Volume 4, Issue 5, May 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Audio Steganography with Various Compression Algorithms to Improve Robustness and Capacity

**Chintan R. Nagrecha***
*Computer Department*
*BVM, Vallabh vidhynagar, GTU*
*India*

**Prof. Prashant B. Swadas**
*Head Computer Department*
*BVM, Vallabh vidhynagar, GTU*
*India*

***Abstract— As attack on data communication become deliberately advance the security of the transmitted data is very important issue. So more efficient methods are chosen which ensure secure data transfer. One of the method is the audio Steganography. One of the most important and widely used approach of audio steganoggraphy is LSB (List significant approach). In this paper we deals with the approach of embedding the bits at higher random layer which leads towards difficult discovery of data. Main aim of this paper is to improve capacity and robustness of this approach. The combination of well known compressive algorithms and given embedding approach gives observable result. This leads to improve the capacity of host audio and robustness.***

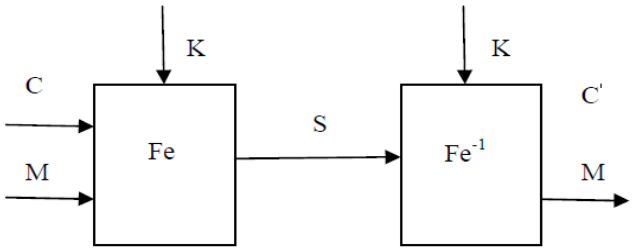
***Keywords— Audio Steganography, Compressive Algorithms, SNR, Capacity, Robustness***

## I. INTRODUCTION

Steganography is the technique to hide the information in some media (cover media) so that third party or attacker can't recognize that information is hidden into the cover media. The information that to be hidden is called stego and the media in which the information is hidden is called host. Various files can be act as a cover media like text, image, audio, video, IP Datagram etc. The main approach of steganography is to make difficult data discovery as much as we can.

The steganography application hides different types of data within a cover file. The resulting stego also contains hidden information, although it is virtually identical to the cover file. What Steganography essentially does is exploit human perception; human senses are not trained to look for files that have information hidden inside of them, although there are programs available that can do what is called Steganalysis (Detecting use of Steganography) [10].

The process of Steganography is as shown in Figure 1.2. The random selection of the samples used for embedding introduces low power additive white Gaussian noise (AWGN). Each time while embedding bits of information, more or less noise introduced. It is well known from psychoacoustics literature that the human auditory system (HAS) is highly sensitive to the AWGN [7].




(Figure 1: The Steganographic operation) [10]

Hiding information into a media requires following elements [7]

- The cover media(C) that will hold the hidden data
- The secret message (M), may be plain text, cipher text or any type of data
- The stego function (Fe) and its inverse (Fe-1)
- An optional stego-key (K) or password may be used to hide and unhide the message.

The embedding process (Fe) embeds the secret message E in the cover data C. The exact position (S) where E will be embedded is dependence on the key K. In some steganography algorithm the bit embedding position is fixed, in such case key is not required. The result of the embedding function is slightly modified version of C: the stego data C'. After the recipient has received C' he starts the extracting process($Fe^{-1}$) with the stego data C' and the key K as parameters. If the key that is supplied by the recipient is the same as the key used by the sender to embed the secret message and if the stego data the recipient uses as input is the same data the sender has produces (i.e., it has not been modified by an adversary), then the extracting function will produce the original secret message E [10].

## II. TECHNIQUES FOR DATA HIDING IN AUDIO

There are several techniques are available for audio steganography. Some of them are as follows:

### *A. Least Significant Bit[LSB] Technique*

Least significant bit (LSB) coding is the simplest way to embed information in a digital audio file. By substituting the least significant bit of each sampling point with a binary message, LSB coding allows for a large amount of data to be encoded. In LSB coding, the ideal data transmission rate is 1 kbps per 1 kHz. In some implementations of LSB coding, however, the two least significant bits of a sample are replaced with two message bits. This increases the amount of data that can be encoded but also increases the amount of resulting noise in the audio file as well [7]. A novel method which increases the limit up to four bits by Nedeljko Cvejic, Tapio Seppben & mediaTeam Oulu at Information Processing Laboratory, University of Oulu, Finland, Further successful research increase the limit up to six bits, this research done by the same researchers [4].

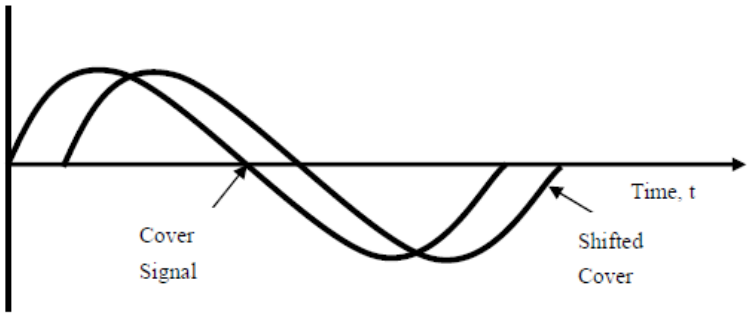
Example:

| Sampled Audio Stream (16 bit) | 'A' in binary | Audio stream with encoded message |
|---|---|---|
| 1001 1000 0011 1100 | **0** | 1001 1000 0011 110**0** |
| 1101 1011 0011 1000 | **1** | 1101 1011 0011 100**1** |
| 1011 1100 0011 1101 | **1** | 1011 1100 0011 110**1** |
| 1011 1111 0011 1100 | **0** | 1011 1111 0011 110**0** |
| 1011 1010 0111 1111 | **0** | 1011 1010 0111 111**0** |
| 1111 1000 0011 1100 | **1** | 1111 1000 0011 110**1** |
| 1101 1100 0111 1000 | **0** | 1101 1100 0111 100**0** |
| 1000 1000 0001 1111 | **1** | 1000 1000 0001 111**1** |

There are two main disadvantages associated with the use of methods like LSB coding. The human ear is very sensitive and can often detect even the slightest bit of noise introduced into a sound file, Second disadvantage however, is that LSB encoding method is not robust. If a sound file embedded with a secret message using either LSB coding was resample, the embedded information would be lost. Robustness can be improved somewhat by using a redundancy technique while encoding the secret message. However, redundancy techniques reduce data transmission rate significantly [7].

### *B. Phase Coding*

Phase coding addresses the disadvantages of the noise inducing methods of audio steganography. Phase coding relies on the fact that the phase components of sound are not as perceptible to the human ear as noise is. Rather than introducing perturbations, the technique encodes the message bits as phase shifts in the phase spectrum of a digital signal, achieving an inaudible encoding in terms of signal-to perceived noise ratio. Original and encoded signal are as shown in Figure 3.2 [10].




(Figure 3.2:Iillustrate the original cover signal and encoded shifted signal of phase coding technique.) [7]

Phase coding is explained in the following procedure [7]:

• The original sound signal is broken up into smaller segments whose lengths equal the size of the message to be encoded.

• A Discrete Fourier Transform (DFT) is applied to each segment to create a matrix of the phases and Fourier transform magnitudes.

• Phase differences between adjacent segments are calculated.

• Phase shifts between consecutive segments are easily detected. In other words, the absolute phases of the segments can be changed but the relative phase differences between adjacent segments must be preserved. Therefore the secret message is only inserted in the phase vector of the first signal segment as follows: [10]

$$\textbf{Phase\_new} = \begin{cases} \pi / 2 \text{ if message bit} = 1 \\ \pi / 2 \text{ if message bit} = 0 \end{cases}$$

• A new phase matrix is created using the new phase of the first segment and the original phase differences.

• Using the new phase matrix and original magnitude matrix, the sound signal is reconstructed by applying the inverse DFT and then concatenating the sound segments back together.

*C. Echo Hiding*

In echo hiding, information is embedded in a sound file by introducing an echo into the discrete signal. Like the spread spectrum method, it too provides advantages in that it allows for a high data transmission rate and provides superior robustness when compared to the noise inducing methods [7].

*D. Spread Spectrum*

In the context of audio steganography, the basic spread spectrum (SS) method attempts to spread secret information across the audio signal's frequency spectrum as much as possible. This is analogous to a system using an implementation of the LSB coding that randomly spreads the message bits over the entire sound file [7].

## III. LOSSLESS COMPRESSION TECHNIQUES

Compression is the conversion of data in such a format that requires few bits usually formed to store and transmit the data easily and efficiently. Compression is used to reduce amount of data and needed to reproduce that data whenever we require it [12]. There are two type of compression methods – lossy compression and lossless compression. In lossy compression some data loss may occurs after compression while in lossless compression no such data loss occurs. Steganography completely deals with data security hence lossless compression techniques are more preferable.

*A. Repetitive Sequence Suppression or Run Length Encoding*

This algorithm is more efficient for the strings with Repetitive occurrence of similar contents. But it is not effective if data file has less repeating of characters [12]. We can compress the run-length symbols using Huffman coding, arithmetic coding, or dictionary based methods.

Method: The first step in this technique is read file then it scans the file and find the repeating string of characters [13].when repeating characters found it will store those characters with the help of escape character followed by that character and count the binary number of items it is repeated [12].

*B. Huffman Coding*

The Huffman coding algorithm works on bottom-up approach is named after its inventor, David Huffman, who developed the method as a student in a class on information theory at MIT in 1950[12]. Below steps shows Huffman coding method [12].

1. Initialization: Put the old nodes in a list sorted according to their frequency counts.
2. Repeat the following steps until the sorted list has only one node left:

(1) From the list pick two nodes with the lowest frequency counts. Form a Huffman sub tree that has these two nodes as child nodes and create a parent node.

(2) Assign the sum of the children's frequency to the parent node and insert it into the list such that the order is maintained.

(3) Delete the children from the sorted list.

3. Assign a 0 and 1 codeword to the two branches of the tree on the path from the root. After the Huffman tree, the method creates a prefix code for each node from the alphabet by traversing the tree from the root to the node. It creates 0 for left node and 1 for a right node [12].

*C. Shannon-Fano Coding technique*

It is used to encode messages depending upon their probabilities [13].

Method [12]:

1. For a given list of symbol create a probability table.
2. Sorting the table based on the probability and places the most frequent symbol at the top of a list.
3. The table is divided into equally two halves upper and lower which having a same probability as much as possible.
4. The upper half of the list defined with „0" digit and the lower half with a "1".

5. Repeat the steps 3 and 4 for each of the two halves then further divide the groups and adding bits to the codes and stop the process when each symbol has a corresponding leaf on the tree.

*D. LZW (Lempel-Ziv Welch) compression method*

LZW is the most popular method. This technique has been applied for data compression [12]. The main steps for this technique are given below:-

Method [12]:

1. Firstly it will read the file and given a code to each character.
2. If the same characters are found in a file then it will not assign the new code and then use the existing code from a dictionary.
3. The process is continuous until the characters in a file are null [12].

## IV. METHODOLOGY

In standard data hiding algorithm in audio file was very easy to extract as attacker may correctly guess about LSB data hiding position hence chances of retrieving may increase this makes this algorithm least popular. This existing approach of embedding data bits at random and higher bit is very important and helpful.

*A. Algorithm*

Step 1: Extract host audio file, evaluate sample rate, sample size etc. according to sample size (16bit or 8bit) read sample.

Step 2: Read message string.

Step 3: According to message length choose compressive algorithm (i.e., for extremely large message prefer Shannon-Fano) and embed output bit stream over audio.

Step 4: Generate new 16 bit samples by inserting message bits into random higher bits using algorithm [3].

Step 5: Embedding message bit at random higher layers such that the distortion can be minimized.

Step 6: Embed layer number (bit position) value with next sample.

Step 7: Convert all sample (16 bit) into regular audio stego file.

*B.Procedure*

As mention in algorithm, it calculates the distortion by embedding at various layers of host sample. The sample with minimum distortion is selected.

First of all convert the host audio file samples in order to extract header information. In following example we have considered only 8 bits of each sample as remaining 8 bit must be remain unchanged.

e.g.; 00111010, 00110101, 11001111, 10101010 …..

Then covert the message into bit stream, suppose the compressed bit stream is 00101011….

Consider first sample and convert it into binary 00111010=58

Consider first bit of message and embed it to various layers and try to minimize the distortion as mention above. Suppose we inserts bit 0 at layer 4 (100), after embedding the binary equivalent will be 00110010=50.

Apply algorithm to reduce distortion: 00110111 = 56 embed 100 (layer position) at next sample.

Result: 0011**0**111, 00110100, 11001111, 10101010 …

*C.SNR (Signal to noise ration)*

1) Mer (mean error rate) = coverfilebits – embeddedfilebits / coverfilebits
2) sizeinfo=size of the cover file

SNR=(20*log10(sizeinfo / mer)) [7]

$$SNR = 10 \cdot \log_{10} \frac{\sum_n x^2(n)}{\sum_n [x(n) - y(n)]^2}$$

[4]

## V. EXPERIMENT RESULTS AND ANALYSIS

Run-Length encoding method is not much efficient as compare to other lossless compression techniques. R-L encoding technique is very less popular because of inefficient results. Sometime the output compressed file may larger then input file hence we have skipped that technique.

TABLE I
COMPARISON OF SNR VALUES

| | Message size | SNR values for Huffman compression | SNR values for Shannon-Fano Compression | SNR values for LZW compression |
|---|---|---|---|---|
| *1* | 5,212 | 57.59 | 56.10 | 56.91 |
| *2* | 10,848 | 53.11 | 52.84 | 53.02 |
| *3* | 21,098 | 46.44 | 46.20 | 46.89 |

As shown in above table, For smaller/medium size messages- Huffman compression algorithm gives better SNR value as compare to others. LZW is very powerful and popular too but this approach is very complex and time consuming. Directory maintenance and updating is require in LZW hence this approach becomes complex. As this research also tries to reduce the computational overhead LZW technique is not much preferable here. Shannon-Fano compression algorithm is most suitable in compression of very large messages.

TABLE II
AVERAGE PERCENTAGE OF IMPROVEMENT OF STORAGE CAPACITY

| Huffman compression | Shannon-Fano Compression | LZW compression |
|---|---|---|
| 35.45% | 33.01% | 35.34% |

Above results shows that host media storage capacity can efficiently increase by various compressive algorithms. However the compressing algorithm efficiency is also depends upon the complexity of message. According to message size and strength of compressive algorithm may differ in their performance

## VI. CONCLUSIONS

In standard LSB algorithm the chances of message discovery is very high. Difficult discovery of message bit in host audio can be achieved by embedding it to random higher bits of sample. After applying compression algorithm the

output message bit stream completely differs then input message. As compare to standard method this approach is more robust. Lossless compression techniques sufficiently improve storage capacity of host audio.

**REFERENCES**

[1] Mazdak Zamani , Hamed Taherdoost, Azizah A. Manaf , Rabiah B. Ahmad , and Akram M. Zeki, "Robust Audio Steganography via Genetic Algorithm",IEEE 2009
[2] Krishna Bhowal,Anindya Jyoti Pal,Geetam S. Tomar,P. P. Sarkar,"Audio Steganography using GA",2010 IEEE
[3] Krishna Bhowal,Debnath Bhattacharyya,Anindya Jyoti Pal,Tai-Hoon Kim, "A GA based audio steganography with enhanced security", Springer Science+Business Media, LLC 2011
[4] Nedeljko Cvejic, Tapio Seppanen,"Increasing Robustness of LSB Audio Steganography Using a Novel Embedding Method",2004 IEEE.
[5] Mazdak Zamani , Azizah A. Manaf , Rabiah B. Ahmad , Akram M. Zeki , and Shahidan Abdullah, "A Genetic-Algorithm-Based Approach for Audio Steganography", World Academy of Science, Engineering and Technology 54 2009
[6] Soumyendu Das,Subhendu Das,Bijoy Bandyopadhyay,Sugata Sanyal"Steganography and Steganalysis: Different Approaches",
[7] K.P.Adhiya,K.P.Adhiya Swati A. Patil,"Hiding Text in Audio Using LSB Based Steganography" IISTE 2011
[8] Pradeep Kumar Singh, R.K.Aggarwal "Enhancement of LSB based Steganography for hiding Image in Audio" IJCSE 2010
[9] Gunjan Nehru ,Puja Dhar, " A Detailed look of Audio Steganography Techniques using LSB and Genetic Algorithm Approach" – IJCSI 2012
[10] Jayaram P, Ranganatha H, Anupama H, "Information hiding using audio steganrgaphy – A Survey" IIJMA 2011
[11] Rupinder Singh Brar,Bikramjeet singh, "A Survey on Different Compression Techniques and Bit Reduction Algorithm for Compression of Text/Lossless Data",IJARCSSE,2013
[12] Rajinder Kaur,Mrs. Monica Goyal "A Survey on the different text data compression techniques." , IJARCET ,2013
[13] Y. M. Kamir, M. Deris. M. Sufian, and A. A.F. Amri, 'Study of Efficiency and Capability LZW++Technique in Data Compression", World Academy of Science, Engineering and Technology 35 2009
[14] Anmol Jyot Maan, "Analysis and Comparison of Algorithms for Lossless Data Compression", IJICT, 2013