la Información por Shannon; el segundo, la aparición del estándar del sistema de cifrado.

Tanto la criptografía clásica como la moderna se clasifican de acuerdo a las técnicas o métodos que se utilizan para cifrar los mensajes. Esta clasificación la podemos ver en la Figura 6.

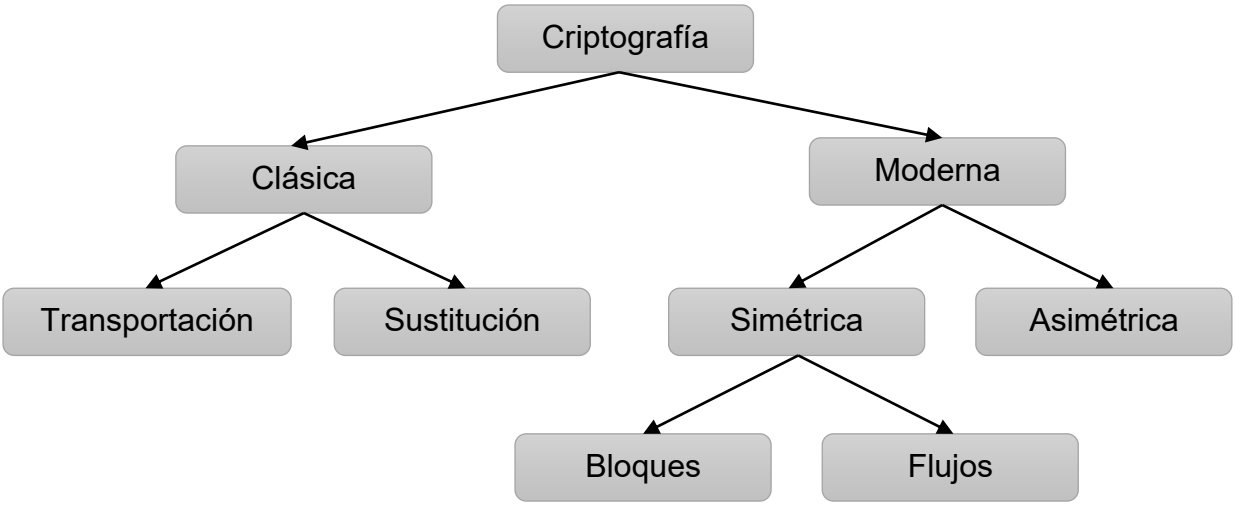


Figura 6 Clasificación de la criptografía

Criptografía clásica

Como se mencionó anteriormente, la criptografía clásica es muy antigua. Las técnicas criptográficas eran muy ingeniosas y se usaban para enviar mensajes secretos entre las personas que tenían el poder o en época de guerra para enviar instrucciones. A diferencia de la criptografía moderna, el algoritmo del sistema criptográfico se mantenía en secreto. La criptografía clásica también incluye la construcción de máquinas las cuales, mediante mecanismos, comúnmente engranes o rotores, transformaban un mensaje en claro a un mensaje cifrado como la máquina Enigma usada en la Segunda Guerra Mundial.

Asimismo, se utilizan cifrados por transposición que usan la técnica de permutación de forma que los caracteres del texto se reordenan mediante un algoritmo específico.
Por otra parte, el método basado en sustitución utiliza la técnica de modificación de cada carácter del texto por otro que corresponde al alfabeto de cifrado. Si el alfabeto de cifrado es el mismo que el del mensaje o bien el único, hablamos entonces de cifradores mono-alfabéticos; es decir, existe un único alfabeto en la operación de transformación del

mensaje en criptograma. De lo contrario, si en dicha operación intervienen más de un alfabeto, se dice que el cifrador es poli-alfabético.

Es realmente interesante analizar cada una de las técnicas anteriores, se describirán dos técnicas en este caso: un cifrado de transposición de grupos, la escítala y un ejemplo de sustitución monoalfabética, monográmica con el alfabeto estándar conocido como el cifrado César.

- La escítala

En siglo V a.c. los lacedemonios, un antiguo pueblo griego, usaban el método de la *escítala* para cifrar sus mensajes. El sistema consistía en una cinta que se enrollaba en un bastón sobre el cual se escribía el mensaje en forma longitudinal, como se muestra en la figura 7:

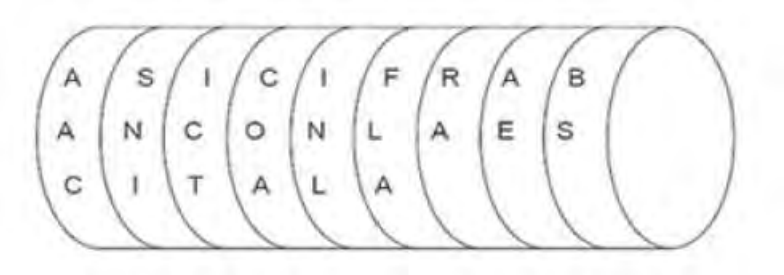


Figura 7 Escítala

Habiendo escrito el mensaje, la cinta se desenrollaba y se entregada al mensajero. Para enmascarar completamente la escritura, aunque la cinta debería tener caracteres en todo su contorno. Como es de esperar, la llave del sistema residía precisamente en el diámetro de aquel bastón, de forma que solamente el receptor autorizado tenía una copia exacta del mismo bastón en el que enrollaba el mensaje recibido y, por tanto, podía leer el texto en claro.

- El cifrado César

El cifrado del César aplica un desplazamiento constante de tres caracteres al texto en claro, de forma que el alfabeto de cifrado es el mismo que el alfabeto del texto en claro, pero desplazado 3 espacios hacia la derecha módulo n, con n el número de letras del mismo. A continuación, se muestra el alfabeto y la transformación que realiza este cifrador por sustitución de caracteres para el alfabeto castellano de 27 letras.

Tabla 2 Alfabeto y transformación del cifrado Cesar

Alfabeto	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
Alfabeto	Z	Y	X	W	V	U	T	S	R	Q	P	O	Ñ	N	M	L	K	J	I	H	G	F	E	D	C	B	A

Así con este alfabeto podemos cifrar el siguiente mensaje:

Mensaje original: MENSAJE DE PRUEBA

Mensaje cifrado: OHPVDM GH SUXHED

Al describir el cifrado de César se utilizó un concepto muy usado en las matemáticas y más en criptografía, el módulo.

El módulo es una operación binaria que se realiza en los enteros positivos y se representa de la siguiente forma: *c* = *a* mod *b* de tal forma que *a*, *b* y *c* son enteros positivos.
El valor de *c* al realizar la operación *c* = *a* modulo *b* es igual al residuo de dividir a entre *b*. Se puede observar claramente que 0 <= c < b.

Con este antecedente podemos escribir en forma matemática el cifrado de César de la siguiente forma:

Para cifrar

$Ci = (3 + Mi) \bmod 27$

con i= 0,1,…,n ; n = número de letras del mensaje

donde *Ci* es la letra cifrada y *Mi* es la letra a cifrar

el alfabeto comienza con A = 0, B=1, …, Z=26

Para descifrar

$Mi = (Ci - 3) \bmod 27 = (Ci + 24) \bmod 27$

con i= 0,1, …, n; n = número de letras del mensaje

donde *Ci* es la letra cifrada y *Mi* es la letra a cifrar

el alfabeto comienza con A = 0, B=1, …, Z=26

Criptografía moderna

La criptografía moderna se puede clasificar en dos grandes grupos: la criptografía de llave secreta o simétrica y la criptografía de llave pública o asimétrica.

- Criptografía simétrica:

La criptografía simétrica o de llave secreta es aquella que utiliza algún método matemático llamado sistema de cifrado para cifrar y descifrar un mensaje utilizando únicamente una llave secreta. Se puede observar en la figura 8, que la línea punteada es el eje de simetría: lo mismo que hay de un lado existe exactamente igual en el otro, esto ilustra el hecho del porqué se le da el nombre de criptografía simétrica [13].

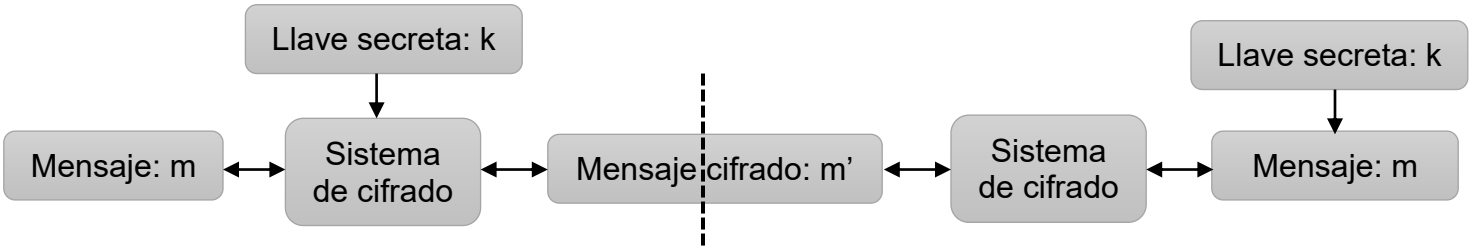


Figura 8 Criptografía simétrica

Este tipo de criptografía sólo utiliza una llave para cifrar y descifrar, esto es: si yo cifro un mensaje *m* con una llave secreta *k* entonces el mensaje cifrado resultante *m'* únicamente lo voy a poder descifrar con la misma llave *k*. Este tipo de llave conocida como secreta se debe de compartir entre las personas que se desea que vean los mensajes.
Con este tipo de criptografía podemos garantizar la confidencialidad porque únicamente quien posea la llave secreta será capaz de ver el mensaje.

Criptografía simétrica por bloques:

Este tipo de criptografía está basado en el diseño propuesto por Horst Feistel en los años 70.

- Diseño de Feistel:

Un bloque de tamaño N bits comúnmente N=64 o 128 bits se divide en dos bloques de tamaño N/2, A y B. A partir de aquí comienza el proceso de cifrado y consiste en aplicar una función unidireccional (muy difícil de invertir) a un bloque B y a una subllave k1 generada a partir de la llave secreta. Se mezclan el bloque A con el resultado de la función mediante un XOR. Se permutan los bloques y se repite el proceso n veces. Finalmente se unen los dos bloques en el bloque original.

Tabla 3 Algoritmos de cifrado de bloque

Algoritmo	Bloque (bits)	Llave (bits)	Vueltas
Lucifer	128	128	16
DES	64	56	16
Loki	64	64	16
CAST	64	64	8
Blowfish	64	Variable	18

Criptografía Simétrica de Flujo

Este tipo de criptografía se basa en hacer un cifrado bit a bit, esto se logra usando la operación XOR, representada con Å. Se utiliza un algoritmo determinístico que genera una secuencia pseudoaletoria de bits que junto con los bits del mensaje se van cifrando utilizando a operación XOR.

- *Criptografía asimétrica*

Si se observa la Figura 9, la cual muestra la idea de criptografía de llave pública, se puede ver claramente que no existe simetría en ella, ya que de un lado de la figura se cifra o descifra con una llave pública y en el otro lado con una privada. De este hecho es de donde la criptografía asimétrica debe su nombre [13].

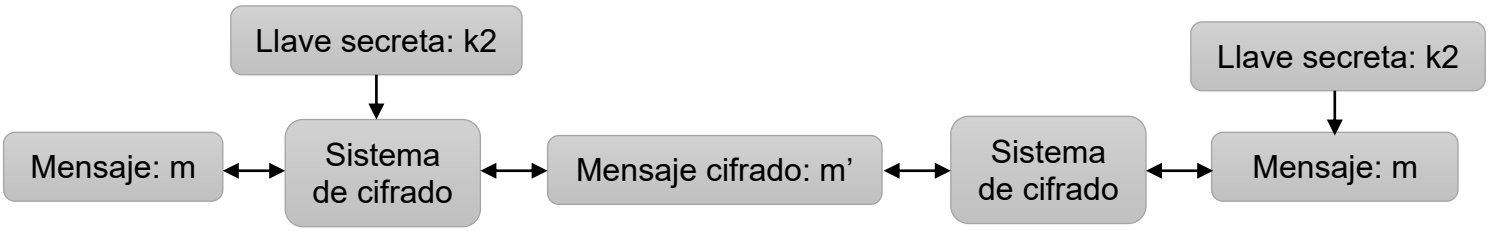


Figura 9 Criptografía asimétrica

Es importante destacar que para este tipo de criptografía lo que se cifra con una llave se puede descifrar con la otra llave. Es decir, yo puedo cifrar con la llave pública y descifrar con la privada y viceversa. Esto es de gran ayuda ya que el número de llaves que debo de poseer se reduce considerablemente.

Proceso criptográfico

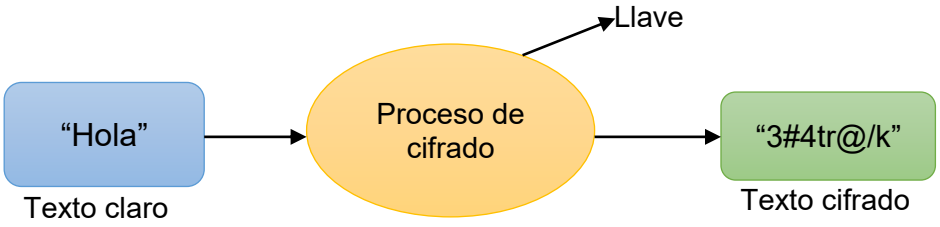


Figura 10 Encriptado o cifrado de datos

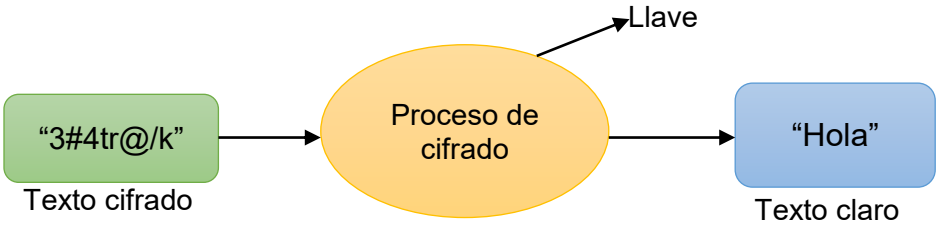


Figura 11 Desencriptado o descifrado de datos

1.3.3 La esteganografía

Del griego steganos (oculto) y grafos (escritura), la esteganografía se puede definir como la ocultación de información en un canal encubierto con el propósito de prevenir la detección de un mensaje oculto.

La esteganografía estudia el conjunto de técnicas cuyo fin es insertar información sensible dentro de otro archivo. A este archivo se le denomina fichero contenedor (gráficos, documentos, programas ejecutables, etc.). De esta manera, se consigue que la información pase inadvertida a terceros, de tal forma que sólo sea recuperada por un usuario legítimo que conozca un determinado algoritmo de extracción de la misma.

Se pueden observar distintos actores implicados en el campo de la esteganografía:

- Objeto contenedor: se trata de la entidad que se emplea para portar el mensaje oculto. Acudiendo al ejemplo de los mensajes sobre el cuero cabelludo, el objeto contenedor es el esclavo en sí.
- Estego-objeto: se trata del objeto contenedor más el mensaje encubierto. Siguiendo con el ejemplo, se trata del esclavo una vez se ha escrito en su cuero cabelludo el mensaje y se le ha dejado crecer el pelo.
- Adversario: son todos aquellos entes a los que se trata de ocultar la información encubierta. El adversario puede ser pasivo o activo. Un adversario pasivo

sospecha que se puede estar produciendo una comunicación encubierta y trata de descubrir el algoritmo que se extrae del estego-objeto, pero no trata de modificar dicho objeto. Un adversario activo, además de tratar de hallar el algoritmo de comunicación encubierta, modifica el estego-objeto con el propósito de corromper cualquier intento de mensajería subliminal.

- Estegoanálisis: ciencia que estudia la detección (ataques pasivos) y/o anulación (ataques activos) de información oculta en distintas tapaderas, así como la posibilidad de localizar la información útil dentro de la misma (existencia y tamaño).

Teniendo en cuenta que pueden existir adversarios activos, una buena técnica esteganográfica debe ser robusta ante distorsiones, ya sean accidentales o fruto de la interacción de un adversario activo [2].

La esteganografía es una técnica para ocultar información secreta en cualquiera de los medios, como imágenes, texto, audio y video. El mensaje que se oculta está oculto en otro archivo llamado cubierta de medios. La combinación de mensaje secreto y archivo de cobertura se llama as - stego. Que se envía a través de la red desde el origen hasta el destino [14].

Tipos de esteganografía

Esteganografía pura

Los algoritmos de ocultación y extracción de la información, que sólo el emisor y el receptor del mensaje deberían conocer. Este tipo de esteganografía es seguro siempre y cuando el medio por el cual se transmite el mensaje sea inexperto en las habilidades de esteganografía, pero en medios más avanzados se requiere el uso de este método combinado con criptografía, ya que, si no se combina podría resultar desastroso para el objetivo que se busca.

Un ejemplo claro de esto ocurre en una cárcel, cuando dos prisioneros planean escapar, pero sus notas tienen que pasar necesariamente por el guardia de seguridad quien las revisa antes de entregarlas, su plan lo realizan mediante notas al parecer indefensas, las cuales, al ser puestas a contraluz, revelaban el mensaje oculto, este es un claro ejemplo de la aplicación de la esteganografía pura, cuando el medio que transporta el mensaje, omite los mensajes ocultos que puede contener los ficheros que transporta (ver figura 12).

Figura 12 mensaje anfitrión y el problema de los prisioneros

Esteganografía de clave privada o secreta

Tomando en cuenta que un atacante podría conocer los algoritmos de ocultación y extracción de la información. Por lo tanto, el mensaje se cifra utilizando un cifrado simétrico antes de ocultarlo. De esta manera, si el atacante intercepta la transmisión y logra extraer la información aún tendrá que enfrentar el proceso de descifrar el mensaje. Como se mencionó anteriormente para evitar el fracaso a la hora de transmitir mensajes, es necesario agregar otro nivel de seguridad más a el mensaje oculto y es aquí donde la esteganografía gana gran fortaleza frente a otras técnicas de ocultamiento de información, tan sólo se agrega un nuevo parámetro al estego-algoritmo el cual es comúnmente conocido como estego-clave, la estego-clave debe ser socializada entre el emisor y el receptor antes de la comunicación, puede llegar a ser desde que metro de la cinta se debe leer, cada cuántas revoluciones de cassette se debe capturar una letra o incluso que intensidad de luz es la óptima para poder ver el mensaje, de esta manera se tiene infinidad de formas de esconder y es por esto que es casi imposible descifrar el contenido de la comunicación si no se cuenta con la estego-clave.

1.3.3.1 Esteganografía de clave pública:

Por último, este tipo de esteganografía utiliza dos claves y su principal característica es que no requiere un intercambio previo de estego-clave. Requiere de dos claves, una secreta que se utiliza al momento de realizar la inserción del mensaje secreto

y una pública, la cual se guarda en las bases de datos públicas. La estego-clave se usa para reconstruir el mensaje.

1.3.4 Propiedades de un sistema de esteganografía robusto

La seguridad de estos sistemas no debe estar basada en la ocultación de los algoritmos utilizados, sino en la fortaleza de los mismos y en la seguridad de la clave.
Entre las propiedades deseables de un sistema de esteganografía se encuentran la robustez, la resistencia a las manipulaciones, imperceptibilidad, el costo computacional y la baja probabilidad de error.

1.3.4.1 Robustez

Los archivos digitales como imágenes, audio y video están expuestos a muchos tipos de modificaciones o distorsiones entre las cuales se encuentran las pérdidas por compresión, los cambios producidos por el mejoramiento de imágenes, la amplificación de las señales de audio, etc. Un sistema de esteganografía es considerado robusto si conserva sus características después de esas operaciones en imágenes y video, también se deben conservar estas características después de aplicar alguna técnica o algoritmo esteganográfico. Es decir, que el texto oculto (mensaje) presente en los archivos debe poder ser recuperado después de las distorsiones. Para consolidar su robustez, los sistemas esteganográficos, deben insertar el texto en regiones perceptualmente significativas de los archivos multimedia.
La robustez no debe exigirse incondicionalmente, ya que el sistema de algoritmos puede necesitar ser robusto respecto a determinados procesos y frágil respecto a otros. Si un sistema esteganográfico requiere que ciertas modificaciones de los archivos dañen la información oculta, se le denomina sistema esteganográfico frágil.

1.3.4.2 Resistencia a manipulaciones

La resistencia a manipulaciones de un sistema esteganográfico es un aspecto que puede relacionarse con la seguridad del mismo; se refiere a su resistencia frente a los ataques

hostiles basados en el total conocimiento de los algoritmos de incrustado y detección y de los archivos marcados, excepto de la clave utilizada.

1.3.5 Modelo general de la esteganografía

Para establecer un modelo de comunicación secreta entre dos partes, muchas de las aplicaciones esteganográficas siguen el modelo que se describe en la figura 13. En el que el emisor escoge aleatoriamente un objeto que servirá de cubierta (C), el cual puede ser transmitido sin levantar sospecha de ningún atacante.

Posteriormente, un mensaje secreto (M) es oculto dentro del objeto (C) mediante el uso de la estego-clave (K) para dar como resultado final el estego-objeto (S); entonces, el estego-objeto es enviado al receptor haciendo uso de un canal público inseguro.

Luego de que el estego-objeto es obtenido por el receptor, puede recuperar el mensaje oculto, haciendo uso de la estego-clave.

Cabe mencionar que en un sistema como el que se describió, es importante no volver a utilizar el mismo objeto de cubierta para enviar varios mensajes ocultos, evitando así llamar la atención de los atacantes.

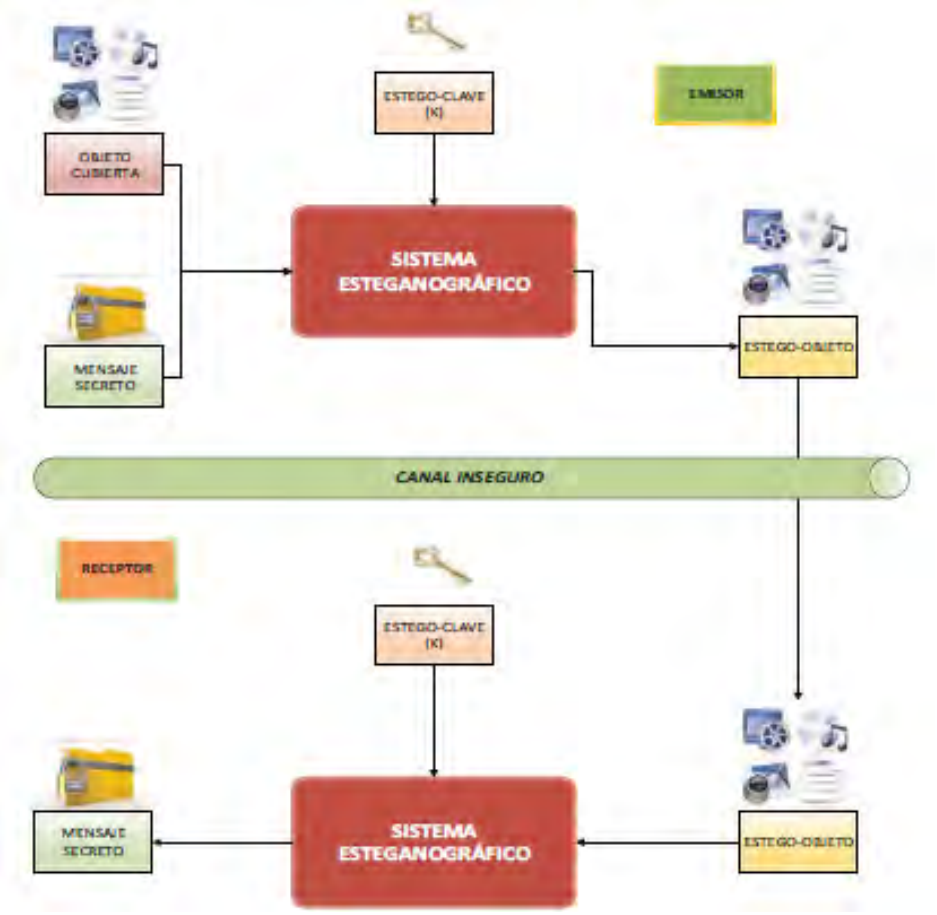


Figura 13 Modelo general de esteganografía

1.3.6 Aplicación de la esteganografía

La esteganografía se utiliza con mejores o peores resultados. Veremos algunos de sus usos reales, como pueden ser las marcas de agua, transmisión de mensajes terroristas, detección de copyright y algún que otro uso. Siendo un medio para almacenar información de forma que oculta la existencia de esa información. Tomando en cuenta los métodos de comunicación existentes, la esteganografía puede ser utilizada para realizar intercambios ocultos. Los gobiernos están interesados en dos tipos de comunicaciones: las que apoyan la seguridad nacional y los que no lo hacen. Esteganografía digital proporciona un amplio potencial para ambos tipos. Negocios pueden tener preocupaciones similares con respecto a secretos comerciales o información sobre nuevos productos. Evitar la comunicación en formas bien conocidas reduce en gran medida el riesgo de información que se filtró en tránsito. [15]

La esteganografía de imágenes tiene muchas aplicaciones en distintos ámbitos dentro de esta era digital, por lo que se estable la siguiente clasificación:

- Integridad y autenticación de objetos.
- Protección frente a copias ilícitas.
- Etiquetas, números de serie y huellas digitales
- Diferentes niveles de acceso a datos.

1.3.6.1 Integridad y autenticación de objetos

Un ejemplo de aplicación de estos procedimientos se encuentra en el campo de la seguridad y vigilancia frente a robos u otros delitos semejantes. Frecuentemente una cámara de video que registra las imágenes del lugar bajo vigilancia, con el propósito de que estas sirvan como prueba en un caso (juicio) pero solo si se demuestra su integridad y autenticidad serán tomadas como válidas.

Ambas propiedades integridad y autenticidad, pueden garantizarse mediante métodos criptográficos, pero también se puede emplear el watermarking o marca de agua digital que es una técnica esteganográfica que consiste en insertar un mensaje (oculto o no) en el interior de un archivo digital, que contiene información sobre el autor o propietario intelectual del objeto digital tratado, la cual tiene como objetivo principal poner el uso ilícito de cierto servicio digital por parte de un usuario no autorizado [16].

1.3.7 Marca de agua

La idea básica de la marca de agua consiste en insertar información en la imagen mediante la realización de modificaciones sobre la misma, con el objetivo de proporcionar pruebas sobre quién es el propietario de la imagen o a quién ha sido vendida o enviada. La realización de este proceso sobre la imagen deberá ser imperceptible para el ojo humano, sin afectar por tanto a su calidad. El objetivo es facilitar el control que se hace de materiales protegidos por derechos de autor, no sólo garantizar que el acceso a la información es el acordado, sino principalmente que no se va a hacer un uso ilegítimo de la misma; por ejemplo, comerciando con segundas copias. Es en este punto donde supone un importante valor añadido para los autores ya que con los sistemas criptográficos podemos autenticar al comprador y garantizar que el material ha sido

entregado a la persona esperada y sin que nadie haya podido realizar copias ilícitas durante la transmisión. Sin embargo, una vez que el material multimedia está en poder del comprador, éste puede usarlo, copiarlo, revenderlo, etc., sin control por parte del autor. Es aquí donde se detecta un grave problema. [17]

Figura 14 Marca de agua

- **Marca de agua sobre papel**

Antes de hablar sobre las marcas de agua digitales y sus aplicaciones en la informática es necesario hablar sobre las marcas de agua sobre papel. Durante el proceso de fabricación del papel y cuando este aún se encuentra húmedo, se emplea una especie de cilindro el cual tiene adherido una rejilla con la señal o la marca que se desea emplear, en muchos casos se utilizan dibujos, escudos o alguna señal que hace distintivo el papel.

En la actualidad esta técnica es usada, por ejemplo, en el papel de los billetes, el objetivo principal de utilizar esta técnica es evitar que los billetes sean falsificados, aunque también se utiliza cuando se desea dar cierta validez a algún libro o estudio impreso en papel además de dotar al libro de información como año en que se elaboró o el lugar de procedencia del mismo.

Figura 15 Marca de agua sobre billete de 50 euros

Una de las principales aplicaciones de las marcas de agua que también se conocen como filigranas es en los billetes para evitar que estos sean falsificados, este método consiste en ocultar información ya que estas marcas no son muy visibles. La marca aparece solo si se observa el billete a contraluz. Se considera una técnica esteganográfica ya que utiliza un objeto para ocultar información.

- **Marca de agua digital**

La era digital ha llegado y la necesidad de manejar información por la red es importante para cualquier persona o compañía, toda la información almacenada en libros y en documentos importantes tienden a convertirse en archivos digitales. La necesidad de las personas de ser reconocidos como los autores originales de todos estos trabajos es un reto para el cual se proponen las marcas de agua digitales como solución.

Esta técnica se creó como solución a los derechos de copia o copyright de archivos como documentos, imágenes, audio, video. La idea principal de este método es crear una marca que sea inseparable de todos estos archivos donde se pueda conservar información como autor, propietario, distribuidor y evitar el plagio de esta información. Las principales características de esa técnica es que debe ser invisible a la persona que quiera observar su contenido, al momento de implementar esta técnica se debe proteger el contenido que se quiere proteger, es decir, no se debe degradar la información.

Esta técnica consiste en insertar un código directamente al archivo ya sea una imagen, audio o video como lo indica. Este código funciona como un identificador que está asociado al autor, en ocasiones se utilizan otros medios como huellas digitales para reforzar la seguridad del sistema. Este modelo debe ser invisible a terceros que quieran hacer plagio del contenido, pero deber ser accesible mediante generalmente el uso de un algoritmo y una contraseña. [17]

1.3.8 Protección referente a copias ilícitas

La protección de los derechos de propiedad intelectual es probablemente la aplicación más habitual de la esteganografía. Por ejemplo, la disponibilidad de equipos que permiten la realización de múltiples copias de discos compactos ha ocasionado un gran incremento en el mercado de copias ilícitas. Con la finalidad de disminuir este incremento se han diseñado muchos sistemas de protección basados en algoritmos esteganográficos.

1.3.9 Etiquetas números de serie y huellas digitales

En esta área la esteganografía se aplica para conocer las identificaciones tanto del transmisor como el receptor, las cuales serán insertadas dentro de un objeto en este caso una imagen digital. Consiste en números o datos de identificación únicos ocultos en el objeto que se va a proteger, que le brindan al propietario de los derechos de autor, conocer que usuario ha corrompido la licencia de uso proporcionándolo a terceras personas. Estos números se insertan de tal forma que es imposible realizar una copia sin incluir en esta los números de serie, de esta forma se determina la presencia de una copia ilícita y quien la realizo.

1.3.10 Diferentes niveles de acceso a datos

Una última e interesante aplicaciones de la esteganografía es el brindar a los usuarios niveles de acceso múltiples a la información. Estas técnicas permiten la creación de canales ocultos de información accesible solo a determinados usuarios. Por ejemplo, una película difundida en un canal de televisión digital podría incorporar diferentes bandas sonoras en múltiples idiomas.

1.3.10.1 Programas de ocultación.

Hay que dejar clara la diferencia entre la criptografía y la esteganografía: cuando solo se utiliza la criptografía, el dato puede ser ilegible, pero es obvio que allí existen datos ocultos. La forma en que actúan juntos criptografía y esteganografía es que la criptografía hace el dato ilegible a quien no conozca la clave y la esteganografía, oculta además la existencia de esos datos. Así los archivos siendo ocultos, hacen que no sea ni leído ni detectado fácilmente. Actualmente, ambas técnicas son perfectamente combinables, complementándose la una a la otra y obteniendo una seguridad aún mayor. Cuando se oculta información en archivos de imágenes se aprovechan los bits menos significativos de los colores para introducir en ellos la información (en la cual se hace una reducción de colores respecto a la imagen original, si es necesario). Si la relación entre la información a ocultar, el tamaño de la imagen y el número de colores es buena, suele ser prácticamente imposible diferenciar la imagen original de la imagen con información oculta. Cuando se usan archivos de sonido, la información oculta aparece como ruido de fondo, pudiendo confundirse fácilmente con una simple grabación con algo de ruido [17]. Otros métodos de esteganografía más comunes, accesibles y a los que todos hemos tenido acceso sin tener ni idea de que era eso de la esteganografía, son la escritura con tintas invisibles (Por ejemplo: leche, jugo de frutas, vinagre), las cuales al ser expuestas al calor se oscurecen, dejando entrever el mensaje oculto. Igualmente, sencillo e inocente resulta enviar un mensaje escribiendo un texto, y luego, tomando la primera letra de cada palabra, se podía leer el mensaje que en realidad se quería transmitir.

En resumen, la esteganografía es el arte de transmitir información de modo que la presencia de la misma, pase inadvertida, típicamente escondida dentro una imagen, archivo de audio o en su caso video digital [17]. La esteganografía clásica consiste en métodos completamente obscuros, es una protección basada en descomponer el canal encubierto especifico que se está usando. La esteganografía moderna consiste en el uso de canales digitales como por ejemplo archivos de texto, audio digital, imágenes, video, protocoles de comunicaciones (TCP/IP).

1.3.11 Esteganografía en imágenes, audio y video digital

Las imágenes digitales, el audio y el video digital, se han postulado como los estegomedios más propicios para ocultar información. En los últimos años se ha publicado múltiples técnicas esteganográficas centradas en este tipo de contenidos [17], así como nomenclatura en función de cómo se trabaje con la información de estos formatos. Los diferentes procedimientos esteganográficos se clasifican en función de las transformadas o criterios matemáticos que se utilizan para la ocultación de información. Por ejemplo, son famosas las técnicas que trabajan en el dominio de la señal, dominios transformados como la transformada discreta de Fourier, la transformada discreta del coseno (por ejemplo, en ficheros JPEG), transformada Mellin-Fourier, transformada discreta Wavelet y sus coeficientes etc. Independientemente de todos estos conceptos, lo que debe tenerse en cuenta para avanzar en procedimientos esteganográficos en contenidos multimedia es conocer el formato del fichero sobre el que se desea trabajar y acceder a las posiciones donde se almacena la información útil, aquella información que puede ser utilizada con fines esteganográficos.

En unos casos, serán pixeles, en otro caso coeficientes de transformadas matemáticas (por ejemplo, ficheros JPEG), etc., Que portador y qué transformada utilizar dependerá de la aplicación que se desee, la seguridad esperada y la invisibilidad requerida.
Es común que en algunas técnicas que trabajan en el dominio espacial proporcionen mayor capacidad de ocultación, mientras que los del dominio transformado (por ejemplo, dominio de la frecuencia) sean más robustos contra ataques, tales como compresión, o procesamiento, en general, de la señal.

Aunque podrían desarrollarse procedimientos de ocultación basados en cualquiera de las técnicas generales de ocultación mencionadas anteriormente, en la práctica las técnicas de ocultación más utilizadas en formatos multimedia son técnicas de inserción que modifican el fichero fuente para ocultar unos datos. En general, la información añadida actuará a modo de ruido adicional que puede provocar perturbaciones en el archivo digital generado. Por este motivo, es común que las técnicas de ocultación se aprovechen de las limitaciones del sistema de percepción humano (vista y oído) para minimizar que un usuario pudiera darse cuenta a simple vista de la presencia de información oculta.

Por ejemplo, la modificación de los colores de una imagen en una región donde hay presente una variedad de colores es menos apreciable que si se modifica el color de ciertos píxeles en regiones uniformes con algunos colores. Del mismo modo, el ojo humano es muy sensitivo a modificaciones en los bordes de una imagen (en el contorno). A continuación, se van a analizar de forma resumida algunos aspectos interesantes de la ocultación de información de formatos digitales de video, audio e imágenes.

- **Las imágenes digitales**

Los principios en los que se fundamentan las técnicas de ocultación sobre este tipo de archivo digital son dos: que la modificación de la imagen no introduzca un ruido visual que levante sospechas a una persona que vea la imagen y que las modificaciones introducidas no proporcionen pistas adicionales a un intruso. Existe una gran variedad de técnicas para diferentes formatos gráficos de fichero (bmp, gif, jpeg, png, etc.)

Las técnicas y variantes más documentadas de estos procedimientos consisten en la modificación de los LSB (Least Significant Bit) de los píxeles de una imagen, de los índices que enlazan a la paleta de colores de un formato GIF (otros procedimientos como el reordenamiento de los colores de la paleta es posible) o de los coeficientes resultantes de aplicar alguna transformación matemática a una imagen, por ejemplo, los coeficientes DCT (transformada discreta del coseno) del formato gráfico JPEG o los coeficientes Wavelet del formato gráfico JPEG2000.

Posteriormente se va a analizar resumidamente algunas de las técnicas más famosas para los formatos de ficheros BMP, GIF y JPEG. En general, cualquier fichero con formato gráfico podrá ser utilizado con fines esteganográficos.

- **Esteganografía en video digital**

Una opción interesante para ocultar información son los formatos digitales de vídeo debido a que presentan un volumen considerable de datos que se pueden modificar para ocultar información. En 1998, A. Westfeld y G. Wolf [7] demostraron cómo un sistema real, en concreto de videoconferencia, se podía utilizar para establecer un canal oculto de información sin que la señal sufriera una degradación grande. En general, un video

puede definirse como un conjunto de marcos (frames) individuales formado por imágenes y audio (y en ocasiones también texto). Los procedimientos de ocultación de información en un vídeo se pueden aprovechar de las distintas técnicas esteganográficas sobre cada uno de esos elementos individuales (estegomedios) que configuran cada marco, algunos de los procedimientos específicos documentados en videos han sido: codificación de información a partir del cálculo de los vectores de movimiento entre una colección de marco, técnicas basadas en corrección de errores, etc.

- **Esteganografía∫ en audio digital**

En la actualidad el avance de la tecnología ha permitido la creación de dispositivos cada vez más pequeños y autónomos, que facilitan la creación, reproducción, replicación y distribución de audio. Dispositivos cotidianos, portátiles o no, que permiten estas tareas son, por ejemplo, los ordenadores personales, PDAs, móviles, reproductores de diferentes soportes y formatos de audio (lectores de CD/DVD, MP3/MP4, iPod, audiostreaming, radios-online personales, etc.), dispositivos sintonizadores de radio (FM/AM), grabadoras de voz, software informático de procesamiento de audio, etc. Por no hablar de los clásicos instrumentos musicales y los soportes para compartir dicha información, alejadas de un mundo claramente digital, como son las clásicas partitura musical.

El estudio de las limitaciones del sistema de audición humano es el punto de partida para el diseño correcto de un algoritmo esteganográfico que oculte información en señales de audio [8]. El sistema de audición humano es bastante más difícil de engañar que otros sentidos, por ejemplo, que el sistema de visión.

El oído presenta una sensibilidad alta a la presencia de un ruido blanco gaussiano añadido a una señal de audio. Esta detección puede ser incluso de unos 70 dB por debajo del nivel de ruido ambiente. Sin embargo, existen diversas situaciones en las que el sistema puede ser engañado, es decir, es impreciso en la detección. Una de estas situaciones consiste en que ante la presencia de sonidos fuertes y sonidos más débiles los primeros tienden a enmascarar a los segundos. Por otra parte, el sistema de audición humano es poco sensible a ciertos cambios de fase en una señal de audio, o a la

supresión de ciertas frecuencias en la señal de audio, modificaciones en las que se basa el estándar MPEG-1 audio Layer III (mp3).

Existen diferentes procedimientos esteganográficos y estegoanalíticos en señales de audio: técnicas basadas en LSB (Least Significant Bit) en muestras de audio, técnicas de ocultación en la fase de una señal (modulación de la fase de una señal y codificación en la fase), técnicas de ocultación en el eco de una señal, ocultación aprovechando las características estadísticas de las señales de audio (por ejemplo, segmentación de la señal de audio de forma adaptativa), ocultación basada en algoritmos de compresión (MP3, WMA, OGG), etc.

Sin embargo, al igual que sucede con otros portadores, que la introducción de modificaciones puede crear patrones no comunes que pueden ser detectados, por ejemplo, mediante estudios estadísticos de las propiedades de la señal empleada, en muchos casos varios ataques serían válidos para distintos formatos.

1.3.12 Imágenes digitales

El componente más pequeño de una imagen digital se llama pixel, cada pixel se codifica mediante un conjunto de bits de longitud determinada. Las longitudes más empleadas son: 8 bits, usado por las imágenes en escala de grises, con lo que un pixel puede tomar un valor entre 0 y 255, 24 bits usado para las imágenes a color o con 48 bits usado por imágenes de alta resolución [6].

Una imagen digital tiene diferentes parámetros que la definen, entre los principales se encuentran los siguientes:

- Resolución
- Dimensiones de la imagen
- Profundidad de bits
- Formato de imagen

1.3.12.1 Resolución de una imagen digital

La resolución de una imagen es el indicador de la cantidad de información que hay en pixeles o puntos de impresión, en un área determinada ya sea en centímetros o en

pulgadas. Las unidades más utilizadas son puntos por pulgada, cuando se refiere a una imagen impresa, o pixeles por pulgada, cuando se refiere a una imagen que será mostrada en una pantalla digital. Las medidas estándar más comunes usadas para resolución de imágenes digitales son 72 ppi (pixels per inch) cuando se refiere a visualización de imágenes en pantalla. La Figura 16 muestra una comparación en la cual se puede apreciar la calidad de la imagen en distintas resoluciones.

Figura 16 Imagen a diferentes resoluciones

1.3.12.2 Dimensiones de una imagen digital

Se refiere a las medidas horizontales y verticales de la imagen expresada en pixeles. Se pueden determinar realizando la multiplicación del ancho y de la altura de la imagen, medida en pulgadas, por la resolución puntos por pulgada.

La profundidad de bits de una imagen digital se determina por la cantidad de bits que se utilizan para representar a un pixel y se mide en bits/pixel (ver figura 17).

A medida que aumenta la profundidad de bits, se puede disponer de una mayor gama de colores. Las imágenes digitales se pueden representar a blanco y negro, escala de grises o a color.

Una imagen en blanco y negro cuenta con un bit para representar a cada pixel en donde el 1 representa el color blanco y el 0 el color negro.

Una imagen en escala de grises puede representar a un pixel con 2 hasta 8 bits, generalmente el valor de cada pixel equivale a una tonalidad diferente de color gris, empezando desde el negro más profundo hasta alcanzar el color blanco.

En las imágenes a color se puede representar un pixel con una profundidad entre 8 y 24 bits o más, dependiendo de los detalles de color que se deseen visualizar en la imagen. Por ejemplo, en una imagen que utilice 24 bits de profundidad y el modelo de color RGB

(Red-Green-Blue), usará 8 bits para el color rojo, 8 bits para el color verde y 8 bits para el color azul, los demás colores se obtendrán al realizar combinaciones de estos bits.

Figura 17 Profundidad de bits de izquierda a derecha:

1.3.12.3 Compresión

Se utiliza para reducir el tamaño del archivo de imagen y de esta manera, facilitar su procesamiento, almacenamiento y transmisión.
Básicamente es reducir o eliminar las componentes de color redundante, existente en una determinada área de la imagen, para esto generalmente se utiliza la codificación Hoffman.

Los sistemas de compresión se caracterizan en compresión sin pérdidas y compresión con pérdidas.

La compresión sin pérdidas representa cierta cantidad de información en un espacio menor, siendo posible la recuperación exacta de la información

En la compresión con pérdidas, se representa cierta cantidad de información, usando una menor cantidad de la misma, siendo imposible realizar una reconstrucción exacta de la información original.

1.3.12.4 Formatos de imagen digital

La mayoría de los formatos de imágenes digitales se componen de una cabecera que contiene atributos, por ejemplo, las dimensiones de la imagen, el tipo de codificación, etc.

A continuación, se detallan los formatos de imágenes digitales más comúnmente utilizados:

- Windows BitMaP (BMP): este formato de imagen consiste en una cabecera, seguida de los valores de cada pixel de la imagen (cada pixel puede tomar valores de 4, 8, 16, 24 y 32 dependiendo de la calidad de la imagen). Es el formato de imagen más simple y aunque soporta compresión de imágenes no es aplicada. Dado que no se aplica ningún tipo de compresión, las imágenes almacenadas con este formato ocupan un tamaño considerable y no son recomendadas para su transmisión a través de una red de comunicaciones.
- Graphics Image Format (GIF): es un formato de compresión excelente para imágenes con grandes áreas homogéneas de color, y es muy sencillo para realizar animaciones vectoriales. Para la compresión utiliza el algoritmo LZW (Lempel-Ziv-Welch). La principal desventaja que posee es que utiliza 8 bits para la representación de cada pixel, con lo que limita su paleta de colores a tan solo 256 posibilidades y lo hace un formato muy malo para imágenes que requieren alta definición en sus detalles.
- Joint Photographic Experts Group (JEPG): es el algoritmo de compresión más popular debido a que se aprovecha de una limitación del ojo humano, la cual impide la completa visualización de la paleta de colores de 24 bits, razón por la que elimina información que el opjo humano no es capaz de procesar. Esto produce una compresión considerable de la imagen digital, pero produce perdida de información.
- Portable Network Graphics (PNG): este formato de compresión resulta debido a la necesidad de sustituir a GIF, ya que se presentaron problemas con la parte del algoritmo (LZW). PNG cubre en su totalidad todos los aspectos de GIF, utilizando un algoritmo de compresión mejor y con una paleta de colores de 216 posibilidades, superior a la usada por GIF. [16]

1.3.13 Clasificación de las técnicas utilizadas para la esteganografía

Se han desarrollado algunas técnicas para ocultar información, algunas con cierto grado más de dificultad que otras, pero de manera general se pueden clasificar de

acuerdo con el tipo de cubierta que se usa para la comunicación secreta o de acuerdo a las modificaciones que se hacen a la cubierta del proceso de inserción.

1.3.13.1 Técnica de sustitución LSB (Least Significant Bit)

La técnica LSB es el mecanismo de ocultación más utilizado en esteganografía. Su utilización es muy común, debido a su facilidad de aplicación, sobre todo a imágenes y audio, permitiendo ocultar grandes cantidades de información. Para comprender su uso vamos a analizarla en primera instancia en uno de los formatos gráficos más sencillos.

Una imagen digital se puede estudiar como un conjunto de unidades de tamaño mínimo, denominados píxeles, que tienen unos determinados valores que reflejan los diferentes colores visibles para una imagen concreta. Dependiendo de la resolución de la imagen y de la codificación empleada para un formato gráfico concreto, los píxeles se representan con 1 o más octetos en dicho formato. Es habitual encontrarse ficheros gráficos con diferentes codificaciones, por ejemplo, ficheros gráficos BMP (BitMaP) con resoluciones 8, 16, 24 o 32 bits. Para una imagen con una resolución de 24 bits cada píxel se representa con 3 octetos (8 bit x 3 = 24 bit). Si estos octetos representan un modelo de color RGB (Red, Green, Blue), cada octeto, respectivamente, almacenará información, sobre el nivel de rojo, verde y azul, que está presente en cada píxel. Esta información permite obtener el color real de un píxel.

El valor concreto que puede tomar cada octeto representa la intensidad de dicho color. Este valor oscila desde el valor 0 (el nivel más oscuro) a 255 (el nivel más luminoso). Por ejemplo, un píxel representado con los siguientes 3 octetos: 11111111 (rojo) 00000000 (verde) 00000000 (azul) da como resultado un píxel de color rojo. Siendo estrictos, la técnica LSB aplicada a imágenes es un procedimiento de sustitución que permite modificar el bit menos significativo de la codificación de cada píxel de una imagen por el bit del mensaje a ocultar.

1.3.13.2 Técnicas esteganográficas basadas en paleta de colores

Los cambios sobre el tratamiento y almacenamiento de imágenes, dio lugar a la necesidad de representar las imágenes en formatos de ficheros gráficos, de tal forma que su tamaño final, en octetos, fuera no excesivamente grande. Lo cual facilitaría no solo su almacenamiento sino también su intercambio por las redes de telecomunicaciones. Para lo cual se propusieron varias opciones, entre ellas, destacó la idea de crear imágenes con un número pequeño de colores. Si el número de colores en una imagen es pequeño se pueden codificar estos valores en una especie de tabla, de tal forma que cada píxel en lugar de almacenar el color que le corresponda apunte a su color presente en esa tabla. El tamaño con el cual se codifica estos índices que apuntan a la tabla es menor que la codificación con la que se almacena el color para cada píxel, con lo que se consigue, finalmente, ficheros gráficos de tamaño más pequeño. Esta ordenación especial, de los octetos de la imagen, permite reducir bastante su tamaño. Esta información adicional, a modo de tabla, se conoce como paleta de colores.

Un ejemplo de uso de paleta de colores puede verse en el formato gráfico GIF aunque también es posible encontrarla en otros formatos gráficos, por ejemplo, en los ficheros BMP. La paleta de colores de los ficheros GIF permite representar hasta 256 colores diferentes, codificando el valor de cada uno con 3 octetos, que indican su valor RGB. Los valores RGB de los píxeles de la imagen no se almacenan directamente en el fichero gráfico, sino que se almacenan índices que enlazan a la paleta de colores. De esta forma, en un fichero GIF, cada píxel se puede representar con 8 bits (28=256 colores posibles de la paleta) pudiendo servir de índice o entrada para indicar cuál es el color real del píxel que está en la paleta.

Una comparación rápida entre este formato y otros formatos gráficos más clásicos se puede ver con imágenes en formato BMP con una resolución de 24 bits. Para esta resolución, cada píxel necesitaría 3 octetos para codificar el valor del color que representa (su valor RGB). Independientemente de la mejor conveniencia de utilizar uno u otro formato gráfico que contenga una imagen, es cierto, que estos formatos que se apoyan en la utilización de una paleta de colores facilitan la creación de diferentes técnicas de ocultación de información.

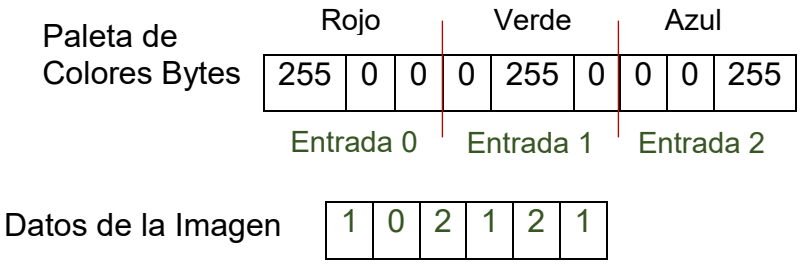


Figura 18 Ejemplo de paleta de colores con 3 colores e imagen de 6 píxeles.

Esta técnica es fácil de comprender en el caso del formato GIF (como se muestra en la Figura). En este formato los datos de la imagen son octetos individuales. En lugar de que el octeto contenga información de los colores del pixel indica la posición en la paleta de colores donde se encontrará el valor real del pixel. Así, por ejemplo, un octeto con valor 0 apunta a la entrada 0 de la paleta de colores, un octeto con valor 1 apunta a la entrada 1, y así sucesivamente para los 256 valores posibles que puede tomar cada octeto. En el ejemplo de la Figura 18 el valor "real de los píxeles de la imagen" sería: color verde (3 octetos) apuntado por el índice (octeto) con valor 1, color rojo (apuntado por el índice con valor 0), color azul (apuntado por el índice con valor 2), etc. Es habitual que se modifique el bit menos significativo de la codificación de cada octeto de los datos de la imagen. Si modificamos algún bit a un índice su valor cambiará, por tanto, ese nuevo índice apuntará a otro color diferente presente en la paleta. Aunque esta técnica es fácil de utilizar debe tenerse en cuenta una serie de consideraciones. Al modificar el bit menos significativo de un índice (ponerlo a 0 o 1) el color actual que corresponde al píxel en cuestión cambia, ya que el nuevo índice apuntará al color adyacente del actual, presente en la paleta de colores. Este hecho hace que si los colores adyacentes en la paleta no son parecidos el "ruido" debido a la manipulación de los LSB será obvio, y producirá una imagen la cual tendrá un resultado visual que hará sospechar una posible ocultación de información.

Para intentar reducir este problema es común que antes de insertar la información a ocultar se realice algún procesado u ordenación previa de los colores presentes en la paleta de colores. Sin embargo, este hecho debe realizarse con precaución ya que puede crear patrones inusuales que no se suelen encontrar en los ficheros gráficos. Por ejemplo, es habitual que los valores de la paleta de colores estén ordenados del más usado al

menos usado, para reducir tiempos de decodificación de la imagen, o que los colores de esta no presenten cambios graduales (por ejemplo, de un bit).

1.3.13.3 Técnicas esteganográficas basadas en coeficientes. JPEG

En la actualidad el procesamiento digital y almacenamiento de imágenes, ha hecho que se utilicen un sinfín de algoritmos y transformaciones matemáticas con diferentes propósitos, entre ellos, su aplicación a la esteganografía, especialmente los procedimientos de ocultación mediante transformaciones en el dominio de la frecuencia. En concreto, la ocultación de información utilizando los coeficientes cuantificados presentes en un fichero gráfico con formato JPEG, obtenidos al aplicar a una imagen original una transformada discreta del coseno (DCT). Esta técnica es la más difundida para ocultar información en imágenes en formato JPEG. Las investigaciones se centran fundamentalmente en la utilización de los coeficientes cuantificados DCTs con fines esteganográficos. El formato JPEG es fácil comprender cuál es el significado real de estos coeficientes cuantificados y por qué pueden utilizarse en esteganografía. Los coeficientes cuantificados, son números que almacenan la información de la imagen a visualizar. En este caso se aplica una técnica LSB de forma secuencial e individualizada a los coeficientes cuantificados DCTs de la imagen, de modo que para ocultar un bit de información se modifica el bit menos significativo del valor de un coeficiente cuantificado. Esta herramienta tiene la ventaja de proporcionar una alta capacidad de almacenamiento de mensajes ocultos, así como resistencia frente a ataques publicados, como los ataques visuales. Sin embargo, la modificación de los bits LSB de los coeficientes cuantificados introduce anomalías que son detectables por estudios estadísticos del histograma.

1.3.14 Sistema de dominio de la transformada.

Estos sistemas tratan de ocultar información secreta en un área significativa de la imagen que sirve como cubierta, con lo cual son más robustos los ataques como compresión o adición de ruido. Entre los métodos más comunes está el usar la transformada (DTC) o también está el uso de la transformada de wavelet.

1.3.15 Relación entre esteganografía y criptografía

Para un atacante que desee emitir mensajes a nombre de otro, en el caso criptográfico, la seguridad radica en la fortaleza de la clave privada del emisor atacado, mientras que en el caso esteganográfico, la seguridad radica en el secreto de la aplicación de esteganografía para transmitir el generador del mensaje. Romper el esquema de autenticación implica vulnerar cada uno de estos aspectos. [9,10,18]

La criptografía intenta proteger el mensaje de un enemigo, pero sin esconder que se realiza una comunicación y una protección a la misma. Mientras que, con esteganografía, se intenta ocultar que se protege la comunicación. Por lo tanto, se pueden enumerar las consecuencias de esta diferencia como las siguientes:

- La existencia de un mensaje cifrado, con aplicación de criptografía, levanta sospecha al enemigo que lo observa: el cual podría amenazar o torturar a los participantes de la comunicación legítima para que revelen el contenido del mensaje. Con la aplicación de esteganografía no se levanta sospecha de una protección a la comunicación.
- Encriptar mensajes alienta desarrollar mecanismos de extorsión para conseguir el mensaje oculto.
- El hecho de encontrar un mensaje encriptado, suele alentar el desafío de saber cuál es el mensaje y esto le motiva a encontrar mecanismos para romper la seguridad y obtener el mensaje original.
- La detección de un mensaje encriptado puede ser suficiente para un enemigo: en ocasiones no es necesario conocer el contenido del mensaje. El sólo hecho de conocer que se efectúa una comunicación puede ser suficiente información.
- El procesamiento de las computadoras aumenta y atenta permanentemente contra las características de los algoritmos criptográficos: un algoritmo seguro hoy en día puede no serlo en un futuro. Si la seguridad se basa en la clave criptográfica, un ataque por fuerza bruta será más rápido de desarrollar con equipos computacionales con mayor capacidad de procesamiento.

- Suele ser difícil demostrar la confiabilidad de algoritmos criptográficos y, en ocasiones, la vulnerabilidad de la seguridad criptográfica puede radicar en el algoritmo utilizado.

- Por último, puede mencionarse una popularidad ampliamente mayor de la Criptografía respecto de la Esteganografía. Esto puede deberse a que la aplicación de técnicas criptográficas es en la práctica más fácil que las técnicas esteganográficas. En particular las técnicas esteganográficas requieren una alta sincronización entre el emisor y el receptor para mantener la comunicación.

Historia

Historia de Histiaeo

Según esta crónica, desde la corte de Persia Histiaeo quería alentar a su yerno Aristágoras de Mileto para que se revelara contra el rey de Persia. Para transmitir sus instrucciones de forma segura, Histiaeo afeitó la cabeza de un mensajero, y tatuó el mensaje en su cuero cabelludo esperando a continuación a que le volviera a crecer el pelo. El mensajero pudo viajar sin levantar sospechas y hacer llegar la información que fue revelada al afeitarse la cabeza. Esta "sutil" comunicación permitió a los griegos percatarse de los planes persas para conquistarlos.

Una variante de esta idea se relata en otra crónica donde el noble Harpagus transmitió una información a Cyrus, rey de Persia, indicándole que recibiría ayuda desde "dentro" para solucionar la opresión que sufría su país. *Para ello vistió a un mensajero de cazador y le proporcionó una liebre en cuyo vientre afeitado había escrito el mensaje a transmitir. Este mensaje permanecería oculto tras crecer el bello*. El cazador con la liebre pasaron desapercibidos.

Historia de Demarato

Demarato, un griego que vivía en la ciudad persa de Susa decidió avisar a Esparta que el rey persa Jerjes estaba planeando invadir Grecia. La dificultad recaía en cómo enviar el mensaje sin que fuera interceptada por los vigías persas, para ello ingenió un mecanismo de comunicación que consistía en retirar la cera de un par de tablillas de madera, escribir la alerta de la proliferación militar en Persia y luego cubrir el mensaje con cera. Esta tabla, aparentemente en blanco, pasó completamente desapercibida, incluso para sus receptores durante un tiempo, hasta que la hija de Cleómenes (esposa

de Leónidas) lo vaticinó y descubrieron el método tan sutil de comunicación. Como resultado obvio de esta advertencia, los griegos comenzaron a armarse.

Un ejemplo esteganográfico curioso en este periodo tuvo lugar en la Inglaterra del siglo XVI donde la esteganografía tuvo una gran importancia en las conspiraciones urgidas entre los nobles católicos ingleses que querían destronar a la reina protestante Isabel I (1533-1603) y entregar el trono a la católica María I de Escocia (1542-1587). La comunicación entre los conspiradores y la reina María debía pasar lo más desapercibida posible ya que cualquier conocimiento de esta implicaría ser acusados de alta traición y condenados a muerte. Por este motivo, emplearon tanto criptografía como esteganografía para ocultar sus mensajes. Un mecanismo recurrido fue la ocultación de mensajes en barriles de cerveza que se transportaban sin levantar la atención.

Siglo XV

El científico italiano Giovanni Battista della Porta descubrió cómo esconder un mensaje dentro de un huevo cocido. El método consistía en preparar una tinta mezclando una onza de alumbre y una pinta de vinagre, y luego se escribía en la cáscara. La solución penetra en la cáscara porosa y deja un mensaje en la superficie de la albúmina del huevo duro, que sólo se puede leer si se pela el huevo. [21]

Personas

Gaspar Schott

El científico alemán *Gaspar Schott* (1608-1666) describiría en su libro Schola Steganographica. En esta obra se describía como ocultar mensajes en partituras de música, haciendo equivaler una nota musical concreta con una letra. En ningún momento se buscó que las notas musicales tuvieran alguna coherencia y su resultado diera una melodía agradable, en cualquier caso, su representación, por ejemplo, en papel, permitía perfectamente ocultar un mensaje. Basado en estas ideas surgieron otros procedimientos de ocultación, como los basados en el número de ocurrencias de las notas. Se han documentando mensajes ocultos en músicos famosos como *J.S. Bach o John Wilkins*. [22]

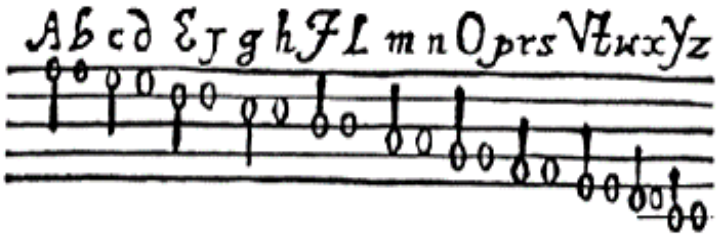


Figura 19 Equivalencia de notas musicales y texto a ocultar

John Wilkin

John Wilkins (1614-1672) desarrolló, además, obras relacionadas directamente con el mundo de la criptografía y esteganografía como Mercury, or The Secret and Swift Messenger en 1641. Es destacable el desarrollo de procedimientos para ocultar información utilizando dibujos geométricos, es decir usando puntos, líneas o triángulos de un dibujo para enmascarar información. [21]

Johannes Trithemius

Johannes Trithemuis (1462-1516) Los primeros usos registrados de la esteganografía se remontan al año 440 aC, cuando Heródoto menciona dos ejemplos en sus Historias. Histiaeus envió un mensaje a su vasallo, Aristágoras, al afeitarse la cabeza de su servidor más confiable, "marcando" el mensaje en su cuero cabelludo, luego, enviándolo a su camino una vez que su cabello haya vuelto a crecer, con la instrucción: "Cuando vengas a Mileto, dile a Aristágoras que se afeite la cabeza y mira al respecto". Además, Demaratus envió una advertencia sobre un próximo ataque a Grecia al escribirlo directamente en el soporte de madera de una tableta de cera antes de aplicar su superficie de cera de abeja. Las tabletas de cera eran de uso común entonces como superficies de escritura reutilizables, a veces utilizadas como taquigrafía.

En su obra Polygraphiae, Johannes Trithemius desarrolló su llamada "Ave-Maria-Cipher" que puede ocultar información en una alabanza latina de Dios. "Auctor Sapientissimus Conseruans Angelica Deferat Nobis Charitas Potentissimi Creatoris", por ejemplo, contiene la palabra oculta VICIPEDIA. [19]

1.3.4 Least Significant Bit (LSB)

Esta es una técnica más simple, popular y antigua utilizada para la esteganografía. La información secreta se oculta en los bits menos significativos de un byte, lo que representa un valor de intensidad de un píxel. Solo podemos utilizar LSB o los últimos dos, tres o cuatro LSB, según la cantidad de datos que se oculten y según la importancia de los datos secretos y también según el tipo de imagen de portada y la frecuencia de los píxeles utilizados en una imagen. Por lo tanto, el mejor algoritmo de esteganografía LSB puede diseñarse considerando los parámetros anteriores de una imagen de portada y la información secreta que se ocultará [1].

La esteganografía de imagen basada en LSB incrusta el secreto en los bits de valores de píxeles menos significativos de la imagen de portada (CVR) en ingles values of the cover image.

Para entender mejor la forma general en la cual se implementa la técnica base de sustitución 1 bit, (observar la Figura 20). En ésta, se aprecia que de cada byte del mensaje se extrae cada bit, y éste a su vez es insertado en cada bit menos significativo de cada componente de color RGB, o en cada byte, que conforma a cada del pixel de la imagen portadora. En la figura podemos ver un ejemplo el cual hace referencia a pixeles de una imagen portadora BMP de 24 bits.

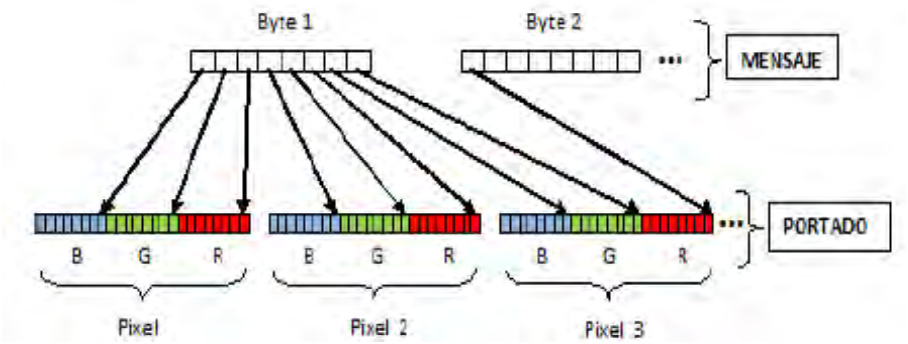


Figura 20 Implementación LSB 1 bit

Es importante tener en cuenta que cuando se opera sobre una imagen desde su archivo binario, el orden de los bytes que representa a las componentes R, G y B de cada píxel, presentan el orden B, G y R. ese orden es el que se muestra en la figuta tratando de la descripción del esquema de implementación de la técnica. Corresponde con el orden que representa cada una de las componentes del color cuando se lee una imagen byte a byte desde su almacenamiento en disco. Siendo también el mismo orden de canales (B, G y R) en el que se trabaja cuando se usa la imagen desde un programa escrito en cualquiera de los lenguajes de programación.

Método A

Ocultamiento

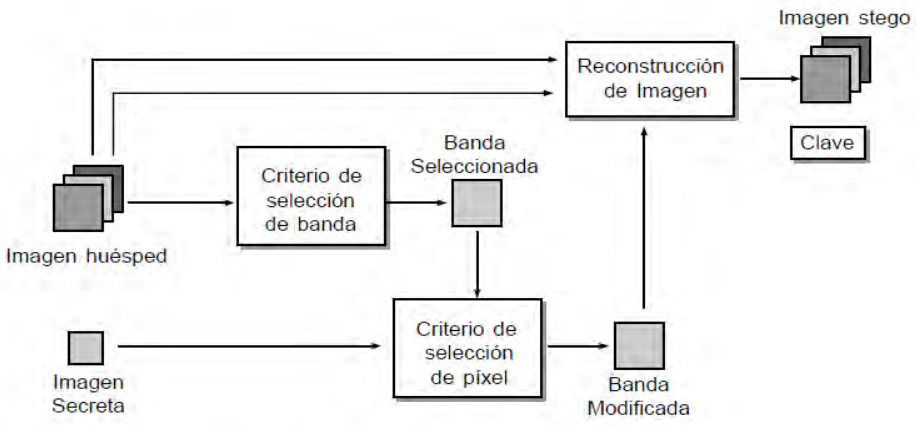


Figura 21 MA Ocultamiento

1. Se inicializa la clave con tres datos: el total de filas de la imagen secreta, el total de columnas de la imagen secreta y el valor promedio de los píxeles de la imagen secreta.
2. Se separan las tres bandas de color de la imagen huésped y se selecciona la banda más idónea para la inserción de los datos secretos. La selección se basa en el criterio de correlación entre el histograma de la imagen secreta y el histograma de cada una de las bandas de la imagen huésped. El histograma de la banda que presente mayor grado de correlación (similitud) con el histograma de la imagen secreta, determinará la banda seleccionada.

3. Se incluye en la cuarta posición de la clave el número de la banda seleccionada en el paso anterior, así: 1 si la banda es la roja, 2 si es la verde y 3 si es la azul.
4. Se realiza el proceso de búsqueda de un píxel de la banda seleccionada de la imagen huésped que sea similar al píxel de la imagen secreta a ocultar. El criterio de similitud implica que no necesariamente deben ser iguales los dos píxeles, sino que se tolera cierto margen de error. Este margen de error se conoce como rango, de tal forma que el valor del píxel que se busca está comprendido entre el valor del píxel de la imagen secreta _ el valor de rango.
5. Cuando se encuentra el píxel que satisface el criterio de búsqueda, se reemplaza por el valor del píxel de la imagen secreta y se guarda en la clave la posición absoluta del píxel modificado. Por ejemplo, si se tiene una imagen huésped de 100 filas _ 80 columnas y el píxel modificado se encuentra en la segunda fila, séptima columna, la posición absoluta del píxel haciendo un barrido en zigzag de izquierda a derecha y de arriba abajo es de 87 (80 posiciones de la primera fila más 7 posiciones de la segunda fila). Con el propósito de modificar solamente una vez el píxel seleccionado de la imagen huésped, esté píxel se bloquea y se omite para búsquedas futuras.
6. Si ningún píxel de la imagen huésped satisface el criterio de búsqueda, entonces se guarda en la clave el valor de 0. En el módulo de extracción se explicará qué valor se sustituye en la imagen recuperada cuando la clave tiene un 0.
7. Los pasos d-f se repiten para cada uno de los píxeles de la imagen secreta. Al finalizar el proceso de búsqueda, se tiene una banda de la imagen huésped que presenta modificaciones en algunos de sus píxeles y dos bandas que no sufrieron cambios en el proceso de ocultamiento de información. El total de posiciones de la clave es igual al total de datos de información complementaria (4 valores correspondientes a N2, M2, promedio, banda seleccionada) más el total de píxeles de la imagen secreta (N2 _M2).
8. Con la banda modificada y las dos restantes sin modificar, se reconstruye la imagen a color la cual corresponde a la stego-imagen. Esta imagen junto con la clave se trasmite por dos canales independientes al usuario autorizado.

Recuperación

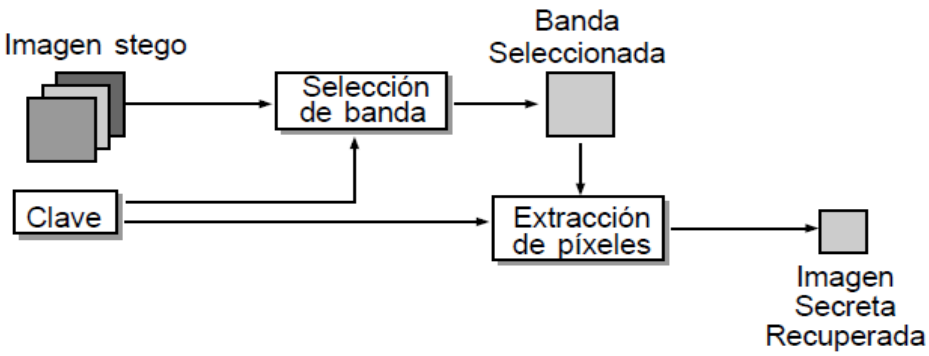


Figura 22 MA Recuperación

1. Identificar la banda en la cual está contenida la imagen secreta: la stego-imagen se descompone en las tres bandas de color (R, G, B) y a continuación se selecciona el número de la banda que fue almacenado en la clave. Esta información quedó registrada en la cuarta posición de la clave en el proceso de ocultamiento.
2. Con la información contenida en la clave, a partir de la quinta posición, se seleccionan los píxeles de la banda de la imagen huésped que fueron modificados y que contienen la información de la imagen secreta. Se hace un barrido en zigzag de izquierda a derecha y de arriba abajo para extraer los píxeles modificados. Este barrido se hace de la misma forma que en el módulo de ocultamiento, ya que el valor de las posiciones de los píxeles modificados corresponde a la posición absoluta dentro de la banda.
3. Si en la clave se encuentra el valor de 0, significa que el píxel de la imagen secreta no se pudo ocultar dentro de la banda seleccionada de la imagen huésped. De tal forma que, se asigna a ese píxel de la imagen secreta el promedio de los píxeles, que previamente fue almacenado en la clave en la tercera posición. Este promedio es muy similar al valor esperado de la imagen (el de mayor probabilidad de ocurrencia), pero computacionalmente de menor coste.

4. Al finalizar el proceso de extracción de píxeles, se obtiene un vector de N2 _ M2 elementos. Este vector se redimensiona de acuerdo a la información contenida en las dos primeras posiciones de la clave (número de filas y columnas de la imagen secreta).

Método B

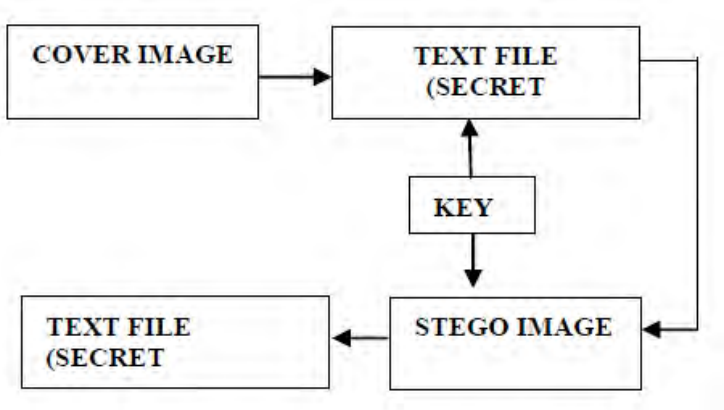


Figura 23 MB Formato

Algoritmo para insertar mensaje secreto:

1. Seleccione una imagen en color como imagen de portada.
2. Ahora elige el archivo de texto que contiene información secreta
3. Introduzca la clave de 8 bits que es clave debe estar entre 0 – 255
4. Convierta cada carácter del mensaje secreto en código ASCII, es decir, 97 para ‘a’, 48 para ‘0’, 32 para ‘barra espaciadora’, etc. En Matlab esto se puede lograr con el comando "doble".
5. Determine la longitud del mensaje y el cero rellenado para que tenga 8 caracteres de longitud y colóquelo en el encabezado del mensaje, lo que significa agregar la longitud del mensaje en el encabezado del mensaje
6. Convierta cada código ASCII a su equivalente binario de 8 bits
7. Tome la imagen de portada y conviértala en un entero de 8 bits sin signo para que su valor de píxeles esté en el rango [0 255]
8. Separe la imagen de portada en el plano RGB e inserte el primer bit del mensaje secreto hasta el último bit del primer píxel del plano rojo, y el segundo bit hasta el último bit del segundo píxel del plano rojo

9. Ahora el tercer bit del mensaje secreto al último bit del primer píxel del plano verde, el cuarto bit al último bit del segundo bit y el quinto bit al último bit del tercer píxel del plano verde
10. Finalmente, sexto, séptimo y octavo bit de mensaje secreto al último bit del primer píxel, segundo píxel y tercer píxel del plano verde respectivamente. Por lo tanto, 8 bits del primer carácter del mensaje de texto secreto se incorporan en 8 píxeles de la imagen de portada (2 rojos, 3 verdes y 3 azules). Cada vez que se incrusta un bit en un píxel de un plano, aumenta su posición en 1 para pasar al siguiente píxel de ese plano. Este proceso continúa hasta que todos los bits se incrustan en la imagen de portada.
11. La imagen resultante es una imagen stego que contiene información.

Método C

Se describe el proceso de realización del algoritmo LSB. En la Figura 24, se muestra un diagrama a bloques, el cual es llevado a cabo para realizar la versión final del algoritmo en el lenguaje de programación Matlab, mostrando el proceso realizado en el transmisor **Tx**

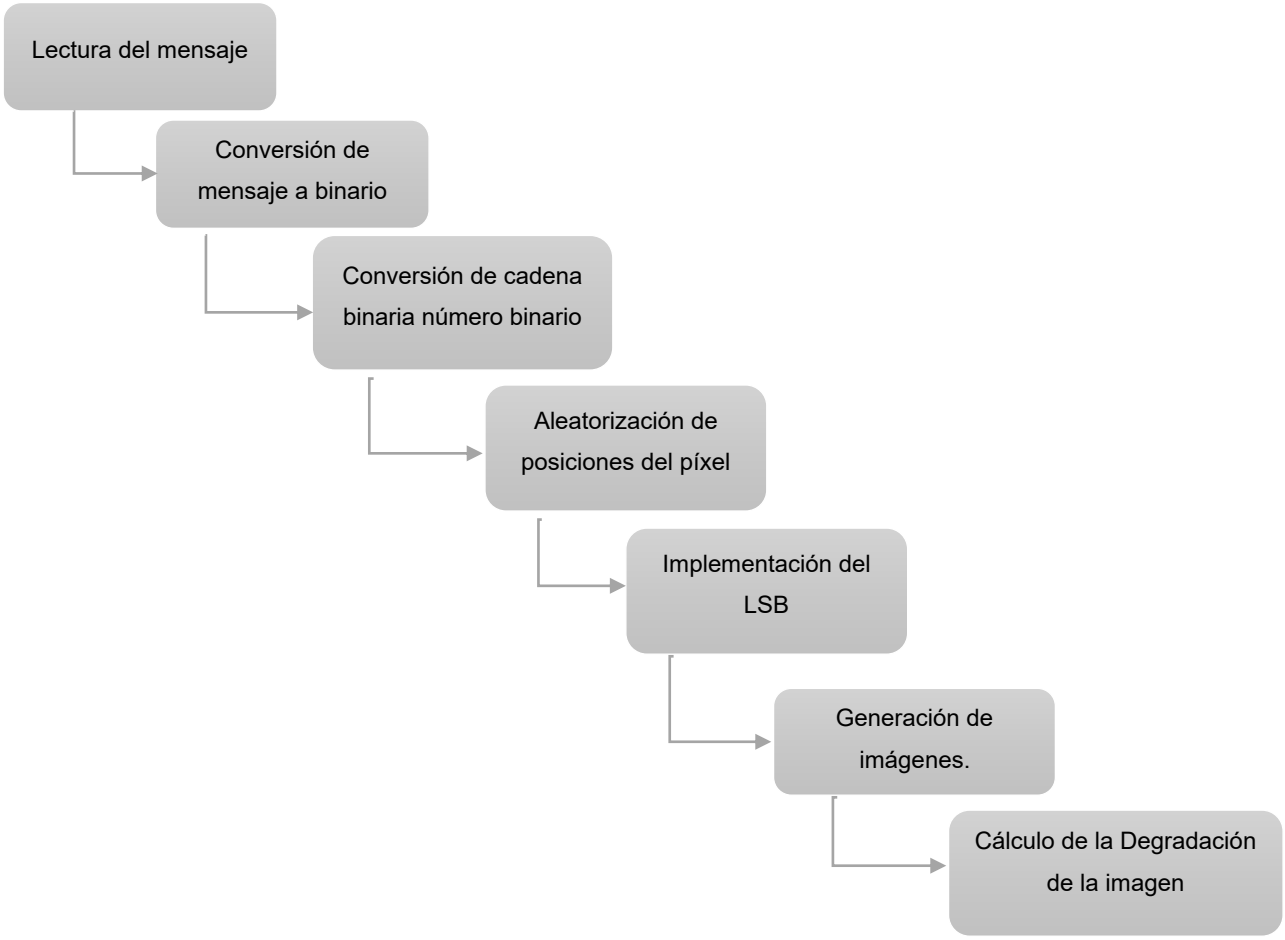


Figura 24 Proceso de transmisión del mensaje

Tx.

De igual manera se muestra el proceso de transmisión de un mensaje, iniciando con la lectura, la conversión, y finalizando con la inserción de bits a la imagen.

La Figura 25 muestra un diagrama a bloques sobre el proceso en el lado receptor Rx del mensaje, el cual consta de 4 bloques.

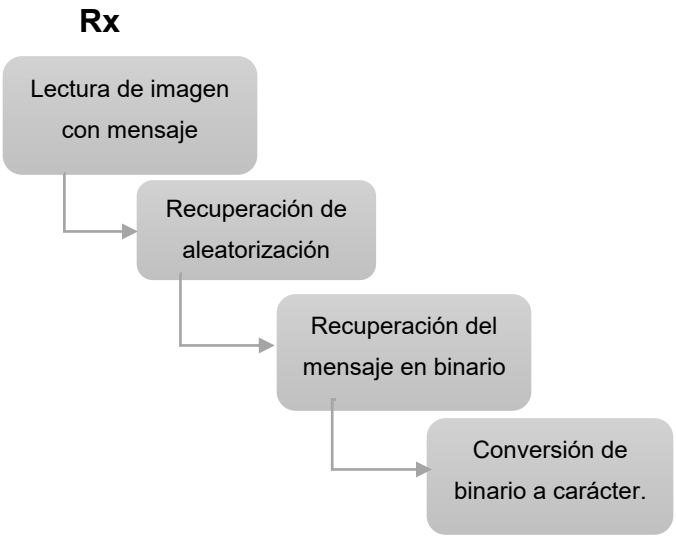


Figura 25Proceso de recepción del mensaje

Capítulo 2 Desarrollo

2.2 Algoritmo en Matlab

La implementación en Matlab del algoritmo LSB fue en una imagen como archivo digital la cual tiene las siguientes características:

- Escala de color
- Tipo de archivo: PNG
- Ancho: 220 pixeles
- Largo: 220 pixeles

El tamaño del mensaje a insertar depende de las características de la imagen especialmente el ancho y largo. Tomando en cuenta las características de la imagen que se está utilizando tenemos un total 241x 239= 48400 bytes de los cuales se utilizara un solo bit de cada pixel, por lo tanto, tendremos un total de 48400 bits para ocultar información teniendo la posibilidad de ocultar un mensaje de 6050 caracteres.

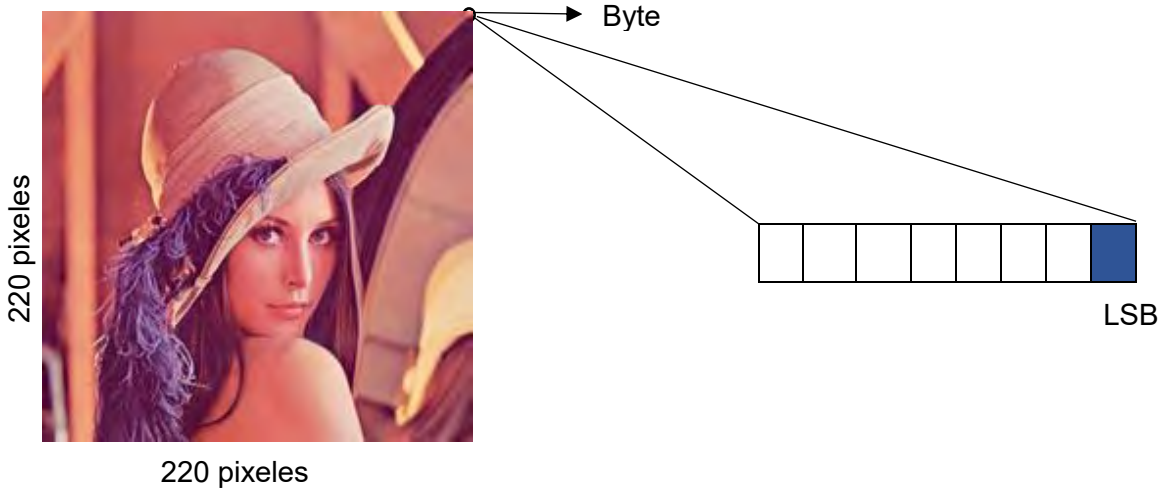


Figura 26 Características de imagen

A continuación, se describe el proceso de realización del algoritmo LSB. En la Figura 27, se muestra un diagrama a bloques, el cual es llevado a cabo para realizar la versión final del algoritmo en el lenguaje de programación Matlab, mostrando el proceso realizado en el transmisor Tx.

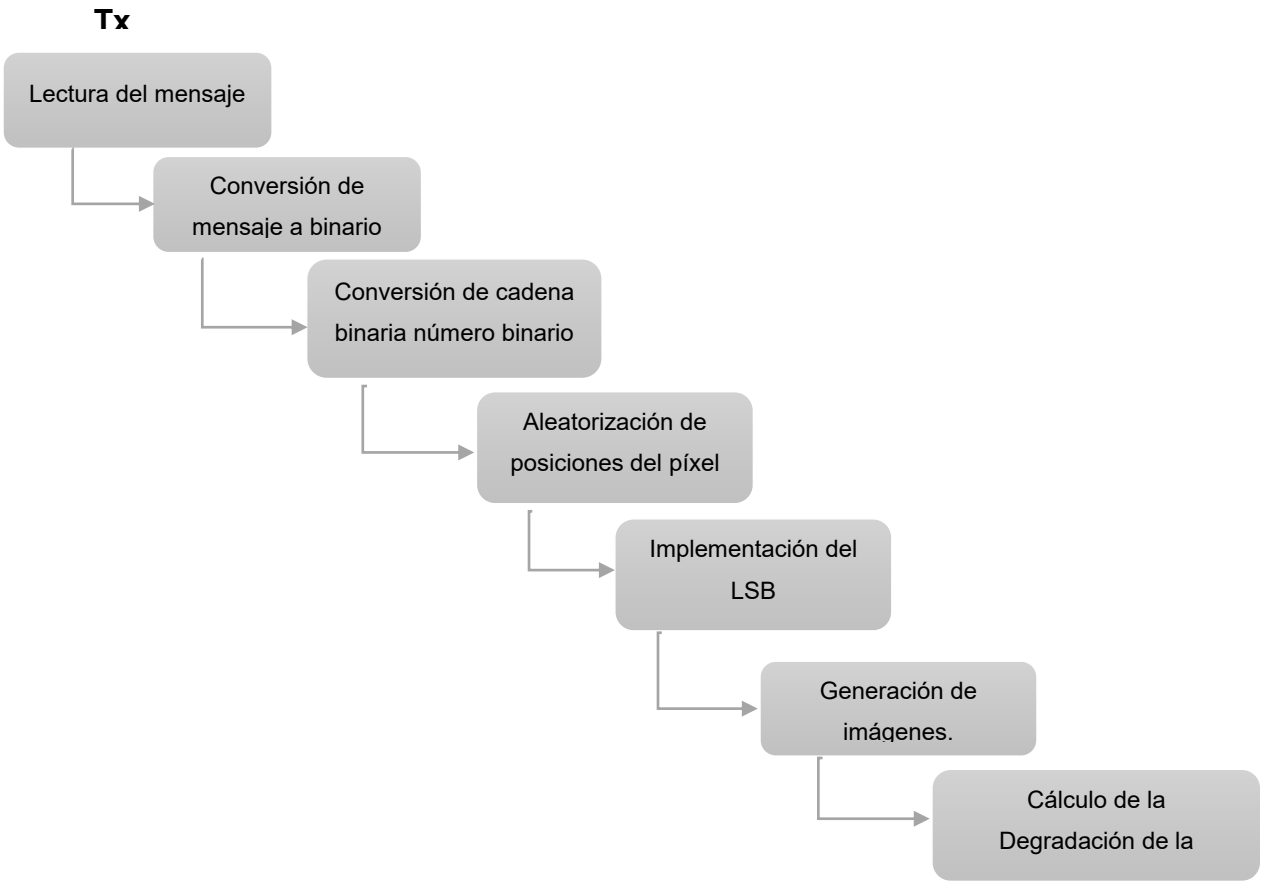


Figura 27 Proceso de transmisión del mensaje

De igual manera se muestra el proceso de transmisión de un mensaje, iniciando con la lectura, la conversión, y finalizando con la inserción de bits a la imagen.

La Figura 28 muestra un diagrama a bloques sobre el proceso en el lado receptor Rx del mensaje, el cual consta de 4 bloques.

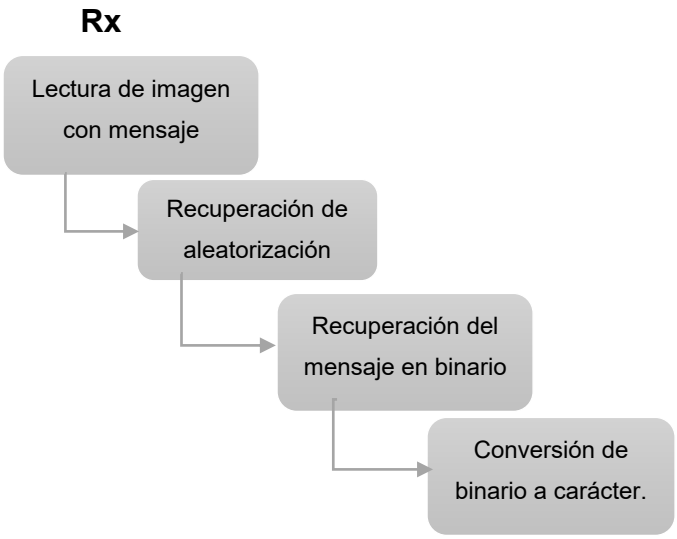


Figura 28 Proceso de recepción del mensaje

2.1.1 Transmisor Tx

La Figura 29 muestra el bloque llamado lectura del mensaje que describe la inicialización del algoritmo implementado en Matlab inicializando con borrar todas las entradas y salidas de la ventana de comandos, después se asigna el mensaje el cual es leído como una cadena y se calcula la longitud de dicho mensaje.

```
1-clear all;
2-clc;
3- c = (imread ('Lenna_color.png'));
4- %   message =
'ABCDEFGHIJKLMNOPQRZ12345678910ABCDEFGHIJKLMNOPQRZ12345678910!"#$%&/()()()(
)()ABCDEFGHIJKLMNOPQRZ1234';
5- message = '- -Esta es una prueba de esteganografia a traves de
substitucion de bit menos significativo o LSB- -';
6- message = strtrim(message);
7-   m = length(message) * 8;
```

Figura 29 Bloque: Lectura de mensaje

La Figura 30 muestra el bloque llamado conversión del mensaje a binario, en donde aún se contiene el mensaje en la línea 8. Así, se hace la conversión de dicha cadena de texto a código ASCII con el fin de representar caracteres alfanuméricos. Para continuar con el

proceso, se procede a realizar una conversión a lenguaje binario (en la línea 9) que es la representación de 1s y 0s. En línea 10, la cadena binaria se genera en un vector fila, y posteriormente se calcula la longitud de este nuevo vector binario.

```
8- AsciiCode = uint8(message);
9- binaryString = transpose(dec2bin(AsciiCode,8));
10- binaryString = binaryString(:);
11- N = length(binaryString);
```

Figura 30 Bloque: Conversión del mensaje a binario

Para ejemplificar el proceso de inserción de un mensaje, supongamos que se desea enviar un carácter A. En alfanumérico la letra A es representada en código ASCII con el número 65 por lo que este número en representación binaria corresponde de la siguiente manera: 01000001.

La figura 31 muestra el bloque llamado conversión de cadena binaria a numero binario, de la línea 12 a la línea 19 se realiza el proceso para cambiar de cadena binaria a número binario. Para lo cual se crea un nuevo vector, el cual será llenado en base a la condición del ciclo **`for`**.

```
12- b = zeros(N,1); %b is a vector of bits
13- for k = 1:N
14-   if(binaryString(k) == '1')
15-         b(k) = 1;
16-   else
17-         b(k) = 0;
18-   end
19- end
```

Figura 31 Bloque: conversión de cadena binaria a número binario

La Figura 32 muestra el bloque llamado aleatorización de posiciones del pixel en el cual de la línea 20 a la línea 26 se realiza el procedimiento de la aleatorización mediante el método Twister, donde se nombra otra variable la cual va a contener la imagen original, para posteriormente calcular la anchura y altura de la imagen. En la línea 24 se indica la inicialización de la semilla, que para este ejemplo inicialice en 1 usando el comando `rng(k,'twister').` En línea 25 y 26 se siembra el generador de números aleatorios uniformes usando una semilla de números enteros no negativos para que la función `randi` produzca una secuencia predecible de números dada la semilla utilizada.

```
20- s = c;
21- height = size(c,1);
22- width = size(c,2);
23- k = 1;
24- rng(1,'twister');   % aleatorizacion de la posicion
25- pos_h = randi(height,1,N+1);
26- pos_w = randi(width,1,N+1);
```

Figura 32 Bloque: Aleatorización de posiciones del pixel

En la Figura 33 se muestra el bloque llamado implementación del algoritmo LSB en sí. Como puede observarse, con el ciclo for se realiza el proceso de lectura y modificación de bits, tomando valores de 1 hasta N en donde N representa la longitud del mensaje en un vector de bits. Posteriormente, se leen las posiciones aleatorias que se generaron anteriormente, (en la línea 31), se realiza la lectura del ultimo bit de la imagen para posteriormente tomar en cuenta la longitud del mensaje y hacer una comparación entre ésta y b(k) en la posición k. Esta posición depende del número binario correspondiente generado en el bloque conversión. La inserción de los bits es realizada con el código descrito de las líneas 35 a 38.

```
28- for l = 1:N
29-   i = pos_h(l);
30-   j = pos_w(l);
31-   LSB = mod(double(c(i,j)), 2);
32-   if (k>m || LSB == b(k))
33-         s(i,j) = c(i,j);
34-         else
35-               if (LSB==1)
36-                     s(i,j) = c(i,j) - 1;
37-               else
38-                     s(i,j) = c(i,j) + 1;
39-               end
40-         end
41-         k = k + 1;
42- end
```

Figura 33 Bloque: Implementación del algoritmo LSB

En la siguiente figura se muestra el bloque llamado generación de imágenes, en el cual de la línea 42 a la línea 46 se crean y muestran 2 figuras de las cuales la primera es la original y la segunda es el resultado de la inserción de información a través del algoritmo LSB. Posteriormente, en la línea 47 se escriben los datos de la matriz imagen s en una imagen llamada ‘112.png’ y de esta forma se genera un nuevo archivo digital.

```
43- figure(1)
44- imshow(c)
45- figure(2)
46- imshow(s)
47- imwrite(s, '112.png');
```

Figura 34 Bloque: generación de imágenes

En figura 35 se muestra el bloque llamado cálculo de la degradación de la imagen que implementa una métrica para el cálculo de la relación de señal a ruido. La relación señal a ruido (SNR) define la proporción existente entre la potencia de la señal que se transmite y la potencia del ruido indeseado, en este caso la imagen original y la imagen que se genera después de la inserción de bits. La medición es en dB y es una medida de cualquier cantidad en relación a un nivel conocido, es una unidad logarítmica que nos permite representar números muy pequeños o muy grandes.

```
48- %% SNR computation
49- Power_c_image     = 0;
50- Power_error_image = 0;
51- for i = 1 : height
52-   for j = 1 : width
53-         dum_c     = double (c(i,j));
54-         dum_s     = double (s(i,j));
55-         dum_noisy = dum_c-dum_s;
56-         Power_c_image     = Power_c_image + dum_c.^2;
57-         Power_error_image = Power_error_image + (dum_noisy.^2);
58-   end
59- end
60- SNR_db = 10 * log10(Power_c_image/Power_error_image)
```

Figura 35 Cálculo de la degradación de la imagen

2.1.2 Receptor Rx

La figura 36 muestra el bloque llamado Lectura de imagen con mensaje. Se inicia leyendo la imagen que se generó en el bloque generación de imágenes del Transmisor Rx, se realiza el cálculo de las dimensiones de la imagen, y se asigna a una variable la longitud del mensaje (en la línea 5).

```
1- s = imread('112.png');
2- height = size(s,1);
3- width = size(s,2);
4- %For this example the max size is 100 bytes, or 800 bits, (bytes * = bits
5- m = N;
6- k = 1;
```

Figura 36 Bloque: Lectura de imagen con mensaje

La figura 37 muestra el bloque llamado Recuperación de la Aleatorización en donde se reinicia el generador de números uniformes aleatorios utilizando el configurado en el transmisor y de esta manera se recuperan las posiciones de los pixeles en donde se realizó la modificación de los bits haciendo la inserción del LSB.

```
7- rng(1,'twister');
8- pos_h = randi(height,1,N+1);
9- pos_w = randi(width,1,N+1);
```

Figura 37 Bloque: Recuperación de Aleatorización

La Figura 38 muestra el bloque llamado Recuperación del Mensaje en Binario, en donde con base en las posiciones que se generaron se recuperan los valores binarios introducidos desde 1 hasta N; donde N corresponde a la longitud del vector unitario del mensaje.

```
10- for l = 1:N
11-   i = pos_h(l);
12-   j = pos_w(l);
13-   if (k <= m)
14-      br(k) = mod(double(s(i,j)),2);
15-      k = k + 1;
16-   end
17- end
```

Figura 38 Bloque: Recuperación del Mensaje en Binario

La figura 39 muestra el bloque llamado Conversión de Binario a Caracter en donde se hace la conversión para lograr recuperar el mensaje que se ocultó en la imagen, se escriben los valores correspondientes a el octeto (en la línea 20), se genera un vector binario. Posteriormente, se realiza la conversión por lo que la función reshape ordena el vector binario para formar los octetos del byte y los almacena en una matriz, finalmente la función char en la línea 26 convierte esta matriz en cadena de caracteres.

```
19- binaryVector = br;
20- binValues = [ 128 64 32 16 8 4 2 1 ];
21- binaryVector = binaryVector(:);
22- if mod(length(binaryVector),8) ~= 0
23-   error('Length of binary vector must be a multiple of 8.');
24- end
25- binMatrix = reshape(binaryVector,8,100);
26- textString = char(binValues*binMatrix);
27- disp(textString);
28- bt = b';
29- Errores = sum (br-bt)
```

Figura 39 Bloque: Conversión de Binario a Carácter.

2.2 Cálculos de Métricas

La correlación es un método para establecer el grado de probabilidad de que exista una relación lineal entre dos cantidades medidas. En 1895, Karl Pearson definió el coeficiente de correlación producto-momento de Pearson r. El coeficiente de correlación de Pearson, r, fue la primera medida de correlación formal y se usa ampliamente en el análisis estadístico, el reconocimiento de patrones y el procesamiento de imágenes. Para imágenes digitales monocromáticas [25]. Por ejemplo, en la figura 40, donde 'c' significa la imagen original, 'cR' muestra los pixeles del vector de la primera columna en la imagen original, 'ccR' se agrega todos los pixeles en una sola columna vertical y 's' es la imagen modificado o el stego-imagen. El 'mean_ccR' y 'mean_ssR' calcula la media del vector, 'corrfac', o 'r', predice los cambios en las dos imágenes, la sigma es la sumatoria de las imágenes y SSIM mide la diferencia entre las dos imágenes como se muestra en figura 42.

```
58- %% Correlation Factor
59- cR = c(:,:,1); %red
60- ccR = cR(:);
61- sR = s(:,:,1); %red
62- ssR = sR(:);
63- mean_ccR = sum(ccR)./length(ccR);
64- mean_ssR = sum(ssR)./length(ssR);
65- corrfac = ( sum((ccR-mean_ccR).*(ssR-mean_ssR)) ) / sqrt( sum((ccR-
mean_ccR).^2) * sum((ssR-mean_ssR).^2) );
66- sigma_cc = sum((ccR-mean_ccR).^2) / (length(ccR) - 1);
67- sigma_ss = sum((ssR-mean_ssR).^2) / (length(ssR) - 1);
68- sigma_cc_ss = sum((ccR-mean_ccR) .* (ssR-mean_ssR)) / (length(ccR) - 1);
69- SSIM = (2.*(mean_ccR * mean_ssR + ccR)) .* (2*(sigma_cc_ss) + ssR) ./
((mean_ccR^2 + mean_ssR^2 + ccR) .* (sigma_cc + sigma_ss + ssR));
```

Figura 40 Cálculo de factor de correlación

PSNR es una medida matemática de la calidad de la imagen basada en la diferencia de píxeles entre dos imágenes. La medida SNR es una estimación de la calidad de la imagen reconstruida en comparación con la imagen original. Por otra parte, MSE mide el error cuadrático medio y se calcula promediando la intensidad al cuadrado de la imagen original (entrada) y los píxeles de la imagen resultante (salida) como se muestra en la Figura 41.

```
70- %% PSNR & MSE
71- cc = c(:);    %imagen original
72- ss = s(:);    %imagen stego
73- mse = sum((cc-ss).^2)./length(cc);
74- PSNR = 10*log10(255^2./mse);
```

Figura 41 Cálculo de PSNR y MSE

SSIM mide la diferencia perceptiva entre dos imágenes similares. No puede juzgar cuál de los dos es mejor: debe inferirse a partir de saber cuál es el "original" y cuál ha sido sometido a un procesamiento adicional, como la compresión de datos o también se llama una mejora para UIQI. Con la función [ssimval, ssimmap] se calcula de forma directa utilizando Matlab (como se presenta en la figura 42) o manualmente como ha sido calculado en la figura 40.

```
75- %% SSIM
76- [ssimval, ssimmap] = ssim(s,c);
```

Figura 42 Cálculo de SSIM

UIQI mide la degradación en imágenes digitales para mejorar la calidad de la imagen resultante. Divide la comparación entre la imagen original y la distorsionada en tres secciones (comparaciones): luminancia, contraste y comparaciones estructurales, como en la figura 43 donde ‘Q’ es la calidad de la imagen (UIQI).

```
77- %% UIQI Universal Image Quality Index
78- cR = c(:,:,1); %red
79- cG = c(:,:,2); %green
80- cB = c(:,:,3); %blue
81- mean_cR = mean(cR(:));
82- mean_cG = mean(cG(:));
83- mean_cB = mean(cB(:));
84- sR = s(:,:,1);
85- sG = s(:,:,2);
86- sB = s(:,:,3);
87- mean_sR = mean(sR(:));
88- mean_sG = mean(sG(:));
89- mean_sB = mean(sB(:));
90- cc = c(:);
91- ss = s(:);
92- Fc = cc-mean_cR;
93- Fs = ss-mean_sR;
94- c_sum = sum((Fc).^2);
95- Sigma_c = c_sum./(length(cc)-1);
96- s_sum = sum((Fs).^2);
97- Sigma_s = s_sum./(length(ss)-1);
98- Sigma_cs = sum(Fc.*Fs)./(length(cc)-1);
99- Q = (4.*Sigma_cs.*mean_cR.*mean_sR)./((Sigma_c + Sigma_s).*(mean_cR.^2
+ mean_sR.^2));
```

Figura 43 Cálculo de UIQI

Después de los cálculos de cada una de las métricas, utilizamos la función fprintf para mostrar en pantalla los resultados de los cálculos (ver la figura 44).

```
101- fprintf('\n The PeakSNR value is %0.4f', PSNR);
102- fprintf('\n The SSIM value is %0.4f.\n',ssimval);
103- fprintf(' The MSE value is %0.4f.\n',mse);
104- fprintf(' The Quality image value is %0.4f.\n',Q);
```

Figura 44 Muestra de resultados

Capítulo 3 Resultados

Los resultados de la implementación de cada uno de los algoritmos fueron obtenidos mediante Matlab utilizando la versión R2015a. Así mismo, se ha implementado en Matlab una función denominada LSB, a través de un archivo (.m), que implementa el algoritmo de la técnica de Sustitución LSB 1 bit, operando a la imagen contenedora del mensaje oculto como una matriz de la imagen. El código desarrollado para el algoritmo de sustitución LSB 1 bit implementado genera como resultado la presentación gráfica por pantallas visualizando o desplegando la imagen original (portadora), así como también, la imagen modificada en donde se realizó la inserción de los bits (estegoportadora). La imagen portadora definida como "Lenna_color.png" de 220x220 pixeles se muestra en la Figura 45.

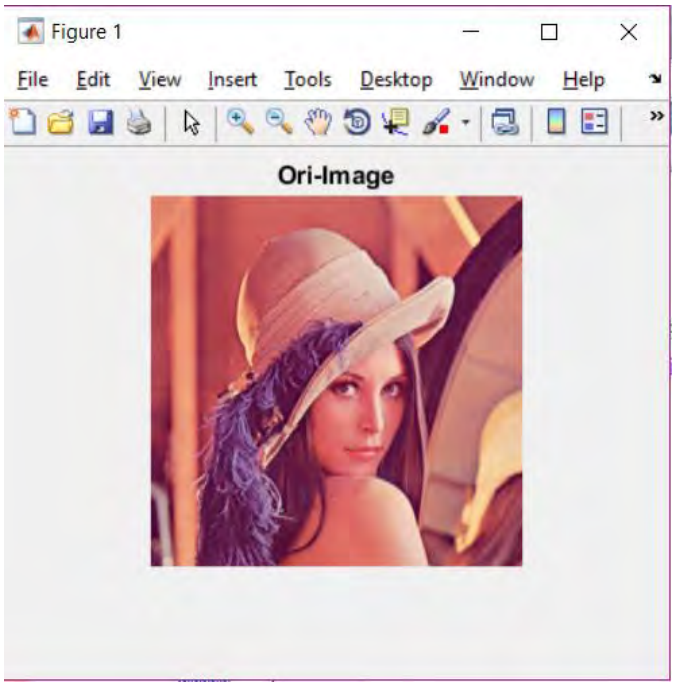


Figura 45 lenna_color portadora (original)

Como mensaje la cadena de texto en la figura 29 línea 4, la siguiente cadena fue elegida con el fin de mostrar la posibilidad de representar números, letras y caracteres. Con un total de 100 caracteres, una representación de 8 bits por caracteres; teniendo un total de 800 bits para ocultar en la imagen portadora.

Otra segunda prueba se utilizó la cadena de texto presentada en la Figura 29, línea 5, con la finalidad de demostrar que los espacios son todas como caracteres en la implementación del LSB y teniendo como resultado al archivo imagen "112.png" con las mismas dimensiones de la imagen inicial, es decir, 220x 220 pixeles. Como se muestra en la Figura 46.

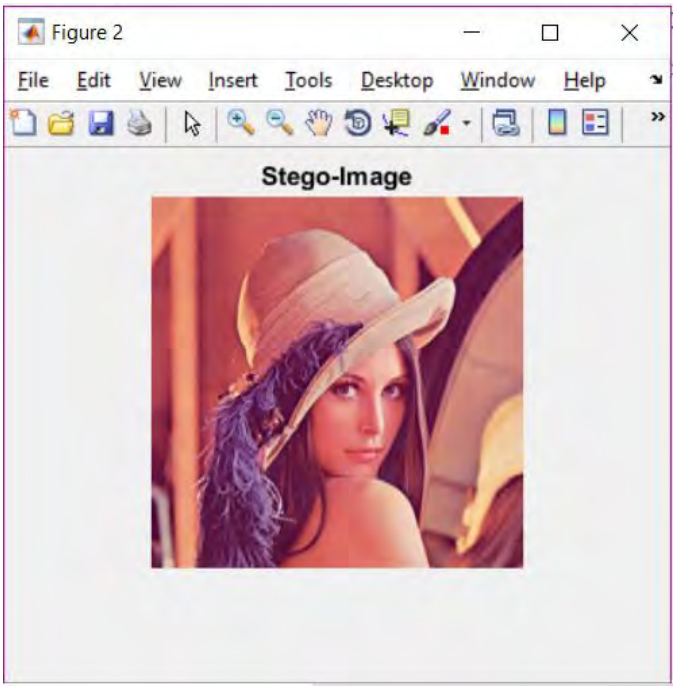
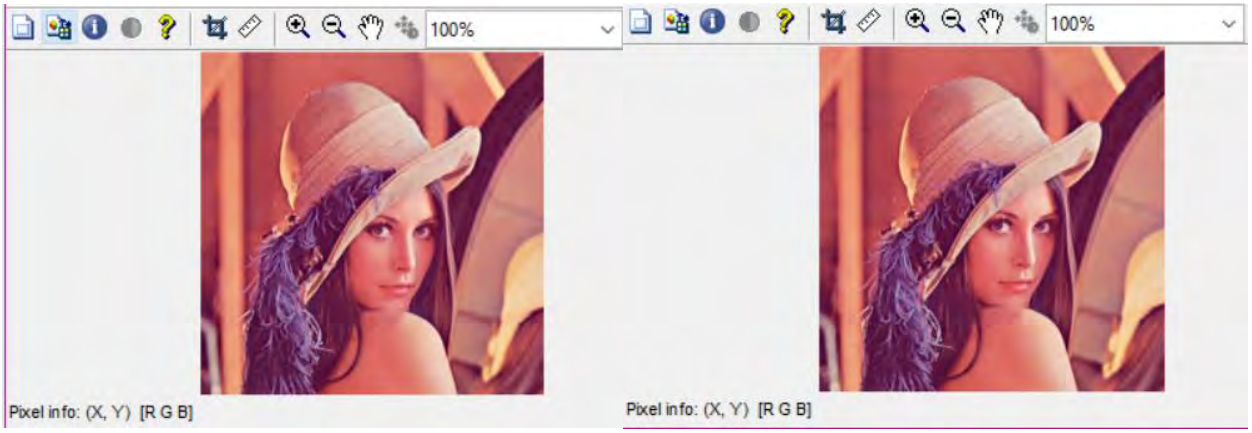
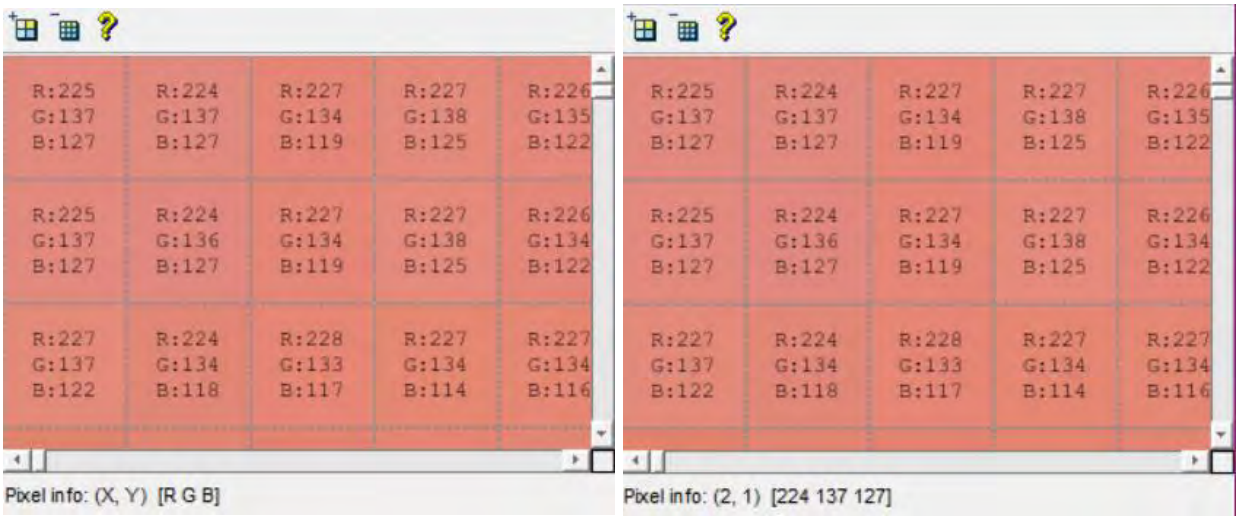


Figura 46 Estego-imagen(112.png)

Matlab cuenta con la función imtool la cual abrirá un visor de imágenes que muestra el nivel de color de la imagen, tomando una región de pixeles y con la función de imhist la cual abrirá un grafo de histograma que muestra el nivel de píxel de la imagen.

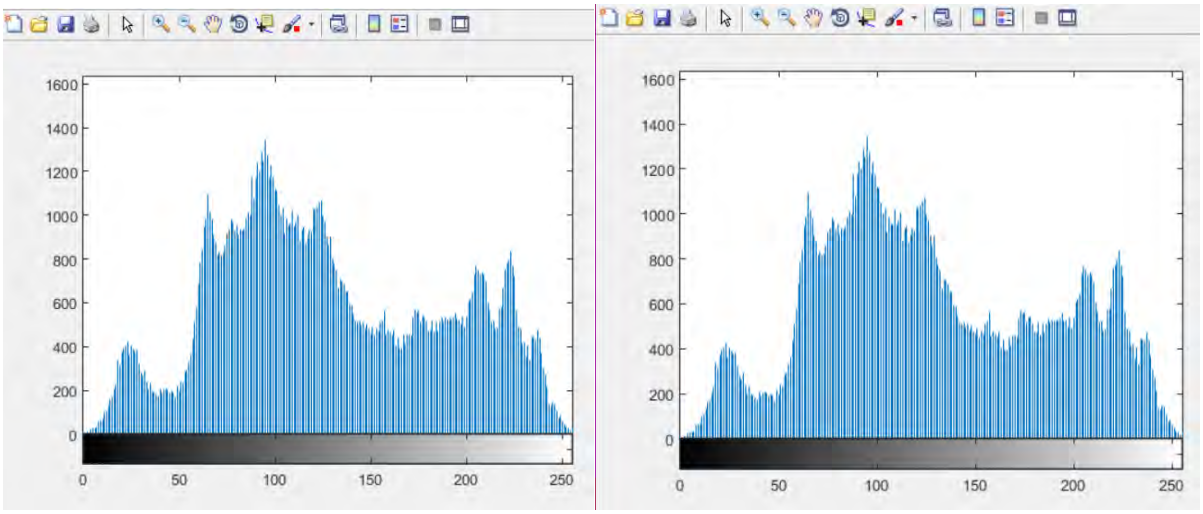


A. Imagen Tool 1 C B. Imagen Tool 1 S



C. Región de pixeles (Imagen Tool 1)

D. Región de pixeles (Imagen Tool 2)



E. Histograma C

F. Histograma S

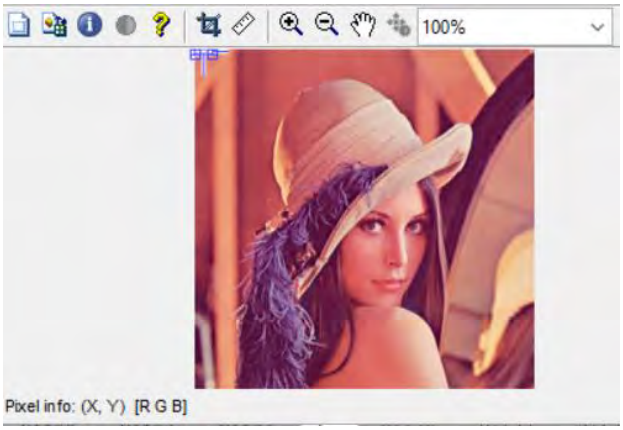
Figura 47 Representación de la modificación de los bits

En la figura 47(A) se muestra la imagen original que fue utilizada para la inserción del mensaje, la figura 47(B) es la figura que se genera una vez que se realiza el proceso de inserción del bit menos significativo (LSB), en la figura 47(C), (imagen portadora) se muestra una región de pixeles de la figura original, que será comparada con la figura 47(D) que es la región en donde ya fueron modificados los bits.

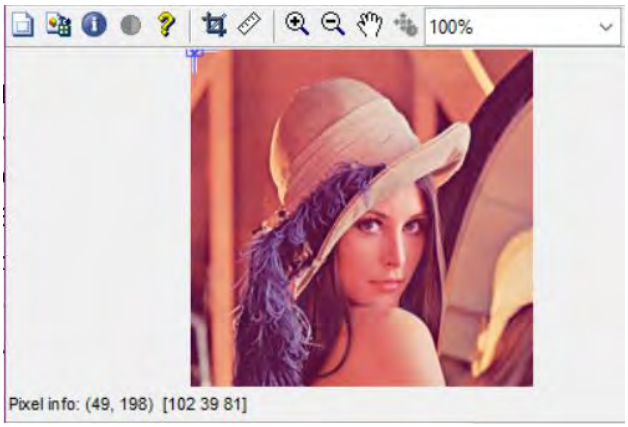
Tomando en cuenta la región de pixeles que fueron modificados, se obtiene como resultado que visiblemente no existe ningún cambio y haciendo la comparación de los valores entre la figura 47(C), la figura 47(D) son iguales y no existe ninguna modificación, o tomando en cuenta en la figura 47(E) y en la figura 47(F). Esto se debe a que la longitud

del mensaje que se inserta (100 caracteres o una representación binaria de 800 bits) es muy pequeña en comparación a la longitud máxima de la información que se puede ocultar. De la misma forma como se realizó el proceso de aleatorización es muy impredecible deducir en donde fueron modificados los bits.

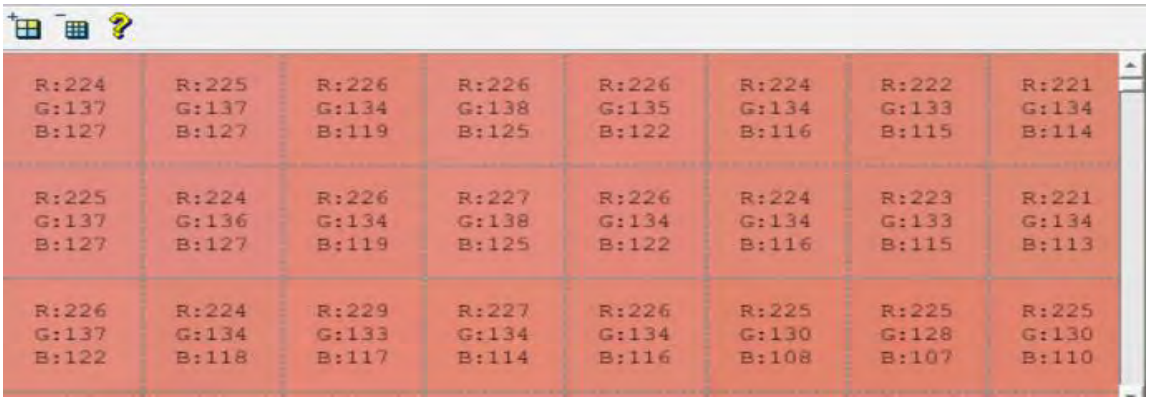
Con el objetivo de probar la funcionalidad del algoritmo LSB se realizaron pruebas sobre este mismo, utilizando el código sin la parte de aleatorización, de tal manera que la inserción se realiza de manera secuencial con el motivo de fácil identificar los cambios de valores de bits que fueron modificados, como se muestra en la figura 47.



A. Imagen Tool 1 C

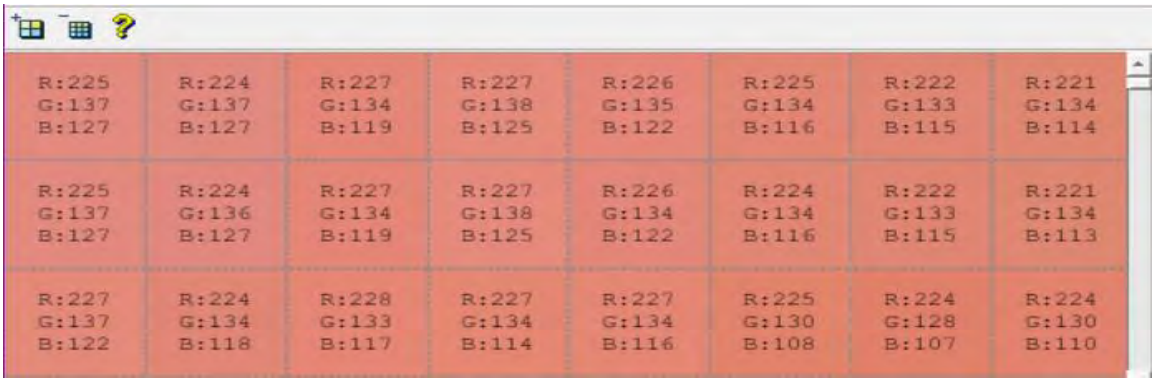


B. Imagen Tool 2 S



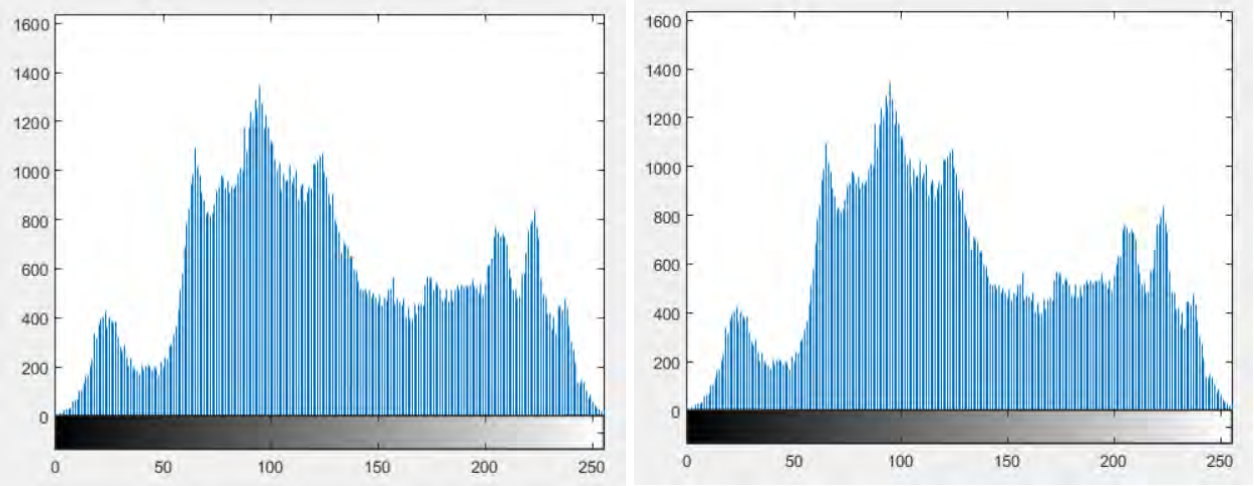
R:224 G:137 B:127	R:225 G:137 B:127	R:226 G:134 B:119	R:226 G:138 B:125	R:226 G:135 B:122	R:224 G:134 B:116	R:222 G:133 B:115	R:221 G:134 B:114
R:225 G:137 B:127	R:224 G:136 B:127	R:226 G:134 B:119	R:227 G:138 B:125	R:226 G:134 B:122	R:224 G:134 B:116	R:223 G:133 B:115	R:221 G:134 B:113
R:226 G:137 B:122	R:224 G:134 B:118	R:229 G:133 B:117	R:227 G:134 B:114	R:226 G:134 B:116	R:225 G:130 B:108	R:225 G:128 B:107	R:225 G:130 B:110

C. Región de pixeles (Imagen Tool 1)



R:225 G:137 B:127	R:224 G:137 B:127	R:227 G:134 B:119	R:227 G:138 B:125	R:226 G:135 B:122	R:225 G:134 B:116	R:222 G:133 B:115	R:221 G:134 B:114
R:225 G:137 B:127	R:224 G:136 B:127	R:227 G:134 B:119	R:227 G:138 B:125	R:226 G:134 B:122	R:224 G:134 B:116	R:222 G:133 B:115	R:221 G:134 B:113
R:227 G:137 B:122	R:224 G:134 B:118	R:228 G:133 B:117	R:227 G:134 B:114	R:227 G:134 B:116	R:225 G:130 B:108	R:224 G:128 B:107	R:224 G:130 B:110

D. Región de pixeles (Imagen Tool 2)



E. Histograma C F. Histograma S

Figura 48 Demostración de funcionalidad del algoritmo LSB

Para una mejor explicación sobre la figura anterior se tomará una cadena de los primeros 8 bits pertenecientes a el vector binario b(k) correspondiente al ciclo **`for`**, en donde realiza la conversión de cadena binaria a número binario del mensaje a insertar en la imagen (ver Anexo C del código sin aleatorización) para demostrar el proceso de inserción.

Para estas pruebas el mensaje a transmitir es el siguiente:

```
'- -Esta es una prueba de esteganografia a traves de substitucion de bit menos significativo o LSB- -'
```

De tal manera que los primeros 8 bits corresponden a el carácter `'-'` que representado en código ASCII equivale al número 45 y en representación binaria equivale al siguiente número binario: 00101101 (ver fgiura 49).

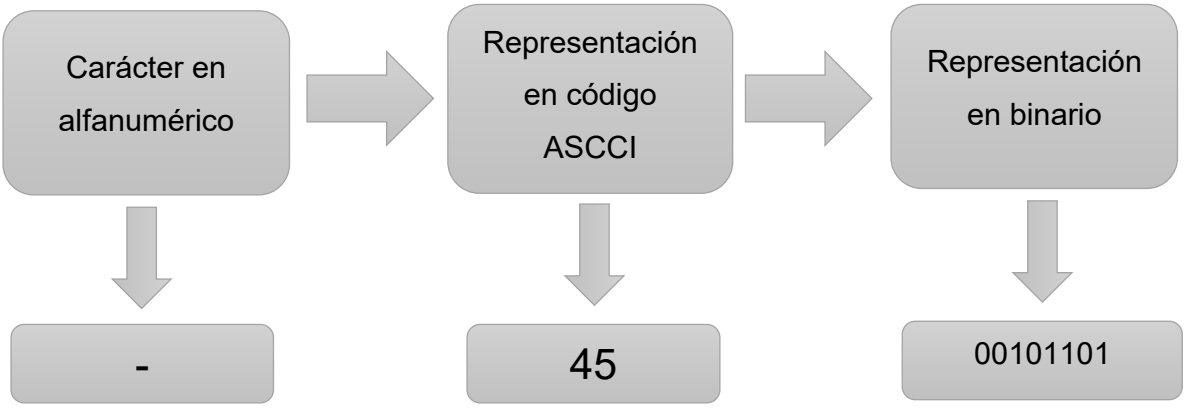


Figura 49 Representación de la conversión del carácter '-' a número binario

En la figura 50 se identifican las posiciones de los pixeles que fueron modificados en la imagen c (imagen portadora), tomando los primeros 8 pixeles de la imagen para explicar la representación de los primeros 8 bits de la cadena del mensaje.

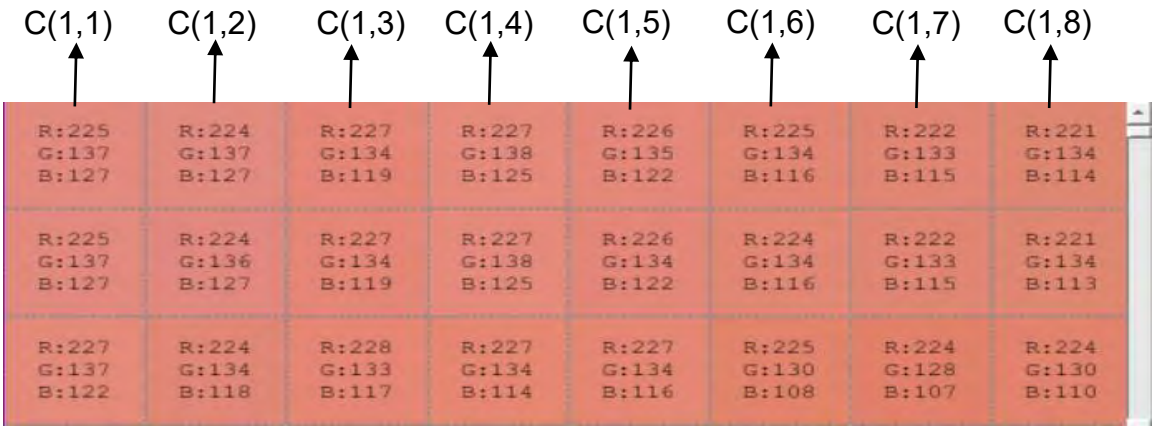


Figura 50 Posición sin aleatorización

En escala de color los valores que pueden tomar van de 0 a 255, en donde el 0 representa al color negro hasta el 255 que corresponde al color blanco y se la aplican en el RGB para obtener diferentes colores obtenibles.

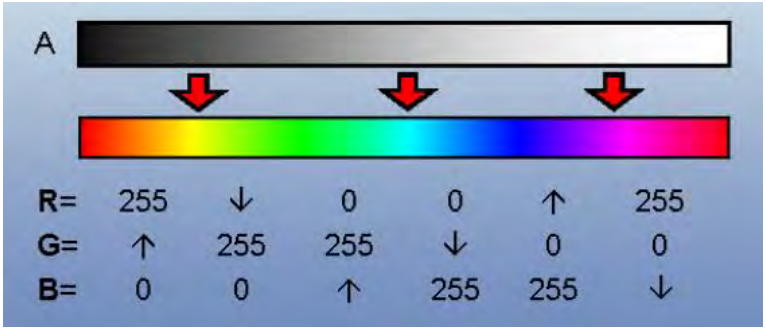


Figura 51 Valores del pixel en escala de color

En la siguiente tabla se explica el proceso en el que se lleva a cabo la inserción de los bits.

Tabla 4 inserción de los primeros 8 bits de la cadena binaria del mensaje

Representación de los primeros 8 bits a insertar	Valor del pixel en la posición `c(i,j)`	Valor en binario de la posición `c(i,j)` Valor	Resultado del pixel en imagens (imagen estegoportadora)
0	156	10011100	10011100
0	155	10011011	10011100
1	156	10011100	10011101
0	153	10011001	10011100
1	154	10011010	10011101
1	158	10011110	10011101
0	155	10011011	10011100
1	154	10011010	10011101

El primer valor de la fila 1 que se desea insertar es **0**, pero el valor del byte en la posición c(1,1) es 156 que en binario equivale al número 1001110**0** y vemos que el ultimo bit es equivalente al valor del bit que se desea insertar por lo tanto se le suma un cero (ver el bloque llamado Implementación del LSB líneas 35-38, del Capítulo 3) de esta forma conserva su valor 1001110**0**.

El segundo valor de la fila 2 que se desea insertar es **0** y el valor del byte en la posición c(1,2) es igual a 155 que en número binario corresponde a 1001101**1**, en este caso el número que

se desea insertar es distinto al valor del bit menos significativo al pixel correspondiente. Por lo que se le resta uno para que el bit menos significativo sea equivalente al bit que se desea insertar, quedando de la siguiente manera 1001110**0**.

```
The PeakSNR value is 75.8766
The SSIM value is 1.0000.
The MSE value is 0.0017.
The Quality image value is 1.0000.
```

Figura 52 Resultado de cada métrica en Tx normal

En la Figura 52 se muestra los resultados de cada cálculo en las figuras 40-44 cuando está cifrado la imagen. El rango dinámico de SSIM y del UIQI (calidad de la imagen, image quality) es [0,1], siendo de mejor calidad cuando vale 1. Se mide qué tan cerca está la luminancia media entre las imágenes o qué tan similares son los contrastes de las imágenes [24].

```
ABCDEFGHIJKLMNOPQRZ12345678910ABCDEFGHIJKLMNOPQRZ12345678910!"#$%&/()()()()
()ABCDEFGHIJKLMNOPQRZ1234

Errores =
     0
```

Figura 53 Resultado Rx normal

La Figura 53 muestra el resultado en Rx normal descifrando la imagen modificada o stego-imagen, mostrando el mensaje que está cifrado y mostrando los errores si se encuentra algunas.

```
SNR_db =
   66.4633
```

Figura 54 Resultado Tx Random

La figura 54 se muestra el SNR (signal to noise ratio) o el ruido, se mide el ruido o la sensibilidad de la imagen, muy parecido a PSNR [23]. Finalmente, la Figura 55 muestra el resultado de Rx random descifrando el mensaje de la imagen recibida.

```
ABCDEFGHiJKLMNOPQRZ02345678910ABCDEFGHIJKLMNOPQRZ12345678910!"#$%&/()()()()
()ABCDEFGHIJKLMNOPQRZ1234

Errores =
     0
```

Figura 55 Resultado Rx random

Capítulo 4 Conclusiones

El propósito de la esteganografía es definir una estrategia relacionada a la ocultación de un mensaje, o datos de un tercero, o información a enviar. Esto difiere de la criptografía, el arte de la escritura secreta, que pretende hacer que un mensaje sea ilegible para un tercero, pero no oculta la existencia de la comunicación secreta. Los resultados experimentales obtenidos indican que el método propuesto será un esquema de esteganografía aceptable. Lo anterior puede comprobarse observando los resultados de cálculo de las métricas utilizadas para medir el desempeño del algoritmo.

En lo que se refiere a mi experiencia con este tipo de implementación de algoritmos, fue necesario realizar una investigación de los antecedentes, trabajos propuestos hasta ahora, etc., y en consecuencia conocer el impacto de este tema en la actualidad referente a la seguridad de la información. De igual manera, entender la forma en que se puede manipular o programar a través de Matlab ya que mi conocimiento de este lenguaje de programación era mínimo. En consecuencia, este proyecto de tesis fue como una nueva experiencia para aprender y conocer más sobre el aprovechamiento de Matlab, sobre esteganografía y cálculo de sus métricas, lo cual fue necesario para medir el desempeño de la implementación.

Referencias

[1] Dipesh Agrawal, Samidha D.Sharma. (2013). Analysis of Random Bit Image Steganography Techniques. International Conference on Recent Trends in engineering & Technology

[2] Araceli De La Cruz Franco. (2017). Implentacion de un algorithmo computacional para esteganografia basado en tecnicas del bit menos significativo. UQROO

[3] Dr. Alfonso M. (2014). Curso de privacidad y protección de comunicaciones digitales. 02/01, de Universidad Politecnica de Madrid Sitio web: http://www.criptored.upm.es/crypt4you/temas/privacidad-proteccion/leccion7/leccion7.html

[4] Inas J. K., Prashan P., Peter J. V., Brendan H., (2018). Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research. Neurocomputing: IJ.

[5] Shiv Prasad and Arup Kumar Pal. (2017). An RGB colour image steganography scheme using overlapping block-based pixel-value differencing. 04/26, de PMC Sitio web: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5414260/

[6] Shashank. (2018). What is Cryptography? – An Introduction to Cryptographic Algorithms. Oct 31, de edureka Sitio web: https://www.edureka.co/blog/what-is-cryptography/#Encrypt

[7] Lee YP, Lee J-C, Chen W-K, Chang K-C, Su I-J, Chang C-P. 2012. High-payload image hiding with quality recovery using tri-way pixel-value differencing. Inf. Sci. 191, 214–225.

[8] Al-Otaibi N, Gutub A. 2014. Flexible stego-system for hiding text in images of personal computers based on user security priority. In *Proc. Int. Conf. on Advanced Engineering Technologies (AET-2014), Dubai, UAE*, pp. 250–256.

[9] Das R, Das I. 2016. Secure data transfer in IoT environment: adopting both cryptography and steganography techniques. In *Proc. 2nd Int. Conf. on Research in Computational Intelligence and Communication Networks, Kolkata, India*, pp. 296–301.

[10] Zhou X, Gong W, Fu W, Jin L. 2016. An improved method for LSB based color image steganography combined with cryptography. In 2016 IEEE/ACIS 15th Int. Conf. on Computer and Information Science (ICIS), Okayama, Japan, pp. 1–4.

[11] Chan CK, Cheng LM. 2004. Hiding data in images by simple LSB substitution. Pattern Recognit. 37, 469–474.

[12] R. D. V. Velasco, «Criptografía, una necesidad moderna,» vol. 7, 2006.

[13] G. G. Paredes, «Introducción a la criptografía,» Revista digital Universitaria, vol. 7, p. 17, 2006.

[14] Muños, «Criptored.upm.es,» 2 enero 2014. [En línea]. Available: http://www.criptored.upm.es/crypt4you/temas/privacidad-proteccion/leccion7/leccion7.html. [Último acceso: 20 abril 2017].

[15] D. Artz, «Digital Steganography: Hiding Data within Data, » IEEE Internet computing, p. 6, 2001.

[16] E. A. M. Checa, JUlio 2014. [En línea]. Available: http://bibdigital.epn.edu.ec/handle/15000/8062. [Último acceso: 3 Julio 2017].

[17] M. Balleste, «Esteganografía en contenido multimedia,» 2007.

[18] P. A. Deymonnaz, «Análisis de vulnerabilidades esteganográficas en protocolos de comunicación IP y HTTP,» Facultad de Ingeniería, Buenos Aires, 2012.

[19] Wikipedia. (.). Johannes Trithemius. Sitio web: https://en.wikipedia.org/wiki/Johannes_Trithemius#Steganographia

[20] Wikipedia. (.). Esteganografía. Sitio web: https://es.wikipedia.org/wiki/Esteganograf%C3%ADa#Siglo_XV

[21] Wikipedia. (.). John Wilkins. Sitio web: https://es.wikipedia.org/wiki/John_Wilkins

[22] Wikipedia. (.). Gaspar Schott. Sitio web: https://en.wikipedia.org/wiki/Gaspar_Schott

[23] Wikipedia. (.). SNR. Sitio web: https://en.wikipedia.org/wiki/Signal-to-noise_ratio_(imaging)

[24] Zhou W., Alan C. B. (2002). A Universal Image Quality Index. IEEE.

[25] Avneet K., Lakhwinder K., and Savita G., (2012). Image RecognitionusingCoefficient of Correlation and Structural SIMilarity Index in Uncontrolled EnvironmentImage RecognitionusingCoefficient of Correlation and Structural SIMilarity Index in Uncontrolled Environment. nternational Journal of Computer Applications: Volume 59–No.5.

Anexos

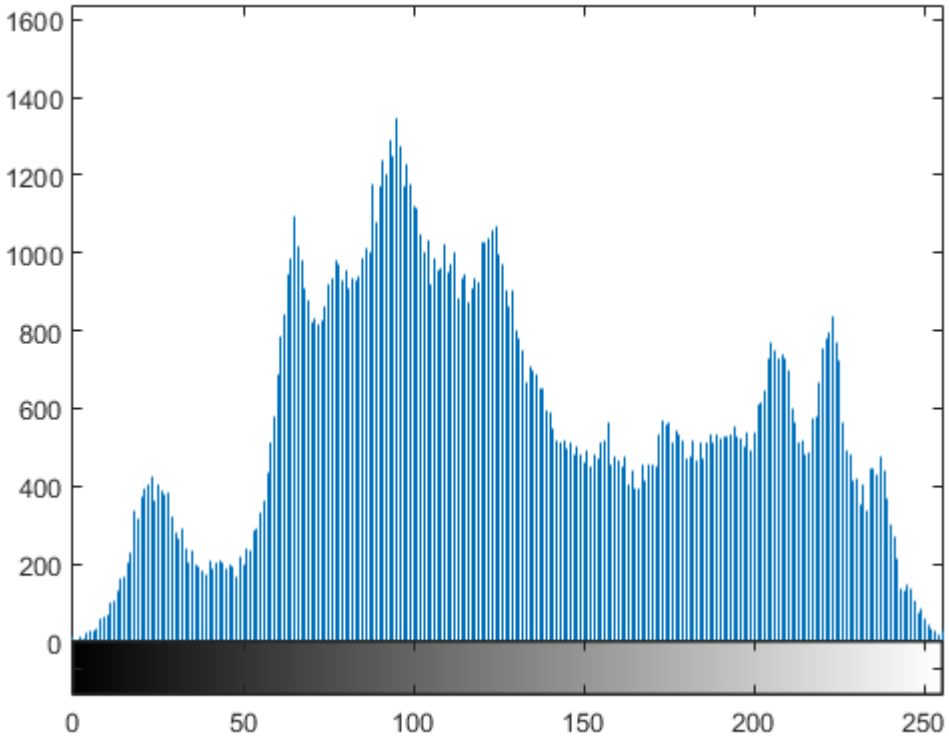
Anexo A Transmisor Tx Normal

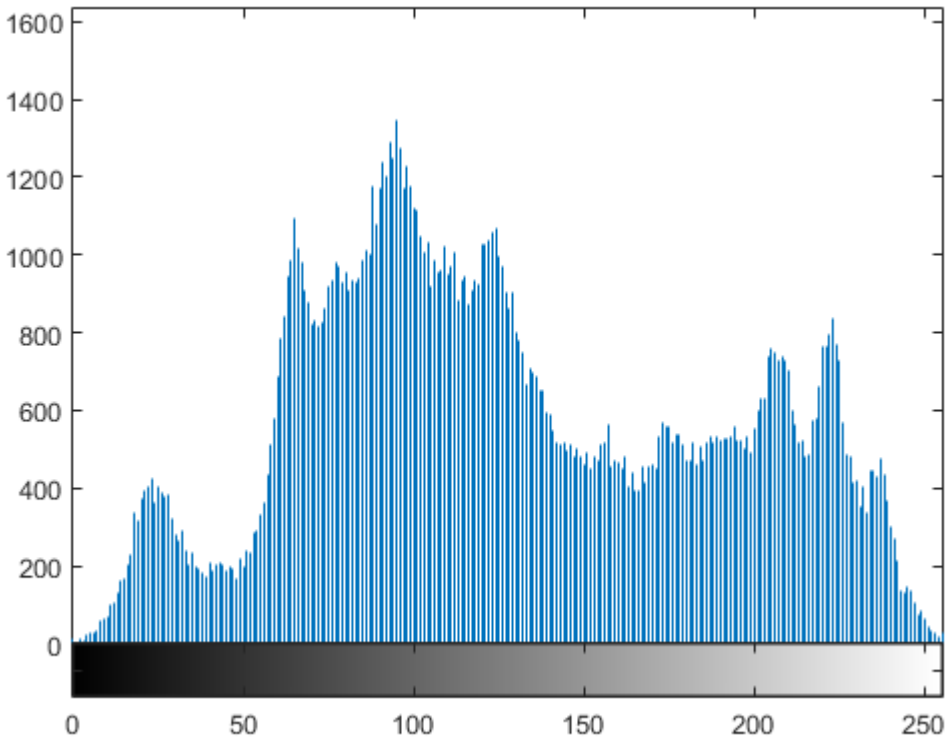
```
clear all;
clc;

c = (imread ('Lenna_color.png'));
   message =
'ABCDEFGHIJKLMNOPQRZ12345678910ABCDEFGHIJKLMNOPQRZ12345678910!"#$%&/()()()()()ABCDEFGHI
JKLMNOPQRZ1234';
%   message = '- -Esta es una prueba de esteganografia a traves de substitucion de bit
menos significativo o LSB- -';

    message = strtrim(message);
    m = length(message) * 8;
    AsciiCode = uint8(message);  %decrypting messages to ascii code(number)
    binaryString = transpose(dec2bin(AsciiCode,8)); %converting ascii code to binary
num
    binaryString = binaryString(:);
    N = length(binaryString);
    b = zeros(N,1); %b is a vector of bits
    for k = 1:N   %changing string to num, removing '' from the '1'
        if(binaryString(k) == '1')
            b(k) = 1;
        else
            b(k) = 0;
        end
    end
    s = c;
    height = size(c,1);
    width = size(c,2);
    k = 1;
    for i = 1 : height
        for j = 1 : width

            LSB = mod(double(c(i,j)), 2);
            if (k>m || LSB == b(k))
                s(i,j) = c(i,j);
            else
                if (LSB==1)
                    s(i,j) = c(i,j) - 1;
                else
                    s(i,j) = c(i,j) + 1;
                end
            end
            k = k + 1;
        end
    end
     figure(1)
     imshow(c)
     title('Ori-Image');
```

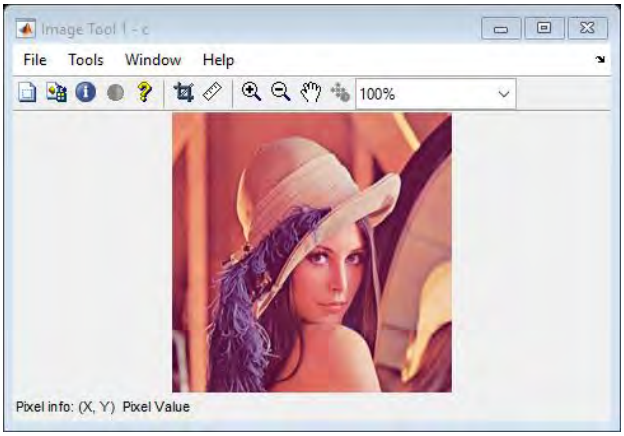
```
    figure(2)
    imshow(s)
    %imwrite(s, '111.png');
    imwrite(s, '112.png');
    title('Stego-Image');
    figure(3)
    imhist(c(:))
    figure(4)
    imhist(s(:))  %histogram graph

    imtool(c)  %RGB scale
    imtool(s)
% ------------------------------------------------------------------------
```

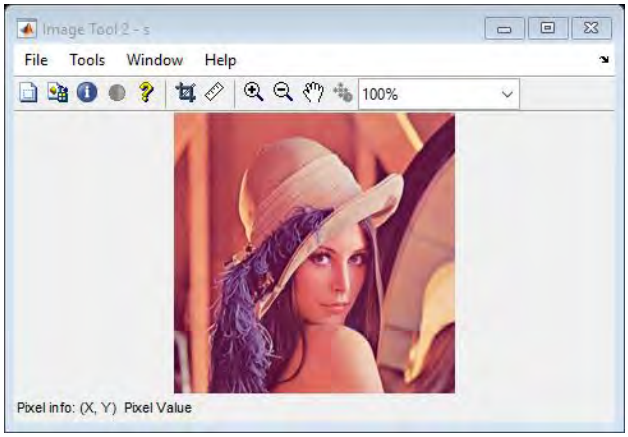

Ori-Image


Stego-Image


1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250







Correlation Factor

```
%corr = xcorr2(numel(c),numel(s));
cR = c(:,:,1); %red
ccR = cR(:);
sR = s(:,:,1); %red
ssR = sR(:);
mean_ccR = sum(ccR)./length(ccR);
mean_ssR = sum(ssR)./length(ssR);
corrfac = ( sum((ccR-mean_ccR).*(ssR-mean_ssR)) ) / sqrt( sum((ccR-mean_ccR).^2) *
sum((ssR-mean_ssR).^2) );

sigma_cc = sum((ccR-mean_ccR).^2) / (length(ccR) - 1);
sigma_ss = sum((ssR-mean_ssR).^2) / (length(ssR) - 1);
sigma_cc_ss = sum((ccR-mean_ccR) .* (ssR-mean_ssR)) / (length(ccR) - 1);
```

```
SSIM = (2.*(mean_ccR * mean_ssR + ccR)) .* (2*(sigma_cc_ss) + ssR) ./ ((mean_ccR^2 +
mean_ssR^2 + ccR) .* (sigma_cc + sigma_ss + ssR));
%--------------------------------------------------------------------------
```

PSNR & MSE

```
%[peaksnr, snr] = psnr(c(:), s(:))
cc = c(:);
ss = s(:);
%mse = immse(c,s);
mse = sum((cc-ss).^2)./length(cc);
%mse = sum((c(:)-s(:)).^2);
%mse = sum((c(:) - s(:)).^2) / numel(c); %numel returns number of elements in the table
%dummy = c(:);
%mse = mse./length(dummy);
PSNR = 10*log10(255^2./mse);

%--------------------------------------------------------------------------
```

SSIM

```
[ssimval, ssimmap] = ssim(s,c);

%--------------------------------------------------------------------------
```

UIQI Universal Image Quality Index

```
%UIQI = img_qi(c, s)
cR = c(:,:,1); %red
cG = c(:,:,2); %green
cB = c(:,:,3); %blue
mean_cR = mean(cR(:));
mean_cG = mean(cG(:));
mean_cB = mean(cB(:));

sR = s(:,:,1);
sG = s(:,:,2);
sB = s(:,:,3);
mean_sR = mean(sR(:));
mean_sG = mean(sG(:));
mean_sB = mean(sB(:));

cc = c(:);
ss = s(:);
Fc = cc-mean_cR;
Fs = ss-mean_sR;

c_sum = sum((Fc).^2);
Sigma_c = c_sum./(length(cc)-1);
s_sum = sum((Fs).^2);
Sigma_s = s_sum./(length(ss)-1);
Sigma_cs = sum(Fc.*Fs)./(length(cc)-1);
Q = (4.*Sigma_cs.*mean_cR.*mean_sR)./((Sigma_c + Sigma_s).*(mean_cR.^2 + mean_sR.^2));
```

```
%-------------------------------------------------------------------------

fprintf('\n The PeakSNR value is %0.4f', PSNR);
fprintf('\n The SSIM value is %0.4f.\n',ssimval);
fprintf(' The MSE value is %0.4f.\n',mse);
fprintf(' The Quality image value is %0.4f.\n',Q);

%-------------------------------------------------------------------------
```

```
 The PeakSNR value is 75.8766

 The SSIM value is 1.0000.

 The MSE value is 0.0017.

 The Quality image value is 1.0000.
```

Anexo B Receptor Rx Normal

```
s = imread('112.png');
    height = size(s,1);
    width = size(s,2);
    %For this example the max size is 100 bytes, or 800 bits, (bytes * = bits
    % m = N;
    m = 800;
    k = 1;
    for i = 1 : height
        for j = 1 : width
            if (k <= m)
                br(k) = mod(double(s(i,j)),2);
                k = k + 1;
            end
        end
    end
    binaryVector = br;
     binValues = [ 128 64 32 16 8 4 2 1 ];
     binaryVector = binaryVector(:);
     if mod(length(binaryVector),8) ~= 0
         error('Length of binary vector must be a multiple of 8.');
     end
     binMatrix = reshape(binaryVector,8,100);
     %display(binMatrix);
     textString = char(binValues*binMatrix);
     disp(textString);

bt = b';
Errores = sum (br-bt)
```

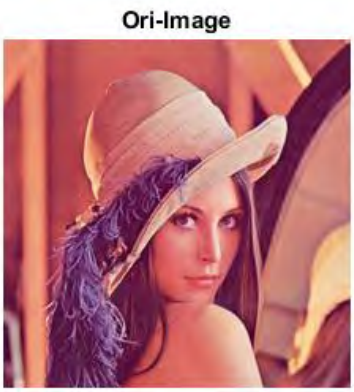
```
ABCDEFGHIJKLMNOPQRZ12345678910ABCDEFGHIJKLMNOPQRZ12345678910!"#$%&/()()()()()ABCDEFGHIJ
KLMNOPQRZ1234

Errores =
```

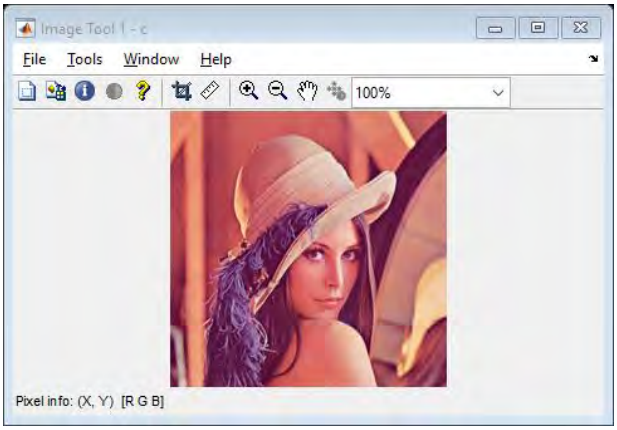
0

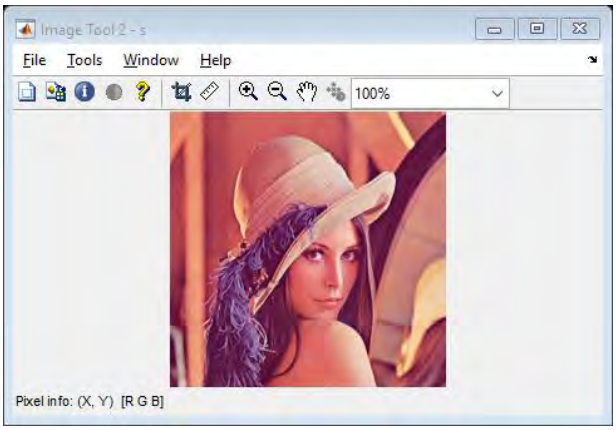
Anexo C Transmisor Tx Aleatorización

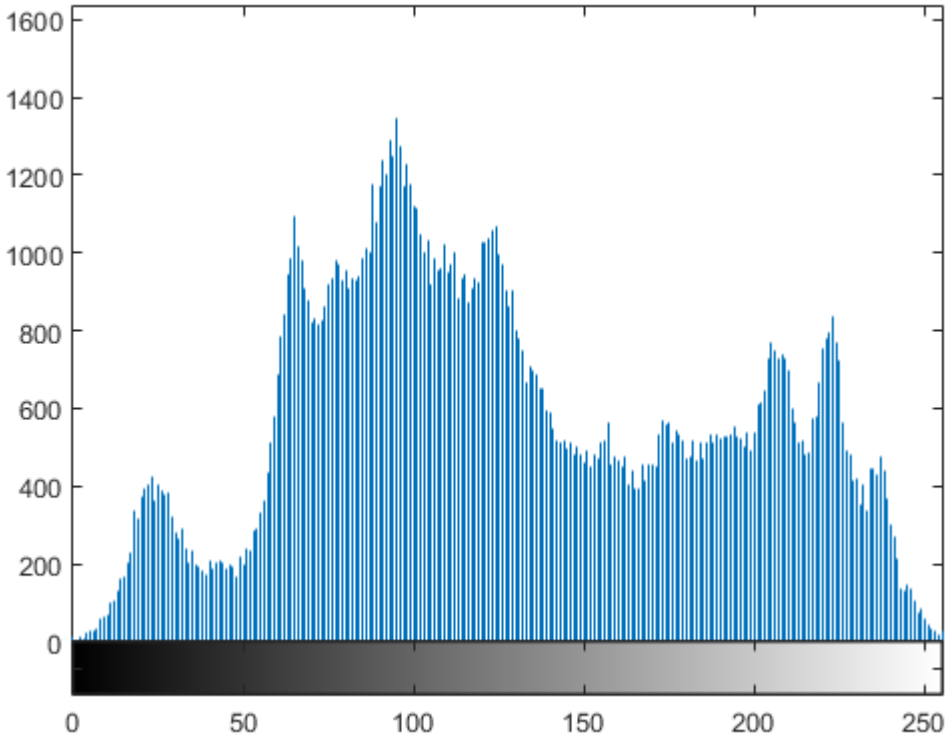
```
clear all;
clc;

c = imread('Lenna_color.png');
     message =
'ABCDEFGHIJKLMNOPQRZ12345678910ABCDEFGHIJKLMNOPQRZ12345678910!"#$%&/()()()()()ABCDEFGHI
JKLMNOPQRZ1234';
 %   message = '- -Esta es una prueba de esteganografia a traves de substitucion de bit
menos significativo o LSB- -';

    message = strtrim(message);
    m = length(message) * 8;
    AsciiCode = uint8(message);
    binaryString = transpose(dec2bin(AsciiCode,8));
    binaryString = binaryString(:);
    N = length(binaryString);
    b = zeros(N,1); %b is a vector of bits
    for k = 1:N
        if(binaryString(k) == '1')
            b(k) = 1;
        else
            b(k) = 0;
        end
    end
    s = c;
    height = size(c,1);
    width = size(c,2);
    k = 1;

    rng(1,'twister');                % aleatorizacion de la posicion
    pos_h = randi(height,1,N+1);
    pos_w = randi(width,1,N+1);



    for l = 1:N
    %for i = 1 : height
     %   for j = 1 : width

            i = pos_h(l);
            j = pos_w(l);
             LSB = mod(double(c(i,j)), 2);
            if (k>m || LSB == b(k))
                s(i,j) = c(i,j);
            else
                if (LSB==1)
                    s(i,j) = c(i,j) - 1;
```

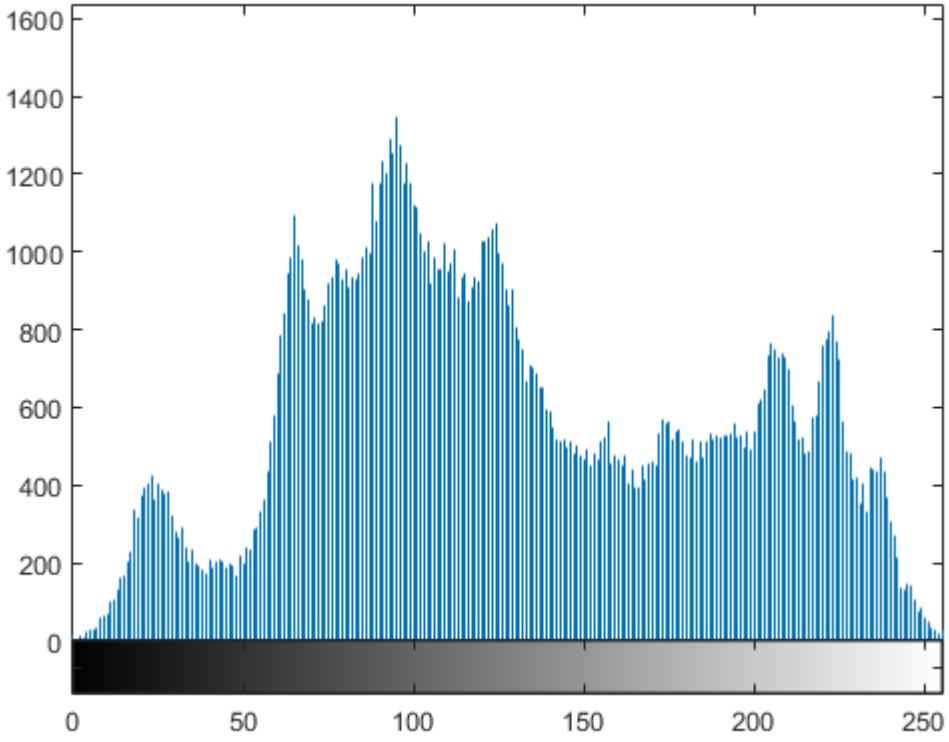
```
            else
                s(i,j) = c(i,j) + 1;
            end
        end
        k = k + 1;
   %  end
%end
end
 figure(1)
 imshow(c)
 title('Ori-Image');
 figure(2)
 imshow(s)
 imwrite(s, '112.png');
 title('Stego-Image');
 figure(3)
 imtool(c)
 figure(4)
 imtool(s)
 figure(5)
 imhist(c(:))
 figure(6)
 imhist(s(:))
```








1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250


1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250

SNR computation

```
    Power_c_image     = 0;

    Power_error_image = 0;



       for i = 1 : height
         for j = 1 : width
             dum_c     = double (c(i,j));
             dum_s     = double (s(i,j));
             dum_noisy = dum_c-dum_s;
             Power_c_image     = Power_c_image + dum_c.^2;
             Power_error_image = Power_error_image + (dum_noisy.^2);
         end
       end

       SNR_db = 10 * log10(Power_c_image/Power_error_image);
```

Correlation Factor

```
%corr = xcorr2(numel(c),numel(s));
cR = c(:,:,1); %red
ccR = cR(:);
sR = s(:,:,1); %red
ssR = sR(:);
mean_ccR = sum(ccR)./length(ccR);
mean_ssR = sum(ssR)./length(ssR);
corrfac = ( sum((ccR-mean_ccR).*(ssR-mean_ssR)) ) / sqrt( sum((ccR-mean_ccR).^2) *
sum((ssR-mean_ssR).^2) );

sigma_cc = sum((ccR-mean_ccR).^2) / (length(ccR) - 1);
sigma_ss = sum((ssR-mean_ssR).^2) / (length(ssR) - 1);
sigma_cc_ss = sum((ccR-mean_ccR) .* (ssR-mean_ssR)) / (length(ccR) - 1);
SSIM = (2.*(mean_ccR * mean_ssR + ccR)) .* (2*(sigma_cc_ss) + ssR) ./ ((mean_ccR^2 +
mean_ssR^2 + ccR) .* (sigma_cc + sigma_ss + ssR));
%-------------------------------------------------------------------------
```

PSNR & MSE

```
%[peaksnr, snr] = psnr(c(:), s(:))
cc = c(:);
ss = s(:);
%mse = immse(c,s);
mse = sum((cc-ss).^2)./length(cc);
%mse = sum((c(:)-s(:)).^2);
%mse = sum((c(:) - s(:)).^2) / numel(c); %numel returns number of elements in the table
%dummy = c(:);
%mse = mse./length(dummy);
PSNR = 10*log10(255^2./mse);
```

```
%-------------------------------------------------------------------------
```

SSIM

```
[ssimval, ssimmap] = ssim(s,c);

%-------------------------------------------------------------------------
```

UIQI Universal Image Quality Index

```
%UIQI = img_qi(c, s)
cR = c(:,:,1); %red
cG = c(:,:,2); %green
cB = c(:,:,3); %blue
mean_cR = mean(cR(:));
mean_cG = mean(cG(:));
mean_cB = mean(cB(:));

sR = s(:,:,1);
sG = s(:,:,2);
sB = s(:,:,3);
mean_sR = mean(sR(:));
mean_sG = mean(sG(:));
mean_sB = mean(sB(:));

cc = c(:);
ss = s(:);
Fc = cc-mean_cR;
Fs = ss-mean_sR;

c_sum = sum((Fc).^2);
Sigma_c = c_sum./(length(cc)-1);
s_sum = sum((Fs).^2);
Sigma_s = s_sum./(length(ss)-1);
Sigma_cs = sum(Fc.*Fs)./(length(cc)-1);
Q = (4.*Sigma_cs.*mean_cR.*mean_sR)./((Sigma_c + Sigma_s).*(mean_cR.^2 + mean_sR.^2));
%-------------------------------------------------------------------------
fprintf(' The SNR value is %0.4f.\n',SNR_db);
fprintf(' The PeakSNR value is %0.4f.\n', PSNR);
fprintf(' The SSIM value is %0.4f.\n',ssimval);
fprintf(' The MSE value is %0.4f.\n',mse);
fprintf(' The Quality image value is %0.4f.\n',Q);
```

```
 The SNR value is 66.4633.

 The PeakSNR value is 76.0030.

 The SSIM value is 1.0000.

 The MSE value is 0.0016.

 The Quality image value is 1.0000.
```

Anexo B Receptor Rx Aleatorización

```
s = imread('112.png');
    height = size(s,1);
    width = size(s,2);
    %For this example the max size is 100 bytes, or 800 bits, (bytes * = bits
    % m = N;
    m = N;
    k = 1;

    rng(1,'twister');
    pos_h = randi(height,1,N+1);
    pos_w = randi(width,1,N+1);

   for l = 1:N
    %for i = 1 : height
    %    for j = 1 : width
            i = pos_h(l);
            j = pos_w(l);

            if (k <= m)
                br(k) = mod(double(s(i,j)),2);
                k = k + 1;
            end
     %   end
    %end
    end


    binaryVector = br;
     binValues = [ 128 64 32 16 8 4 2 1 ];
     binaryVector = binaryVector(:);
     if mod(length(binaryVector),8) ~= 0
         error('Length of binary vector must be a multiple of 8.');
     end
     binMatrix = reshape(binaryVector,8,100);
     %display(binMatrix);
     textString = char(binValues*binMatrix);
     disp(textString);


bt = b';

Errores = sum (br-bt)
```

```
ABCDEFGHiJKLMNOPQRZ02345678910ABCDEFGHIJKLMNOPQRZ12345678910!"#$%&/()()()()()ABCDEFGHIJ
KLMNOPQRZ1234

Errores =

     0
```