Trabajo Práctico: Esteganografía

Criptografía y Seguridad (72.44)

Arce, Julian Francisco 60509
Conca, Maria Victoria 58661
Domingues, Paula Andrea 60148

24 de junio de 2022

Índice

1. Discutir los siguientes aspectos relativos al documento.

1.1. Organización formal del documento.

El formato del documento está dividido en 4 secciones: Introducción, Implementación, Resultados y Conclusiones. En la introducción se explica en qué consiste la esteganografía. En la sección de implementación se muestra el algoritmo clásico LSB, y se proponen dos nuevos algoritmos para mejorar el proceso de esteganografiado. Luego, la sección de resultados analiza estas dos nuevas propuestas y compara los resultados, evidenciando la cantidad de bits que cambian en cada caso. Por último, se realizan conclusiones y proyecciones a futuro.

1.2. La descripción del algoritmo.

El primer esquema propone agarrar el 2do y 3er bit menos significativos y guardar información en torno a las 4 combinaciones posibles de estos bits. La información guardada corresponde a si se debe invertir o no el 1er bit menos significativo (que es el que contiene la información esteganografiada). Es decir, si la cantidad de bits para la combinación ij que cambia es mayor a la que no cambia, debemos guardar esos bits invertidos, para cambiar una menor cantidad. Y almacenar de alguna manera que para esa combinación ij se debe invertir el valor del lsb para obtener la información correcta.

1.3. La notación utilizada, ¿es clara? ¿hay algún error o contradicción?

La notación utilizada es confusa cuando explica cómo generar los patrones y almacenarlos, ya que supongamos que se toma el siguiente byte 0000 0010, donde los bits 2 y 3 corresponden a un patrón, generó confusión si se deben tomar como 01 o invertir y representar 10. En base a la implementación, se terminó considerando que corresponde a 01.

2. Esteganografiar un mismo archivo en un .bmp con cada uno de los tres algoritmos, y comparar los resultados obtenidos. Hacer un cuadro comparativo de los tres algoritmos estableciendo ventajas y desventajas.

Para analizar los algoritmos se va a utilizar una imagen bmp blanca (white.bmp), y luego una imagen bmp real (a4.bmp). De esta manera podremos notar mejor en la imagen blanca la incidencia del esteganografiado y en la imagen real, la incidencia práctica del mismo.

Figura 1: Imagen Original (white.bmp).

Figura 2: Imagen con LSB1 (whiteLSB1.bmp).

Figura 3: Imagen con LSB4 (whiteLSB4.bmp).

Figura 4: Imagen con LSBI (whiteLSBI.bmp).

Figura 5: Imagen Original (a4.bmp).

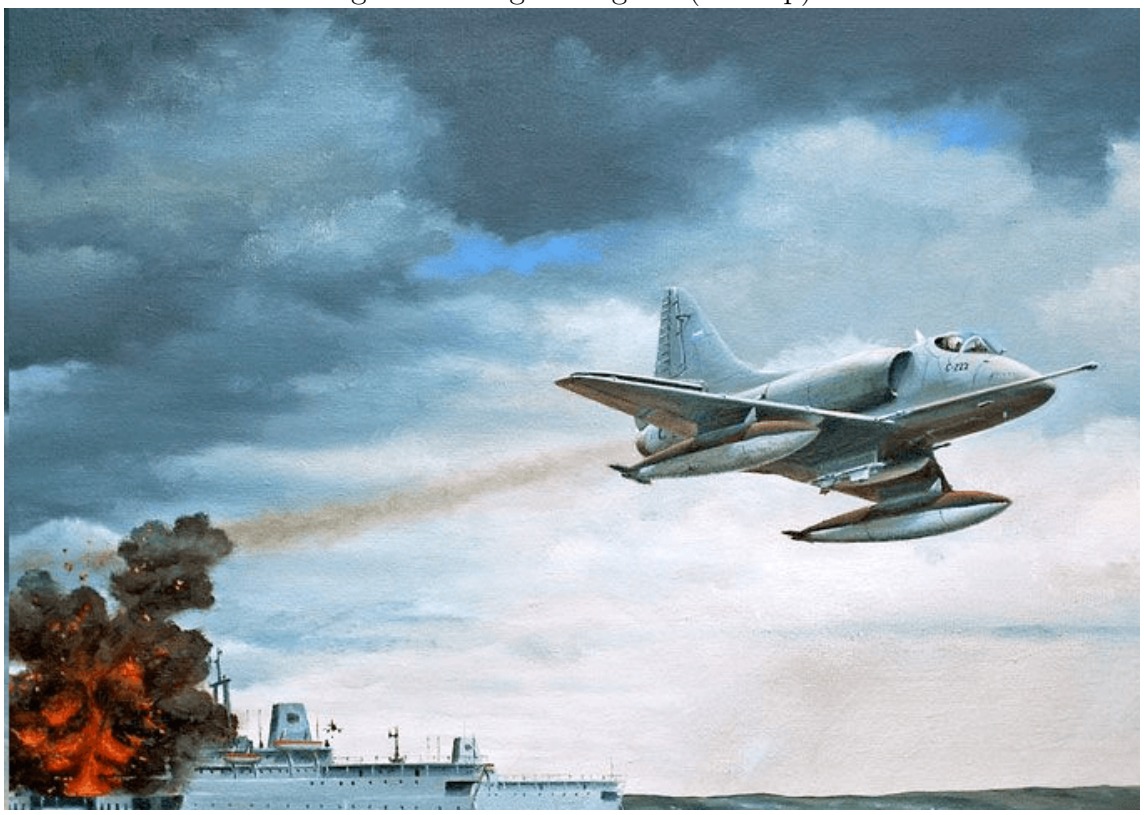


Figura 6: Imagen con LSB1 (a4LSB1.bmp).

Figura 7: Imagen con LSB4 (a4LSB4.bmp).

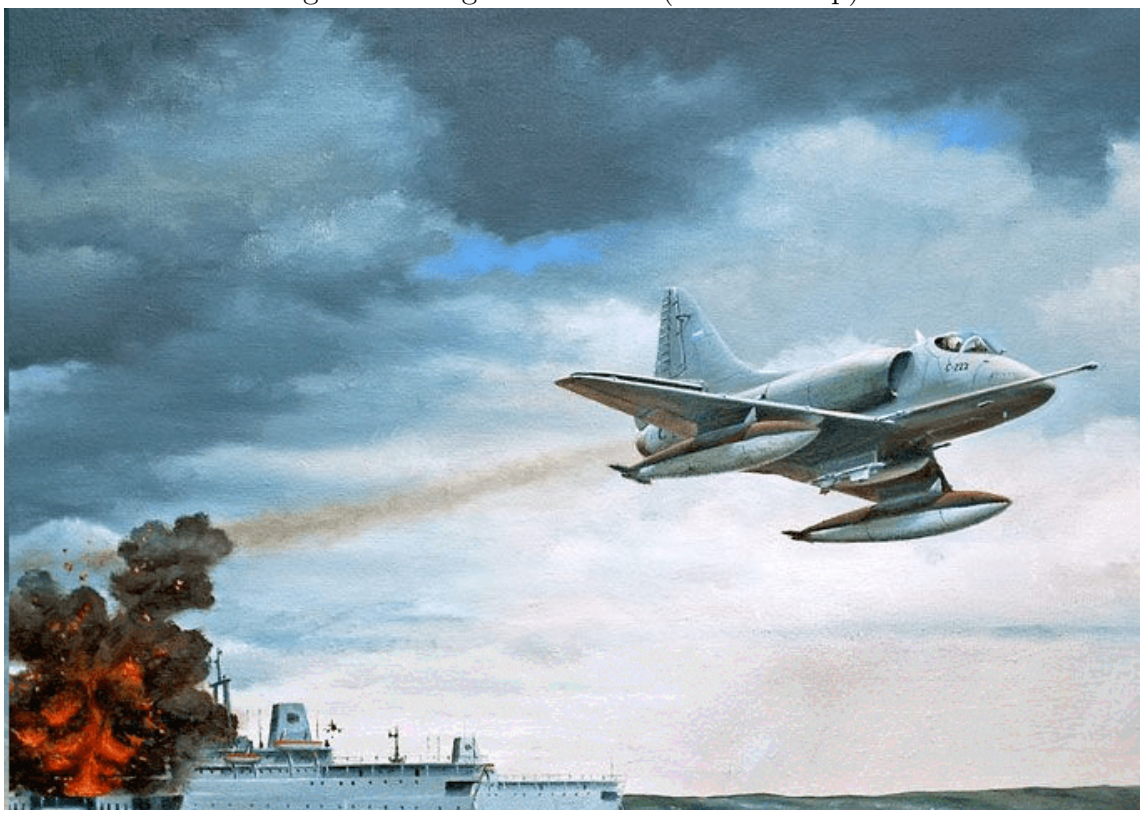


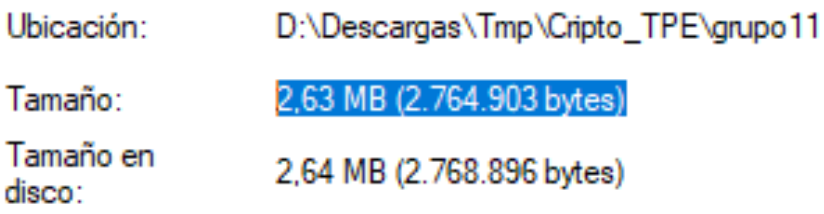
Figura 8: Imagen con LSBI (a4LSBI.bmp).

Cuadro 1: Comparación entre modos.

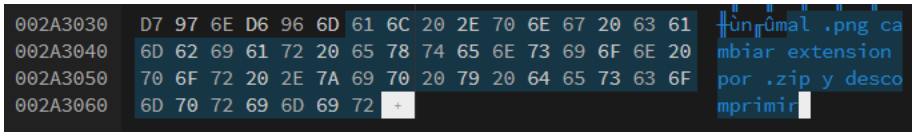
	LSB1	LSB4	LSBI
Tamaño de imagen de entrada	$\frac{1}{8}$ de la imagen portadora	$\frac{1}{2}$ de la imagen portadora	$\frac{1}{8}$ de la imagen portadora (-4 bytes para almacenar el patrón de inversión).
Inferencia de la imagen de entrada	Imperceptible	Poco perceptible si se comparan a la par de la original. Si no, no se percibe a simple vista. Ligero cambio de tonalidad	Imperceptible

3. Explicar detalladamente el procedimiento realizado para descubrir qué se había ocultado en cada archivo y de qué modo. Indicar qué se encontró en cada archivo.

Archivo 1: **buenosaires.bmp:**
Este archivo fue analizado en el programa hexed.it que permite analizar el contenido hexadecimal. Notamos que el tamaño de la imagen no coincide con la información del header. El tamaño de la imagen es 2,63 MB (2.764.903 bytes):



Si calculamos el tamaño que debería tener, hacemos la siguiente cuenta:
54 bytes (header) + 1280 * 720 * 3 = 2.764.854 bytes.
Esto nos da una diferencia de 49 bytes.
Lo que encontramos es que al final del archivo, se encuentra incrustado un texto plano que dice lo siguiente: “al .png cambiar extension por .zip y descomprimir”. Mensaje que tiene exactamente 49 bytes.



Archivo 2: **lado.bmp:**
Para este archivo, debemos usar el modo LSB4 para realizar la extracción de contenido, a través del siguiente comando:
`./stegobmp -extract -p ./grupo11/lado.bmp -out ./resultados/lado_out -steg LSB4`
Obtenemos la siguiente imagen:

Y de acuerdo a lo encontrado en buenosaires.bmp, cambiamos la extensión del archivo `lado_out.png` a `lado_out.zip`. Lo descomprimimos y obtenemos el archivo `sol11.txt` que contiene lo siguiente: "cada mina es un 1. cada fila forma una letra. Los ascii de las letras empiezan todos en 01. Asi encontraras el algoritmo que tiene clave de 256 bits y el modo La password esta en otro archivo Con algoritmo, modo y password hay un .wmv encriptado y oculto."
Con esta información, la imagen anterior queda de la siguiente manera:

Entonces, ya sabemos que debemos usar AES256 (porque el txt dice que tiene clave de 256 bits) y modo cfb. Nos queda encontrar la password.

Archivo 3: **lima.bmp:**
Luego, usando LSBI, extraemos la información de `lima.bmp`, con el siguiente comando: `./stegobmp -extract -p ./grupo11/lima.bmp -out ./resultados/lima_out -steg LSBI`.
Esto nos extrae un archivo pdf que contiene lo siguiente:

Archivo 4: **silence.bmp:**

Finalmente, ya contamos con todos los datos para realizar la desencripción y la extracción. Estos son AES256 cfb y password solucion. Además, sabemos que debemos usar LSB1 porque es el único modo que no usamos hasta ahora. Por lo tanto, extraemos el archivo estaganografiado con el siguiente comando: `./stegobmp -extract -p ./grupo11/silence.bmp -out ./resultados/silence_out -steg LSB1 -a aes256 -m cfb -pass solucion`
Esto nos da como salida un video en formato .wmv.

4. Algunos mensajes ocultos tenían, a su vez, otros mensajes ocultos. Indica cuál era ese mensaje y cómo se había ocultado.

El archivo obtenido de lado.bmp (lado_out.png) tiene adentro un zip. Al cambiar la extensión del mismo a lado_out.zip, de acuerdo a lo indicado en buenoaires.bmp, y luego extraerlo se obtiene un txt. El txt es sol11.txt y contiene el siguiente mensaje:
“cada mina es un 1.
cada fila forma una letra.
Los ascii de las letras empiezan todos en 01.
Asi encontraras el algoritmo que tiene clave de 256 bits y el modo
La password esta en otro archivo
Con algoritmo, modo y password hay un .wmv encriptado y oculto.”

5. Uno de los archivos ocultos era una porción de un video, donde se ve ejemplificado una manera de ocultar información ¿Qué se ocultaba según el video y sobre qué portador?

En esta porción de video se muestra una forma de ocultar información no tan común, donde se oculta un mensaje en un tejido. Lo que se hace es mirar los hilos del tejido. Es decir, si el hilo vertical pasa por encima de los otros, se lee el código como un 1 y si pasa por debajo, se lee un 0, formando así un código binario.

6. ¿De qué se trató el método de estenografiado que no era LSB1 ni LSB4 ni LSBI? ¿Es un método eficaz? ¿Por qué?

El método de esteganografía que no fue ninguno de los implementados para este trabajo, consiste en agregar al final del archivo portador un texto plano con

la información a ocultar. Podría decirse que no es un método muy eficaz ya que si leemos al final del archivo podemos encontrar el texto plano que se quería ocultar en un principio. Además, también se puede obtener fácilmente el tamaño de este ya que podemos ver que no concuerda el tamaño del archivo portador que indica en el header de este con el tamaño real que en realidad ocupa.

7. ¿Por qué la propuesta del documento de Akhtar, Khan y Johri es realmente una mejora respecto de LSB común?

En un principio, Akhtar, Khan y Johri[1] definen a una buena técnica de esteno-grafiado mediante tres aspectos principales:

- Capacidad: la cantidad de información que puede ser guardada dentro de la imagen cubierta
- Imperceptibilidad: la calidad visual de la estego-imagen luego de haberse escondido la información, y por último
- Robustez

En base a esto, podemos determinar que LSB común es buena en cuanto a imperceptibilidad, pero dado que solamente se utiliza un bit por pixel para cubrir la información la capacidad del método es baja. A su vez, tampoco es robusto dado que el mensaje puede ser recuperado de forma muy sencilla una vez que se ha identificado que la imágen tiene algún tipo de información secreta guardado dentro al recuperar el LSB.
Otra desventaja que cabe destacar, si bien no esta relacionada con los ítems anteriormente mencionados, es que para esteganografiar un byte de la imagen original, serán necesarios 8 bytes de la imágen cobertora, por lo tanto serán requeridos una mayor cantidad de recursos (por ejemplo, mayor ancho de banda) para envíar esta imágen.
Para intentar solucionar todos estos puntos, se han desarrollado variaciones de LSB común, como por ejemplo modificar dos o mas bits de la imágen cobertora con el objetivo de esconder mayor cantidad de información, llegando a usar hasta cuatro LSBs para esconder mensajes, lo cual producía un resultado aceptable. Sin embargo, el deterioro de la imágen resultaba muy notorio y el mismo se incrementaba acorde a la cantidad de LSBs utilizados [2].
Con el fin de mejorar significativamente de la calidad de la estego-imágen los autores proponen dos nuevos métodos o esquemas, los cuales serán brevemente explicados a continuación:

1. En el primer esquema, se deben obtener las posibles combinaciones del segundo y tercer LSB, es decir, las combinaciones (00, 01, 10, 11) y observar a partir de esto, si el LSB se debe cambiar o no. Esto se determina en función de los LSBs que han sido modificados en comparación con la imágen portadora original. Si el número de bits menos significativos que cambia es mayor al que no cambia, se determina que es necesario invertir esos bits para el patrón correspondiente del segundo y tercer LSB.
   Esto es una estrategia utilizada para reducir la cantidad de bits modificados. Finalmente, el objetivo es quedarse con una secuencia de bits que difiera lo menos posible del original, por lo que el Peak signal-to-noise ratio (PSNR) disminuiria, mejorando así la calidad de la estego-imágen.

Es importante destacar que para luego poder realizar la desteganografía de forma correcta, se debe guardar el patrón que fue utilizado para determinar cuáles LSB se invertirían.

2. En el segundo esquema se asume que la imágen portadora ya ha sido enviada al destinatario. Esto genera que se pueda mejorar aún mas la estegoimágen con la técnica de inversión de bits. En este caso, ademas de considerarse el segundo y tercer bit menos significativo y establecer un patrón a tener en consideración como en el primer esquema, también se tiene en cuenta el LSB. A partir de esto, se consideran cuatro casos distintos a sobre los cuales se contara la cantidad de ocurrencias:

   a) el número de bytes en los cuales el LSB cambió de 0 a 1,

   b) el número de bytes en los cuales el LSB era originalmente 0 y no ha cambiado,

   c) el número de bytes en los cuales el LSB cambió de 1 a 0,

   d) el número de bytes en los cuales el LSB era originalmente 1 y no ha cambiado.

   Llamaremos a estos A, B, C y D respectivamente. Si A es mayor a B, invertimos el LSB de aquellos bytes que tienen el patrón determinado previamente en el segundo y tercer bit menos significativo, así como también aquellos cuyo LSB era originalmente 0. De la misma forma si C es mayor a D, invertimos el LSB de aquellos bytes que tienen el patrón en el segundo y tercer bit menos significativo, así como también aquellos cuyo LSB era originalmente 1. De esta forma, se asegura que se están modificando la menor cantidad de bits posibles. Para un correcto desteganografiado, se necesita conservar esos patrones mediante los cuales el LSB fue modificado. Como se han considerado todos las posibles combinaciones con el segundo y tercer bit menos significativo y con el LSB mismo, sería necesario guardar hasta 8 combinaciones. A su vez, el segundo y tercer bit no han cambiado, pero no se puede decir lo mismo del LSB, el cual es necesario para analizar estos patrones. Por lo tanto es necesario poseer la imágen portadora original para revertir este proceso.

Finalmente, los esquemas de inversión de bits propuestos mejoran la calidad de la estegoimágen. Sin embargo, la mejora del PSNR no es proporcional, esta puede ser muy grande para algunas imágenes, mientras que en otros casos puede ser mas pequeña o nula. Para el primer esquema, dada la imagen a esconder, se puede tratar de minimizar el PSNR eligiendo una imágen portadora en la cual esta mejora sea mayor (es decir, seleccionar una imágen portadora en base al mensaje cuya distribución de bits minimice el PSNR).
Estos métodos son una mejor propuesta dado que, más allá de mejorar la calidad de la estegoimágen, dado que si bien un agente externo podría determinar si los bits del mensaje han sido manipulados de forma tal de esconder un mensaje mediante métodos de esteganografía, dificultan la recuperación del mensaje original dado que algunos de los LSB fueron invertidos. Esto añade una complejidad adicional al dificultar el proceso de desteganografiado, que ya no es tan lineal como con el LSB común. El método de inversión de bits mejora la esteganografía al mejorar su seguridad y calidad de imagen.

8. ¿De qué otra manera o en qué otro lugar podría guardarse el registro de los patrones invertidos?

Podría guardarse, al igual que se realiza en el trabajo de implementación, en una sección de la imágen, en nuestro caso, la información se encuentra en los primeros bytes, luego del header, guardándolo en el LSB.
Otra posible opción sería guardar el registro de los patrones invertidos en los bytes reservados del header que no tienen uso. Es una opción posible, pero queda atada a que en un futuro, esos bytes reservados del header no se utilicen.

9. Leer el Segundo esquema y analizar (sin implementar) cuáles serían las ventajas que pueden verse.

Como fue previamente mencionado en el ítem 7.2, el segundo esquema cuenta con cuatros casos en base a los cuales se decide si es necesario invertir los bits menos significativos o no. A partir de esto, es posible determinar como ventaja que al ser necesario contar con la imágen portadora para obtener el LSB al momento de desteganografiar, cualquier atacante que descubra de la presencia del mensaje oculto no podrá revertir el esteganografiado con facilidad. Esto ocurrirá dado que le faltarán los LSBs de la portadora original para realizar este proceso.
A su vez también, con este método se intenta modificar la menor cantidad de LSB de la imagen posibles. Persiguiendo este objetivo, la manipulación de la imágen portadora es menor, por lo que la imágen original (portadora) es alterada en menor medida que en otros esquemas.

10. Leer el Segundo Esquema e indicar qué desventajas o inconvenientes podría tener su implementación.

Una posible desventaja de esta implementacion es, nuevamente, la necesidad de contar con la imágen portadora original para poder realizar el proceso de desteganografiado satisfactoriamente. Esto producira que sea mandatorio trabajar con dos archivos en simultaneo, leyendo de uno y escribiendo en el otro para esteganografiarlo y luego, leyendo de ambos para desteganografiarlo. Esto podría resultar tedioso para la implementación.
A su vez, se puede observar que este esquema presenta mas condicionales y casos, por lo que es más complejo que los demás analizados. Esto podría ser un inconveniente a la hora de implementarlo, dado que al aumentar la complejidad de un sistema crece paralelamente la probabilidad de que el mismo posea errores o bugs.

11. ¿Qué dificultades encontraron en la implementación del algoritmo del paper?

Las dificultades encontradas a la hora de implementar el algoritmo del paper fueron:

- Interpretar el correcto orden para almacenar los flags de inversión de cada patrón.

12. ¿Qué mejoras o futuras extensiones harías al programa stegobmp?

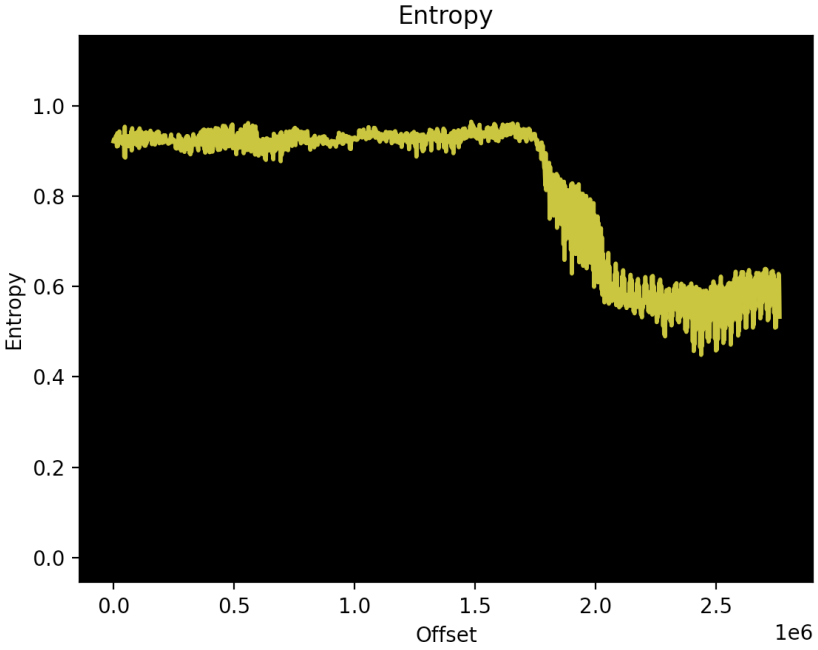
Las mejoras o futuras extensiones a agregar podrían ser:

- Soportar distintos tipos de imágenes portadoras, como png o jpg.
- Interfaz gráfica que permita usar el programa como una aplicación desktop sin necesidad de usar la terminal.
- Actualmente cuando se leen los archivos, se levantan completamente en memoria. Esto no supone un problema porque se trabaja con archivos en el orden de los MB. Pero, en el caso de necesitar manejar archivos en el orden de los GB, sería mejor mantenerlos abiertos e ir leyéndolos de a bloques.
- A la hora de alocar memoria para la encripción/desencripción se usa una constante de 64MB. Se podría calcular el tamaño final de encripción en base al tamaño a encriptar.
- Otra mejora sería realizar el proceso de embed o extract lanzando varios hilos que se encarguen de distintas partes de la imagen a ser embebida o extraída y de esta manera paralelizar el trabajo mejorando la performance (de nuevo, esto sería notorio en el caso de trabajar con imágenes en el orden de los GB). Se pueden analizar distintas estrategias a la hora de crear los threads, por ejemplo, crear una catidad fija y asignales sectores de tamaño variable o crear una cantidad variable de hilos que trabajen una cantidad fija de bytes de la imagen.

Anexos

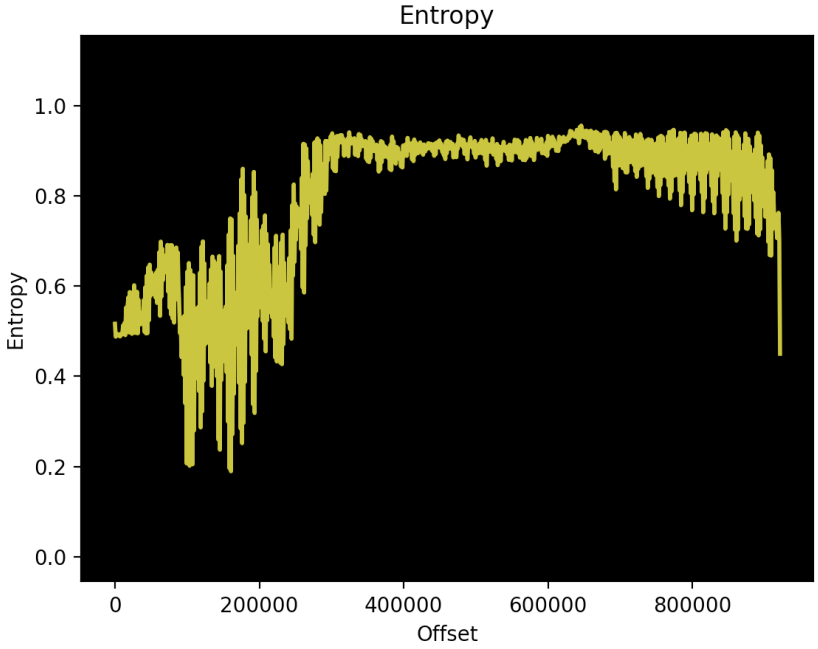
A. Análisis de Entropía

A.1. Entropía de buenosaires.bmp



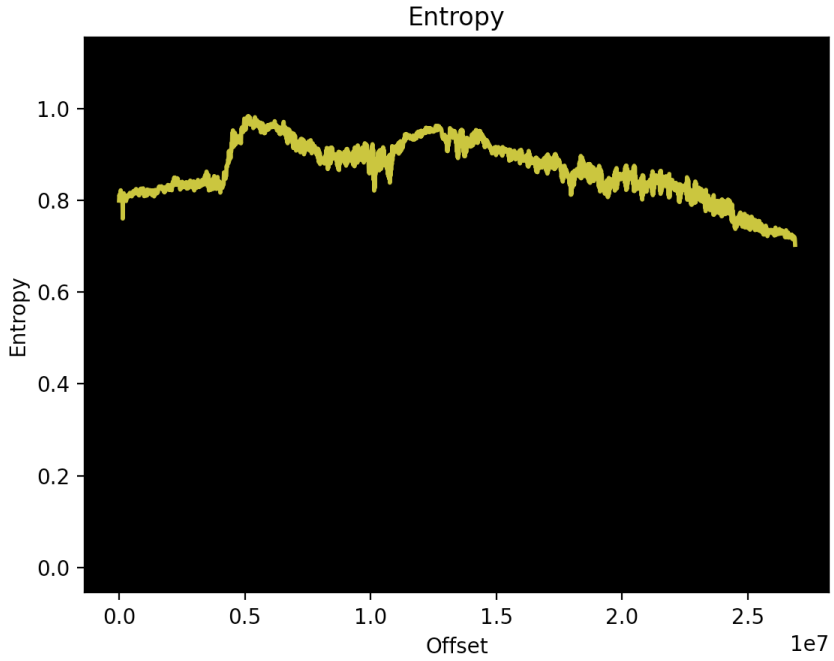
En este grafico se puede observar que la entropía mantiene una linea casi recta gran parte del archivo hasta llegar al final, donde se produce una caída brusca. Este salto indica que al final del archivo hay algo que produce esa variación.

A.2. Entropía de lado.bmp



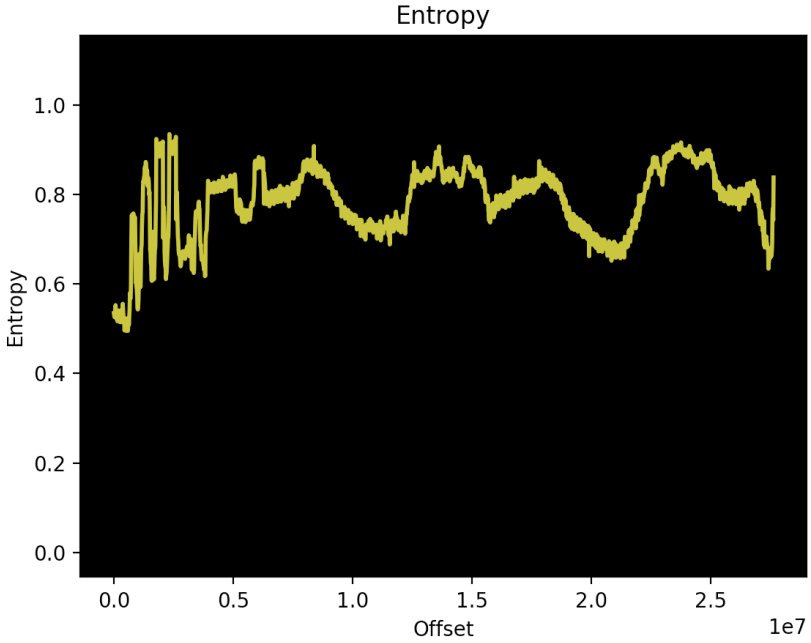
Para el caso de lado.bmp se observn variaciones al principio y al final. Las variaciones tan alta al principio pueden indicar que se trata de un esteganografiado con LSB4 ya que este altera 4 bits por cada byte.

A.3. Entropía de lima.bmp



Luego, con lima.bmp, la entropía se mantiene sin grandes variaciones, lo que puede indicar que se trata de un esteganografiado con LSB1 o LSBI.

A.4. Entropía de silence.bmp



Por último, en silence.bmp la entropía tiene bastantes variaciones, a pesar de que es la imágen que contiene el video encriptado.

Referencias

[1] Akhtar, Nadeem, Shahbaaz Khan, and Pragati Johri "An improved inverted LSB image steganography," *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pp. 1, IEEE, 2014.

[2] Akhtar, Nadeem, Shahbaaz Khan, and Pragati Johri "An improved inverted LSB image steganography," *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pp. 2, IEEE, 2014.