Tecnológico de Monterrey

w

Esteganografía de audio

Raymundo Romero Arenas - A00570654
Fátima Delenne Zapata - A01635041
Juan Arturo Cruz Cardona - A01701804
Seguridad Informática - Grupo 1
Profesor: Jesús Arturo Pérez Díaz
Lunes 5 de abril del 2021
Equipo = RSA

Tabla de contenidos

Introducción

¿Qué es la esteganografía?

La esteganografía es una rama de la criptología que estudia el ocultamiento de mensajes dentro de 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, dejó que le creciera el pelo y lo mandó al receptor con instrucciones de que le rasuraran la cabeza. Durante las dos guerras mundiales, se usaban técnicas como tinta invisible, micropunto, código navajo y cifrado nulo para esconder instrucciones y comunicaciones militares. Con la llegada de la esteganografía moderna en 1985, los primeros intentos se basaron en marcas de agua en imágenes, enmascaramiento y codificación LSB (Least Bit Significant). Existen casos recientes sobre su uso para espionaje y terrorismo. Por ejemplo, en 2001 se estudió la posibilidad de que los terroristas del 11 de septiembre se coordinaran con mensajes ocultos en la red, mientras que en 2010 el FBI reveló una investigación en la que implicaba a diez espías rusos de transmitir material clasificado del gobierno de Estados Unidos mediante esteganografía digital.

Actualmente, el medio más común son los archivos multimedia (imágenes, audios, videos, etc…) debido a su tamaño y capacidad. Debido a esto, se ha vuelto muy popular entre los cibercriminales para realizar campañas de spyware o insertar código malicioso en memes. Incluso se pueden usar algoritmos de criptografía para 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 mediante esteganografía no tiene la capacidad de robar datos de una computadora. Aun así, se recomienda ser consciente a la hora de descargar archivos en línea para proteger tus dispositivos. Usa sólo fuentes confiables, evita cualquier tipo de material sospechoso e instala medidas de seguridad para evitar componentes maliciosos

.

Técnicas esteganográficas

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 (siguiendo el modelo RGB), dando la posibilidad de generar hasta 16,777,216 colores. Incluso se puede agregar un cuarto bit más para la transparencia. Mientras la imagen tenga mayor calidad y resolución, más fácil y eficiente de ocultar y revelar un mensaje será. Durante el proceso, se debe evitar 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. Si bien es el más popular de implementar, es muy poco seguro debido a su inserción de ruido blanco (falta de correlación entre píxeles) y facilidad 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 pixeles de incrustación dependiendo de la intensidad del valor de este. Se revisa si los seleccionados o sus vecinos se encuentran dentro de los límites de la imagen o no, 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:

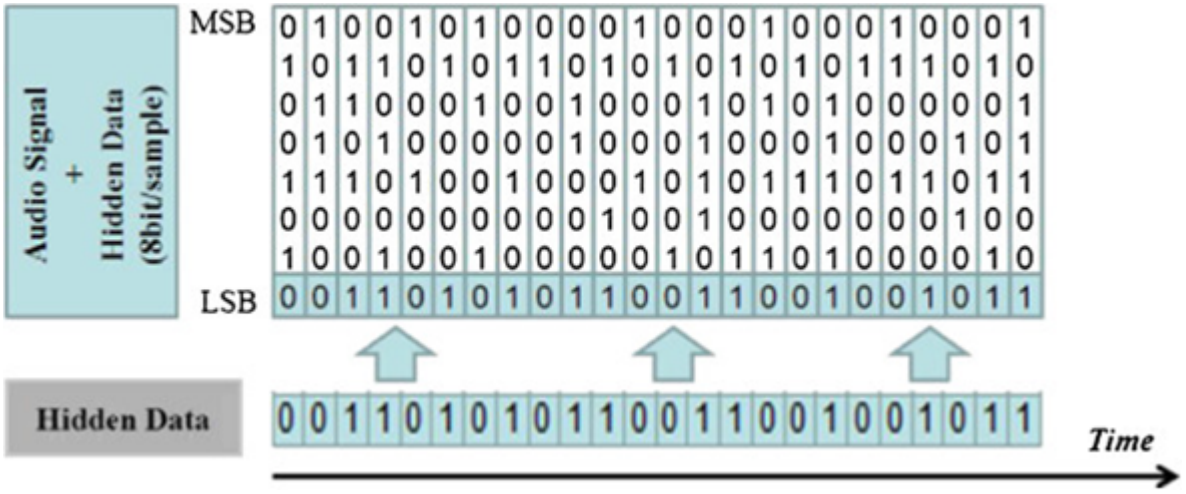
El audio sucede cuando un conjunto de moléculas vibran por una onda de sonido que viaja por un medio natural (aire, agua, tierra, etc…) y llegan a nuestro oído. Mediante una conversión análoga a digital, se puede graficar cada momento de la onda de sonido (llamado sample). Cada uno de estos tiene un canal, ritmo y tamaño. Cuando los samples de varios canales se agrupan en una instancia, se genera un frame, el cual se calcula multiplicando el número de canales por el tamaño del sample en bytes, y cuya cantidad en un segundo depende del ritmo de sampleo. Así, un frame en modo 5.1 (con 6 canales) de 4 bytes de longitud contendrá 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 uno no coincide con el bit secreto para ser codificado, el proceso invierte el LSB de una de las muestras en la región, dándole al remitente tiene más de una opción en la codificación de 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 esconder a través del espectro de frecuencia de la señal del audio usando un código independiente a este. Como resultado, la señal ocupa un ancho de banda que es más de lo que realmente se requiere para la transmisión.
- ➔ Eco oculta = Divide la señal en bloques, incrusta un bit del mensaje a esconder en cada uno si y sólo si produce un eco en ella y las 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 LSB (Least Bit Significant) 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

Ocultamiento

Algoritmo:

1) Obtén el nombre del archivo de audio
2) Abre el archivo de audio en modo lectura
3) Lee los bytes del audio
4) Guarda los bytes en un array
5) Escribe el mensaje a ocultar
6) Completa el mensaje con información basura
    - ➔ Información basura a insertar = '#'
    - ➔ Se agrega para que los bits del mensaje cubran todos los bytes del audio
    1) Multiplica el tamaño del mensaje por 8
        - ★ Representa los bits en bytes
    2) Multiplica el resultado por 8
        - ★ Representa los bytes del audio que serán cubiertos
    3) Resta el número de frames en el audio con el resultado
        - ★ Calcula el número de bytes restantes en el audio
    4) Divide el resultado entre 8
        - ★ Calcula el número de caracteres basura a insertar
    5) Multiplica el resultado por el carácter
        - ★ Genera los caracteres basura a insertar
    6) Anexa la información basura al mensaje a ocultar
7) Transforma el mensaje a bits
    1) Convierte cada carácter del mensaje a su forma ASCII
    2) Convierte cada forma ASCII a su equivalente binario
    3) Elimina la identificación binaria ('0b')
    4) Añade un 0 para completar EL byte
    5) Mapea cada byte con su numeración
    6) Añade cada elemento del mapeo a un array
8) Realiza la codificación LSB entre el audio y mensaje
    1) Realiza un ANDing entre el byte con el número 254
        - ★ Elimina el último bit del byte

2) Realiza un ORing entre el resultado con el bit del mensaje
    - ★ Añade un bit del mensaje a ocultar al byte del audio

9) Genera un nuevo archivo con el audio que tiene el mensaje oculto
    1) Reagrupa los bytes del audio en un array
    2) Asigna un nombre al nuevo archivo
    3) Abre el nuevo archivo en modo escritura
    4) Settea los parámetros del audio
    5) Escribe los bytes del audio en el nuevo archivo
10) Cierra el archivo de audio original

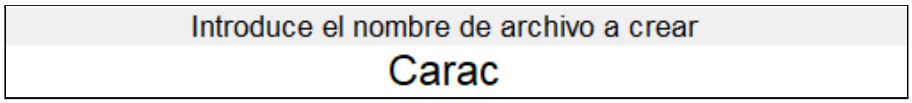
Recuperación

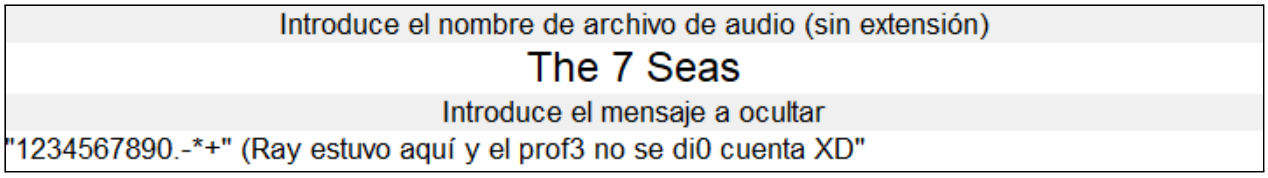
Algoritmo:

1) Abre el archivo de audio con el mensaje oculto en modo lectura
2) Lee los bytes del audio
3) Guarda los bytes en un array
4) Extrae el LSB de cada byte
    - ➔ Realiza un ANDing entre el byte con el número 1
5) Agrupa los LSB en grupos de 8
6) Convierte los grupos en binario
7) Convierte los grupos a su carácter en ASCII
8) Inserta cada carácter encontrado en un string
9) Elimina la información basura del mensaje
    - ➔ Detecta la primera instancia de un carácter basura (‘#’)
    - ➔ Divide el mensaje en dos partes
        - ★ Añade la 1° parte (el mensaje) en la 1° localidad del array
        - ★ Añade la 2° parte (la información basura) en la 2° localidad del array
10) Devuelve el mensaje oculto
    - ➔ Imprime el mensaje en consola
    - ➔ Devuelve la primera localidad del array
        - ★ Contiene el mensaje oculto sin información basura
11) Cierra el archivo de audio con el mensaje oculto

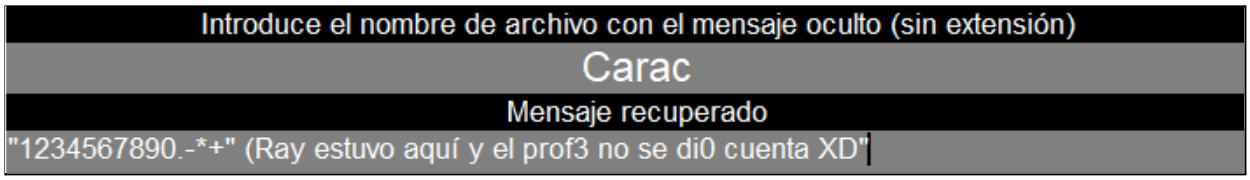
Interfaz gráfica

1) Ocultamiento
    - ➔ Archivo a crear = Cómo se llamará el archivo de audio con el mensaje oculto
    - ➔ Archivo de audio = Donde se ocultará el mensaje
    - ➔ Mensaje a ocultar = Cadena de caracteres a ocultar





2) Recuperación
    - ➔ Archivo oculto = Nombre del archivo con el mensaje oculto
    - ➔ Mensaje recuperado = Cadena de caracteres recuperada del archivo



Conclusión

Con base en lo que vimos a lo largo del proyecto, pudimos apreciar una nueva perspectiva sobre la ocultación de mensajes. Aprendimos que la criptografía no es el único método para esconder datos de manera segura y efectiva. Sin duda la esteganografía tiene aplicaciones muy interesantes viendo la cantidad de medios bajo los cuales se puede aplicar. Incluso sería viable combinar ambos rubros para buscar algoritmos de seguridad más robustos.

Por el otro lado, este proyecto nos hizo más conscientes sobre la gran cantidad de vulnerabilidades a la que estamos expuestos en esta era digital. Ahora más que nunca debemos prestar más atención a los riesgos que existen en línea para proteger nuestra información personal. La esteganografía es una técnica en desarrollo pero con un alcance infinito, por lo que vale la pena ver cómo evoluciona para adaptarse a ella. En este caso, el de audio aún tiene muchas facetas por explorar, así que sería de gran utilidad intentar otros algoritmos más allá del LSB para estar mejor entrenados en el área y generar métodos de detección de mensajes en un futuro.

Referencias

1) Kaspersky, 2019, “¿Qué es la esteganografía digital?” (09/04/2021), 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” (09/04/2021), 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) Wikipedia, 2021, “Esteganografía” (09/04/2021), Recuperado de: https://es.wikipedia.org/wiki/Esteganografía
4) Moreira et al. 2017, “Análisis de técnicas de esteganografía aplicadas en archivos de audio e imagen” (09/04/2021), Recuperado de: https://polodelconocimiento.com/ojs/index.php/es/article/download/10/pdf
5) IICybersecurity, 2018, “¿Cómo ocultar mensajes secretos en archivos de música? (09/04/2021), Recuperado de: https://www.iicybersecurity.com/audio-esteganografia.html
6) MDN, 2020, “Digital audio concepts” (09/04/2021), Recuperado de: https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Audio_concepts