



Full length article

A new method for hiding a secret file in several WAV files depends on circular secret key

Aamer Tahseen Suhail, Harith Ghanim Ayoub *

Ninaveh Technical Institute, Northern Technical University, Mosul, Iraq



ARTICLE INFO

Article history:

Received 2 March 2022

Revised 13 June 2022

Accepted 21 June 2022

Available online 19 July 2022

Keywords:

Information hiding

Encryption

Circular secret key

Extracting

Cover file

WAV

LSB (Least Significant Bit)

ABSTRACT

The technique of information hiding is very important in achieving security of important and sensitive data, especially those that are transmitted through various digital communication channels because it has robust characteristics that distinguish it from encryption techniques, the most important of which is that it is not perceived by protrusive and hackers, as it is based on the principle of hiding those data inside other digital media as a carrier cover. On the other hand, with the development of hiding techniques, techniques developed to try extracting its hiding to hacking its contents, for this reason and to increase the efficiency of these methods, the encryption and hiding techniques was combined together.

The proposed method depends on distributing secret data on several digital audio files of type WAV in random way depends on circular secret key values that is generated in random order that be agreed upon between the sender and the receiver. This technique has achieved high durability, and in the event of suspicion of the existence of secret content within any WAV file, the multiplicity of carrier files and their different length, as well as random distribution of data in them, all these characteristic eliminates the suspicion of any confidential content, as well as complicates the file retrieved process if the suspicion be occurs.

© 2022 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Computers and Artificial Intelligence, Cairo University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

With the Internet developing, the users started to share, transmit, and distribute their private information one to others through various communication channels. As a result, protecting these information became a critical issue for everyone. Information hiding and cryptography are two security technique that provide secret information confidentiality [1,2]. In essence, cryptography encrypts the secret information so that it cannot be understood by eavesdroppers; whereas the hiding conceals it into another media as a cover file [3,4]. For this reason, information hiding is frequently viewed as a stealthy approach for transferring impor-

tant data in total secrecy via public channels in such a way that no one may know about the communication except the transformation parties, namely the sender and receiver [3,5]. Basically, hiding information in audio files is a type of steganography that hides digital data into digital audio files as a carrier such WAV, MP3, and WMA files without damaging the contained of this uninteresting audio file, so that it cannot be seen by eavesdroppers [6,7,8]. An encrypted file would immediately imply a secret communication to a third party. A hidden file, on the other hand, would not attract attention and so would not arouse suspicions, nor would changing its size [9,10,11].

Hiding and encryption techniques are both used to achieve information security and prevent unauthorized from realizing its content, but there is a difference between them, which is that the encryption does not eliminate the principle of suspicion in the data, while the coverage it is achieves this because it is not perceived. Therefore, the hiding technique has become of great importance in achieving the confidentiality of important and sensitive files, especially those that are transmitted through various communication channels [1,4,5].

With the advancement of the hiding techniques, and because it has become very common after achieving the efficiency and

* Corresponding author.

E-mail addresses: aamir@ntu.edu.iq (A. Tahseen Suhail), harithga@ntu.edu.iq (H. Ghanim Ayoub).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.



Production and hosting by Elsevier

desired its goals, the suspicion has come to those protrusive and hackers in any file that exchanged through the different channels, especially those that exchanged between important parties such as security governmental, organizations and financial institutions. Therefore, to prevent such intruders from attempting to benefit from the content of that files has become insufficient to use the pure methods of file hiding, that is, simply by sequential and regular hiding of it inside this carrier covers. Rather, it is necessary to find common hybrid technique and algorithms that mix between encryption and hiding techniques together to increase the efficiency of hiding technique and archive higher durability of the used methods [8,12,13].

The insertion of confidential data into the cover file should not cause any visible modifications to the secret media, so that the authenticity of the file should not disturbed. The goal of the audio hiding view is to ingrain valuable secret information into an audio file in such a way that the human auditory system (HAS) is unable to identify the change caused by the data ingrain. In the audio hiding, Least Significant Bit (LSB), Echo and Spread spectrum hiding approaches along with other current applications that have been developed in recent years [14,15,16,17].

The goal is to embed the secret data into a WAV file so that the difference between the original and embedded files is unnoticeable. Here when applying hiding, it must be ensured that the format of the file is not be changed by tampering with the content of the header of the file located in the first 44 bytes of it because of the header gets corrupted and the audio file will also corrupt [7,18].

In general, the most important criteria of any efficient hiding technique on any WAV. file are [4,8,19,20]:

- a- Robustness and resilience of strength hiding with inability to extract its contents by the attacker, especially in the event of suspicion of this.
- b- Imperceptibility and unexpected which meaning that no perceived change or distortion takes place that carries it after the hiding process.
- c- Survivability and resilience in front of attempts to recover the hiding in the event of suspicion among the attackers.
- d- The largest percentage of the data that can be hidden in cover file depends on its size with no any distortion can be feel it in the features of the cover after the hiding.

2. Different types of coverage media

Generally, the majority of formats for computer digital files can be used as covers, but the most suitable files that achieve the highest percentage of hiding compared to its size, which can be defined as the amount of bits in the digital entity that can be manipulated or replaced without causing any distortion can be realized. All image, sound, video file types are the suitable media to achieves these mission [2,7,13].

2.1. Hiding in audio files

Sound files, like image files, can be modified in such a way that they contain hidden information, and those modifications must be done in such a way that it should be imperceptible from others. Therefore, the methods that embeds data in sound files must use the properties of the Human Auditory System (HAS). The HAS can identify disturbances in a sound file as well as additive random noise.

However, there are several flaws that we can exploit. [10,19,20].

Basically, Audio steganography is a sort of digital steganography in which digital data is hidden in digital audio files like WAV, MP3, and WMA [9,11]. Audio steganography takes advantage of the

human auditory system's (HAS) inability to detect small differences in audio frequencies in the case of simple manipulation of the least significant bits of audio samples; thus, these bits can be used to hide secret data without compromising the audio file's quality or changing its size [8,10].

2.2. WAVE file format

One of the most popular file formats for digital sounds is the **Windows Audio Visual (WAV)** and the **Audio Interchange File Format (AIFF)**. The quantization converts each input sample value into one of a discrete value, and the number of repeated sound samples during one second represents the sound frequency or sampling rates. The popular sampling rates for sound include 8 kHz, 9.6 kHz, 10 kHz, 12 kHz, 16 kHz, 22.05 kHz and 44.1 kHz. Which mean frequently samples per second and each sample represented by one or two bytes of one or two channels (mono or stereo)[6,19].

2.3. Hide in the least significant bit (LSB)

It is important to know that in order to manipulate any of the bytes of the WAV file, the first 44 bytes must be exceeded from the beginning of the audio file, as it contains the header of it, which includes all the information of that file, and any manipulation or change in its values will lead to corruption [11,12].

Audio files are competent to convey hidden information due of their popularity and abundance. As a result, several researchers have begun to look at how audio signals and properties might be exploited in the domain of information concealment [8]. Least Significant Bit [8,9], Echo concealing [14,15], Hiding in Silence Interval [21], Phase Coding [14,15], Amplitude Coding [15,17,18], Spread Spectrum [17,18], and Discrete Wave Transform [15,16] are some of the most prominent methods.

Basically, the LSB approach is based on hiding each bit from the secret data in the carrier WAV file's rightmost bits of every audio sample. The LSB approach takes use of the HAS, which is unable to detect small changes in audio frequencies on the high frequency portion of the auditory spectrum. The LSB approach allows for a high embedding rate without sacrificing audio quality. It's also relatively successful and simple to put into practice [11,16,19]. However, the main disadvantage is that the secret data are hidden in a predictable manner, making them easy to recover by attackers, and to bypass this weak point, encryption of the secret file to be hidden is resorted to by one of the encryption methods before the completion of the hiding process, or by using the proposed method that will be detailed in this paper.

3. Suggested method

The proposed method is intended for use with 16-bit uncompressed digital audio files like WAV files. In a 16 bit WAV file, each sample is made up of two bytes that represent the amplitude of the audio sample [16]. The technique hides the secret data using two LSBs in each of these audio samples, resulting in a hiding capacity of two bits out of 16 bits. or 2/16 of the total size of the carrier audio file ($2/16 = 0.125 = 12.5\%$).

The art of hiding and its algorithms has proven its efficiency in the transmission of confidential files for its enjoyment of those characteristics that encryption lacks. However, on the other hand, techniques and counter methods have emerged to decode and extract that hiding. With the development of the use of hiding techniques, suspicion has become among workers in the field of piracy and techniques for extracting any of the media. It can be used as carrier cover for hiding of audio files. And to miss the opportunity for the abusers hackers and intruders, the trend was

towards finding methods of merging and combining more than one technique to achieve this [22,23].

A method has been proposed for distributing secret messages within several WAV files and sending them as separate files, depending on algorithm that is agreed upon between the sender and the receiver.

The proposed algorithm is based on a technique for selecting several audio files of type WAV as carrier covers for important secret files, provided that those files are in sizes whose sum is at least greater or equal to eight times the size of the secret file required to be hidden because of the replacing will take place in the 2 least significant bits of 16 bits for each sample.

After that, the circular secret key is generated, whose numbers are represent the number of WAV files selected as carrier covers for hiding. For example if five WAV files carriers, the long of that key will be five in size.

3.1. Circular secret key and its characteristic

The circular secret key contains the numbers of the selected WAV files as covers of hiding which are arranged randomly, for example, if there are five files, then the key will be five in length, which are the numbers of those files, which must be inserted randomly and not sequentially. For example, if the file numbers are (1, 2, 3, 4, 5), a random key arrangement is created as in Fig. 1.:

To increase the efficiency of secret key we repeat the file numbers on one cycle of key by doubling it twice or more in randomly sequence, this can increase the randomly of the message distribution, as well, as in Fig. 2. For example.

The selected covers are different of sizes and not equal. Therefore, the repetition percentage of its numbers on the key are not equal, then, this percentage of repetition of each cover number is depends on its file size to the total sizes of the all, see Table 1. Below for example of five cover files:-.

From the above table, it becomes clear that the secret key has a length of 20 numbers, represented the five files which repeated according to ratio of each file size to total size. Therefore, the number of first cover is repeated two times in one cycle of the secret key while the number of second cover is repeated four times, the third one time, the fourth is repeated six times and the last one repeated seven times during one cycle, then those numbers used to generate the secret key of length 20 in random way to increase efficiency see Fig. 3. and Table 2. below, as example. Table 3.

The secret key must be send in an independent form with a specific code to be agreed upon between the sender and the receiver. Then the receiver check that secret key of the number of WAV files containing the hidden message, then well be is distinguished from many of the WAV files received based on a hidden code of its number.

3	1	5	2	4
---	---	---	---	---

Fig. 1. Sample of circular secret key of five values in random distribution.

4	2	3	5	4	1	3	1	5	2
---	---	---	---	---	---	---	---	---	---

Fig. 2. Sample of circular secret key of five values, the number of files repeated tow times randomly in one cycle.

Table 1

Example of deferent covers sizes.

Number of file iteration	Percentage of file	cover size	Cover
in one cycle of the key	size to total sizes	In Byte	number
2	10%	15,000	1
4	20%	30,000	2
1	5%	7,500	3
6	30%	45,000	4
7	35%	52,500	5
20(key length)	100%	1,50,000	Total

3.2. Characteristics of selected covers

The selected WAV files as a cover caring a hidden message characterized by:

a) As the hiding takes place in the two least significant bits of each WAV file sample, the percentage of hiding achieved will be 1/8 of the file size, then the total size of the files (without its headers length (44bytes)) should be at least equal to eight times the size of the secret file to be hidden, see Eqs. (1) and (2) in the following:

$$\sum_{i=1} (cover\ No.\ [i]) - (n * 44) \geq (size\ of\ secret\ file * 8) \quad (1)$$

or it can be checked as:

$$\sum_{i=1} Size\ of\ secret\ file \leq \sum (cover\ No.\ [i] - 44) / 8 \quad (2)$$

$n =$ number of WAV files.

b) The selected WAV files should be selected in different sizes to increase the efficiency of the method.

c) Must be hide a code in each cover file to indicates that these file carry hidden message, as well as hide the number of that file in certain locations of the carrier file to be agreed between the sender and the receiver, in order to know which of the file carry a hidden message and its number within the secret key, as a guide in hiding and extracting process.

d) The carrier file numbers not necessary be sequential but it better to be random.

e) To increase the efficiency of this method, it is preferable to send carrier files of hiding in different paths or at different times to prevent attempts in the case of discovery and to avoid the doubt.

e) The higher number of cover files lead to higher efficiency of the method, because the size of the circular secret key will be greater, and To ensure that the secret message is dispersed as much as possible.

f) The number of bits allocated to represent the numbers of file does not have to be eight (one byte), but can be based on the number of files selected, for example, if the number of files are four, it is 2bits are enough to represent it as (00, 01, 10, 11), while if the number of 8 files we need 3bits to represent each one (000, 001, 010, 011, 100, 101, 110, 111). Therefore, the size of the secret key will be determined by the required bytes, so in the first example, the secret key can be represented by only one byte (00011000) or can be represented by three bytes in the second example (000001010011100101110111), and so on.

5	4	2	1	5	2	4	3	5	4	1	5	4	5	5	4	4	2	2	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Fig. 3. Generation of circular secret key of five different size covers in one cycle.

Table 2
Number of file iterations.

Number of file iterations	file number
2	1
4	2
1	3
6	4
7	5

3.3. Properties of secret file

The proposed method depend on technique of converting the secret file into a stream of bits before the hiding process, to be hidden in the form of sequential bits, therefore, it does not require a specific type of file to hide, so it is valid for any type of digital file.

3.4. Hiding algorithm

After determine the WAV files that selected as covers of hiding and creating the circular secret key, the hiding process is performed by taking one number of circular secret key each time to indicate the WAV file number in which a bit of the secret file to be hidden in it after converting it to a stream of bits. When all circular key values are finished, the loop is repeated by rotating the process, and the technique of hiding detailed by the following steps:

Step 1: Determine the size of the secret file to hide, with adding two bytes at beginning to represent its size.

Step 2: Convert the secret file to a sequence of bits stream.

Step 3: Select the WAV files to be used as covers.

Step 4: Calculate the total WAV file sizes.

Step 5: Check whether summation of cover files sizes are greater than or equal to eight times the size of a secret file, if not, another cover file will be added or replacing one or more of it with the larger ones.

Step 6: Hide an agreed code between the sender and the receiver indicate that the file is carrying a hidden information, as well as hide the cover file number.

Step 7: Generate the circular secret key based on the numbers of covers taking into account that each cover should be repeated as much as its size to the rest of the others.

Step 8: Read a value from the secret key to specify the number of the cover to be hide.

Step 9: Hide 2bits of secret file indicated in the two least significant bits of the selected sample from the selected cover.

Step 10: Shifting the pointer of the this cover in which you hide is skipped.

Step 11: If all bits of the secret file have been hidden go to **The end**, otherwise: -.

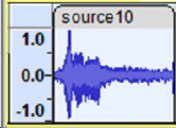
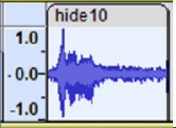
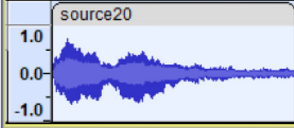
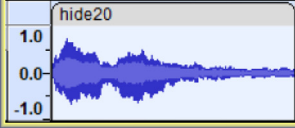
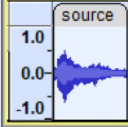
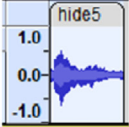
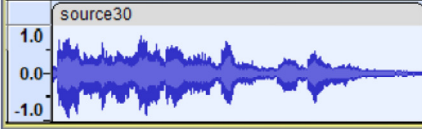
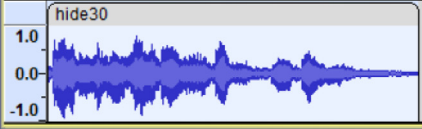
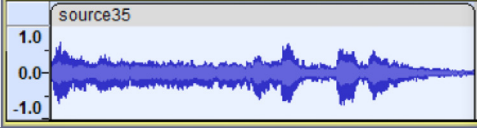
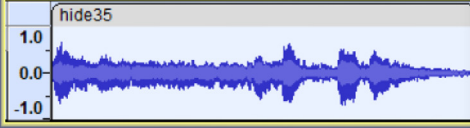
Step 12: Shifting the secret file pointer to the next 2bits.

Step 13: Shifting the secret key pointer to the next value.

Step 14: Go to **step 8**.

The end.

Table 3
Figures of WAV files before and after hiding.

Source files	Hiding files	File No.
		1
		2
		3
		4
		5

The detail flowchart of this hiding algorithm is in Fig. 4:

3.5. Extracting algorithm

The process of extract hiding from covers is a completely opposite process to what was done in the stage of hiding. After determining the WAV files which contained the secret information and specifying their numbers within the secret key agreed upon between the sender and the receiver, the details of the extract hiding process are shown in the following algorithm steps:

- Step 1: Reading the circular secret key to determine the numbers and sequence of WAV files that contained the hiding.
 - Step 2: Identification the covers that contained the hiding from the collection of files received by checking the existence of the secret code, as well as determine the number of each cover.
 - Step 3: Create a counter to extract the 16 bits of secret file size..
 - Step 4: Reading a value of the circular secret key to determine the number of the cover to be extract from it.
 - Step 5: Read the two least significant bits of the indicator from cover sample.
 - Step 6: Shifting the current cover pointer to the next sample, and shifting the circular secret key pointer to the next value.
 - Step 7: Increase the value of the counter by two.
 - Step 8: Did the value of the counter equal to 16, if so, the 16 bits extracted represent the size of the hidden file, otherwise refer to **step 4**.
 - Step 9: Create a new counter starting at zero and ending with eight times the value of secret file size.
 - Step10:Reading a value from the circular secret key to determine the number of the cover to be extracted from hiding.
 - Step 11: Read the two least significant bits of the sample pointed from the cover.
 - Step 12:Shifting the current cover pointer to the next sample and shifting the circular secret key pointer to the next value.
 - Step 13: Increase the counter value by two.
 - Step 14: If the value of the counter not equal to eight times size of the hidden file, repeat the operation from **step 10**.
 - Step 15: Collect the bits of the extracted file by converting every eight bits to byte to review the secret file.
- The end.

The details of the extracting process are shown in the following flowchart in Fig. 5:

4. Result discussion

The method was applied by selecting five WAV files of different sizes as a covers and distributing the bits of the secret message after converting it to a binary stream which indicated below, based on the suggested secret key values (4,5,2,1,4,5,3,4,5,2,5,4,5,4,5,4,2,1,2).

4.1. The secret message

“ The technique of information hiding is very important in achieving security of important and sensitive data, especially those that are transmitted through various digital communication channels because it has robust characteristics that distinguish it from encryption techniques, the most important of which is that it is not perceived by protrusive and hackers, as it is based on the principle of hiding those data inside other digital media as a carrier cover. On the other hand, with the development of hiding techniques, techniques developed to try extracting its hiding to hacking its contents, for this reason and to increase the efficiency of these

methods, the encryption and hiding techniques was combined together.”.

4.2. The binary stream of the secret message

```

00100000 00100000 01010100 01101000 01100101 00100000
01110100 01100101 01100011 01101000 01101110 01101001
01110001 01110101 01100101 00100000 01101111 01100110
00100000 00100000 01101001 01101110 01100110 01101111
01110010 01101101 01100001 01110100 01101001 01101111
01101110 00100000 01101000 01101001 01100100 01101001
01101110 01100111 00100000 01101001 01110011 00100000
01110110 01100101 01110010 01111001 00100000 01101001
01101101 01110000 01101111 01110010 01110100 01100001
01101110 01110100 00100000 01101001 01101110 00100000
01100001 01100011 01101000 01101001 01100101 01110110
01101001 01101110 01100111 00100000 01100111 01100101
01100011 01110101 01110010 01101001 01110100 01111001
00100000 01101111 01100110 00100000 01101001 01101101
01110000 01101111 01110010 01110100 01100001 01101110
01110100 00100000 01100001 01101110 01100100 00100000
01110011 01100101 01101110 01110011 01101001 01110100
01101001 01110110 01100101 00100000 01100100 01100001
01110100 01100001 00101100 00100000 01100101 01110011
01110000 01100101 01100011 01100101 01101001 01110100
01101100 01111001 00100000 01110100 01101000 01101111
01110011 01100101 00100000 01110100 01101000 01100001
01110100 00100000 01100001 01110010 01100101 00100000
01110100 01110010 01100001 01101110 01110011 01101101
01101001 01110100 01110100 01100101 01100100 00100000
01110100 01101000 01110010 01101111 01110101 01100111
01101000 00100000 01110110 01100001 01101100 00100000
01100011 01101111 01101101 01101101 0110101 01100111
01101001 01100011 01100001 01110100 01101001 01101111
01101110 00100000 01100011 01101000 01100001 01101110
01101110 01100101 01101100 01110011 00100000 01100010
01100101 01100011 01100001 01110101 01110011 01100101
00100000 01101001 01110100 00100000 01101000 01100001
01110011 00100000 01110010 01101111 01100010 01110101
01110010 01100001 01100011 01110100 01100101 01110010
01101001 01110011 01110100 01101001 01100011 01110011
00100000 01110100 01101000 01100001 01110100 00100000
01100100 01101001 01110011 01110100 01101001 01101110
01100111 01110101 00100000 01110011 01101000 00100000
01101001 01110100 00100000 01100110 01110010 01101111
01101101 00100000 01100101 01101110 01100011 01110010
01111001 01110000 01110100 01101001 01101111 01101110
00100000 01110100 01100101 01100011 01101000 01101110
01101001 01110001 01110101 01100101 01110011 00101100
00100000 01110100 01101000 01100101 00100000 01101101
01101111 01110011 01110100 00100000 01101001 01101101
01110000 01101111 01110010 01110100 01100001 01101110
01110100 00100000 01101111 01100110 00100000 01101111
01101000 01101001 01100011 01101000 00100000 01101001
01110011 00100000 01110100 01101000 01100001 01110100
00100000 01101001 01110100 00100000 01101001 01110011
00100000 01101110 01101111 01110100 00100000 01110000
01100101 01110010 01100011 01100101 01101001 01110110
01100101 01100100 00100000 01100010 01111001 00100000
01110000 01110010 01101111 01110100 01110010 01110101
01110011 01101001 01110110 01100101 00100000 01100001
01101110 01100100 00100000 01101000 01100001 01100011
01101011 01100101 01110010 01110011 00101100 00100000

```

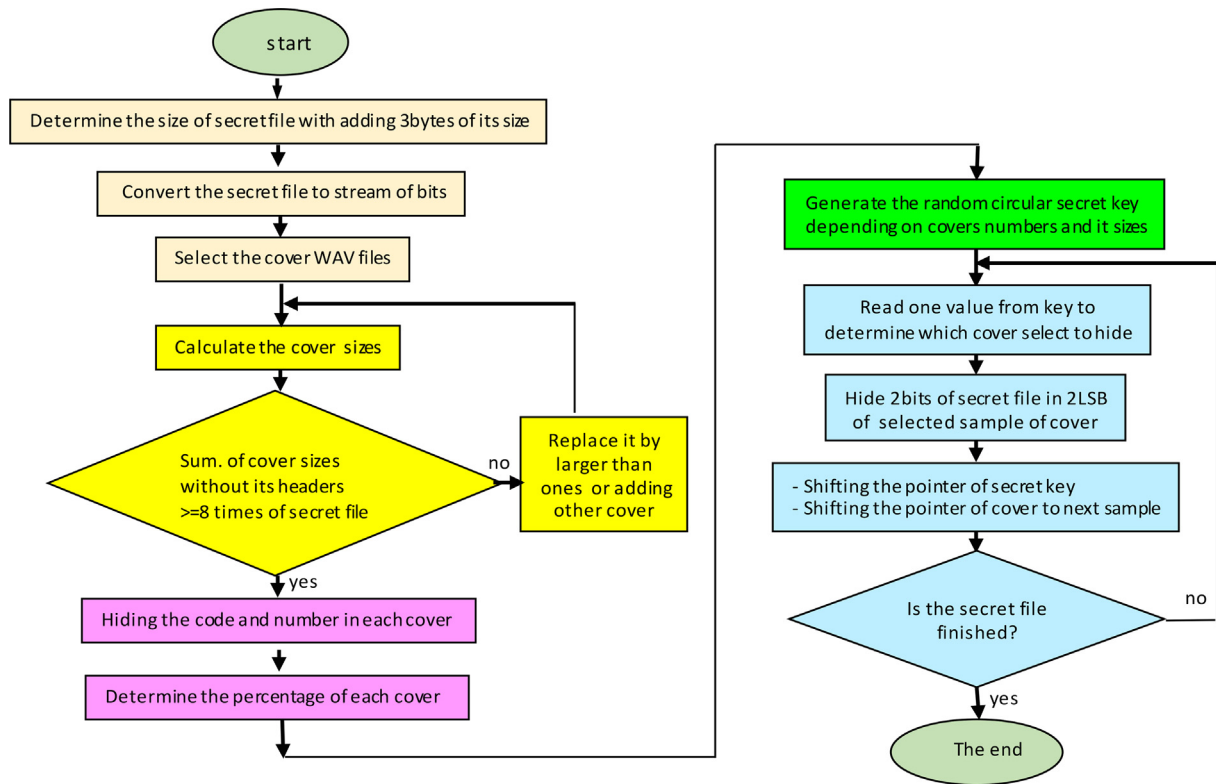



Fig. 4. Flowchart of the hiding.

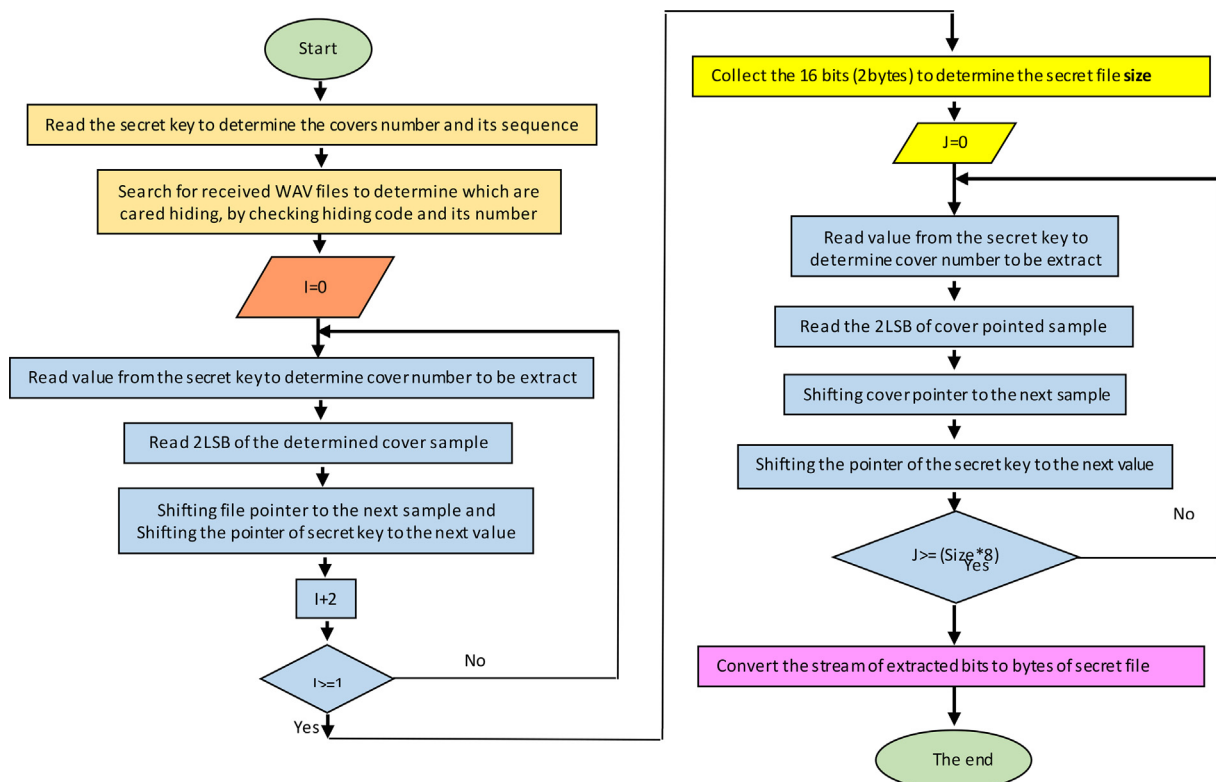


Fig. 5. Flowchart of the extracting.

```

01100001 01110011 00100000 01101001 01110100 00100000
01101001 01110011 00100000 01100010 01100001 01110011
01100101 01100100 00100000 01101111 01101110 00100000
01110100 01101000 01100101 00100000 01110000 01110010
01101001 01101110 01100011 01101001 01110000 01101100
01100101 00100000 01101111 01100110 00100000 01101000
01101001 01100100 01101001 01101110 01100111 00100000
01110100 01101000 01101111 01110011 01100101 00100000
01100100 01100001 01110100 01100001 00100000 01101001
01101110 01110011 01101001 01100100 01100101 00100000
01101111 01110100 01101000 01100101 01110010 00100000
01100100 01101001 01100111 01101001 01110100 01100001
01101100 00100000 01101101 01100101 01100100 01101001
01100001 00100000 01100001 01110011 00100000 01100001
00100000 01100011 01100001 01110010 01110010 01101001
01100101 01110010 00100000 01100011 01101111 01101110
01100101 01110010 00101110 00100000 01001111 01101110
00100000 01110100 01101000 01100101 00100000 01101111
01110100 01101000 01100101 01110010 00100000 01101000
01100001 01101110 01100100 00101100 00100000 01110111
01101001 01110100 01101000 00100000 01110100 01101000
01100101 00100000 01100100 01100101 01110110 01100101
01101100 01101111 01110000 01101101 01100101 01101110
01110100 00100000 01101111 01100110 00100000 00100000
01101000 01101001 01100100 01101001 01101110 01100111
00100000 01110100 01100101 01100011 01101000 01101110
01101001 01110001 01110101 01100101 01110011 00101100
00100000 01110100 01100101 01100011 01101000 01101110
01101001 01110001 01110101 01100101 01110011 00100000
01100100 01100101 01110110 01100101 01101100 01101111
01110000 01100101 01100100 00100000 01110100 01101111
00100000 01110100 01110010 01111001 00100000 01100101
01111000 01110100 01110010 01100001 01100011 01110100
01101001 01101110 01100111 00100000 01101001 01110100
01110011 00100000 01101000 01101001 01100100 01101001
01101110 01100111 00100000 01110100 01101111 00100000
01101000 01100001 01100011 01101011 01101001 01101110
01100111 00100000 00100000 01101001 01110100 01110011
00100000 01100011 01101111 01101110 01110100 01100101
01101110 01110100 01110011 00101100 00100000 01100110
01101111 01110010 00100000 01110100 01101000 01101001
01110011 00100000 01110010 01100101 01100001 01110011
01101111 01101110 00100000 01100001 01101110 01100100
00100000 01110100 01101111 00100000 01101001 01101110

```

```

01100011 01110010 01100101 01100001 01110011 01100101
00100000 01110100 01101000 01100101 00100000 01100101
01100110 01100110 01101001 01100011 01101001 01100101
01101110 01100011 01111001 00100000 01101111 01100110
00100000 01110100 01101000 01100101 01110011 01100101
00100000 01101101 01100101 01110100 01101000 01101111
01100100 01110011 00101100 00100000 01110100 01101000
01100101 00100000 01100101 01101110 01100011 01110010
01111001 01110000 01110100 01101001 01101111 01101110
00100000 01100001 01101110 01100100 00100000 01101000
01101001 01100100 01101001 01101110 01100111 00100000
01110100 01100101 01100011 01101000 01101110 01101001
01110001 01110101 01100101 01110011 00100000 01110111
01100001 01110011 00100000 01100011 01101111 01101101
01100010 01101001 01101110 01100101 01100100 00100000
01110100 01101111 01100111 01100101 01110100 01101000
01100101 01110010 00101110 00100000.

```

When trying to recover the hiding from any of the five files, random data was obtained that is not meaningful, and from which it is impossible to obtain the secret message or any of its part, as long as the secret key is preserved from hackers or any unauthorized one. And the following details shows the data extracted from each cover file and its corresponding random segments after converting it to the text.

4.3. Cover1

4.3.1. Extracted binary stream:

```

00011101 11010001 10011001 11001001 10100100 10110001
11010001 11011000 11010000 10110100 11011000 11001001
11010000 11001001 10100000 10000000 11011100 10000000
10011001 10100001 01101111 11000100 10010001 10110011
10010111 11000010 00001000 01001011 01100001 11101100
10011001 00110111 01100010 00011010 10011101 01000011
00100000 10001111 01000000 11011000 11001001 01100010
00101110 10010101 00010000 11110110 01000000 01001000
00011011 01110010 01101110 10110001 10111001 00000111
11001010 11011100 00101011 11010101 01010111 01111000
10110001 00001010 10000111 11001011 10000110 10000011
00100011 10110111 01110000 01100101 11100001 10010100
00110100.

```

4.3.2. Converted code

Ń™É±NØÐ'ØÉÐÉ □Ü□™;oÄ'³—Â Kai™7b • C • @ØÉb. • ö@H rn±¹ÊÜ+ÖWx±‡Ë†f#•peá”4

4.4. Cover2

4.4.1. Extracted binary stream:

```

01110111 01100101 10010000 11000110 01000010 11100110
11001110 11101100 10100100 11000110 10001101 10000000
10001010 11000101 01001110 11001101 10000100 10000110
10000101 10000110 11001111 10101100 11000101 11001100
00011010 00101001 01110011 10101101 10010001 10111001
00000110 11011100 00001000 10100011 01100101 00001110
10010101 10110010 10010111 11001010 10001011 01101000
00101101 10000001 00110100 10010111 01000000 10011001
10011011 01100001 01001100 10100000 00111010 10000110
01000000 00011010 10011011 01110010 00101100 11010101
00110100 01100100 00100101 00111110 00110000 01100100
00110100 01110000 00100100 00100100 00101100 00101100
01100111 01100000 00100100 00100000 00101110 01111100
01100100 01101110 00111010 10011010 00110100 10110010
00111010 00010110 00010110 10110010 00010000 00111010
00110010 00011010 00011001 00110110 10111010 00110110
00110010 00111010 00010010 10000011 01000110 11000110
00000010 00100010 01000111 01000011 01000000 00000111
01100110 01000011 01000010 11000001 01000111 11100010
01000110 01100110 01000110 01100000 11110110 01000000
11011001 00000011 01100001 10001100 10000001 00110111
00000110 11001010 00011010 01001011 01100100 10101101
10000001 10110111 00000110 01000000 11011001 11000001
10010000 11110110 01000000 10010011 10000001 01110111
00101110 10000001.

```

4.4.2. Converted code

4.6. Cover4

4.6.1. Extracted binary stream

```

10110010 00000110 11001010 10011001 10100011 00100000
01001100 11010001 01100001 01101011 01101010 11000111
00111000 11001100 11000110 11001000 11000110 00101110
11001001 11000110 00100000 01101010 01101011 01101010
01100001 01100100 00101010 00100110 00100110 01011110
00101000 00100101 01101010 01101010 01111010 01100010
01100110 00111010 01101100 01101000 01101001 01101100
00100110 11000001 11001000 11001000 00100101 01011011
11000111 11001000 11001000 01000011 00100110 00101010
00100101 10000000 01100011 00000110 11001000 11000100
11001001 11000100 11001001 11001001 11001001 00100000
01111011 00111010 11110111 11000100 10000000 01100001
11110111 11001000 11000100 11000001 00101111 00101111
11000100 11001000 11000100 11000101 01000111 01000110
00100010 01101100 01101011 01111001 01101111 01101001
01110101 11001000 11000100 00101000 11001000 11000111
00100100 00100101 11110111 00100000 11000001 11000100
01111011 01001011 01010101 01001100 01000110 01011001
01010101 01100110 01100100 01110011 00111011 00101110
00101100 00101111 00101110 00101100 01101101 11001001
00101001 00100000 11001000 11000101 10001010 00101000
00101110 01101011 01110011 01101000 01110111 01100101
11001000 11001001 11001010 11001000 11000011 11000101
11000101 11110101 01110111 01101001 01110101 01110000

```

we• ÆBæîî□Æ□□ŠĀNÍ.,†...†İ-ÂÌ)s-‘¹ Ü Ğe•²—Êħh• 4—@™)aL :†@ >r,Ö4d%>0d4p\$\$.g`\$.|d6:š4²: ²:2 6°62:
fFÆ" GC@fCBÁGâ FfF`ö@Ù aCE• 7 Ê Kd-□• @ÜÁ• ö@“• w.•

4.5. Cover3

4.5.1. Extracted binary stream

```

10110010 11110111 11101000 01011001 00000011 01101110
00001110 10000101 00110010 11110110 00100000 01101100
11100100 00111010 10000110 11101000 00000011 01010011
01110010 10101100 11010001 10110010 00000111 11000010
10011001 01111011 01101100 00100100 10000101 00110010
00000110 11101000 00011000 01110001 01100001 11101110
10010101.

```

4.5.2. Converted code

•²÷èY n...2ö lä:†è Sr¬Ñ²Â™{l\$...2 è qai•

```

00110100 01110000 00100100 00100100 00101100 00101100
01100111 01100000 00100100 00100000 00101110 01111100
01100100 00110110 01100001 01001100 10100000 00111010
10000110 01000000 00011010 10011011 01110010 00101100
11010101 00110100 10110010 11110111 11101000 01011001
00000011 01101110 00001110 10000101 00110010 11110110
00100000 01101100 11100100 00111010 10000110 11101000
00000011 01010011 01110010 10101100 11010001 10110010
00000111 11000010 10011001 01111011 00011010 00101001
01110011 10101101 10010001 10111001 00000110 11011100
00001000 00101100 00101100 01100111 01100000 00100100
00100000 00101110 01111100 01100100 00110110 00111010
10011010 00110100 10110010.

```


4.6.2. Converted code

² Ê™£ LÑakjÇ8lÆÈÆ.ÉÆ jkjad*&&^(%jjzbf:lhil&ÁÈÈ%[ÇÈÈC&*%€ c ÈÄÉÄÉÉÉ
 {:÷Ä€ a÷ÈÄÄ//ÄÈÄÄGF"lkyoiuÈÄ(ÈÇ \$%÷ ÄÄ{KULF YUfds;./.,mÉ) ÈÄŠ(.kshweÈÈÈÈÄÄöwiup4p\$\$.g`\$.|d6aL
 :†@ ›r,Ö4²÷èY n...2ö lä:†è Sr¬Ñ²Â™{ }s -‘¹ Ü ,,g`\$.|d6:š4²

4.7. Cover5

4.7.1. Extracted binary stream

00111010 00010110 00010110 10110010 00010000 00111010
 00110010 00011010 00011001 00110110 10111010 00110110
 00110010 11010100 01010000 10010010 00100100 01001000
 10000001 01010100 10001001 00000100 11110101 00101010
 10101101 10010001 01001011 01000010 11010101 00010101
 01010101 01010101 00010101 01011101 01010100 10010001
 00101010 10101001 01010101 01000100 01111001 11001010
 00000100 01101001 10001100 10101010 10001011 11111010
 11010101 01010110 10101011 10001001 01010101 01010101
 01000101 01010001 00101010 10101010 10100010 11001010
 01010101 00101000 10101010 01010100 01010001 01000010
 00111101 11011111 10010101 01010101 01000101 00100100
 10001101 01101010 10001010 10101010 10101010 11011101
 01111001 01010001 01111010 10010111 01010101 01010101
 01101001 01001010 11010101 01010101 01001011 01010101
 00101010 10001101 11101010 10101010 11111111 10101001
 00010001 01011010 10101001 01010101 01001010 01010110
 01010110 10101010 10011000 10110100 11010101 01100000
 10111010 10101000 01010101 01010111 00101010 10001010
 01010100 10010001 01000100 10010100 10101110 01001010
 10100100 11010101 00101001 00001001 00100100 00001001

00100100 10010000 10010100 00010010 01010010 00101111
 11010101 01001110 11010101 11111000 10101000 10010101
 00010111 01010111 11011101 00100100 01001010 01010010
 01000101 01010101 00000100 00101010 01010010 01000000
 10010111 11101101 01001010 10010100 10010010 01000101
 01010100 10101001 01010101 00110100 01010010 10101010
 10111100 11101010 11010010 10101010 01011001 01100101
 01100101 01010101 01010101 00101010 10100101 10001010
 10010101 00100100 10010010 01011001 01110100 10101010
 10101011 01001001 01010100 10111010 10110101 01001010
 10101101 01110011 10100010 10101000 01001001 01000110
 10010100 10010010 01001000 10100100 00010010 01001001
 01010101 01010001 00100100 00010010 01000001 01110100
 10011010 10100101 01001011 01010011 00110101 00010100
 10100010 01000010 01001010 01000010 11011100 10100110
 10101101 00000101 11101101 00100111 10111011 10101001
 01000010 10001000 10101000 00001010 01100100 11100101
 01000100 10010101 01000010 10101010 10100000 10111101
 10100101 00010100 01000100 01011101 01001010 01000100
 10000100 01000010 00000100 10100101 01000010 01110010
 00001001 01000100 10000100 01011101 11101010 01001011
 10011111 10001010 01000100

4.7.2. Converted code

```
: 2:2 6°62ÔP'$H• T%ø δ*-‘KBÕ UU ]T‘*©UDyÊ iEª·úÕV«%UUEQ*ª¢ÊU(ªTQB=ß•UE$• jšªªYyQz—
UijÔUKU*• ¢ªÿ© Z©UJVvª~' Õ`o"UW*ŠT'D"®JªÕ)$ $□" R/ÕNÕø"• WÝ$JREU *R@—ij"ET©U4Rª¼êÔªYee
UU*¥Š$'Ytª«TªµJ-s¢ "IF" Hª IUQ$ Atš¥KS5 ¢B JBÜ |· í»©B"ªdªD•Bª ½¥ DJJD,,B ¥BrD,,jêKÝŠD
```

5. Conclusions and future works

The proposed method has proven its effectiveness and fulfillment of the most important criteria necessary for the success of the hiding technique model by integrating this technique with taking advantage of the characteristics of encryption technique and the secret key, which can be visualized through the following conclusions, as well as suggestions for its development as a future work.

5.1. Conclusions

At the first, the histogram charts of the five WAV files shown in Table 3 of figures indicate that no any distortion can be distinguished in them when comparing the samples before and after the hiding process, also the hiding in the least significant bits of the audio file samples, as mentioned earlier, that does not occur any voice distortion that is perceptible or recognizable by the human ear.

When applied this method to several cases that have shown their efficiency compared to the previous methods referred to, since it was difficult to reach the hidden secret file because of the following characteristics:

1. In the case of suspicion of the existence hiding, the process of extracting the secret file is very complex for the following reasons:
 - a. The extraction of the hidden from any one cover file will produce a random data, because what was hidden in it is one fragmented part of several parts.
 - b. The numbers of WAV files containing the hiding and their arrangement is unknown. so, no part of the secret file can be obtained in an of attempt to extract the hiding.
 - c. The theft of the secret key alone is not sufficient to reach the hidden file, it is impossible to know which of the WAV files are contained hiding, as well as the random numbering and the secret location of its code.
2. The circular secret key achieves the randomness of the distribution, and the more random distribution of the numbers of covers files mean the more random there are, the same is true for increasing the repetition of those randomized in one cycle of that key. So that the efficiency of the method is greater.
3. The multiple covers in which the hiding achieves a state of randomness as well as the difference in its size, which leads to inconsistency and regularity of the distribution the bits of secret file, which increases the efficiency of the method by increasing random distribution.

5.2. Future work

1. In stereo samples representation of WAV files, there is greater flexibility by considering each channel as an independent audio file, as well as the possibility of merging different types of audio to increase the durability of the method.
2. In the absence of a number of WAV files as covers, can be applied the method on one file by slicing its samples into several parts and each part numbered randomly which treated as a separate file, then apply the same algorithm of hiding.
3. It is possible to choose different types of audio file formats such as MID and MP3 with WAV at the same time, or use of images files with WAV files together, which increases the efficiency of the method and increase the complexity of extract its hiding.
4. The percentage of hiding can be increased by including three or four least significant bits of sixteenth sample representing, especially in noisy WAV files that are difficult to perceived or distinguished by the human ear, and thus the hiding ratio is three or four out of sixteen (3/16 or 4/16), and it should be noted here that it is preferable to avoid silent positions in the audio models.
5. We can encrypt the secret file with one of the encryption algorithms adopted before the process of hiding to increase the efficiency of the method by increasing the random of hidden file in the case of trying to extracting it.

References

- [1] Provos N, Honeyman P. Hide and seek: introduction to Steganography. IEEE Secur Privacy J 2003.
- [2] Shashidhar R, Kumar MS, Arunakumari BN, Kumar RS, Patilkulkani S. Novel Approach for Steganography to Camouflage Digital Information Using Least Significant Bit. In: Computational Intelligence in Pattern Recognition. p. 551–62.
- [3] Zaidan BB, Zaidan AA, Al-Farajat AK, Jalab HA. On the differences between hiding information and cryptography techniques: An overview. J Appl Sci (Faisalabad) 2010;10:1650–5.
- [4] Taha MS, Rahim MSM, Iafta SA, Hashim MM, Alzuabidi HM. Combination of Steganography and Cryptography: A short Survey. 2nd International Conference on Sustainable Engineering Techniques (ICSET) IOP Publishing IOP Conf. Series: Materials Science and Engineering 518, 2019.
- [5] Gowda NCh, Srivastav PSV, G. P. R.. "Steg Cryp. "Encryption using steganography". Int J Eng Adv Technol (IJEAT) 2019;8(55).
- [6] Djebbar F, Ayad B, Abed-Meraim K, Hamam H. A view on latest audio steganography techniques. 7th IEEE International Conference on Innovations in Information Technology, 2011.
- [7] Gadicha AB. Audio Wave Steganography. Int J Soft Comput Eng 2011;1(5).
- [8] Padmashree G, Venugopala PS. Audio Steganography and Cryptography: Using LSB algorithm at 4th and 5th LSB layers. Int J Eng Innov Technol (IJEIT) October 2012;2(4).

- [9] Asad M, Gilani J, Khalid A. An Enhanced Least Significant Bit Modification Technique for Audio Steganography. International Conference on Computer Networks and Information Technology (ICCNIT). IEEE; 2011.
- [10] Ramesh Yadav P, Usha Shree V, Padmapriya K. "Hiding Data in Audio Using Audio Steganography", International Journal of Computer Applications. Eng Sci 2012;ISSN:2231–4946.
- [11] Adhiya KP, Patil Swati A. Hiding Text in Audio Using LSB Based Steganography ISSN 2224-5758 X(Online) Vol 2, No.3. Information and Knowledge Management, 2012.
- [12] Jayaram P, Ranganatha HR, Anupama HS. Information Hiding Using Audio Steganography – a Survey. Int J Multimedia Appl (IJMA) August 2011;3 (3):86–96.
- [13] Karthikeyan B, Kosaraju AC, Gupta S. Enhanced security in steganography using encryption and quick response code. In: Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference on. IEEE; 2016. p. 2308–12.
- [14] Yin-Cheng Qi, Liang Ye, Chong Liu. Wavelet Domain Audio Steganalysis for Multiplicative Embedding Model. Proceedings of the 2009 International Conference on Wavelet Analysis and Pattern Recognition, 2009.
- [15] Jain MP, Trivedi PV. Effective Audio Steganography by using Coefficient Comparison in DCT Domain. Int J Eng Res Technol 2013;2(8).
- [16] Antony J, Sobin CC, Sherly AP. Audio Steganography in Wavelet Domain A Survey. Int J Comput Appl August 2012;52(13):33–7.
- [17] Adeboje O, Adetunmbi AO, Gabriel AJ. Embedding Text in Audio Steganography System using Advanced Encryption Standard, Text Compression and Spread Spectrum Techniques in Mp3 and Mp4 File Formats. Int J Comput Appl March 2020;177(41):46–51.
- [18] Djebbar F, Ayad B, Abed-Meraim K, Habib H. Unified phase and magnitude speech spectra data hiding algorithm. J Secur Commu Networks 2012.
- [19] Akoum A. New Method for Hiding Secret Message and Book in Audio ISSN 0973–4562. Int J Appl Eng Res 2018;13(18):13697–701.
- [20] Parthasarathi M, Shreekala T. Secured Data Hiding in Audio Files Using Audio Steganography Algorithm. Int J Pure Appl Math (ISSN) 2017;116(21):619–28.
- [21] Shirali-Shahreza S, Shirali-Shahreza M. Steganography in Silence Intervals of Speech. In: Proceedings of the Fourth IEEE International Conference on Intelligent Information Hiding and Multimedia Signal. p. 605–7.
- [22] Sinha N, Bhowmick A, Kishore B. Encrypted Information Hiding using Audio Steganography and Audio Cryptography. Int J Comput Appl February 2015;112 (5):0975–8887.
- [23] Karim M, Rahman S, Hossain I. A new approach for LSB based image steganography using secret key. In: Proceedings of 14th International Conference on Computer and Information Technology (ICCIT-2011). p. 286–91.