

Development of an Audio Steganography System Using Discrete Cosine Transform and Spread Spectrum Techniques

Olawale Timothy Adeboje, Arome Junior Gabriel(✉), and Adebayo Olusola Adetunmbi

Federal University of Technology, P.M.B. 704, Akure, Nigeria
{adebojeot,ajgabriel,aoadetunmbi}@futa.edu.ng

Abstract. In today's electronic society, security of information is an issue of major concern especially in the communication domain. Indeed, there is a very serious need for schemes that guarantee security of information exchanged between entities, over public networks. Several techniques such as cryptography and/or steganography have been proposed in existing literature. Audio steganography has certain advantages over conventional steganography. Existing audio steganography systems suffer from limitations that includes, their inability to embed information in multiple audio file formats, high distortion rate and low level of robustness of their resultant stego-files. To overcome these limitations, this current research focused on the development of an efficient and robust audio steganography system for the security of information whether in store or on transit across the Internet. This is achieved by first encrypting a given secret information, compressing the resultant cipher-text, then embedding the compressed file in a suitable cover (audio file) using frequency hopping spread spectrum technique, to yield as output, an audio stego file which is not distinguishable from the original audio cover file. To allow for the embedding of the secret compressed file, the selected audio cover file is first broken into frames using one dimensional Discrete Cosine Transform (DCT). Results of performance evaluation of this developed system shows that it has very low level of distortion as revealed by the Signal to Noise Ratio (SNR). The compression ratio obtained is also equal to one (1), which shows that the cover audio file is identical to the resultant stego-file.

Keywords: Cryptography · Cyber security · Encryption · Information security · Steganography

1 Introduction

Over the years, there have been significant growth in computer networks and more specifically, the Internet. Fundamentally, computers (or Information Technology) finds application in communications [1], health [2], electoral decision making (voting), governance and human resource management [3–5], crime control [6], prediction of weather/climate conditions [7], Web Services Security [8], and other facets of human existence. All these, coupled with the exponential increase of computer performance,

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12254, pp. 412–427, 2020.
https://doi.org/10.1007/978-3-030-58817-5_31

has expedited the distribution of multimedia data such as images, audio, video etc. Data transmission has been made very simple, fast and accurate using the internet. However, one of the main problems associated with transmission of data over the internet is that it may pose a security threat, that is, personal or confidential data can be stolen or hacked in many ways. Users may be reluctant to distribute data over the internet due to lack of security. One of the ways to conceal information is Cryptography. In cryptography, the structure of a message is scrambled to make it meaningless and unintelligible unless the decryption key is available. Unfortunately, it is sometimes not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret [3, 9, 10]. A method referred to as steganography is used for achieving these. Steganography is defined as the art and science of writing concealed messages in a manner that no one, except the owner/sender as well as the intended recipient, suspects the presence of any embedded secret message. The major objective of steganography is to allow for secure information exchange in an absolutely imperceptible fashion and to avoid drawing suspicion to the transmission of a hidden data. This research work combined both Cryptography and Steganography to hide text file into an audio signal. The use of Huffman algorithm and Lempel-Ziv-Welch algorithm was adopted for text compression in order to reduce the distortion in the audio file and Frequency Hopping Spread Spectrum technique to increase the robustness of the stego file.

The remainder of this paper is organized in a way that, Sect. 2 contains review of related existing research works. The design of the new system is reported in Sect. 3. Then Sect. 4 presents results and findings of the experiment conducted. Section 5 presents performance evaluation of the system. Section 6 contains the conclusion.

2 Related Works

Although Information hiding techniques like watermarking and steganography have shown some qualities as promising solutions towards addressing security concerns, they still exhibit some teething problems or shortcomings. The research work presented by [11] and [12] for instance, has a shortcoming of yielding distorted stego-files, which arouses suspicion of attackers. The work titled, “Increasing the Hiding Capacity of Low-Bit Encoding Audio Steganography Using a Novel Embedding Technique” in [13], presented LSB technique as an efficient strategy for concealing information in sound signals. Be that as it may, as the size of the secret message increases, the embedding process gets complex or complicated.

Another novel method for embedding data in the audio stream was suggested in the paper presented in [12]. This method/system however could only embed information in dot-wav audio file. Other file formats were not considered for suitability as cover-image at all.

In a bid to combine encryption and steganography techniques so as to provide a more powerful message concealing technique, and to improve the quality of the resultant stego-file, towards completely eliminating suspicion of covert communication, the authors of the work reported in [14] developed a tool for encrypting a message and hiding the resultant cipher-text in a digital object. The encrypted message is

embedded in the homogenous frames of mp3 audio file. The limitations of this research work are in two fold; first, the fact that the quality of the final output (stego-file) is dependent on file size and message length, has undesirable implications. Secondly, aside the mp3 audio file format, other audio file formats were not considered.

The authors of [15] in their work, "*analysis and design of three LSB techniques for secure audio steganography*" presented an audio steganography solution that uses encrypted image. The objectives were to embed image file in an audio file using 3rd LSB technique, and to analyze the quality of the resultant stego file after the embedding process is completed. Least significant bit (LSB) method was used. The limitation of the research work is that there was a big difference in the quality of the audio before and after embedding of information. This implies low robustness, and arouses a lot of suspicion on the part of their steganography system. The limitations of the research work are: the audio file used in the research is less (952 KB) and the research was only limited to .wav audio file.

The authors of the research paper in [18] carried out a survey of audio steganography techniques for secure data transmission. They opined that LSB is the simplest of steganography techniques, but spread spectrum technique offers more robustness to attacks.

The authors of the research work in [19] went further in their work to develop a more efficient audio steganography system that has increased robustness against attack, and is able to embed text messages in both MP3 and MP4 audio file formats, with impressive efficiency

In fact, other authors proposed the idea of more robust information systems that combine post quantum cryptography with steganography. Such systems can be used to protect information against common classical and even quantum attacks on information in the cloud [10], and even electronic voting systems [3].

In order to achieve improvement in the properties of images with embedded secrets and at the same time eliminate the propensity of high computational complexity in steganography systems, the authors of [20] adopted the use of particle swarm optimisation algorithm as well as AVX instructions available in modern CPUs. These as they claimed, accelerates data parallel operations required in image steganography with advanced optimizations.

The authors of the work in [21] proposed an Optical Character Recognition (OCR) based Steganography technique where, messages, in their feature forms, are embedded in a cover image. Character level features were extracted from images which contain the textual messages, and embedded in the cover image, thus strengthening the data hiding objective of steganography. They reported impressive results.

In the study titled "Image Steganography and Steganalysis based on LSB" as presented in [22], the authors proposed a new LSB steganography method which is an improvement on the 1-byte least significance method. The results they presented showed impressive performance of their model. This work however considered images alone, and could require some extension to be able to handle audio files as cover media.

3 The Proposed Audio Steganography System

Audio file format is a container format for storing audio data and metadata on a computer system. This format is divided into three major groups namely: the uncompressed audio formats, (e.g., WAV), formats with lossless compression, (e.g., WMA) and formats with lossy compression (e.g., MP3). In this research work, the selected cover media which is the audio signal undergoes different stages; the selected audio file irrespective of its format will be converted into .Wav (Wave File Audio) format using a function in MATLAB.

Many types of audio file are available, such as Windows Audio Visual (WAV), Windows Media Audio (WMA), and MPEG (MP3) exist. The type used in this study is WAV file format of type Pulse Code Modulation (PCM), because it is an uncompressed audio format, which gives more flexibility for data hiding. A stego-object (WAV file) with high sampling rate and sampling resolution may draw suspicion, because of its large size, especially if its subjective quality is not high. Usually, it is easy to hide more secret data in the high quality audio data (for example, the use of least significant bit encoding to embed one bit in each sample, consist of 16 bits, sample has less effect on the stego object than adding one bit in a sample consist of 8 bits). In the developed system the wave files, with 8-bit samples resolution, are used as cover media for hosting the secret data.

There are some basic parameters to understand when discussing on digital audio files. This parameter includes:

A. **Sampling rate (f_s)** is the number of samples of audio carried per seconds. It is measures in Hertz. In this research, 44100 Hz is used in wav audio format.
B. **Frame size:** The frame size is the amount of bits in each frame.

$$\text{Frame Size } = \frac{1}{\textit{Partitioned frame size}} \tag{1}$$

C. **Frequency Resolution (k_s):** It is the scaling factor, which ensures the cover medium is embedded below the audibility threshold. It is measured in Hertz. Human Auditory System (HAS) works dynamically in a wide range of frequencies between 20 Hz–20000 Hz. It is calculated using Eq. 2;

$$k_s = \frac{f_s}{2 \times \frac{N}{2}} \tag{2}$$

where f_s is the sampling rate and N is the total number of frames.

The proposed/developed audio steganography system is broadly divided into two modules which are: the Embedding Module and the Extraction Module.

A. The Embedding Module

Embedding module is a module at which the secret message (text file) is embedded into the selected cover medium (audio Signal). This is depicted in the Fig. 1.

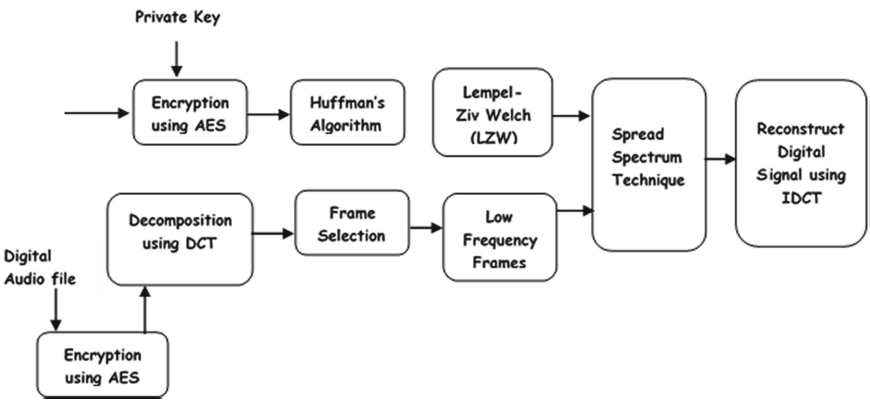

Fig. 1. The embedding module

The embedding module involves three processes which include: plain text (secret message) encryption and compression, audio signal decomposition as well as combined signal construction processes.

The secret message which is in plain text is encrypted in order to enhance the security of the secret message from the intruders by using a public key and private key to avoid unauthorized access of the text. This research work makes use of Advanced Encryption Standard (AES) over other encryption algorithms because it uses higher length key sizes such as 128, 192 and 256 bits for encryption, it uses 128-bit block size and also has 10, 12 or 14 rounds of bits depending on the key size used. Hence it makes AES algorithm more robust against hacking. The general encryption procedure is mathematically represented in Eq. 3;

$$E(K, M) = \{C\}K \tag{3}$$

where E represents the Encryption function, M denotes the Plain text (Secret message), K stands for Encryption Key, and C is the cipher text.

The encrypted text will be compressed using a two stage compression technique that uses the Huffman algorithm as well as the Lempel-Ziv Welch (LZW). The Huffman encoding algorithm starts by constructing a list of all the alphabet symbols in descending order of their probabilities. It then constructs, from the bottom up, a binary tree with a symbol at every leaf. This is done in steps, where at each step two symbols with the smallest probabilities are selected, added to the top of the partial tree, deleted from the list, and replaced with an auxiliary symbol representing the two original symbols. When the list is reduced to just one auxiliary symbol (representing the entire alphabet), the tree is complete. The tree is then traversed to determine the code words of the symbols. The C (Cipher text) in Eq. 5 would be subjected to compression using Huffman algorithm presented in equation.

The Lempel Ziv Welch algorithm is one of the many algorithms used for compression. It is typically used to compress certain text files, image files command among others. It relies on reoccurring pattern to save data space. Lempel ziv welch

compression was adopted over Huffman lossless data compression because its construct its dictionary on the fly, going through the data once that is when reading through the data for compression it goes through it once unlike Huffman that goes through it data twice. The Lempel-Ziv algorithm compresses by building a dictionary of previously seen string. It codes groups of characters of varying length.

Data compression is a method of encoding rules that allows substantial reduction in the total number of bits to store or transmit a file and this will help to reduce distortion in the stego file. The feature of both LZW (Lempel-Ziv-Welch) and Huffman algorithms are combined to improve the compression ratio. The Huffman algorithms is represented as;

$$b_{Huff} = \sum_{a=1}^{n} f(a_i) L(\{C\}) \tag{4}$$

where $L(\{C\})$ denotes the length of the cipher text, $f(a_i)$ is the word character and n is the number of bit.

Lempel-Ziv Welch is represented as;

$$c_{lzw} = b_{Huff}(\log_2 b + \log_2 N + 2) \tag{5}$$

where b_{Huff} represents the compressed cipher text using Huffman model, N denotes the fixed codes and c_{lzw} denotes the LZW compression.

The main advantage of this combined algorithm is that the percentage of data reduction increases more compared to the existing text compression techniques.

The audio format irrespective of its audio file format is converted into .wav format. This is achieved using an audio write function in MATLAB. The reason for this is that it will be easier to decompose the digital audio signal into analog signal using the One Dimensional Discrete Cosine Transform (DCT).

The DCT is one of the powerful compact transforms. It relocates most of the signal energy into the first transform coefficients, lesser energy or information is relocated into other (i.e., high frequency) coefficients. The frames thus created are queued based on the energies of the frames. DCT is applied on the voiced blocks that have power less than the predefined second threshold value (T′). The block size was taken small to avoid the high computational complexity of DCT calculations which makes the system slow. Thereafter, selection of low frequency frames was carried out. The selection of the low frequency frames from other frames is in order to ensure the secret message to be embedded does not introduce audible distortion in the audio signal;

$$f_{dct}(x) = \sum_{u=1}^{N-1} \alpha(u) c(u) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \tag{6}$$

for $x = 0, 1, \ldots\ldots\ldots\ldots N-1$

where;

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} for\ u = 0 \\ \sqrt{\frac{2}{N}} for\ u \neq 0 \end{cases} \tag{7}$$

where $f_{dct}(x)$ represents the original sequence of the audio signal, N denotes the Last frame in the audio file, x is the Number of frames in an audio file and u denotes the frame size.

Spread Spectrum technique was used to hide the encrypted and compressed secret message (Text file) into the digital Audio signal. Spread spectrum is a method by which energy generated in particular bandwidth is deliberately spread in the frequency domain, resulting in a signal with a wider bandwidth. Spread spectrum ($s_{sprectrum}$) systems encode data as a binary sequence which sounds like noise but which can be recognized by a receiver with the correct key. There are two types of spread spectrum techniques which are: Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS). In Direct Sequence Spread Spectrum, data to be transmitted is divided into small pieces and each piece is allocated to a frequency channel across the spectrum. In this research work, Frequency Hopping Spread Spectrum is used. In Frequency-Hopping Spread Spectrum, the audio file's frequency spectrum is altered so that it hops rapidly between frequencies. Spread spectrum combined the compressed text file with the low frequencies of the audio signal using;

$$s_{sprectrum} = c_{lzw}.\ f_{dct}(low) \tag{8}$$

The embedded signal is added to the other frames of high frequency using:

$$f_{frame}(t) = s_{sprectrum} + f_{dct}(high) \tag{9}$$

The analog signal generated is then converted into digital signal using the Inverse Discrete Cosine Transform (IDCT) as given in Eq. 10;

$$c_{dct}(u) = \alpha(u) \sum_{x=1}^{N-1} f_{frame}(t) \cos\left[\frac{\pi(x+1)u}{2N}\right] \tag{10}$$

where $c_{dct}(u)$ is the new audio signal (stego file).

B. The Extraction Module

This is the reverse of embedding module. At this stage, the secret message is extracted from the stego file (the embedded audio file). The stages to accomplish this are depicted in Fig. 2.

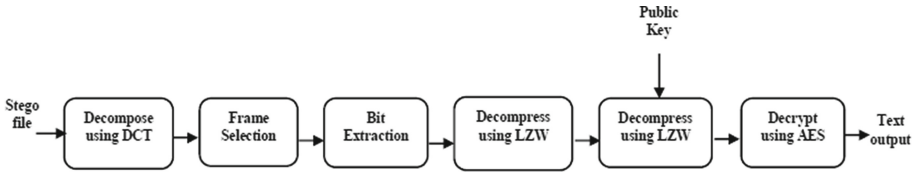

Fig. 2. The extraction module

The audio format irrespective of its audio file format is converted into .wav format. The reason for this is that it will be easier to decompose the wave file using the Discrete Cosine Transform (DCT) using the header of .wav file format. This is achieved using an audio write function in MATLAB.

At this stage, the stego file which is the embedded audio file is decomposed in to frames using one dimensional Discrete Cosine Transform (DCT).

$$c_{dct}(u) = \alpha(u) \sum_{x=1}^{N-1} f_{frame}(t) \cos\left[\frac{\pi(x+1)u}{2N}\right] \tag{11}$$

for $x = 0, 1, \ldots\ldots, N-1$

where;

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} for\ u = 0 \\ \sqrt{\frac{2}{N}} for\ u \neq 0 \end{cases} \tag{12}$$

where;

$f_{dct}(x)$ represents the original sequence of the audio signal, N denotes the last frame in the audio file, x stands for the number of frames in an audio file while, u denotes the frame size.

The decryption process is the process of converting the cipher text back to plain text. In addition, it has an edge over static compression methods because no dictionary or other pre-existing information is necessary for the decoding algorithm. Both encoding and decoding programs must start with same initial dictionary, in this scenario, all the 256 ASCII characters.

Here's how it works; The LZW decoder first reads in an index, looks up the index in the dictionary, and returns the substring associated with the index. The first character of this substring is appended to the current working string. This new concatenation is added to the dictionary. The decoded string then becomes the current working string (the current index, that is the substring, is remembered), and the process repeats.

For decomposition by Huffman Algorithm, this is the inverse of Huffman encoding module. To decode the encoded data, we require the Huffman tree. We iterate through the binary encoded data to find character corresponding to current bits, the following steps is used:

a. Start from the root and do the following until a leaf is found
b. If current bit is 0, we move to left node of the tree.

c. If the bit is 1, move to right node of the tree.
d. If during transversal, we encounter a leaf node, we print character of that particular leaf node and then again continue the iteration of the encoded data starting from step 1.

4 Results and Discussion

Implementation was done using MATLAB programming language on Windows 8.1 Operating System platform with hardware configuration of 3 GB RAM, 1.6 Hz Intel processor speed and 250 GB of hard disk.

In this developed system, three (3) different digital audio file formats were used as the cover media and different ranges of secret text were hidden into them for evaluation. The three different audio file formats used in the experiment include; .wav, .oga, and .m4a.

The stego files (embedded audio file) were evaluated. The stego file (audio file) retains its initial size after evaluating the proposed approach and the amount of information that the developed system can hide is very high (500 KB).

In order to cater for the evaluation of the research work, standard performance metrics as highlighted in Sect. 4.1 were used.

4.1 System Performance Evaluation Metrics

The proposed/developed system was evaluated based on the following metrics; *computational time*, *bit per character*, *compression ratio* and *signal to noise ratio*. These metrics are further highlighted as follows;

A. Compression Ratio

This is the ratio of the cover medium before and after the secret message is embedded into it to its ratio when the secret message is embedded into it.

$$\text{Compression Ratio} = \frac{\textit{output file size}}{\textit{input file size}} \tag{13}$$

From the result of experiment carried out in this research work, the system has a compression ratio of 1, which means the size of the audio file before the secret message is embedded into it is still the same size after the secret message is embedded into it.

B. Signal to Noise Ratio

Signal to noise ratio is a parameter used to know the amount by which the signal is corrupted by the noise. It is defined as the ratio of the signal power to the noise power.

Alternatively, it represents the ratio of desired signal (say a music file) to the background noise level. It is measured in decibel (db). SNR can be calculated by Eq. 14.

$$\text{SNR (db)} = 10 \log \frac{\sum_n I_n}{\sum n(E_n - I_n)^2} \tag{14}$$

where E_n = Stego file and I_n = Original Audio Signal

C. Computation Time

This is the time taken for the system to execute its function.

This times were recorded for the three (3) audio files used as cover-media in this experiment.

The columns of all table of records in this sub-section are labeled with letters A to J. This is to allow enough space for the tables to be captured. A = Plaintext Size (in kb), B = Audio cover file size (in mb), C = Length of Audio cover file (in minutes), D = Compressed Plaintext Size (in kb), E = Compression Ratio (in %), F = Entire System Operation time (in seconds), G = Signal-to-Noise ratio (in db), H = Bit Per Character, I = Stego Audio file size after embedding (in mb), and J = Extraction time.

4.2 System Performance Evaluation Results

The columns of all table of records in this sub-section are labeled with letters A to J. This is to allow enough space for the tables to be captured. A = Plaintext Size (in kb), B = Audio cover file size (in mb), C = Length of Audio cover file (in minutes), D = Compressed Plaintext Size (in kb), E = Compression Ratio (in %), F = Entire System Operation time (in seconds), G = Signal-to-Noise ratio (in db), H = Bit Per Character, I = Stego Audio file size after embedding (in mb), and J = Extraction time.

A. Using .Wav Audio Format as Cover File

For wav file, an audio size of 50.5 Mb and the length of 5 min were used for the evaluation. Table 1 shows the result of the evaluation using .wav file format.

Table 1. Results of experiment when .Wav audio file was used as cover

A	B	C	D	E	F	G	H	I	J
50	50.5	5	41	82.7	3	59.8	8	50.5	2
100	50.5	5	82	81.7	3	57.8	8	50.5	2
150	50.5	5	123	81.7	3	55.7	8	50.5	2
200	50.5	5	180	89.8	3	52.8	8	50.5	2
250	50.5	5	212	84.8	3	46.5	8	50.5	2
300	50.5	5	245	81.7	3	40.5	8	50.5	2
350	50.5	5	297	84.8	3	33.9	8	50.5	2
400	50.5	5	355	88.9	3	27.9	8	50.5	2
450	50.5	5	386	85.8	3	22.7	8	50.5	2
500	50.5	5	439	87.8	3	17.1	8	50.5	2

From Table 1, the values of the SNR are more than 50 db when the size of the text file to be embedded ranges from 50 kb to 200 kb, and this indicates that there will be no distortion in the stego file. But from 250 kb to 500 kb, the values of the SNR began to decrease; making the values to be less than 50 db and this shows that there will be distortion as the value decreases from 50 db. This can be represented graphically in Fig. 3 and Table 4.

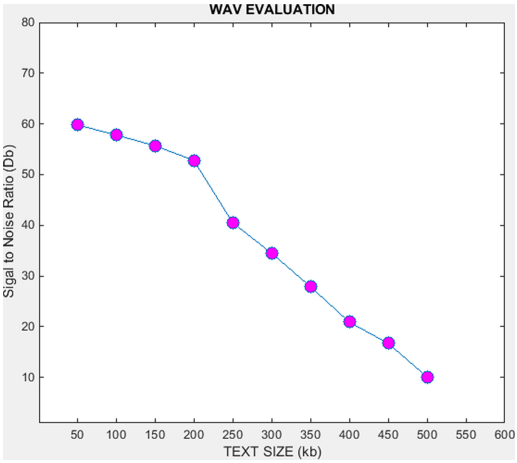

Fig. 3. Graph of WAV file format

B. Using the .OGA Audio Format as Cover

.OGA file format, with size 8.10 Mb and a length of 5 min was used as an experiment to perform the performance metrics. Table 2 shows the result of the evaluation.

Table 2. Results of experiment when OGA audio file was used as cover

A	B	C	D	E	F	G	H	I	J
50	8.10	5	45	89.8	6	56.4	8	8.10	4
100	8.10	5	83	82.7	6	54.7	8	8.10	4
150	8.10	5	132	87.8	6	51.5	8	8.10	4
200	8.10	5	168	83.8	5	44.1	8	8.10	4
250	8.10	5	227	90.8	6	39.6	8	8.10	4
300	8.10	5	272	90.9	6	31.0	8	8.10	4
350	8.10	5	290	82.7	6	28.8	8	8.10	4
400	8.10	5	364	91	6	19.5	8	8.10	4
450	8.10	5	409	90.9	6	13.9	8	8.10	4
500	8.10	5	450	89.8	6	8.8	8	8.10	4

From Table 2, the values of the Signal to Noise ratio are more than 50 db when the size of the text file to be embedded ranges from 50 kb to 100 kb, but the value of SNR for 150 kb is 48.5 which van be approximately to 50 db can also be accepted as good SNR vale. This indicates that there will be no distortion in the audio with a secrete message that's up to 150 kb. But from 200 kb to 500 kb, the values of the Signal to Noise Ratio began to decrease; making the values to be less than 50 db and this implicates that there will be distortion as the value decreases from 50 db. This can be represented graphically in Fig. 4.

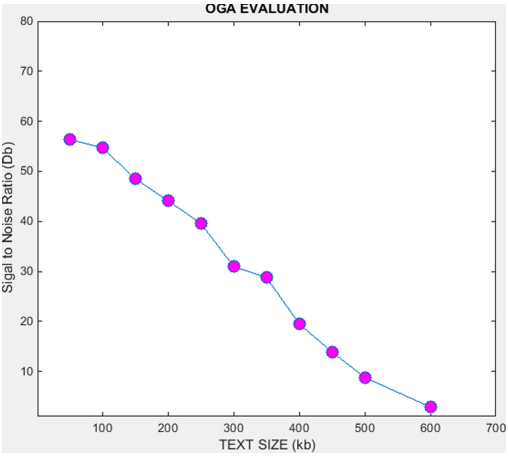

Fig. 4. Graph of OGA file format

C. Using the .M4A Audio Format as Cover

.M4A file format, with size 4.81 Mb and a length of 5 min was used in this experiment to evaluate the performance of the proposed system. The results of the evaluation are shown in Table 3.

Table 3. Results of experiment when M4A audio file was used as cover

A	B	C	D	E	F	G	H	I	J
50	4.81	5	43	14.2	9	59.7	8	4.81	7
100	4.81	5	89	11.1	8	55.9	8	4.81	6
150	4.81	5	132	12.2	8	51.6	8	4.81	6
200	4.81	5	163	18.3	8	49.8	8	4.81	6
250	4.81	5	225	10.1	8	42.6	8	4.81	6
300	4.81	5	273	9.1	8	37.6	8	4.81	6
350	4.81	5	307	12.2	8	31.7	8	4.81	6
400	4.81	5	355	11.2	8	28.7	8	4.81	6
450	4.81	5	400	11.1	8	22.0	8	4.81	6
500	4.81	5	424	15.2	8	18.6	8	4.81	6

From Table 3, the values of the Signal to Noise ratio goes beyond 50 db when the size of the text file to be embedded ranges from 50 kb to 100 kb, but the value of SNR for 150 kb is 48.6 which is approximately 50 db can also be accepted as good SNR value. This indicates that there will be no distortion in the audio with a secrete message that's up to 150 kb. But from 200 kb to 500 kb, the values of the Signal to Noise Ratio began to decrease; making the values to be less than 50 db and this implicates that there will be distortion as the value decreases from 50 db. This can be represented graphically in Fig. 5.

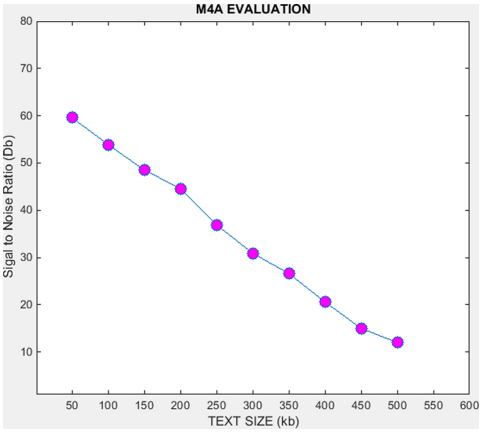

Fig. 5. Graph of M4A file format

Table 4. Results of the evaluated audio formats

Digital audio format	Maximum text sizewithout distortion (KB)	Average SNR (DB)	Average computation time (seconds)	Compression ratio
WAV	200	58.7	3	1
M4A	150	54.1	6	1
OGA	150	54.5	6	1

5 Performance of the New System Versus Existing Related Works

In this sub-section, we compared the performance of the current research work with other existing related ones in literature. Table 5 presents the details of this comparison.

As shown in Table 5 is the comparison of the developed model with some existing works in terms *Computation time*, *number of file formats*, *signal to noise ratio*, *compression ratio*, and *maximum text file that can be embedded without causing distortion in the cover media*. From Table 5, it is obvious that the current developed model

supports five (5) different audio file formats as compared to other existing systems on the table that support one (1) audio format. Besides, the developed model has better computational time per file size. The compression ratio of the developed model is one (1) making stego-files difficult to be discerned. It makes the quality of stego-file can be maintained and reduces the suspicion towards the stego-file. The developed model has minimal distortion at an average signal to noise ratio of 59.8 db at 200 kb text file size.

Table 5. General performance comparison with some existing works

Name and year of research	Average computational time (sec)	Maximum number of file format	Average signal to noise ratio (DB)	Compression ratio	Maximum text file size before distortion begins (KB)
Current Research work	2.2	5	59.8	1	200
[13]	–	1	54.5	1	12.2
[16]	0.3	1	52	>1	8.2
[17]	–	1	–	>1	27.5
[15]	–	1	56	–	40

6 Conclusion and Recommendation

There are a number of proven methods for applying steganography to hide information within audio data. In this research work, an audio steganography system that uses Discrete Cosine Transform (DCT) and spread spectrum techniques was developed. It was shown through implementation and subjective experimentation that the proposed audio steganography system supports three different digital audio formats (WAV, OGA, and M4A). The system developed has the ability to embed a secret message of size that is up to 200 kb with respect to any digital audio length or size without any distortion and has the ability to retain the same size after embedding text into it.

Furthermore, the Wav, and M4A files have the SNR ratio that's greater than or approximately equal to 50 db at 200 kb. This implies that large file of size up to about 200 kb can be hidden without causing audible distortion in the cover medium.

However, the SNR value of OGA audio file format exceeds 50 db once the payload (secret text to be hidden) gets above 150 kb in size. This means that the highest text file that can be embedded into these files without any distortion is 150 kb.

This current work have been able to develop a robust stenographic system that would be very useful in securing and sharing large amount of sensitive data or information without arousing suspicion. The developed system is therefore recommended for security agencies and other organization that consider information security as being of uttermost priority. This system is a useful means for transmitting covert battlefield information via an innocuous cover audio signal.

For future work, the use of post-quantum crypto schemes to strengthen audio steganography systems could be considered.

References

1. Kuboye, B.M., Gabriel, A.J., Thompson, A.F., Joseph, V.O.: Analysis of algorithms in long term evolution (LTE) network. J. Comput. Sci. Appl. **25**(2), 59–71 (2018)
2. Alabi, O., Thompson, A.F., Alese, B.K., Gabriel, A.J.: Cloud application security using hybrid encryption. Commun. Appl. Electron. **7**(33), 25–31 (2020)
3. Gabriel, A.J., Alese, B.K., Adetunmbi, A.O., Adewale, O.S., Sarumi, O.A.: Post-quantum crystography system for secure electronic voting. Open Comput. Sci. **9**, 292–298 (2019)
4. Gabriel, A.J., Egwuche, S.O.: Modelling the employees' activities of public service sector using production rules. Anale. Seria Informatică **13**(2), 65–68 (2015). Vol. XIII fasc. 2 – 2015 Annals. Computer Science Series, University of Timisoara, Romania
5. Iwasokun, G.B., Egwuche, O.S., Gabriel, A.J.: Neural network-based health personnel monitoring system. Afr. J. Comput. ICT **8**(1), 79–87 (2015)
6. Alese, B.K., Gabriel, A.J., Olukayode, O., Daramola, O.A.: Modelling of risk management procedures for cybercrime control systems. In: Proceedings of the World Congress on Engineering (WCE 2014), London, United Kingdom (2014). ISBN 978-988-19252-7-5
7. Agboola, A.H., Gabriel, A.J., Aliyu, E.O., Alese, B.K.: Development of a fuzzy logic based rainfall prediction model. Int. J. Eng. Technol. **3**(4), 427–435 (2013). ISSN 2049-3444
8. Daodu, M.N., Gabriel, A.J., Alese, B.K., Adetunmbi, A.O.: A data encryption standard (DES) based web services security architecture. Ann. Comput. Sci. Ser. **14**(2), 53–58 (2016)
9. Gabriel, A.J. Alese, B.K., Adetunmbi, A.O., Adewale, O.S.: Post-quantum crystography; a combination of post-quantum cryptography and steganography. In: The 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), Technically Co-Sponsored by IEEE UK/RI Computer Chapter, London, UK, 9–12 December 2013, pp. 454–457 (2013)
10. Gabriel, A.J., Alese, B.K., Adetunmbi, A.O., Adewale, O.S.: Post-quantum crystography based security framework for cloud computing. J. Internet Technol. Secur. Trans. (JITST) **4** (1), 351–357 (2015)
11. Wheeler, D. Johnson, D., Yuan, B., Lutz, P.: Audio steganography using high frequency noise introduction. Thomas Golisano College of Computing & Information Sciences Rochester Institute of Technology, Rochester, NY, RIT Scholar Works (2012). http://scholarworks.rit.edu/other/302. Accessed 14 Sept 2018
12. Ghanwat, D., Rajan, R.S.: Spread spectrum-based audio steganography in the transformation domain. Glob. J. Adv. Eng. Technol. **2**(4), 66–77 (2013)
13. Olanrewaju, R.F., Othman-Khalifa, H.A., Suliman, R.: Increasing the hiding capacity of low-bit encoding audio steganography using a novel embedding technique. World Appl. Sci. J. **21**(26), 79–83 (2013)
14. Kresnha, P.E., Mukaromah, A.: A robust method of encryption and steganography using ElGamal and spread spectrum technique on MP3 audio file. In: Proceeding of Conference on Application of Electromagnetic Technology, vol. 3, no. 9, pp. 11–15 (2014)
15. Kumari, L., Goyal, D., Gyan, S.: Analysis and design of three LSB techniques for secure audio steganography. Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET) **2**(2), 44–55 (2013)
16. Pratibha, A.N., Belagali, P.P.: Data concealing using speech steganography. Int. Res. J. Eng. Technol. (IRJET) **3**(11), 55–66 (2016)
17. Sakthisudhan, K., Prabhu, P., Thangaraj, P.: Secure audio steganography for hiding secret information. Int. J. Comput. Appl. **7**(8), 55–87 (2012)

18. Balgurgi, P.P., Jagtap, S.K.: Audio steganography used for secure data transmission. In: Aswatha Kumar, M., Selvarani, R., Suresh Kumar, T.V. (eds.) Proceedings of International Conference on Advances in Computing. AISC, vol. 174, pp. 699–706. (2013). https://doi.org/10.1007/978-81-322-0740-5_83
19. Adeboje, O.T., Adetunmbi, A.O., Gabriel, A.J.: Embedding text in audio steganography system using advanced encryption standard and spread spectrum. Int. J. Comput. Appl. **177** (41), 46–51 (2020). https://doi.org/10.5120/ijca2020919. (0975-8887)
20. Snasel, V., Kromer, P., Safarik, J., Platos, J.: JPEG steganography with particle swarm optimization accelerated by AVX (2019). https://doi.org/10.1002/cpe.544. Wiley: https://onlinelibrary.wiley.com
21. Chatterjee, A., Ghosal, S.K., Sarkar, R.: LSB based steganography with OCR: an intelligent amalgamation. Multimed. Tools Appl. **79**, 11747–11765 (2020). https://doi.org/10.1007/s11042-019-08472-6
22. Rachael, O., Misra, S., Ahuja, R., Adewumi, A., Ayeni, F., Mmaskeliunas, R.: Image steganography and steganalysis based on least significant bit (LSB). In: Singh, P.K., Panigrahi, B.K., Suryadevara, N.K., Sharma, S.K., Singh, A.P. (eds.) Proceedings of ICETIT 2019. LNEE, vol. 605, pp. 1100–1111. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-30577-2_97