International Journal of Foundations of Computer Science

Audio Steganography Using Mod 4 Method (M4M)

Souvik Bhattacharyya
Department of Computer Science and Engineering
,University Institute of Technology
, The University of Burdwan
Burdwan, West Bengal/India
souvik.bha@gmail.com

Arko Kundu
Department of Computer Science and Engineering
,University Institute of Technology
, The University of Burdwan
Burdwan, West Bengal/India
arkokundu190789@gmail.com

Kaushik Chakraborty
Department of Computer Science and Engineering
,University Institute of Technology
, The University of Burdwan
Burdwan, West Bengal/India
kc.tech.uit@gmail.com

Gautam Sanyal
Department of Computer Science and Engineering
,National Institute of Technology,
Durgapur, West Bengal/India
nitgsanyal@gmail.com

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

Steganography is a process that involves hiding a message in an appropriate carrier like image or audio. It is of Greek origin and means "covered or hidden writing". The carrier can be sent to a receiver without any one except the authenticated receiver only knows existence of the information. Steganography is an emerging area which is used for secured data transmission over any public media. Considerable amount of work has been carried out by different researchers on steganography. The M4M approach, based on M16M [10] approach, works on **wav** and **mp3** format audio files. The proposed approach works by selecting the embedding positions using some mathematical function and maps each two bit of the secret message in each of the selected positions in a specified manner. A pseudo random number generator is used here to locate the embedding positions of the message bits randomly. This solution is independent of the nature of the data to be hidden and

produces a stego audio with minimum degradation.

Keywords: Cover Audio, Mod 4 Method (M4M), Stego Audio .

1. Introduction

STEGANOGRAPHY is the art and science of hiding information by embedding messages within other, seemingly harmless messages. Steganography means "covered writing" in Greek. As the goal of steganography is to hide the presence of a message and to create a covert channel, it can be seen as the complement of cryptography, whose goal is to hide the content of a message. Another form of information hiding is digital watermarking, which is the process that embeds data called a watermark, tag or label into a multimedia object such that watermark can be detected or extracted later to make an assertion about the object. The object may be an image, audio, video or text only. A famous illustration of steganography is **Simmons' Prisoners' Problem** [1].An assumption can be made based on this model is that if both the sender and receiver share some common secret information then the corresponding steganography protocol is known as then the secret key steganography where as pure steganography means that there is none prior information shared by sender and receiver. If the public key of the receiver is known to the sender, the steganographic protocol is called public key steganography [2, 3] and [4].For a more thorough knowledge of steganography methodology the reader may see [5, 6].Some Steganographic model with high security features has been presented in [7, 9] and [8].Almost all digital file formats can be used for steganography, but the image and audio files are more suitable because of their high degree of redundancy [6]. Fig. 1 below shows the different categories of steganography techniques. Fig. 1 below shows the different categories of steganography techniques.

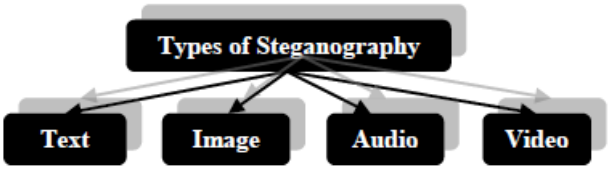

Fig. 1. Types of Steganography

A block diagram of a generic audio steganographic system is given in Fig. 2. A message is embedded in a cover audio through an embedding algorithm, with the help of a secret key. The resulting stego audio may be transmitted over a channel to the receiver where it is processed by the extraction algorithm using the same key. During transmission the stego audio, it can be monitored by unauthenticated viewers who will only notice the transmission of an audio without discovering the

existence of the hidden message. In this work a specific audio based steganographic method has been proposed. In this method instead of embedding the secret message into the cover audio a mapping technique has been incorporated to generate the stego audio. This method is capable of extracting the secret message without the presence of the cover audio.

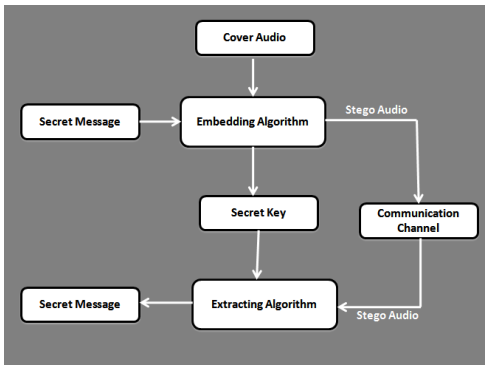

Fig. 2. Generic form of Audio Steganography

This paper has been organized as following sections: Section II describes Proposed call admission control algorithm, Section III deals with mathematical analysis.Results and discussions are presented in Section IV and work is concluded in Section V.

2. Audio Steganography

Like the document images, the sound files may be modified in such a way that they contain hidden information, like copyright information; those modifications must be done in such a way that it should be impossible for a pirate to remove it, at least not without destroying the original signal. The methods that embeds data in sound files use the properties of the Human Auditory System (HAS). The HAS perceives the additive random noise and also the perturbations in a sound file can also be detected. But there are some holes we can exploit. While the HAS have a large dynamic range, it has a fairly small differential range. As a result, loud sounds tend to mask out quiet sounds. And there are also some distortions that are so common that the HAS ignores them. The digital sound is obtained from the analog sound by converting it to digital domain. This process implies two sub processes: sampling and quantization. Sampling is the process in which the analogue values are only captured at regular time intervals.Quantization converts each input value into one of a discrete value. Popular sampling rates for audio include 8 kHz, 9.6 kHz, 10 kHz, 12 kHz, 16 kHz, 22.05 kHz and 44.1 kHz. The most popular file formats for sounds are the Windows Audio-Visual (WAV) and

the Audio Interchange File Format (AIFF).There are also compression algorithms such as the International Standards Organization Motion Pictures Expert Group-Audio (ISO MPEG-AUDIO). When developing a data hiding method for audio, one of the first considerations is the likely environments the sound signal will travel between encoding and decoding. There are two main areas of modification which we will consider. First, the storage environment, or digital representation of the signal that will be used, and second the transmission pathway the signal might travel [7, 8]. In order to conceal secret messages successfully, a variety of methods for embedding information in digital audio have been introduced. These methods range from rather simple algorithms that insert information in the form of signal noise to more powerful methods that exploit sophisticated signal processing techniques to hide information. This section of the paper is organized as follows. First, the clarification of the Audio Environment. Secondly, this section describes a wide range of techniques that have been used in Audio Steganography.

2.1. *Digital representation*

There are two critical parameters to most digital audio representations: sample quantization method and temporal sampling rate. The most popular format for representing samples of high-quality digital audio is a 16-bit linear quantization e.g., Windows Audio-Visual (WAV) and Audio Interchange File Format (AIFF) [4, 11]. Another popular format for lower quality audio is the logarithmically scaled 8-bit m-law. These quantization methods introduce some signal distortion, somewhat more evident in the case of 8-bit m-law. Popular temporal sampling rates for audio include 8 kHz (kilohertz), 9.6 kHz, 10 kHz, 12 kHz, 16 kHz, 22.05 kHz, and 44.1 kHz. Sampling rate impacts data hiding in that it puts an upper bound on the usable portion of the frequency spectrum (if a signal is sampled at 8 kHz, it is not desirable to introduce modifications that have frequency components above 4 kHz). For most data-hiding techniques developed, usable data space increases at least linearly with increased sampling rate [11]. There are many different transmission environments that a signal might experience on its way from encoder to decoder. Four general classes for illustrative purposes shown in Figure 3 [11]. The first is the digital end-to-end environment in Figure 3.1. This is the environment of a sound file that is copied from machine to machine, but never modified in any way. As a result, the sampling is exactly the same at the encoder and decoder. This class puts the least constraints on data-hiding methods [11, 12]. The next consideration is when a signal is re sample to a higher or lower sampling rate, but remains digital throughout shown in Figure 3.2 This transform preserves the absolute magnitude and phase of most of the signal, but changes the temporal characteristics of the signal. The third case is when a signal is played into an analog state, transmitted on a reasonably clean analog line and re sample given in Figure 3.3 Absolute signal magnitude, sample quantization, and temporal sampling rate are not preserved. In general, phase will be preserved. The last case is when the signal is played into the

air and re sample with a microphone given in Figure 3.4 The signal will be subjected to possibly unknown nonlinear modifications resulting in phase changes, amplitude changes, drift of different frequency components, echoes, etc. Signal representation and transmission pathway must be considered when choosing a data-hiding method. Data rate is very dependent on the sampling rate and the type of sound being encoded. A typical value is 16 bps, but the number can range from 2 bps to 128 bps.

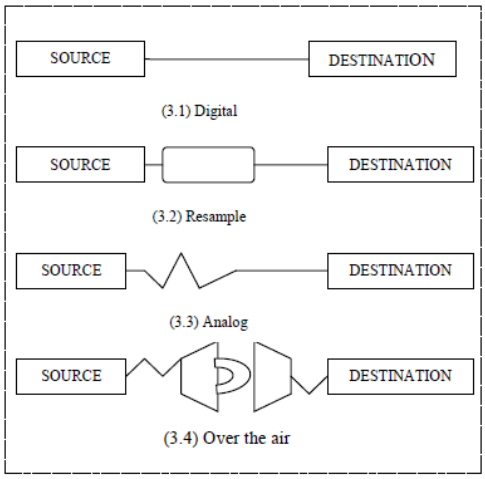

Fig. 3. Data Transmission Medium

3. Techniques of Data hiding in audio

3.1. *Least Significant Bit Encoding*

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. 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 [13, 16]. Thus, one should consider the signal content before deciding on the LSB operation to use. For example, a sound file that was recorded in a bustling subway station would mask low-bit encoding noise. On the other hand, the same noise would be audible in a sound file containing a piano solo. To extract

a secret message from an LSB encoded sound file, the receiver needs access to the sequence of sample indices used in the embedding process. Normally, the length of the secret message to be encoded is smaller than the total number of samples in a sound file. One must decide then on how to choose the subset of samples that will contain the secret message and communicate that decision to the receiver. One trivial technique is to start at the beginning of the sound file and perform LSB coding until the message has been completely embedded, leaving the remaining samples unchanged. This creates a security problem, however in that the first part of the sound file will have different statistical properties than the second part of the sound file that was not modified. One solution to this problem is to pad the secret message with random bits so that the length of the message is equal to the total number of samples. Yet now the embedding process ends up changing far more samples than the transmission of the secret required. This increases the probability that a would-be attacker will suspect secret communication. A more sophisticated approach is to use a pseudo random number generator to spread the message over the sound file in a random manner. One popular approach is to use the random interval method, in which a secret key possessed by the sender is used as a seed in a pseudo random number generator to create a random sequence of sample indices. The receiver also has access to the secret key and knowledge of the pseudo random number generator, allowing the random sequence of sample indices to be reconstructed. Checks must be put in place, however, to prevent the pseudo random number generator from generating the same sample index twice. If this happened, a collision would occur where a sample already modified with part of the message is modified again. The problem of collisions can be overcome by keeping track of all the samples that have already been used. Another approach is to calculate the subset of samples via a pseudo random permutation of the entire set through the use of a secure hash function. This technique insures that the same index is never generated more than once. 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 this is not robust. If a sound file embedded with a secret message using either LSB coding was re sample, 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.

3.2. *Phase Coding*

Phase coding [11, 16] 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 4.

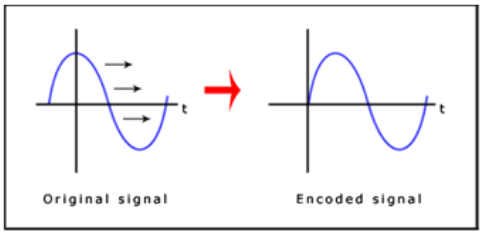

Fig. 4. The original signal and encoded signal of phase coding technique.

Phase coding is explained in the following procedure:

- 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:

$$phase\_new = \begin{cases} \pi/2 & if \ \ message \ \ bit = 0 \\ -\pi/2 & if \ \ message \ \ bit = 1 \end{cases}$$

- 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.

To extract the secret message from the sound file, the receiver must know the segment length. The receiver can then use the DFT to get the phases and extract the information. One disadvantage associated with phase coding is a low data transmission rate due to the fact that the secret message is encoded in the first signal segment only. This might be addressed by increasing the length of the signal segment. However, this would change phase relations between each frequency component of the segment more drastically, making the encoding easier to detect. As a

result, the phase coding method is used when only a small amount of data, such as a watermark, needs to be concealed.

3.3. *Echo Hiding*

In echo hiding [14, 15, 16], 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. If only one echo was produced from the original signal, only one bit of information could be encoded. Therefore, the original signal is broken down into blocks before the encoding process begins. Once the encoding process is completed, the blocks are concatenated back together to create the final signal.

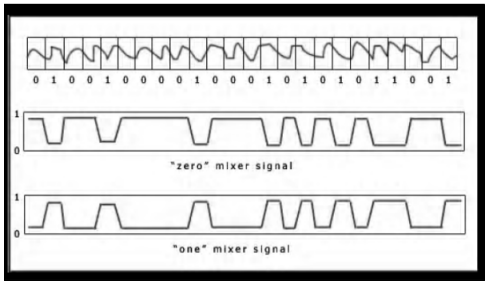

Fig. 5. Echo Hiding Methodology.

To extract the secret message from the stego-signal, the receiver must be able to break up the signal into the same block sequence used during the encoding process. Then the autocorrelation function of the signal's cepstrum (the cepstrum is the Forward Fourier Transform of the signal's frequency spectrum) can be used to decode the message because it reveals a spike at each echo time offset, allowing the message to be reconstructed.

3.4. *Spread Spectrum*

In the context of audio steganography, the basic spread spectrum (SS) [16] 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. However, unlike LSB coding, the SS method spreads the secret message over the sound file's frequency spectrum, using a code that is independent of the actual signal. As a result, the final signal occupies a bandwidth in excess of what is actually required for transmission. Two versions of SS can be used in audio steganography: the direct-sequence and frequency-hopping schemes. In direct sequence SS, the secret message

is spread out by a constant called the chip rate and then modulated with a pseudo random signal. It is then interleaved with the cover-signal. In frequency-hopping SS, the audio file's frequency spectrum is altered so that it hops rapidly between frequencies. The SS method has the potential to perform better in some areas than LSB coding, parity coding, and phase coding techniques in that it offers a moderate data transmission rate while also maintaining a high level of robustness against removal techniques. However, the SS method shares a disadvantage with LSB and parity coding in that it can introduce noise into a sound file.

4. Proposed Scheme

In this section the authors propose a new method for imperceptible audio data hiding for an audio file of **wav** or **mp3** format.This approach based on the Mod 16 Method(M16M) [10] named Mod 4 Method(M4M) along with a Number Sequence Generator Algorithm to avoid embedding data in the consecutive indexes of the audio, which will eventually help in avoiding distortion in the audio quality.The input messages can be in any digital form, and are often treated as a bit stream. Embedding positions are selected based on some mathematical function which depends on the data value of the digital audio stream. Data embedding is performed by mapping each two bit of the secret message in each of the seed position,based on the remainder of the intensity value when divided by 4. Extraction process starts by selecting those seed positions required during embedding. At the receiver side other different reverse operation has been carried out to get back the original information.

4.1. *Data Embedding Method*

Mod 4 Method(M4M) Sending Algorithm is described as:

- Input : Sampled Audio Data Matrix(a), Message.
- msg = Message converted to binary ;
- Initialize m = n = cnt = x =1 and l=cnt;
- Begin for loop starting with i=1, incrementing 2 and till msize;
- Increment cnt and l by 1 and assign i to count;
- msg0=0; msg1=1;
- let cvr contains the value at a(m,n);
- it cvr is negative then sgn = -1 else sgn = 1;
- R is the absolute remainder after dividing cvr by 4;
- msgx1=binmsg(count) and increment count by 1;
- msgx2=binmsg(count) and increment count by 1;
- If(msgx1=msg0 and msgx2=msg0)
- cvr = cvr - R;
- Elseif (msgx1=msg0 and msgx2=msg1)
- cvr = cvr - R + 1;
- Elseif (msgx1=msg1 and msgx2=msg0)

- cvr = cvr - R+ 2;
- Elseif (msgx1=msg1 and msgx2=msg1)
- cvr = cvr - R +3;
- Divide cvr by 1000;
- If sgn = -1 then cvr = cvr * -1;
- Set the value of cvr at a(m,n);
- Let r be the remainder after dividing x by 4;
- If r = val then m = m + r+1; where val = 0, 1 , 2 and 3;
- x = x + 1;
- End For loop

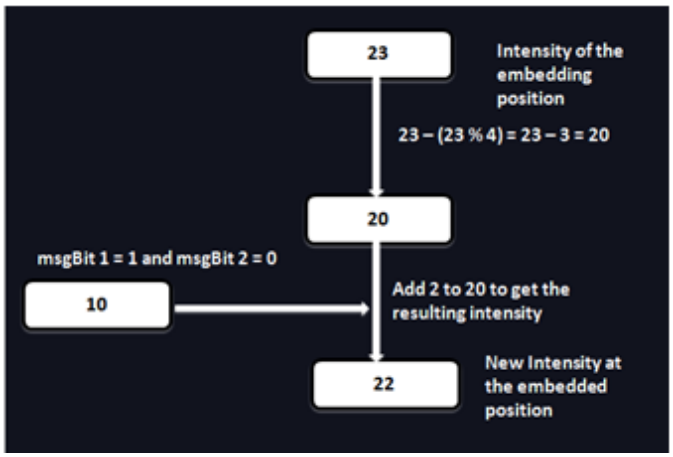

Fig. 6. A snapshot of data embedding process.

4.2. *Data Extraction Method*

The process of extraction proceeds by selecting those same seed positions. The extracting process will be finished when all the bits of every bytes of secret message are extracted.Mod 4 Method(M4M) Receiving Algorithm is described as:

- Input: Sampled Audio Matrix(a), Message size
- Initialize m = n = x = count =1;binmsg1=";
- Begin for loop starting with i=1, incrementing 2 and till msgsize
- V = value of a(m,n) ;
- let R be the remainder after dividing V by 4;
- if(R==0)
- binmsg1(count)=char (0);
- count=count+1;

- binmsg1(count)=char (0);
- count=count+1;
- elseif(R==1)
- binmsg1(count)= char(0);
- count=count+1;
- binmsg1(count)= char(1);
- count=count+1;
- elseif(R==2)
- binmsg1(count)= char(0);
- count=count+1;
- binmsg1(count)= char (1);
- count=count+1;
- elseif(R==3)
- binmsg1(count)= char(0);
- count=count+1;
- binmsg1(count)= char (1);
- count=count+1;
- End if
- Let R1 be the remainder after dividing x by 4;
- If R1 = val then m = m + R1 +1; where val = 0, 1 , 2 and 3;
- x = x + 1;
- Initialize msgx=msg1="and k=0;
- Begin for Loop with incrementing i by 1 and till Message size
- Begin for Loop with incrementing j by 1 and till 8
- Increment k by 1;
- msgx(j) = char(binmsg1(k));
- End For loop
- End For loop

5. Solution Methodology

The proposed system consists of following two windows, one at the SENDER SIDE and the other at the RECEIVER SIDE.The user should be able to select secret message as a file, another audio file has to be used as the carrier (cover audio) and then use the proposed M4M method, which will hide the selected message in the selected carrier audio and will form the stego audio. The user at the receiver side should be able to extract the secret message from the stego audio with the help of different reverse process in sequential manner.

5.1. *Computer Algorithm*

In this section the two algorithimic approach is discussed one for the function of the Sender Side and another for the Receiver Side.

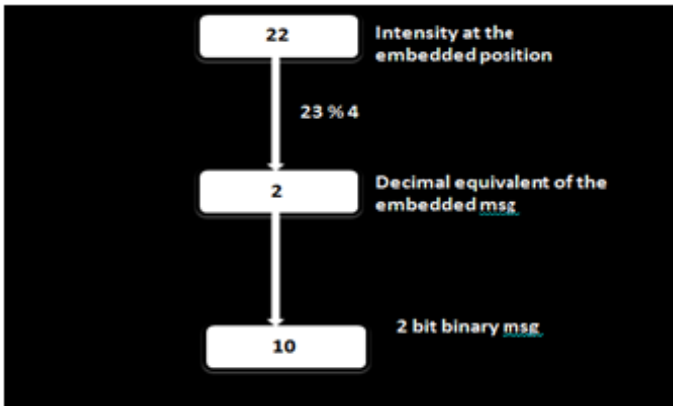

Fig. 7. A snapshot of data extraction process.

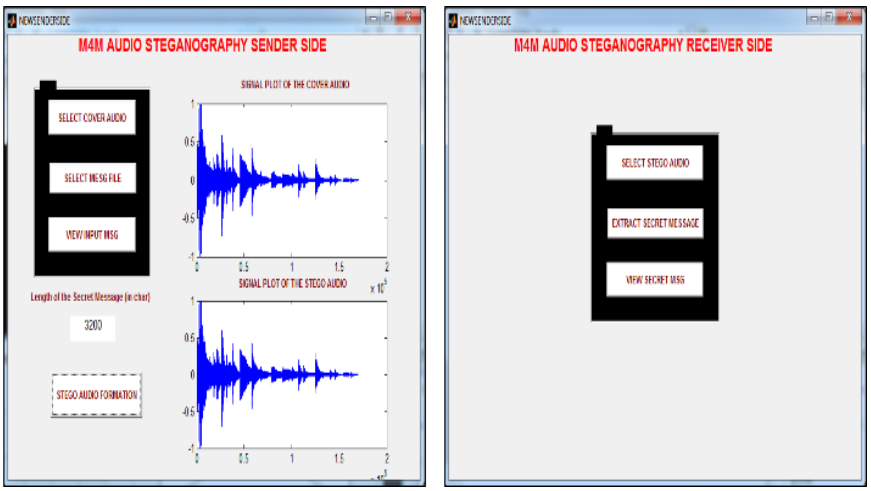

Fig. 8. GUI of the proposed steganography system.

5.1.1. *Sender Side*

- Select the cover audio from the set of audio files.
- Select the secret message in text form.
- Embed the secret message through the M4M sending method to generate the Stego Audio.

5.1.2. *Receiver Side*

- Select the stego audio from the set of audio files.
- Extract the secret message through the M4M receiving method.

- Display the secret message.

6. System Evaluation

In this section the authors present the experimental result of the proposed method based on two benchmarks techniques to evaluate the hiding performance. First one is the capacity of hiding data and another one is the imperceptibility of the stego audio, also called the quality of stego audio. The quality of stego-audio should be acceptable by human ears. The authors also present detailed study of the proposed method by computing embedding capacity, mean square error (MSE) and peak signal-to noise ratio (PSNR).In this section experimental result of stego audio are shown based on two audio formats viz. wav and mp3, having a total of different six audio files, three of each format. Fig. 8 show the length and maximum embedding capacity of each of the audio files.

Audio file	Length	Maximum Embedding Capacity
chimes.wav	00:00:07	8498
heartbeat.wav	00:00:13	31583
johncena.wav	00:01:51	38850
gaanwala.mp3	00:02:45	139308
jagorone.mp3	00:03:46	188440
yaaron.mp3	00:04:25	212901

Fig. 9. Maximum Embedding Capacity Varying with format and size of the audio.

6.1. *Peak Signal to Noise Ratio (PSNR)*

PSNR measures the quality of the image by comparing the original image or cover image with the stego-image, i.e. it measures the percentage of the stego data to the image percentage. The PSNR is used to evaluate the quality of the stego-image after embedding the secret message in the cover. Assume a cover image C(i,j) that contains N by N pixels and a stego image S(i,j) where S is generated by embedding / mapping the message bit stream. Mean squared error (MSE) of the stego image as follows:

$$MSE = \frac{1}{[N \times N]} \sum_{i=1}^{N} \sum_{j=1}^{N} [C(ij) - S(ij)]^2$$

The PSNR is computed using the following formulae:

$$PSNR = 10 \log_{10} 255^2 / \ MSE \ db.$$

Audio		Data Size					
		100	500	1000	2500	5000	10000
chimes.wav	PSNR	67.1593	60.8245	57.9941	54.0638	51.0484	N.A
	MSE	0.0125	0.0538	0.0538	0.2551	0.5108	
heartbeat.wav	PSNR	74.0891	66.8539	63.7689	59.6869	56.7102	53.6966
	MSE	0.0025	0.0134	0.0273	0.0699	0.1387	0.2776
johncena.wav	PSNR	73.1188	67.0395	64.2669	60.3983	57.4135	54.4213
	MSE	0.0032	0.0129	0.0243	0.0593	0.1180	0.2349
gaanwala.mp3	PSNR	79.3587	72.9449	70.0694	66.1954	63.1914	60.1326
	MSE	7.5372e-004	0.0033	0.0064	0.0156	0.0312	0.0631
jagorone.mp3	PSNR	80.6707	74.2569	71.3813	67.5074	64.5033	61.4194
	MSE	5.5721e-004	0.0024	0.0047	0.0115	0.0231	0.0469
yaaron.mp3	PSNR	80.4089	74.1413	71.5498	67.7883	64.8967	61.9353
	MSE	5.9182e-004	0.0025	0.0046	0.0108	0.0211	0.0416

Fig. 10. PSNR and MSE values of six audio files at different message sizes.

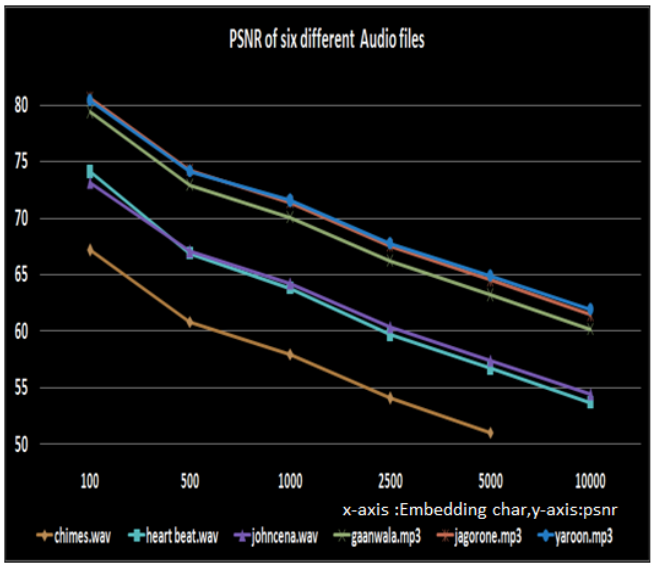

Fig. 11. PSNR of different audio.

6.2. *Similarity Measure Of The Cover Audio And Stego Audio Through Correlation*

The most familiar measure of dependence between two quantities is the Pearson product-moment correlation coefficient [32], or"Pearson's correlation." It is obtained by dividing the covariance of the two variables by the product of their standard deviations. Karl Pearson developed the coefficient from a similar but slightly different idea by Francis Galton. The Pearson correlation is +1 in the case of a perfect positive (increasing) linear relationship (correlation), -1 in the case of a perfect decreasing (negative) linear relationship (anti correlation) , and some value between -1 and 1

in all other cases, indicating the degree of linear dependence between the variables. As it approaches zero there is less of a relationship (closer to uncorrelated). The closer the coefficient is to either -1 or 1, the stronger the correlation between the variables. If the variables are independent, Pearson's correlation coefficient is 0, but the converse is not true because the correlation coefficient detects only linear dependencies between two variables. If we have a series of n measurements of X and Y written as xi and y_i where i = 1,2,,n then the sample correlation coefficient can be used in Pearson correlation r between X and Y. The sample correlation coefficient is written as where and are the sample means of X and Y, s_x and s_y are the sample

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$$

standard deviations of X and Y.

Secret Message Size(in char)	Cover Audio	Correlation-Coefficient
100	Chimes.wav	1.000
500	Chimes.wav	0.9999
1000	Chimes.wav	0.9994
2500	Chimes.wav	0.9980
5000	Chimes.wav	0.9976
8000	Chimes.wav	0.9970
10000	heartbit.wav	0.9951
10000	gaanwala.mp3	0.9950

Fig. 12. Similarity Measure of The Cover and Stego Through Correlation

6.3. *Comparison of M4M with other Audio Steganography Methods*

- No previous work focuses on keeping the increasing size of the embedding capacity and similarity between cover audio and stego audio generated based on different message sizes.
- The human ear is very sensitive and can often detect even the slightest bit of noise introduced into a sound file and most of the existing methods does not come much closer to making the introduced noise inaudible.

Proposed M4M method in Audio steganography has been designed keeping in mind to overcome the above mentioned short comings.

- Embedding capacity has been increased by mapping of two bits at a time instead of one.

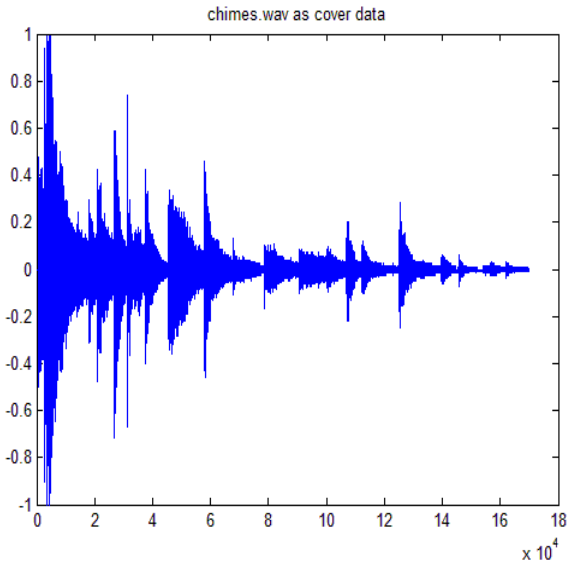

Fig. 13. Signal plotting of the cover audio Chimes.wav.

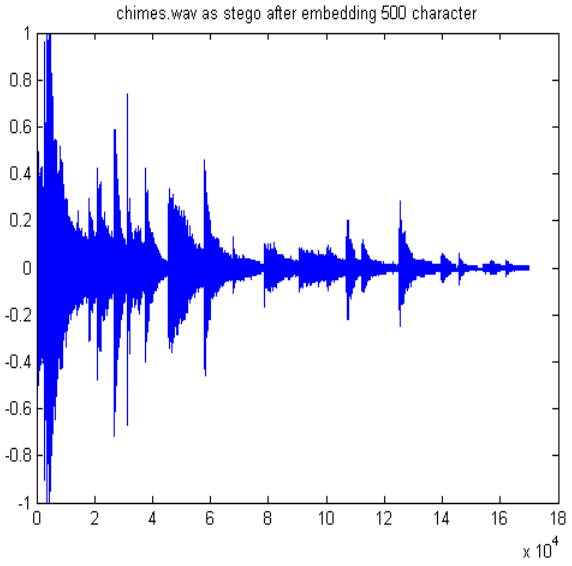

Fig. 14. Signal plotting of the Chimes.wav after embedding 500 char.

- Similarity measure between cover audio and stego audio has been inducted here through correlation method and this method is capable of producing

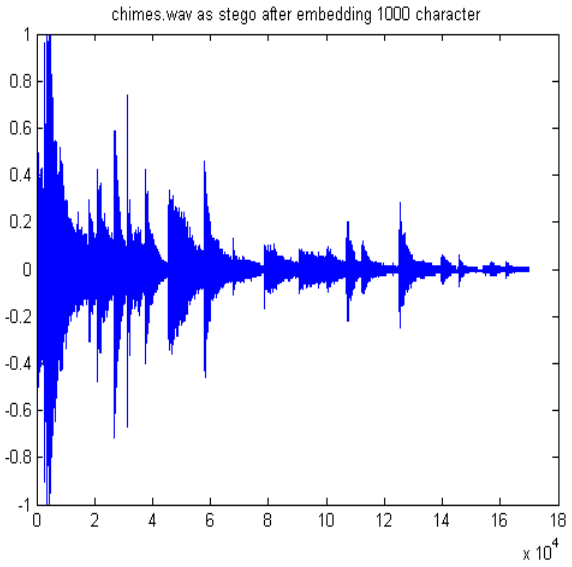

Fig. 15. Signal plotting of the Chimes.wav after embedding 1000 char.

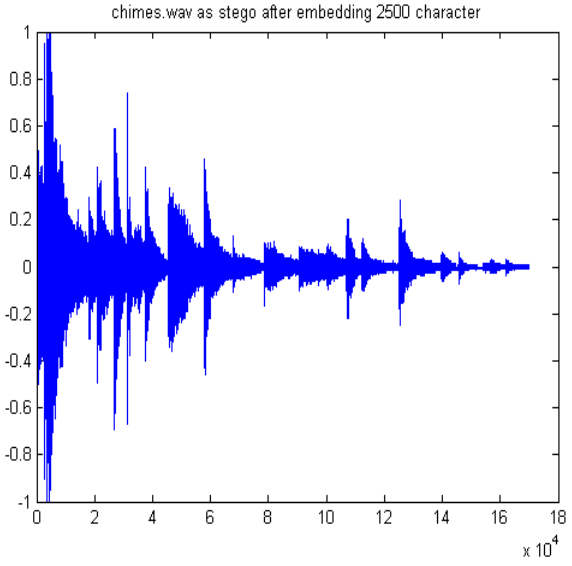

Fig. 16. Signal plotting of the Chimes.wav after embedding 2500 char.

stego audio with minimum or zero degradation.

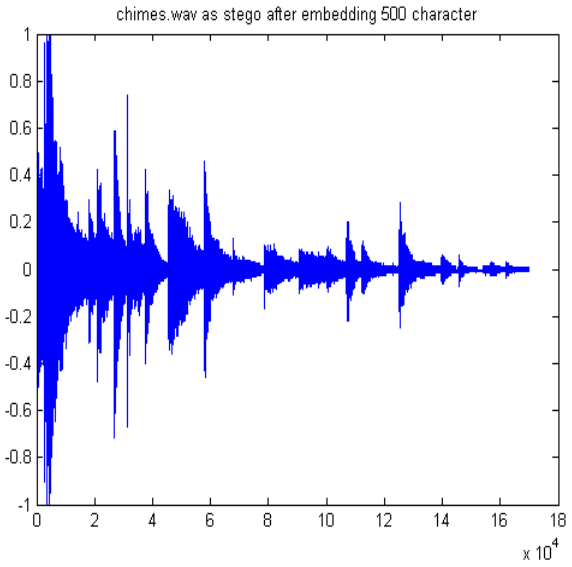

Fig. 17. Signal plotting of the Chimes.wav after embedding 5000 char.

7. Conclusion

In this paper author's have introduced a new and efficient method of imperceptible audio data hiding of wav or mp3 format. Comparison has been shown with some other existing methods also. From the experimental results in can be seen that the embedding capacity of the proposed method is better compared to the other methods because this method can map each two bit of the secret message in the embedding positions instead of one of those existing methods.From the similarity measure point of view the proposed method can be considered as best by producing stego audio with minimum or zero degradation in terms of audio quality as well as the size of the audio.Figure [13-17] shows the signal level of the audio **chimes.wav** as a cover and at various embedding rate. Also as the message bits are not directly embedded,steganalysis may be able to find out the embedded bits but can not be able to extract the original message bits.Besides PSNR value of the proposed method for various size of the secret message is very good. This system is to provide a good, efficient method for hiding the data from vernal able effect of hostile eavesdropping, theft, wiretapping etc.Although this method has been designed for wav and mp3 format but this method can be extended for any type of audio file format.

References

[1] Gustavus J. Simmons, The Prisoners' Problem and the Subliminal Channel, *Proceedings of CRYPTO*,**83**(1984) 51–57.

[2] RJ Anderson, Stretching the Limits of Steganography, *Information Hiding, Springer Lecture Notes in Computer Science*,**1174** (1996) 39–48.
[3] Scott. Craver, On Public-key Steganography in the Presence of an Active Warden, *Proceedings of 2nd International Workshop on Information Hiding.*, (1998) 355–368.
[4] Ross J. Anderson. and Fabien A.P.Petitcolas,On the limits of steganography, *IEEE Journal on Selected Areas in Communications (J-SAC), Special Issue on Copyright and Privacy Protection*,**16**(1998) 474–481.
[5] N.F.Johnson. and S. Jajodia, Steganography: seeing the unseen, in *IEEE Computer*, **16**(1998) 26–34.
[6] T Mrkel., JHP Eloff. and MS Olivier, An Overview of Image Steganography, in *Proceedings of the fifth annual Information Security South Africa Conference*, (2005).
[7] Souvik Bhattacharyya. and Gautam Sanyal, Study of Secure Steganography model, in *Proceedings of International Conference on Advanced Computing and Communication Technologies*,(2008).
[8] Souvik Bhattacharyya. and Gautam Sanyal, Implementation and Design of an Image based Steganographic model, in *Proceedings of IEEE International Advance Computing Conference*,(2009).
[9] Souvik Bhattacharyya. and Gautam Sanyal,An Image based Steganography model for promoting Global Cyber Security, in *Proceedings of International Conference on Systemics, Cybernetics and Informatics*,(2009).
[10] Arko Kundu, Kaushik Chakraborty and Souvik Bhattacharyya,Data Hiding in Images Using Mod 16 Method, in *In the Proceedings of ETECE 2011*,(2011)
[11] W. Bender. and D. Gruhl, Steganography: Techniques for data hiding, in *IBM SYSTEMS JOURNAL*, **35**(1996).
[12] Fabien A. P. Pettitcolas, Ross J. Anderson, and Markus G. Kuhn,Information Hiding-A Survey, in *Proceedings of The IEEE*, **87**(1999).
[13] Nedeljko Cvejic and Tapio Seppben,Increasing the capacity of LSB-based audio steganography, in *IEEE 2002*, (2002).
[14] Samir Kumar Bandyopadhyay, Debnath Bhattacharyya, Poulami Das, Debashis Ganguly and Swarnendu Mukherjee,A tutorial review on Steganography, in *In the Proceedings of International Conference on Contemporary Computing*, (2008).
[15] Souvik Bhattacharyya, Indradip Banerjee and Gautam Sanyal,A Survey of Steganography and Steganalysis Technique in Image, Text, Audio and Video as Cover Carrier, in *Journal of Global Research in Computer Science (JGRCS) VOL 2, NO 4 (2011) APRIL-2011.*
[16] Natarajan Meghanathan and Lopamudra Nayak,STEGANALYSIS ALGORITHMS FOR DETECTING THE HIDDEN INFORMATION IN IMAGE, AUDIO AND VIDEO COVER MEDIA, in *at International Journal of Network Security & Its Application (IJNSA), Vol.2, No.1, January 2010.*