72.44 - Criptografía y Seguridad

Esteganografía

25 de junio de 2024

Alumnos:

Iván Chayer, 61360
Thomas Mizrahi, 60154
Camila Di Toro, 62576
Abril Occhipinti, 61159

Profesores:

Pablo Eduardo Abad
Ana Maria Arias Roig
Rodrigo Ramele

ITBA
INSTITUTO TECNOLOGICO DE BUENOS AIRES
•UNIVERSIDAD PRIVADA•

Índice

Sobre el paper

Organización formal del documento

El documento está organizado de la siguiente forma:

Introducción

- **Contexto y justificación del paper**: Presenta la importancia de la esteganografía, resaltando su relevancia en el campo de la seguridad de la información.
- **Propuesta**: Introduce brevemente la propuesta de mejora en la técnica de esteganografía basada en el bit menos significativo.

Fundamentos Teóricos

- **Técnica estándar del LSB**: Describe el funcionamiento de la técnica LSB estándar y discute sus limitaciones en términos de seguridad y calidad de la imagen.
- **Modelo de color RGB en imágenes de 24 bits**: Explica las propiedades del modelo de color RGB y cómo se representan las imágenes de 24 bits.

Metodología

- **Técnica de inversión de bits**: Presenta la idea para mejorar la técnica LSB estándar mediante la inversión de bits, explicando los principios y razones detrás de esta mejora.
- **Algoritmo de inversión de bits**: Describe paso a paso el algoritmo para implementar la técnica de inversión de bits, incluyendo su pseudocódigo.

Experimentos y Resultados

- **Simulaciones**: Describe cómo se llevaron a cabo los experimentos, incluyendo la configuración del entorno, las imágenes utilizadas, y los criterios de evaluación.
- **Análisis de resultados**: Presenta los resultados de los experimentos, comparando la eficacia de la técnica propuesta con la técnica LSB estándar.

Discusión

- **Ventajas de la técnica propuesta**: Resalta las mejoras logradas en términos de seguridad y calidad de la imagen.
- **Limitaciones**: Identifica limitaciones de la técnica propuesta y posibles desafíos para su implementación.

Conclusión

- **Resumen de resultados**: Resume los principales resultados y contribuciones del estudio y sugiere posibles mejoras adicionales que se podrían explorar.

Descripción del algoritmo

Como se menciona en la sección anterior, el paper propone una mejora a la esteganografía LSB estándar mediante la técnica de inversión de bits. El proceso se puede resumir en los siguientes pasos:

1. **Calcular patrones de ocurrencia**: Se calculan las ocurrencias de los patrones de los dos últimos bits (2º y 3º menos significativos) en la *cover image*.
2. **Aplicar LSB estándar**: Se aplica el método estándar de LSB1 para obtener la imagen esteganografiada (stego-image).
3. **Comparar patrones**: Se comparan los patrones de la *cover image* con los de la imagen esteganografiada.
4. **Invertir bits**: Se invierten los bits LSB si el número de píxeles modificados es mayor que los no modificados.
5. **Almacenar patrones**: Se almacena la información sobre los patrones de bits que fueron invertidos para poder recuperar el mensaje secreto posteriormente.

Sobre las contradicciones y claridad del documento

En primer lugar, encontramos que el documento tiene algunos errores gramaticales y de sintaxis. Por otra parte, el paper **no especifica cómo almacenar la información sobre los patrones de bits que fueron invertidos**. Describe que se necesita almacenar esta información en una "ubicación específica" dentro de la imagen esteganografiada, pero no proporciona detalles sobre cómo se realiza este almacenamiento ni cómo se codifican los patrones para su recuperación posterior. La falta de claridad en esté aspecto nos trajo dificultades que mencionamos en otra sección de este informe. Por último, en la sección 5, donde se explica la técnica LSBI, en el paper se menciona lo siguiente:

> *For pattern ‘01’ (C), we cannot check how many pixels were changed and how many were not, because there is only one pixel, comparison cannot occur.*

Esto no tiene mucho sentido. Si solamente hay un pixel, basta con indicar que la totalidad de los pixeles fueron invertidos o se mantuvieron iguales.

Esteganografía

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

Vamos a esteganografiar una imagen del famoso personaje Mario Bros usando como carrier el escudo de River Plate. Todas las pruebas y los archivos utilizados se encuentran en este link.

LSBI

```
-embed -in "...\stegobmp\src\bmp\examples\mario.png"
-p "...\stegobmp\src\bmp\examples\river.bmp"
-out "riverLSBI.bmp" -steg LSBI

-extract -p "...\stegobmp\src\bmp\examples\riverLSBI.bmp"
-out "mario" -steg LSBI
```

LSB1

```
-embed -in "...\stegobmp\src\bmp\examples\mario.png"
-p "...\stegobmp\src\bmp\examples\river.bmp"
-out "riverLSBI.bmp" -steg LSB1

-extract -p "...\stegobmp\src\bmp\examples\riverLSB1.bmp"
-out "mario" -steg LSB1
```

LSB4

```
-embed -in "...\stegobmp\src\bmp\examples\mario.png"
-p "...\stegobmp\src\bmp\examples\river.bmp"
-out "riverLSBI.bmp" -steg LSB4

-extract -p "...\stegobmp\src\bmp\examples\riverLSB4.bmp"
-out "mario" -steg LSB4
```

	LSB1	LSB4	LSBI
Complejidad	Baja, fácil de implementar.	Moderada, más compleja que LSB1 pero aún relativamente simple.	Alta, requiere comparaciones y cálculos adicionales.
Capacidad de Ocultación	Un bit por píxel.	Cuatro bits por píxel	⅔ bit por pixel
Calidad de la Imagen Estego	Buena, mínimamente alterada.	Menor calidad comparada con LSB1 debido a la mayor cantidad de bits modificados.	Mejor calidad que LSB1 y LSB4 debido a la optimización del PSNR (Peak Signal-to-Noise Ratio) y al no utilizar los bytes del color rojo

Tabla 1: *Comparación de los métodos de esteganografía*

LSB1 sustituye el bit menos significativo de cada píxel con bits del mensaje secreto, siendo muy fácil de implementar y revertir, pero también fácilmente detectable y con baja capacidad de ocultación. LSB4, por otro lado, reemplaza los cuatro bits menos significativos de cada píxel, permitiendo ocultar más información pero a costa de una mayor alteración de la imagen.

LSBI, el método más avanzado, invierte el último bit de cada píxel basándose en comparaciones entre los bits de la imagen que fueron alterados, además, no se oculta información en. Esto permite obtener una imagen de mejor calidad a costa de mayor procesamiento y menor capacidad de ocultar información.

Archivos de la cátedra

Nota: todo lo extraído se encuentra en el repositorio.

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.

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

1. El primer paso fue probar obtener la información oculta probando con los 3 métodos implementados, sin utilizar encriptación.

a. Dentro del archivo roma, encontramos con LSB4 un png con la imagen de un buscaminas

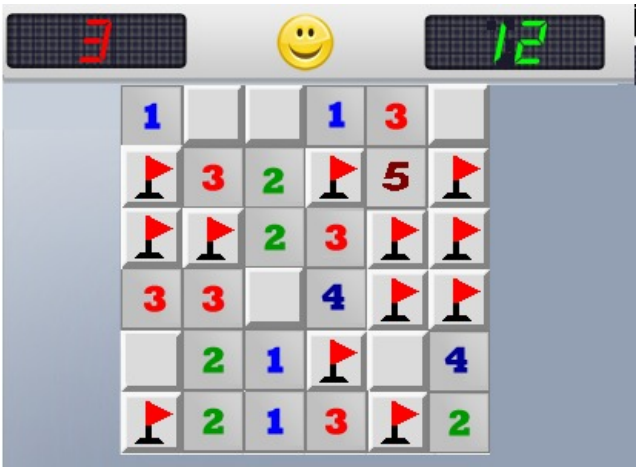

Figura 1: *Imagen del buscaminas extraído*

b. Dentro del archivo back, encontramos con LSBI un pdf con las siguientes instrucciones: *al .png cambiarle la extensión por .zip y descomprimir*
c. No pudimos extraer información significativa de los archivos sherlock.bmp y kings.bmp en este primer análisis.

2. Pasamos a png el zip y encontramos las siguientes instrucciones dentro del zip:
   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 192 bits y el modo
   La password esta en otro archivo
   Con algoritmo, modo y password hay un .wmv encriptado y oculto.
3. Luego de completar el buscaminas, obtuvimos el ASCII de las letras que nos daban el algoritmo de encriptación:
   01 000001 A
   01 100101 e
   01 110011 s
   01 000011 C
   01 100110 f
   01 100010 b
   Por lo que obtuvimos que el algoritmo es Aes de 192 bits Ofb.
4. Utilizando la herramienta https://hexed.it/, encontramos en el archivo kings.bmp la pass *desafio* en plaintext, directamente concatenada al final del archivo.
5. Finalmente pudimos extraer el archivo .wmv de sherlock.bmp utilizando LSB1, Aes de 192 bits Ofb y la pass *desafio*.

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

En el extracto de la película se muestra cómo se utilizaban los tejidos como portadores para ocultar mensajes, codificando ceros y unos. Si el hilo vertical del tejido pasa por encima del horizontal, representa un uno, si pasa por debajo, es un cero.

Al final del video se menciona que se oculta *un nombre, un objetivo*

Preguntas adicionales

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

El método consistió en agregar directamente texto plano al final del archivo kings.bmp. Aunque agregar texto plano al final de un archivo es un método de esteganografía muy simple y preserva la integridad del contenido original, es altamente ineficaz en términos de seguridad. La información añadida es fácilmente detectable y el incremento en el tamaño del archivo puede levantar sospechas rápidamente.

¿Por qué la propuesta del documento de Majeed y Sulaiman es realmente una mejora respecto de LSB común?

La propuesta de Majeed y Sulaiman mejora la técnica LSB común en varios aspectos: **aumenta la seguridad** al invertir bits en ciertos patrones específicos, lo que añade una capa extra de protección y reduce la previsibilidad de los cambios, complicando el análisis de patrones. Además, **mejora la calidad de la imagen** al distribuir los cambios de manera más uniforme gracias a un análisis previo de patrones y ajustando solo cuando es necesario, minimizando así la distorsión visual y preservando la integridad de la imagen portadora. La técnica también es más **resistente al estegoanálisis**, ya que es menos susceptible a la detección mediante análisis de patrones repetitivos o anomalías en la distribución de bits, y la inversión de bits junto con el almacenamiento de patrones específicos añade complejidad, dificultando la reversión del proceso.

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

A la hora de implementar el algoritmo que propone el paper encontramos las siguientes dificultades: la primera fué **cómo almacenar la información sobre los patrones de bits que fueron invertidos** ya que el documento no lo especifica. Menciona que se necesita almacenar esta información en una "ubicación específica" dentro de la imagen esteganografiada, pero no proporciona detalles sobre cómo se realiza este almacenamiento ni cómo se codifican los patrones para su recuperación posterior.
Por otro lado, implementar una técnica que modifique los bits de manera precisa fue una cuestión que tuvimos que analizar con detenimiento ya que cualquier error en la inversión de bits o en el almacenamiento de patrones nos llevaba a la pérdida o corrupción de datos.

En la implementación se optó por guardar los patrones invertidos antes del mensaje ¿de qué otra manera o en qué otro lugar podría guardarse el registro de los patrones invertidos?

Otras maneras de guardar el registro de los patrones invertidos podrían ser:

- **En un prefijo del mensaje oculto**: En lugar de almacenarlo en los primeros bytes de la imagen, los patrones invertidos podrían ser guardados en los primeros bytes del mensaje oculto en sí. Esto haría que el proceso de extracción primero recupere estos patrones antes de proceder con el mensaje.

- **Intercalados en el mensaje**: Los patrones invertidos podrían ser intercalados periódicamente dentro del mensaje oculto. Por ejemplo, después de cada bloque de datos del mensaje, se podría insertar un bit indicando si el bloque siguiente ha sido invertido.

- **Encabezado de la imagen**: El encabezado del archivo BMP podría modificarse para incluir los patrones invertidos. Este método tendría la ventaja de mantener los datos esteganográficos juntos y posiblemente fuera de la vista de análisis más básicos.

¿Qué mejoras o futuras extensiones harías al programa stegobmp?

Para mejorar y extender el programa stegobmp, se podrían implementar varias acciones. Primero, ampliar la compatibilidad para manejar diferentes formatos y resoluciones de imagen (mayores a 24 bits) y adaptar el algoritmo para que funcione con sistemas existentes de transmisión y almacenamiento de imágenes. Además, sería útil desarrollar un método estándar para almacenar y recuperar la información de los patrones de bits invertidos. Por último, mejorar la interfaz de usuario facilitaría el uso de la herramienta para personas con distintos niveles de experiencia técnica.