ALGORITMOS PARA ESTEGANOGRAFÍA EN ARCHIVOS DE AUDIO BASADO EN CAPAS

Fernando Chávez[1], Arthur Meza[2], Abel Reyes[3], Yuber Velazco[4].
Universidad Nacional de San Agustín
[1]fchavezme@unsa.edu.pe, [2]amezapa@unsa.edu.pe, [3]areyesro@unsa.edu.pe, [4]yvelazco@unsa.edu.pe

Abstract. Given the growing use of digital information, the need to preserve its security has been increasing, making it necessary to use different concealment and encryption techniques for the proposed purpose. This article seeks to solve this need, combining the encryption and concealment of information together with the encoding and compression of data in text to audio files, carried out through layers with Huffman coding, encryption by Advanced Encryption Standard (AES) and hiding through Least Significant Bit (LSB), maintaining the original quality of the audio and the imperceptibility of the modification. The entire program has been tested and good results were obtained.

Keywords. steganography, audio steganography, LSB, encryption.

I. INTRODUCCIÓN

La esteganografía, arraigada en la historia de la comunicación, ha sido un arte milenario dedicado a ocultar información en contextos aparentemente ordinarios. Desde los ingeniosos métodos empleados por civilizaciones como la antigua Grecia para cifrar mensajes secretos, hasta su adaptación en la era digital en la ocultación de información en textos, audios o videos. Su aplicación abarca desde la protección de la privacidad en comunicaciones digitales hasta la seguridad en la transmisión de datos confidenciales, ofreciendo un método encubierto para resguardar información en entornos digitales.

El desafío que afronta es preservar la privacidad digital, el de guardar información que solo está destinada para un grupo selecto de personas que tengan conocimiento acerca de la técnica utilizada. Tomando como ejemplo para la investigación del presente artículo, se tiene el envío de información mediante un archivo de audio, donde lo que se busca es cifrar y almacenar dicha información en los bits del audio sin llegar a alterar significativamente su calidad, para lo cual puede utilizarse diversas técnicas, entre las más conocidas se encuentra la búsqueda de bits menos significativos (LSB), es decir, aquellos bits que no generar un cambio perceptible en la audición humana.

La presente investigación busca reforzar la robustez del algoritmo de ocultación LSB basándolo en capas, asimismo, evaluar la distorsión y calidad del audio después del procesamiento. En este marco, el artículo contribuye al campo de la esteganografía auditiva mediante la implementación y evaluación de un algoritmo específico basado en la técnica del LSB, la cual será reforzada con otros algoritmos previos a su ocultación. Al ofrecer una visión integral de la eficacia y las limitaciones de esta técnica en archivos de audio, se presenta como un recurso meritorio para la protección de información confidencial en entornos digitales.

Para entender el propósito de la investigación, se presenta algunos trabajos previos.

En primer lugar, la investigación realizada por Rivadeneira y Halak [1], titulado “Desarrollo de software esteganográfico con criptografía asociada”, se aborda la necesidad de una herramienta que permita ocultar información dentro de otros archivos multimedia de manera segura y eficiente, un campo conocido como esteganografía. Dicha investigación tiene como objetivo presentar una propuesta de software esteganográfico que permita realizar esta tarea de forma segura y eficiente, usando algoritmos de compresión, cifrado y esteganografía. Dicho software se desarrolló utilizando el modelo de desarrollo en espiral, que es un proceso iterativo que abarca las fases de análisis, diseño, implementación y prueba. Las principales contribuciones del artículo son el diseño e implementación de este software esteganográfico, su evaluación a través de pruebas de funcionalidad, seguridad y rendimiento, y la comparación con otras herramientas similares para destacar sus ventajas y desventajas.

En segundo lugar, la investigación “Ocultamiento de información confidencial en imágenes BMP y audio WAV mediante el método LSB” de los autores Chávez, Gutiérrez y Casanova [2] aborda el problema de la necesidad de ocultar información en archivos de audio e imagen de manera segura y eficiente, un campo conocido como esteganografía. Para abordar este problema, los autores realizan un análisis descriptivo de las principales técnicas de esteganografía utilizadas en estos medios digitales. Este análisis se basa en una revisión literaria de dominios, métodos y técnicas que son parte de este conjunto, identificando su funcionamiento, cualidades y debilidades. Las principales contribuciones del artículo son la identificación de la relación entre las técnicas de esteganografía de audio e imagen en su forma de implementación, la determinación de que con este método surgen la necesidad de combinarlo con la criptografía simétrica o asimétrica.

En tercer lugar, Sánchez, Munive y Jaramillo en el trabajo “LSB Algorithm to Hide Text in an Audio Signal” [3], abordan la necesidad de transmitir o almacenar información confidencial de forma segura, evitando que sea detectada o

alterada por terceros. El objetivo del estudio es desarrollar una técnica de esteganografía que permita ocultar información de texto en archivos de audio sin modificar su calidad o percepción. La solución al problema es el uso del algoritmo LSB, que consiste en reemplazar el bit menos significativo de cada byte de un archivo de audio WAV por un bit de la información a ocultar. Las principales contribuciones del artículo son la implementación y evaluación del algoritmo LSB en tres melodías de diferentes géneros, la comparación de los resultados obtenidos mediante el SNR, y las recomendaciones para mejorar la seguridad y la capacidad de la técnica.

Por ende, en la sección 2 se mostrará un marco teórico, en la sección 3 se abordará la metodología utilizada en la implementación del algoritmo esteganográfico, en la sección 4 se realizará el desarrollo del algoritmo, en la sección 5 se mostrará los resultados obtenidos y, por último, en la sección 6 se darán a conocer las conclusiones a las que se ha llegado.

II. ESTEGANOGRAFIA EN ARCHIVOS DE AUDIO

A. ESTEGANOGRAFÍA

El origen del término *esteganografía* proviene del griego *steamos* y *graphos* que significan oculto y escritura respectivamente. Se trata de una técnica versátil y útil para ocultar información importante dentro de algún otro elemento. Esta técnica, en la actualidad, se puede utilizar en áreas como la seguridad de la información, protección de los derechos de autor, la vigilancia y el espionaje y la censura, aunque, también crea problema a la hora de detectar y prevenir el ocultamiento de información confidencial [4]. Un sistema esteganográfico se compone de: un mensaje o secreto, un objeto de cubierta, una función o técnica de esteganografía, un canal inseguro, una clave esteganográfico y un objeto esteganográfico [1].

Como también es un modelo información, este tiene algunas ventajas y desventajas [4].

Ventajas:

- permite la transmisión segura de información confidencial
- puede ser transmitida a distintos tipos de archivos dado su invisibilidad a simple percepción
- protege la privacidad en las redes sociales
- se utiliza en el campo de la seguridad para la detección de posibles amenazas

Desventajas:

- se puede utilizar en el ámbito ilegales como la transmisión de información confidencial
- se puede dar evasión de censura en Internet o la difusión de propaganda
- facilita la propagación de malwares y virus informáticos

Tipos de Esteganografía

1) De texto

Consiste en ocultar información dentro de los archivos, aquí los datos secretos se ocultan detrás de cada enésima letra de cada palabra del mensaje de texto [5].

2) De imagen

Se usa intensidades de pixeles para ocultar datos, ya que la imagen es fuente de cobertura amplia debido que hay una gran cantidad de bits presentes en la representación digital de una imagen [5].

3) De audio

Consiste en incrustar mensajes secretos en una señal de audio este siendo modificada por una secuencia binaria este a comparación de otros es un poco más difícil su proceso [6].

4) De video

Se usa para ocultar cualquier tipo de archivo en uno de formato de video digital utilizando generalmente la transformada de coseno directa (DCT), la cual altera los valores que se usa para ocultar los datos en cada una de las imágenes en el video, donde el ojo humano no lo puede percibir [5].

B. ESTEGANOGRAFÍA EN EL DOMINIO DEL AUDIO

1) Importancia y aplicaciones de la esteganografía de audio

La eficacia en la obtención de confidencialidad se logra mediante la aplicación de técnicas esteganográficas, especialmente en el ámbito del audio, donde se han propuesto métodos innovadores y flexibles. El propósito subyacente de estos sistemas esteganográficos es establecer una forma segura y robusta de ocultar grandes volúmenes de información confidencial [7].

Una aplicación evidente de este sistema esteganográfico es la comunicación encubierta a través de una señal de audio que aparenta ser inofensiva, como en conversaciones telefónicas o videoconferencias.

2) Desafíos en la esteganografía de audio

A pesar de que la ocultación de datos en archivos de audio presenta desafíos, como la sensibilidad del Sistema Auditivo Humano (HAS), este aún tolera alteraciones comunes en pequeños rangos diferenciales. Por ejemplo, los sonidos agudos suelen enmascarar los sonidos graves, y hay distorsiones ambientales que, en la mayoría de los casos, pasan desapercibidas para los oyentes [7]. Estas características han llevado a los investigadores a explorar la utilización de señales de audio como portadoras para ocultar datos. No obstante, es esencial analizar las compensaciones entre las alteraciones introducidas para incrustar datos y la calidad resultante de estas señales de audio.

3) Técnicas para esteganografía basado en capas

a. Compresión mediante el algoritmo de Huffman

Dicho algoritmo consiste en la creación de un árbol binario en el que se etiquetan los nodos hoja con los caracteres, junto a sus frecuencias, y de forma consecutiva se van uniendo cada pareja de nodos que menos frecuencia sumen, pasando a crear un nuevo nodo intermedio etiquetado con dicha suma. Se procede a realizar esta acción hasta que no quedan nodos hoja por unir a ningún nodo superior, y se ha formado el árbol binario [8].

b. Cifrado mediante el algoritmo Rijndael

Rijndael o Advanced Encryption Standard (AES) es un algoritmo de cifrado simétrico en bloques, desarrollado por los criptógrafos belgas Joan Daemen y Vincent Rijmen. El algorítmo opera sobre bloques de datos de 128, 192 o 256 bits utilizando claves de 128, 192 o 256 bits.

Este consiste en una serie de iteraciones o ciclos llamados rondas. La cantidad de rondas depende del tamaño del bloque de datos y de la longitud de la clave, dichas rondas se operan sobre una matriz de bytes llamada “Estado”.

Cada ronda de encriptado efectúa cuatro transformaciones sobre la matriz: SubBytes - ShiftRows - MixColumns – AddRoundKey [9].

c. Ocultación de información mediante LSB

El método LSB, corresponde a la modificación del bit menos significativo, el cual corresponde a un método simple y sencillo de fácil implementación en código de programación [2]. Esto refiere a la modificación del bit con menor valor, el cuál es el del extremo derecho, puesto que está destinado a contener los valores más pequeños.

Además, este tipo de técnica permite la ocultación de una gran cantidad de datos [3], ya que comúnmente se trabaja la ocultación bit x bit y los archivos donde se ocultarán presentan una gran cantidad de muestras de bytes donde colocar dichos bits.

III. METODOLOGÍA

A. PARA LA OCULTACIÓN DE INFORMACIÓN:

Para la implementación del algoritmo de esteganografía se escogió trabajar con un archivo de texto y uno de audio, donde la información del texto será la que se va a ocultar.

Dado que aplicar únicamente el algoritmo de LSB sería poco seguro, es que se decidió por dividir este proceso de ocultación en 3 capas: compresión, encriptación, ocultación. Donde cada capa provee un nivel de complejidad extra a la ocultación de la información y así ser menos propensa a ataques.

1) PRIMERA CAPA: CODIFICACIÓN Y COMPRESIÓN DEL ARCHIVO DE TEXTO

Para esta capa se utilizará el algoritmo de compresión de Huffman, el cual nos proveerá dos ventajas a la hora de ocultar la información. La primera ventaja es que codifica los caracteres en 0 y 1, lo que brinda mayor comodidad al pasar a las siguientes capas, asimismo, comprime la información, por lo que da mayor posibilidad a más cantidad de datos de entrada.

Lo interesante, es que codificar y comprimir la información con este algoritmo brinda una seguridad muy alta, puesto que, si no se tiene el árbol binario generado en la compresión, lo cual será un archivo de entrada para el desencriptado, no se puede saber que cadena de bits corresponde a que carácter.

Para ilustrar de mejor manera el funcionamiento, se presenta la figura 1.

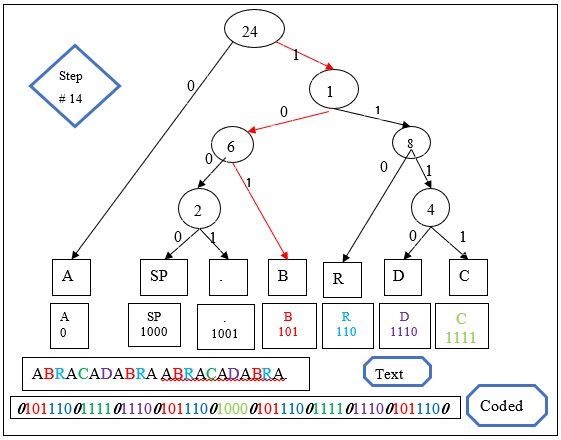

Fig. 1. Huffman coding, the final tree of the code. [10]

2) SEGUNDA CAPA: ENCRIPTACIÓN DE LA CODIFICACIÓN RESULTANTE

Luego de codificar la información en 0 y 1, se procede a encriptarla mediante el algoritmo de Rijndael o AES, el cual trabajará con datos de tipo hexadecimal, y aunque suene contraproducente, esto puede servir como método de ofuscación que dificulta el análisis de datos para un observador casual o un atacante inexperto, además de contar con una clave tanto para el cifrado como para el descifrado.

Luego de aplicar el cifrado y obtener otros datos en hexadecimal, se procede a volverlos cadenas de bits nuevamente, para así poder continuar con la siguiente capa.

3) TERCERA CAPA: OCULTACIÓN DE DATOS ENCRIPTADOS EN EL AUDIO

En este proceso los datos a ocultar deben estar en forma de bits, por ello se dio la conversión en la anterior capa.

El algoritmo a utilizar para la ocultación de información es del Bit Menos Significativo (LSB), dicho bit será el de la memoria más baja o, en términos simples, el del extremo derecho.

La ocultación consistirá en reemplazar los dos bits menos significativos de cada muestra del audio con los bits obtenidos, ello hasta terminar de leer todos los bits resultantes.

Dado que este algoritmo es poco robusto y seguro de realizar, se propuso aplicar un método de selección de muestras pseudo-aleatorias mediante una semilla designada para la generación de los números. Dicha semilla es producto de la concatenación de los últimos valores en bits de cada codificación de carácter en el árbol de Huffman, dicha cadena será de 32 bits, la cual se convierte a su valor entero sin signo para establecer la semilla que, en el descifrado, dará los mismos números/índices para búsqueda en las muestras.

Las figuras 2 y 3 ilustran de mejor manera este procedimiento.

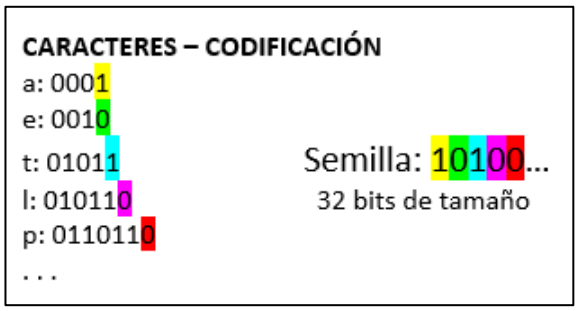

Fig. 2. Obtención de la semilla para generar índices pseudo-aleatorios.

Fig. 3. Explanation of the LSB (right) and the MSB (left) concepts. [11]

B. PARA LA RECUPERACIÓN DE INFORMACIÓN:

El proceso de desencriptado es aplicar los algoritmos de manera inversa, es decir, teniendo como entrada el audio con la información oculta, el árbol de Huffman y la clave para desencriptado, lo que se realiza es la obtención de los bits menos significativos de las muestras, donde la semilla para generar los índices se encuentra en el archivo del árbol de Huffman, luego se agrupan los bits en cadenas de 8 y se convierten a su valor hexadecimal correspondiente para pasar al desencriptado, donde se obtendrá los hexadecimales originales que al convertirlos a sus valores en bits se tendrá la cadena de información codificada por el algoritmo de Huffman, la cual mediante el árbol ingresado por un archivo, se procede a decodificar la información.

IV. Desarrollo

Al momento de aplicar los algoritmos, se ha tomado como mensaje de texto un archivo con 4810 caracteres, el cual será ocultado en un archivo de audio .wav de aproximadamente 2 segundos. La librería utilizada para el procesamiento de los audios es libsndfile.

A. PRIMERA CAPA: CODIFICACIÓN Y COMPRESIÓN DEL ARCHIVO DE TEXTO

Esta capa trabaja únicamente con el mensaje de texto de entrada y realiza la tabla de codificación solo con los caracteres presentes en el texto.

Respecto a la compresión obtenida, suponiendo una codificación normal de cada carácter del texto, el resultante en bits sería 4810*8 = 38480, sin considerar aquellos caracteres que tienen pesos mayores a la de un byte, como las tildes; en cambio, mediante el algoritmo de Huffman se obtuvo un resultante de 25344 bits, donde la codificación más corta fue de 4 bits y la más larga de 14, llegando a comprimir el mensaje original en un 34.1372%, porcentaje significativo si lo que se quiere lograr es una mayor capacidad de ocultación en los archivos de audio.

B. SEGUNDA CAPA: ENCRIPTACIÓN DE LA CODIFICACIÓN RESULTANTE

La segunda capa también trabaja con el mensaje de entrada, aunque ya no en caracteres sino en bits. Suponiendo que se llega a filtrar la codificación correspondiente en el árbol de Huffman, dicha información puede ser decodificada fácilmente, por ello, se encripta los bits obtenidos mediante Rijndael para obtener otra representación de los originales.

Para el encriptado se utilizó una llave de 256 bits y una entrada de datos de 128 bits (16 hexadecimales), siendo 14 el número de rondas para el encriptado. Cabe resaltar que hasta el momento no se ha registrado un algoritmo que pueda romper dicho cifrado, lo que lo hace sumamente seguro a posibles ataques e intentos de descifrado.

Por otro lado, dado que para la encriptación mediante Rjindael es necesario tener una cantidad de bits múltiplo de 128, se realizó un algoritmo para el completado de bits, en el apartado de Huffman, con el código de un carácter muy poco común en textos, el ‘´’, el cual tendrá una codificación de “1”, para poder tener mejor exactitud al momento de concatenar la cadena resultante hasta ser múltiplo de 128.

Luego del encriptado mediante la clave de 256 bits, los valores resultantes se duplican, es decir, se retornan 32 hexadecimales, los cuales se volverán cadenas de bits para ser ocultadas en el audio seleccionado, teniendo un total del doble de bits que el mensaje original antes del cifrado.

C. TERCERA CAPA: OCULTACIÓN DE DATOS ENCRIPTADOS EN EL AUDIO

Luego de aplicar ambas capas se puede ocultar los bits resultantes dentro del archivo de audio. En este caso, el audio utilizado tiene una resolución de 16 bits por muestra, una tasa

de muestreo de 44.1 kHz y dos canales de audio (estero); es decir, se tiene 88200 muestras por segundo, dado que son dos canales de audio, y cada muestra contiene un valor de 16 bits.

Luego de procesar la información del audio y obtener las muestras dentro de un buffer, se puede aplicar la técnica de LSB con la selección de muestras propuesta. Dado que se ocultará 2 bits por muestra, el número de muestras necesarias es igual a la cantidad de bits obtenida antes de la codificación de Rijndael, siendo más óptimo para ocultar mayor cantidad de información; además, el cambio de los dos bits menos significativos implica un aumento o disminución máximo de 3 respecto al valor original, cambio realmente imperceptible al oído humano.

Respecto a la duración mínima que debería tener dicho audio, se calcula mediante el número de bits obtenidos después de la codificación de Huffman más la completación hasta ser múltiplo de 128, y puesto que se ocultan 2 bits por muestra, dicha cantidad obtenida no cambia. En el presente caso se obtuvieron 25251 bits, lo que completado hasta ser múltiplo de 128 resulta en 25344, siendo igual el número de muestras necesarias, y dado que el audio utilizado tiene una tasa de muetreo de 44.1 kHz por segundo y dos canales de audio, el mínimo de segundos necesarios para ocultar la información es de 25344/88200 ≈ 0.287.

A continuación, se muestra la figura 4 y 5, donde se representa la amplitud del audio original y el que contiene el mensaje oculto, respectivamente. Comparando ambas gráficas, no se logra apreciar una diferencia detectable en sus amplitudes.

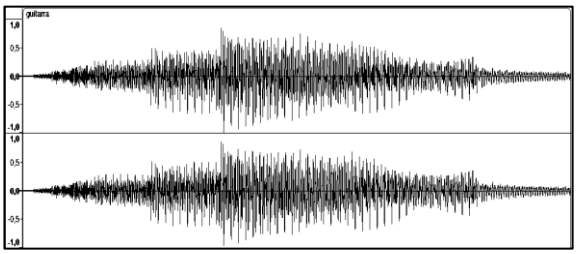

Fig. 4. Mensaje original.

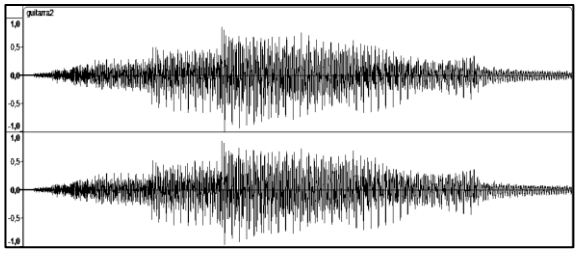

Fig. 5. Audio modificado.

V. Resultados y Discusión

Para la evaluación de los resultados y determinar la eficacia del método utilizado respecto a la imperceptibilidad y mantenimiento de calidad en los audios, se utilizaron métricas objetivas como *Mean Square Error* (MSE) y *Signal to Noise Ratio* (SNR), las cuales medirán el cambio en las señales de audio y la relación entre la señal de audio deseada y el ruido de fondo.

A. MEAN SQUARE ERROR (MSE)

El MSE es la división de la sumatoria de las diferencias de señales (audio original con audio modificado) elevado al cuadrado, entre la cantidad de muestras. Esta métrica mide la distorsión del audio después de aplicar el algoritmo de ocultación, la cual mientras más cerca este de cero, las señales del audio de entrada y salida serán más similares, por ende, menos distorsión ocasionada por el algoritmo, que indica la efectividad y desempeño del mismo.

La definición de la ecuación es la siguiente:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(s1_i - s2_i)^2 \quad [12]$$

Donde s1 y s2 son las señales del audio, original y modificada respectivamente, y N es la cantidad de muestras de señales del audio.

B. SIGNAL TO NOISE RATIO (SNR)

Esta métrica mide la diferencia de la potencia de señal emitida y el nivel de ruido que lo afecta. En este caso se mide entre dos audios, el de entrada y de salida. Es así, que dicha métrica evalúa la calidad de la señal de salida luego de aplicar los algoritmos de ocultación correspondiente. Por ende, si se mide la diferencia entre la señal emitida con el ruido, mientras mayor sea el resultado, mayor es la calidad de la señal de salida, siendo aceptable un valor de 20 y muy bueno si es mayor a 60.

En la aplicación del algoritmo, mientras mayor sea el resultado del SNR, indica que es mayor la calidad de ocultación del mismo, puesto que una persona ajena no podría identificar que ha habido una modificación en el audio original.

La ecuación utilizada se define de la siguiente manera:

$$SNR = 10\,log_{10}\frac{\sum_{i=1}^{N}(s1_i)^2}{\sum_{i=1}^{N}(s1_i - s2_i)^2} \quad [12]$$

Donde s1 y s2 son las señales de audio, original y modificada respectivamente, y N es la cantidad de muestras de señales del audio.

Al aplicar dichas métricas al audio modificado se obtuvieron los siguientes resultados:

Muestra	Original	Modificada	Original^2	Original - Modificada)^2
0	0.0001373	0.0001373	1.88593E-08	0
4500	0.0105743	0.0105591	0.000111817	2.32831E-10
9000	0.0052795	0.0052795	2.78736E-05	0
13500	-0.0325775	-0.0325470	0.001061294	9.31323E-10
18000	-0.0607147	-0.0607147	0.003686277	0
22500	0.0577850	0.0577850	0.00333911	0
27000	0.3660126	0.3660126	0.133965204	0
31500	-0.2799377	-0.2799377	0.078365141	0
36000	0.5151672	0.5151672	0.265397281	0
40500	0.3476105	0.3476105	0.120833041	0
45000	0.2485962	0.2485962	0.061800066	0
49500	-0.2806396	-0.2806396	0.078758612	0
54000	-0.2075195	-0.2075195	0.043064356	0
58500	0.0526428	0.0526428	0.002771267	0
63000	0.2426300	0.2426300	0.058869319	0
67500	-0.1413574	-0.1413574	0.019981921	0
72000	0.0629425	0.0629425	0.003961759	0
76500	0.0597534	0.0597534	0.003570471	0
81000	-0.0408020	-0.0408020	0.001664803	0
85500	-0.0839844	-0.0839844	0.007053375	0
90000	0.0385742	0.0385742	0.00148797	0
Total			3912.404389	1.51477E-05
MSE	**1.68099E-10**			
SNR	**84.12096215**			

Tabla 1. Datos resultantes de la aplicación de las métricas MSE y SNR en el audio modificado.

Si bien en la tabla 1 mostrada se aprecian 21 muestras, los datos que se han procesado fueron las 90112 muestras presentes en el audio, considerando las muestras de los dos canales como una, la tabla solo es una representación de muestras entre 4500 espacios cada una, por ello es que se ve mucha una cantidad notable de 0 en la última columna, sin embargo, los datos totales y del MSR y SNR son en base a todas las muestras.

En dicha tabla, los valores ingresados que van desde -1 a 1, representan la amplitud normalizada de las señales de audio, ello para poder procesar y analizar de mejor manera los datos obtenidos.

El valor obtenido del MSE es de 1.68099E-10, el cual es un valor muy cercano a cero, dando a entender que la distorsión o variación en las señales del audio modificado son mínimas, casi imperceptibles. Por otro lado, el valor obtenido en SNR es de 84.12096215 un valor bastante bueno que indica la buena relación entre potencia de señal emitida y el ruido que produce. Dichos datos obtenidos se deben a dos factores principalmente; primero, las coincidencias entre los dos bits menos significativos en los cambios de las muestras, dando una diferencia de 0 en cuanto su modificación; y segundo, la gran cantidad de muestras respecto a los bits ingresados, pues no llega ser tan significativo.

Para una mejor visión de ello, se presenta la gráfica 6.

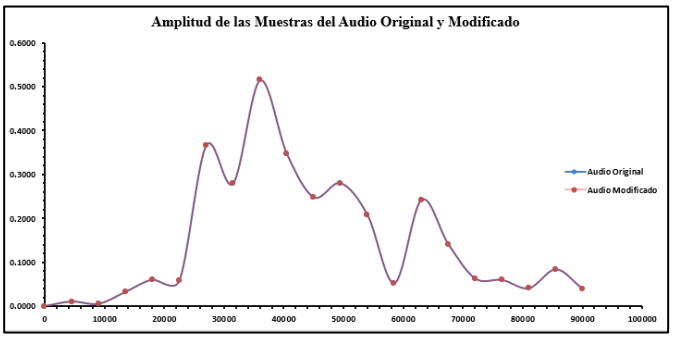

Fig. 6. Amplitud normalizada del audio original y modificado.

Para entender mejor los cambios se cambió sus señales a sus valores absolutos antes de realizar la gráfica. Como se observa, prácticamente es imperceptible la modificación realizada en el audio, es como si ambas frecuencias se solaparan, pero ello se debe a que es la representación de todo y los cambios mínimos no se logran apreciar. Por ello, se realizaron las mismas métricas, pero a un segmento más pequeño del audio. Los resultados obtenidos se presentan en la tabla 2.

Muestra	Original	Modificada	Original^2	Original - Modificada)^2
0	0.0001373	0.0001373	1.88593E-08	0
1	0.0000000	0.0000305	0	9.31323E-10
2	0.0000000	0.0000000	0	0
3	0.0000000	0.0000458	0	2.09548E-09
4	-0.0001526	-0.0001373	2.32831E-08	2.32831E-10
5	0.0001373	0.0001678	1.88593E-08	9.31323E-10
6	0.0001373	0.0002136	1.88593E-08	5.82077E-09
7	-0.0001373	-0.0001678	1.88593E-08	9.31323E-10
8	0.0001373	0.0001526	1.88593E-08	2.32831E-10
9	0.0000000	0.0000000	0	0
10	-0.0002747	-0.0003052	7.54371E-08	9.31323E-10
11	0.0000000	0.0000305	0	9.31323E-10
12	0.0001373	0.0001678	1.88593E-08	9.31323E-10
13	-0.0001373	-0.0001068	1.88593E-08	9.31323E-10
14	0.0000000	0.0000000	0	0
15	0.0001373	0.0001831	1.88593E-08	2.09548E-09
16	0.0001373	0.0001678	1.88593E-08	9.31323E-10
17	-0.0001373	-0.0001526	1.88593E-08	2.32831E-10
18	0.0000000	0.0000153	0	2.32831E-10
19	-0.0001526	-0.0001526	2.32831E-08	0
20	0.0001373	0.0001678	1.88593E-08	9.31323E-10
21	0.0000000	0.0000458	0	2.09548E-09
22	-0.0001373	-0.0000916	1.88593E-08	2.09548E-09
23	0.0000000	0.0000000	0	0
24	0.0000000	-0.0000153	0	2.32831E-10
25	-0.0001373	-0.0001068	1.88593E-08	9.31323E-10
26	0.0002747	0.0003052	7.54371E-08	9.31323E-10
27	-0.0002899	-0.0002594	8.40519E-08	9.31323E-10
28	0.0002747	0.0003052	7.54371E-08	9.31323E-10
29	-0.0002747	-0.0003052	7.54371E-08	9.31323E-10
30	0.0001373	0.0001221	1.88593E-08	2.32831E-10
Total			6.96396E-07	2.86382E-08
MSE	**9.23812E-10**			
SNR	**13.85911302**			

Tabla 2. Datos resultantes de la aplicación de las métricas MSE y SNR a las primeras 31 muestras del audio modificado.

A comparación de la tabla anterior, esta tabla muestra una mayor variedad de datos cambiados, y aunque se vean varias muestras con una modificación, los valores del MSE y SNR son buenos. Empezando con el MSE, al tener un valor muy cercano al 0, se demuestra que los cambios realizados son mínimos y no perceptibles a primera instancia; mientras que el SNR resulta en un valor mayor a 10 pero menor a 20, lo cual es moderado, donde la potencia de la señal de audio aún predomina sobre el

ruido generado. Aunque se logre evidenciar con estos datos el haber modificado el audio original, dichos cambios son imperceptibles al oído humano, por lo que no son significativos a la hora de calificar la calidad y desempeño de ocultación realizada.

Para una mejor visión de ello, se muestra la gráfica 7.

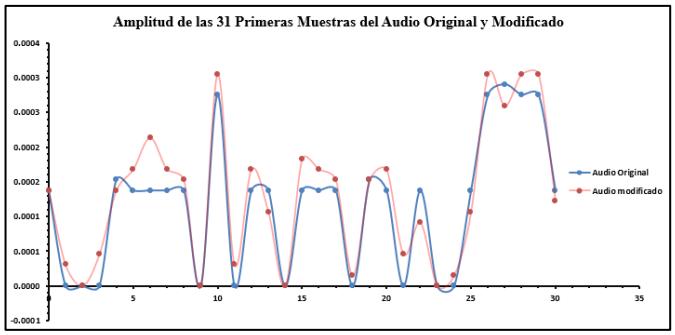

Fig. 7. Amplitud normalizada del audio original y modificado en las primeras 31 muestras.

Respecto a investigaciones relacionadas, la realizada por Mahmoud y Elshoush [12], quienes propusieron otro método mejorado del LSB, obtuvieron un valor promedio del SNR de 99.98 dB, el cual es un valor mayor al obtenido por nuestro método pero ambos se consideran buenos niveles; respecto al MSE, obtuvieron valores entre 3.5693E-02 dB y 3.9788E-03 dB, los cuales son valores menores a los obtenidos en la presente investigación, pero, al igual que el SNR, son valores buenos que indican la eficacia del método de ocultación.

VI. Conclusiones

Dado que el método tradicional LSB es poco robusto y propenso a extracción de información, se realizó un complemento/mejora de tal técnica, dividiéndola en 3 capas, donde primero se realiza una compresión y codificación del mensaje de texto, luego se encripta para sustituir sus valores iniciales, y finalmente se encripta en muestras pseudo-aleatorias de audio. Este conjunto de capas provee al algoritmo mayor capacidad de ocultación y mayor robustez, dado que para poder obtener la información es necesario tanto la llave de encriptación como la codificación obtenida mediante Huffman. De igual manera, se logró mantener la calidad del audio después de ocultar la información, así como su imperceptibilidad; sin embargo, existen ciertas limitaciones, como la necesidad de compartir la codificación de Huffman que también contiene la semilla para la generación de las posiciones pseudo-aleatorias, además, se requiere una notable capacidad de reserva en pila durante la decodificación de los caracteres, ya que usa un método de regresión que llega a ocupar una mayor cantidad que la reservada predeterminadamente en la memoria virtual. Esta investigación se propone como un punto de inicio para desarrollar mejores y más sofisticadas técnicas de esteganografía.

VII. Referencias

[1] J. Rivadeneira and B. Halak, “Desarrollo de software esteganográfico con criptográfia asociada”, MSKN, vol. 8, pp. 255–264, Nov. 2017.

[2] P. Chávez-Lugo, G. A. Gutiérrez-Carreón, S. A. Casanova-Valencia, “Ocultamiento de información confidencial en imágenes BMP y audio WAV mediante el método LSB,” Revista de la Facultad de Contaduría y Ciencias Administrativas, vol. 5, no. 10, pp. 92-99, Dec. 2020.

[3] B. E. Sánchez Rinza, L. G. Munive Morales y A. Jaramillo Núñez, “LSB Algorithm to Hide Text in an Audio Signal”, Comput. y Sist., vol. 26, no. 1, pp. 39-44, Ago. 2022.

[4] J. Guaña Moya, «Usos y aplicaciones de la esteganografía en la era digital Uses and applications of steganography in the digital age»., RRI, vol. 2, n.º 1, pp. 61–75, jun. 2023.

[5] G. E. Naranjo Viteri, “Técnicas, mecanismos de seguridad y encriptación de la información y desarrollo de aplicaciones: estudio de mecanismos de detección de mensajes ocultos utilizando modelos estadísticos,” Tesis de grado, Fac. de Ing. Eléctrica y Electrónica, Escuela Politécnica Nacional, Quito, Ecuador, 2022.

[6] Kaspersky, “¿Qué es la esteganografía? Definición y explicación”. Accedido el 2 de febrero de 2024. [En línea]. Disponible en: https://latam.kaspersky.com/resource-center/definitions/what-is-steganography

[7] F. Djebbar, B. Ayad, K. Abed-Meraim y H. Hamam, “Comparative study of digital audio steganography techniques”, EURASIP Journal on Audio, Speech, and Music Processing, vol. 2012, no. 1, p. 25, 2012.

[8] J. Lezama, “Compresión de imágenes - codificación de Huffman,” Revista de Educación Matemática, vol. 32, no. 1, pp. 25-36, 2017.

[9] A. Segredo, E. Zabala y G. Bellora, “Diseño de un procesador criptográfico Rijndael en FPGA”, X Workshop Iberchip, pp. 64, 2004.

[10] E. Abu-taieh, "The Pillars of Lossless Compression Algorithms a Road Map and Genealogy Tree," International Journal of Applied Engineering Research, vol. 13, 2018.

[11] S. Rahman, J. Uddin, H. U. Khan, H. Hussain, A. A. Khan y M. Zakarya, “A Novel Steganography Technique for Digital Images using the Least Significant Bit Substitution Method”, IEEE Access, vol. 9, pp. 133-149, 2022.

[12] M. M. Mahmoud and H. T. Elshoush, "Enhancing LSB Using Binary Message Size Encoding for High Capacity, Transparent and Secure Audio Steganography–An Innovative Approach," in IEEE Access, vol. 10, pp. 29954-29971, 2022.