Tecnológico de Monterrey

Esteganografía de audio

Seguridad Informática - Grupo 1
Profesor: Jesús Arturo Pérez Díaz
Lunes 7 de mayo del 2022
Equipo:

Adriana Paola Salinas García - A01703675
Diana Arteaga Díaz - A01703727
Fernando Amézquita - A01421954

Tabla de contenidos

Introducción

¿Qué es la esteganografía?

La esteganografía es una rama de la criptología que estudia el ocultamiento de mensajes en un recipiente con el fin de esconder información a plena vista sin que otras personas se den cuenta. A diferencia de la criptografía, que cifra el mensaje con el fin de que no se pueda leer sin la clave, el objetivo de la esteganografía es ocultar la existencia del mensaje de miradas indiscretas.

La historia de la esteganografía se remonta a Heródoto en 484 a.c, donde en su libro *Las Historias,* cuenta cómo un personaje escondió un mensaje en un tablón recubierto con cera, y otro lo tatuó en la calva de su esclavo para que le creciera el pelo y lo mandaran con instrucciones de rasurarse. Durante las guerras mundiales se usaban técnicas como tinta invisible, micropunto y código navajo para esconder órdenes y comunicaciones militares. Con la llegada de la esteganografía moderna en los años 80, los primeros intentos se basaron en marcas de agua, enmascaramiento y codificación LSB (Least Bit Significant). En 2001, se estudió que los terroristas del 11 de septiembre pudieron haberse coordinado con mensajes ocultos en la red, mientras que en 2010 el FBI investigó a diez espías rusos de transmitir material clasificado del gobierno de Estados Unidos con esteganografía.

Actualmente, el medio más común de esteganografía son archivos multimedia (imágenes, audios, videos) debido a su tamaño y capacidad, y se ha vuelto muy popular entre los cibercriminales para realizar campañas de spyware o malware. Incluso se pueden cifrar los mensajes previos al proceso.

El estegoanálisis es la disciplina que estudia la detección de mensajes ocultos con esteganografía. Se puede realizar de manera manual o estadística, buscando cambios en la estructura o distribución de colores del archivo a analizar. Sin embargo, resulta difícil de detectar debido a la gran cantidad de métodos de integración y limitaciones del ojo humano. En el mejor de los casos llega a mostrar la probabilidad de la existencia del mensaje.

Afortunadamente, la información ocultada con esteganografía no tiene la capacidad de robar datos de una computadora. Aun así, se recomienda tener mucho cuidado a la hora de descargar archivos, usar sólo fuentes confiables, evitar cualquier material sospechoso e instalar medidas de seguridad para evitar componentes maliciosos.

.

Técnicas de esteganografía

1) Esteganografía de imágenes

Una imagen es una matriz numérica que representa una rejilla rectangular de píxeles, los cuales se componen de tres bytes que definen su color RGB, permitiendo generar hasta 16,777,216 colores. Mientras la imagen sea de mayor calidad y resolución, más fácil será ocultar y revelar un mensaje. Durante el proceso, se evita cambiar el formato de la imagen para no alterar o dañar la información insertada. Existen varios métodos para realizar esteganografía en imágenes:

- ★ LSB (Least Significant Bit): Reemplaza el bit menos significativo de cada píxel con el del mensaje a esconder. Es el más popular de implementar, pero es inseguro puesto que inserta ruido blanco (falta de correlación entre píxeles) y es fácil de detectar.
- ★ RGB Based: Un canal de indicación elige un píxel aleatorio de la imagen para insertar los bits del mensaje a esconder en función de los valores que dependen de esta. Similar a LSB, tiene la misma seguridad, pero ofrece más capacidad de almacenamiento.
- ★ PVD (Diferenciación de Valores de Píxeles): Sustituye los valores de la diferencia de los bloques de dos pixeles con el mensaje a esconder. Una de sus ventajas es que aprovecha la sensibilidad de la vista humana para los tonos de grises.
- ★ PMM (Método de Mapeo de Píxeles): Se define un píxel semilla y se eligen los píxeles de incrustación dependiendo de la intensidad de este. Se revisa si los elegidos o sus vecinos se encuentran dentro de los límites de la imagen, y se incrusta el mensaje a esconder mediante un mapeo de cada dos o cuatro bits del mensaje en cada píxel vecino.

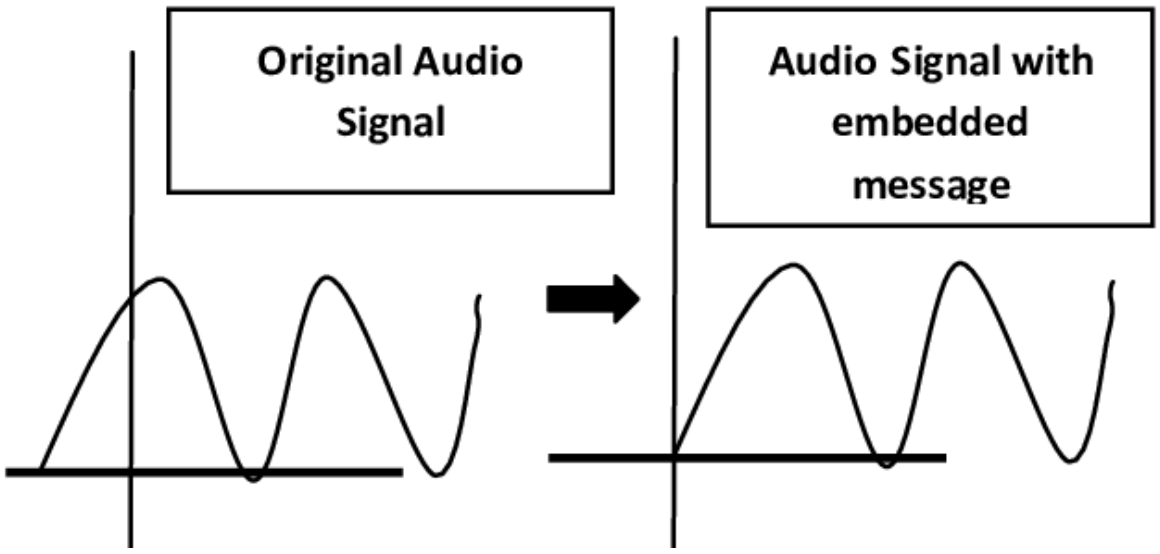
2) Esteganografía de audio:

El audio sucede cuando un conjunto de moléculas vibran por una onda de sonido que viaja por un medio natural (aire, agua, tierra) y llegan a nuestro oído. Con una conversión análoga a digital se puede graficar cada momento de la onda de sonido (samples), y cada uno de estos tiene un canal, ritmo y tamaño. Agrupar samples de varios canales generan un frame, el cual es igual al número de canales por el tamaño del sample en bytes, y la cantidad en un segundo depende del ritmo de sampleo. Por ejemplo, un frame con 6 canales de 4 bytes de longitud tendrá 24 bytes en total.
Existen varios métodos para realizar esteganografía en audio:

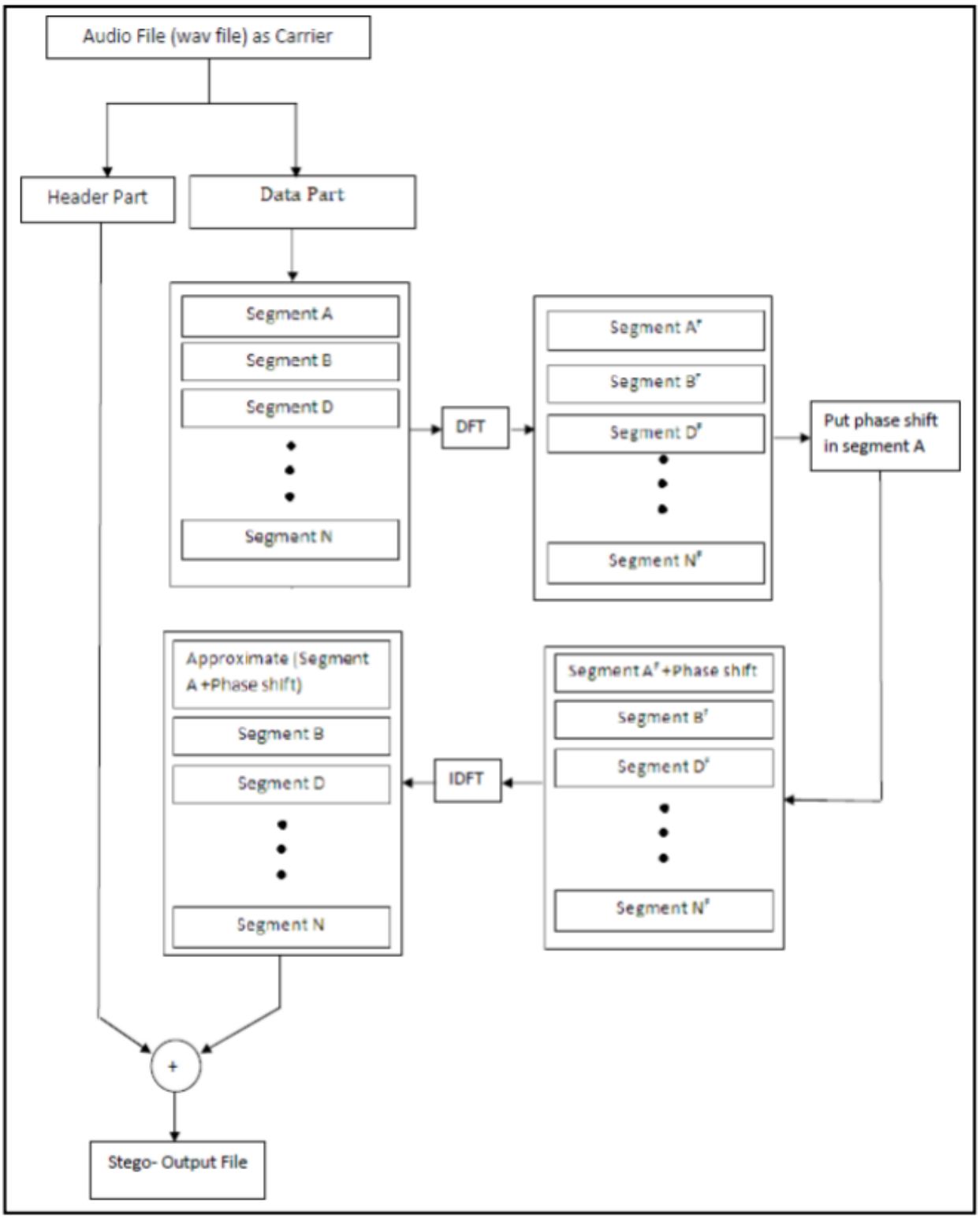
- ★ LSB (Least Significant Bit): Reemplaza el bit menos significativo de cada frame de audio con el del mensaje a esconder. Si bien es el más fácil de implementar, su inserción de ruido blanco en forma de pitidos lo hace detectable para el oído humano.
- ★ Codificación de paridad: Rompe la señal en regiones e incrusta cada bit del mensaje en un bit de paridad. Si no coincide con el bit secreto para ser codificado, el proceso invierte el LSB de una de las muestras en la región, permitiendo más de una opción para codificar la información secreta.
- ★ Codificación de fase: Sustituye la fase de un segmento del audio con una de referencia del mensaje a esconder. Los restantes se ajustan para preservar la fase relativa entre ellos.
- ★ Amplio espectro: Difunde el mensaje a través del espectro de frecuencia de la señal usando un código independiente. Así, la señal ocupa un ancho de banda mayor al requerido para la transmisión.
- ★ Eco oculta: Divide la señal en bloques, incrusta un bit del mensaje en cada uno si produce un eco en ella y concatena al final. Proporciona una velocidad alta de transmisión de datos y robustez superior en comparación con otros métodos.

Objetivo del proyecto

El objetivo del proyecto es desarrollar un programa en Python que realice esteganografía mediante codificación de fase sobre un archivo de audio .WAV. El usuario podrá elegir el archivo sobre el cual insertar o recuperar un mensaje, ya sea un audio o canción.



Desarrollo


Audio File (wav file) as Carrier
Header Part
Data Part
Segment A
Segment B
Segment D
Segment N
DFT
Segment A'
Segment B'
Segment D'
Segment N'
Put phase shift in segment A
Approximate (Segment A +Phase shift)
Segment B
Segment D
Segment N
IDFT
Segment A'+Phase shift
Segment B'
Segment D'
Segment N'
+
Stego- Output File

La esteganografía por codificación de fase consiste en sustituir la fase de un segmento o chunk de audio con una con un bit del mensaje a esconder y ajustar las demás para preservar la fase relativa entre ellos. Esto se logra aplicando una Transformación Discreta de Fourier (DFT) a los segmentos para crear una matriz de fases e insertando el mensaje con la siguiente fórmula:

$$\text{New Phase} = \begin{cases} \text{Old Phase} + \pi/2 & \text{if message bit} = 0 \\ \text{Old Phase} - \pi/2 & \text{if message bit} = 1 \end{cases}$$

La nueva matriz de fases se crea con la nueva fase y la matriz original. Una vez insertado el bit del mensaje, se reconstruyen los segmentos de audio con la Transformación Discreta de Fourier (DFT) inversa y se concatenan con el header del audio.

Ocultamiento

1) Lee el archivo de audio a usar

Se extrae el audio a manera de array junto con su frecuencia en Hz

2) Lee el mensaje a esconder
3) Aplica el formato necesario al mensaje para el ocultamiento

El mensaje a esconder tiene una longitud de 2000 caracteres, el cual consiste del mensaje insertado por el usuario y 2000 - (Longitud del mensaje) ondulaciones (~) para completarlo. Posteriormente, se calcula el tamaño del mensaje en bits.

4) Calcula los chunks del audio

Puesto que el audio se va a dividir en chunks para ocultar los bits del mensaje, debemos calcular el tamaño de los chunks, así como la cantidad a generar.

5) Copia el audio para generar el archivo con el mensaje oculto
6) Transforma el audio para dividirlo en chunks

Se cambia el tamaño del array de audio para acomodar los chunks (Tamaño x Cantidad)

   a) Si el audio es Mono, entonces se añade un segundo eje para que tenga dos canales
   b) Si el audio es Stereo, entonces solo se le aplica la transpuesta a la matriz del audio

7) Descompone el audio en la cantidad de chunks con el tamaño especificado

8) Aplica la Transformación Discreta de Fourier (DFT) a los chunks

Se calcula la magnitud, fase y diferencia de fase entre los chunks

9) Convierte el mensaje a binario
10) Inserta los bits del mensaje en la fase de los chunks

Primero se checa si el bit a insertar es 0 o 1, esto determina si para calcular la nueva fase del chunk se va a sumar o restar pi/2. Luego se procede a calcular la nueva fase en el punto medio del chunk. Una vez aplicada, esta se normaliza en los valores aledaños al punto medio del chunk mediante la suma diferencia de fase. Al final se vuelven a crear los chunks con la nueva fase transformada.

11) Aplica la Transformación Discreta de Fourier (DFT) a los chunks

Esto devuelve el array de audio a sus valores originales y con las nuevas fases

12) Junta los chunks transformados en un solo array de audio
13) Escribe el nuevo archivo de audio con el mensaje oculto

Recuperación

1) Lee el archivo de audio a usar

Se extrae el audio a manera de array junto con su frecuencia en Hz

2) Define el tamaño en bits del mensaje a recuperar

Como el mensaje a recuperar tiene una longitud de 2000 caracteres, tiene un tamaño de 16000 bits

3) Calcula el punto medio de los chunks de audio

Primero se calcula el tamaño de los chunks, luego se saca el punto medio

4) Extrae el header del audio
    a) Si el audio es Mono, se extrae directamente
    b) Si el audio es Stereo, se extrae bajo una dimensión
5) Extrae el bit del mensaje de la fase de los chunks

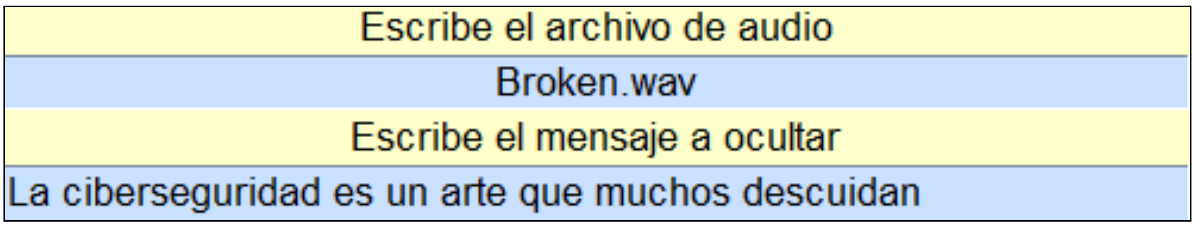
Primero se aplica la Transformación Discreta de Fourier (DFT) a los chunks de audio. Después se checa el valor de la fase para determinar si se sumó o restó pi/2 a la misma. Esto determina si el bit insertado fue un 0 o 1

6) Recupera el mensaje

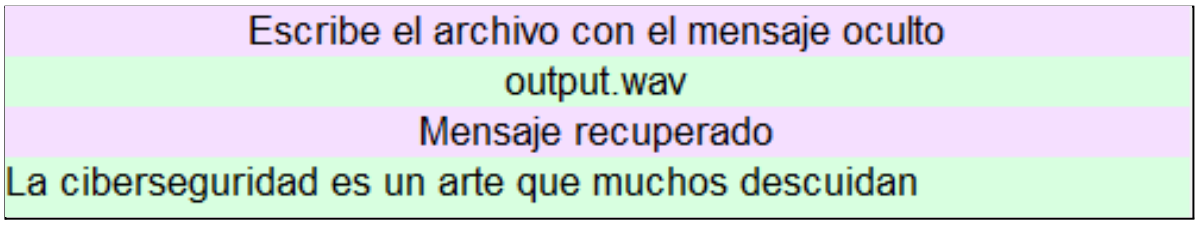
Primero se convierte el mensaje a binario, luego a entero y finalmente a caracteres. Se eliminan las ondulaciones (~) de la parte final del mensaje para imprimirlo en la interfaz gráfica.

Interfaz gráfica

1) Ocultamiento
    - ➔ Archivo de audio en el que se ocultará el mensaje
    - ➔ Mensaje a ocultar en el audio



2) Recuperación
    - ➔ Archivo de audio con el mensaje oculto
    - ➔ Mensaje recuperado del archivo



Conclusión

Con lo que trabajamos en el proyecto, apreciamos una nueva perspectiva sobre la esteganografía. Aprendimos que la criptografía no es el único método de esconder información de manera segura. La esteganografía tiene una gran cantidad de medios para aplicarse y combinarse con varias áreas de la ciberseguridad para crear nuevos algoritmos.

Este proyecto igualmente nos hizo ser más conscientes sobre las vulnerabilidades a la que estamos expuestos al navegar en línea. Debemos prestar más atención a los riesgos que existen en Internet para proteger nuestra información personal. La esteganografía tiene un gran alcance, pero debemos desarrollarla junto con las facetas inexploradas del audio y los algoritmos ya existentes para crear nuevos métodos de detección de mensajes en el futuro. La codificación por fase es una buena señal de que vamos en la dirección correcta, pero aún se puede llegar a más.

Referencias

1) Kaspersky, 2019, “¿Qué es la esteganografía digital?” (22/05/2022), Recuperado de: https://latam.kaspersky.com/blog/digital-steganography/14859/
2) Xataka, 2016, “Cuando una imagen oculta más información de lo que parece: qué es y cómo funciona la esteganografía” (22/05/2022), Recuperado de: https://www.xataka.com/historia-tecnologica/cuando-una-imagen-oculta-mas-informacion-de-lo-que-parece-que-es-y-como-funciona-la-esteganografia
3) Moreira et al. 2017, “Análisis de técnicas de esteganografía aplicadas en archivos de audio e imagen” (22/05/2022), Recuperado de: https://polodelconocimiento.com/ojs/index.php/es/article/download/10/pdf
4) IICybersecurity, 2018, “¿Cómo ocultar mensajes secretos en archivos de música? (22/05/2022), Recuperado de: https://www.iicybersecurity.com/audio-esteganografia.html
5) MDN, 2020, “Digital audio concepts” (22/05/2022), Recuperado de: https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Audio_concepts
6) Kumar, S. et al. (2012), “LSB Modification and Phase Encoding Technique of Audio Steganography Revisited” (22/05/2022), Recuperado de: https://ijarcce.com/wp-content/uploads/2012/06/11_LSB-Modification-and-Phase-Encoding-Technique-of-Audio-Steganography-Revisited.pdf
7) Katta, A. (2021), “Audio Steganography using Phase Encoding” (22/05/2022), Recuperado de: https://medium.com/@achyuta.katta/audio-steganography-using-phase-encoding-d13f100380f2