A Novel Audio Steganography Technique by M16MA

Souvik Bhattacharyya
Department of CSE
University Institute of Technology,
The University of Burdwan
West Bengal, India

Arko Kundu
Department of CSE
Bengal Engineering and Science University
West Bengal, India

Gautam Sanyal
Department of CSE
National Institute of Technology, Durgapur
West Bengal, India

ABSTRACT

Though security is nothing new, the way that security has become a part of our daily life is unprecedented. Attacks, misuse or unauthorized access of information is of great concern today which makes the protection of documents through digital media is a priority problem. This urges users to devise new data hiding techniques through steganography principle to protect and secure the data of vital significance. Considerable amount of work has been carried out by different researchers on steganography. In this work the authors propose a novel audio based steganographic method for wav and mp3 format for hiding information. The proposed approach works by selecting the embedding positions using some mathematical function and maps each four 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.

General Terms

Audio Steganography, Information Security.

Keywords

Cover Audio, Mod 16 Method for Audio (M16MA), Stego Audio.

1. INTRODUCTION

Steganography is the art and science of hiding information by embedding messages with in 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 is advised to see [5], [6].Some Steganographic model with high security features has been presented in [7-9]. 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.

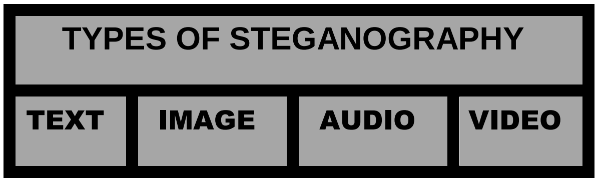


Figure 1: Types of Steganography

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.

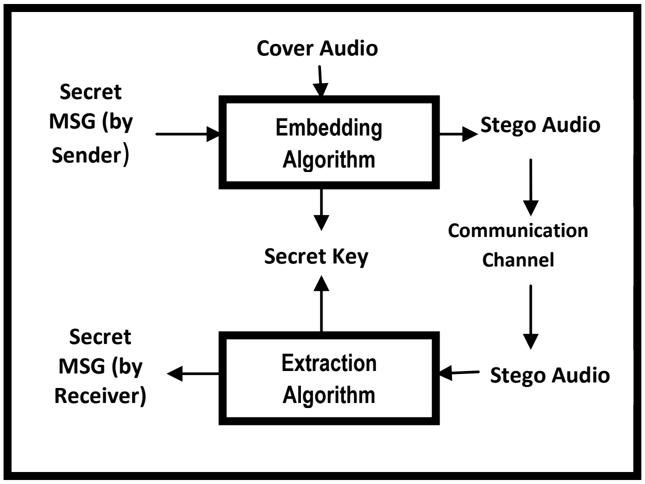


Figure 2: Generic form of Audio Steganography

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.

This paper has been organized as following sections: Section II describes basics of audio steganography. Section III reviews the previous work on audio steganography. Section IV describes the proposed scheme along with data embedding and extraction methodology .Section V presents the solution methodology along with the system algorithm. Evaluation of the system has been done in Section VI and Section VII draws the conclusion.

2. AUDIO STEGANOGRAPHY

The audio files may be used for information hiding like other digital media such as image, text or video. The HAS (Human Auditory System) property of any audio carrier enables itself for information hiding. The HAS perceives the additive random noise and also the perturbations in an audio file can also be detected. Although the HAS have a large dynamic range, but it has also a fairly small differential range which makes the loud sounds to mask out the quiet sounds. The process of digitization involves two steps sampling and quantization. In sampling the analogue values are only captured at regular time intervals where as 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 audio file formats suitable for information hiding are Windows Audio-Visual (WAV) format and the Audio Interchange File Format (AIFF). The two main areas of modification in an audio file for data embedding are the storage environment, or digital representation of the signal that will be used, and the transmission pathway the signal might travel [4, 11]. In order to conceal secret messages successfully, a variety of methods for embedding information in digital audio have been presented by different researchers. 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.

2.1 Digital Representation of Audio

Sample quantization method and temporal sampling rate are the two critical parameters involved in digital audio representations. Windows Audio-Visual (WAV) and Audio Interchange File Format (AIFF) [4, 11] uses the most popular format for representing samples of high-quality digital audio namely 16-bit linear quantization.8-bit m-law which is logarithmically scaled is another popular format for lower quality audio. These quantization methods introduce some signal distortion, which is more evident in the case of 8-bit m-law. Some 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. For most of the data-hiding techniques usable data embedding space increases with the increased rate of sampling [11].

There are mainly four different transmission environments [11, 14, 21] that a signal might experience on its way from the source to the destination as shown in figure 3. **Figure 3.1** depicts the first case which is the digital end-to-end transmission environment between the sender and receiver. This class puts the least constraints on data-hiding methods [11, 12]. The next case is considered when a signal needs to be re-sampling to a higher or lower sampling rate, but remains digital throughout. This situation is 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. **Figure 3.3** shows the case is when a signal works into an analog state, transmitted through an analog line and needs to be re-sampled. In this case although the absolute signal magnitude, sample quantization, and temporal sampling rate are not preserved but the phase will be preserved. The last case is when the signal is transmitted into the air and re-sample with a microphone which shown in **figure 3.4**. Here the signal will be subjected to possibly unknown nonlinear modifications resulting in phase changing, amplitude changing, drift of different frequency components, echoes, etc.

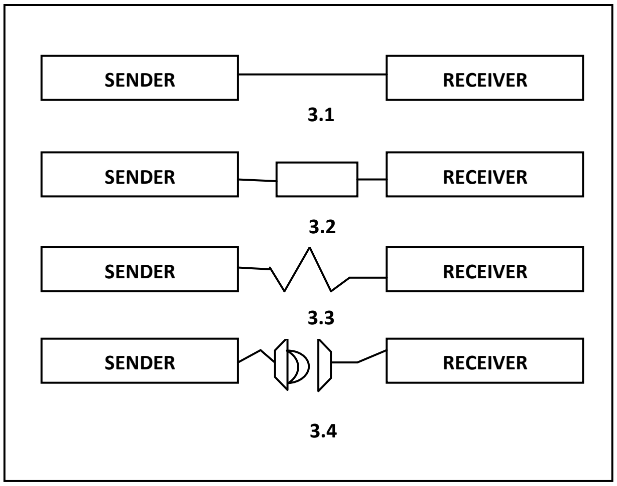


Figure 3: Data Transmission Medium

3. REVIEW OF AUDIO DATA HIDING TECHNIQUES

This section presents some existing techniques of audio data hiding namely Least Significant Bit Encoding, Phase Coding, Echo Hiding and Spread Spectrum techniques. There are two main areas of modification in an audio for data embedding. First, the storage environment, or digital representation of the signal that will be used, and second the transmission pathway the signal might travel [4, 11].

3.1 Least Significant Bit Encoding

The simple way of embedding the information in a digital audio file is done through Least significant bit (LSB) coding. By substituting the least significant bit of each sampling point with a binary message bit, LSB coding allows a data to be encoded and produces the stego audio. In LSB coding, the ideal data transmission rate is 1 kbps per 1 kHz. The main disadvantage of LSB coding is its low embedding capacity. In some cases an attempt has been made to overcome this situation by replacing the two least significant bits of a sample with two message bits. This increases the data embedding capacity but also increases the amount of resulting noise in the audio file as well. A novel method of LSB coding which increases the limit up to four bits is proposed by Nedeljko Cvejic Et al. [13, 16]. To extract secret message from an LSB encoded audio, the receiver needs access to the sequence of sample indices used during the embedding process. Normally, the length of the secret message to be embedded is smaller than the total number of samples done.

There are other two disadvantages also associated LSB coding. The first one is that human ear is very sensitive and can often detect the presence of single bit of noise into an audio file. Second disadvantage however, is that LSB coding is not very robust. Embedded information will be lost through a little modification of the stego audio.

3.2 Phase Coding

Phase coding [11, 16] overcomes the disadvantages of noise induction method of audio steganography. Phase coding designed based on the fact that the phase components of sound are not as audible to the human ear as noise is. This method embeds the message bits as phase shifts in the phase spectrum of a digital signal, achieving an inaudible encoding in terms of signal-to-noise ratio. In figure 4 below original audio signal and audio signal with message through phase coding method has been presented.

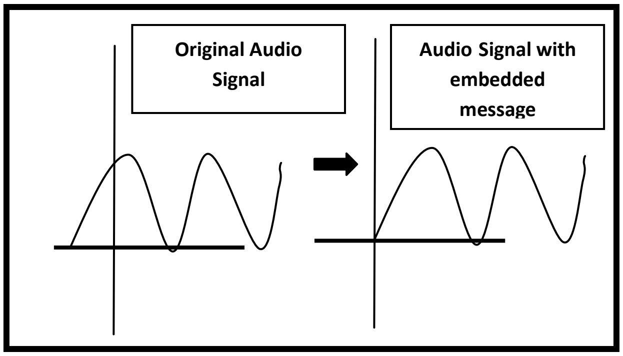


Figure 4: The original signal and encoded signal of phase coding technique.

Phase coding principles are summarized as under:

- The original audio signal is broken up into smaller segments whose lengths equal the size of the message to be embedded.
- 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 next.
- 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.

Thus the secret message is only inserted in the phase vector of the first signal segment as follows:

$$phase_{new} = \begin{cases} \frac{\pi}{2} \; if \; message \; bit \; is \; 0 \\ -\frac{\pi}{2} if \; message \; bit \; is \; 1 \end{cases}$$

- 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 audio signal is reconstructed by applying the inverse DFT and by concatenating the audio segments.

To extract the secret message from the audio file, the receiver needs to know the segment length. The receiver can extract the secret message through different reverse process.

The disadvantage associated with phase coding is that it has a low data embedding rate due to the fact that the secret message is encoded in the first signal segment only. This situation can be overcome by increasing the length of the signals segment which in turn increases the change in the phase relations between each frequency component of the segment more drastically, making the encoding easier to detect. Thus, the phase coding method is useful only when a small amount of data, such as a watermark, needs to be embedded.

3.3 Echo Hiding

In echo hiding [14, 15, 16] method information is embedded into an audio file by inducing an echo into the discrete signal. Like the spread spectrum method, Echo Hiding method also has the advantage of having high embedding capacity with superior robustness 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 form the final signal.

To extract the secret message from the final stego audio 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 which 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.

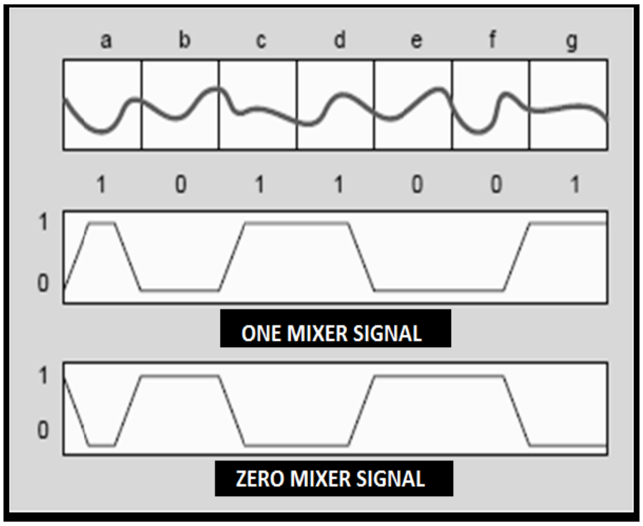


Figure 5: Echo Hiding Methodology.

3.4 Spread Spectrum

Spread Spectrum (SS) [16] methodology attempts to spread the secret information across the audio signal's frequency spectrum as much as possible. This is equivalent to a system using the LSB coding method which randomly spreads the message bits over the entire audio file. The difference is that unlike LSB coding, the SS method spreads the secret message over the audio file's frequency spectrum, using a code which is independent of the actual signal. As a result, the final signal occupies a more bandwidth than actual requirement for embedding. Two versions of SS can be used for audio steganography one is the direct sequence where the secret message is spread out by a constant called the chip rate and then modulated with a pseudo random signal where as in the second method frequency-hopping SS, the audio file's frequency then interleaved with the cover-signal spectrum is altered so that it hops rapidly between frequencies. The Spread Spectrum method has a more embedding capacity compared to LSB coding and phase coding techniques with maintaining a high level of robustness. However, the SS method shares a disadvantage common with LSB and parity coding that it can introduce noise into the audio file at the time of embedding.

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] designed for image named Mod 16 Method for audio (M16MA) 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 de-ends on the data value of the digital audio stream. Data embedding is performed by mapping each four bit of the secret message in each of the seed position, based on the remainder of the intensity value when divided by 16. 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 16 Method (M16MA) Sending Algorithm is described as

- Input: Audio Data Matrix(a), Message
- msg = Message converted to binary from decimal with at least 8 bits;
- Initialize m=n=x=count=1;
- **Begin for loop** starting with i=1, incrementing 4 till i=size of the binary message;
- msg0=0; msg1=1;
- let cvr contains the value at a(m,n);
- If cvr is negative then sgn = -1 else sgn = 1;
- **r is the remainder after dividing cvr by 16;**
- msgx1=binmsg(count) and increment count by 1;
- msgx2=binmsg(count) and increment count by 1; msgx3=binmsg(count) and increment count by 1;
- msgx4=binmsg(count) and increment count by 1;
- If(msgx1=msg0 and msgx2=msg0 and msgx3=msg0 and msgx4=msg0)
- cvr = cvr – r;
- Else If(msgx1=msg0 and msgx2=msg0 and msgx3=msg0 and msgx4=msg1)
- cvr = cvr – r+1;
- Else If(msgx1=msg0 and msgx2=msg0 and msgx3=msg1 and msgx4=msg0)
- cvr = cvr – r+2;
- Else If(msgx1=msg0 and msgx2=msg0 and msgx3=msg1 and msgx4=msg1)
- cvr = cvr – r+3;
- Else If(msgx1=msg0 and msgx2=msg1 and msgx3=msg0 and msgx4=msg0)
- cvr = cvr – r+4;
- Else If(msgx1=msg0 and msgx2=msg1 and msgx3=msg0 and msgx4=msg1)
- cvr = cvr – r+5;
- Else If(msgx1=msg0 and msgx2=msg1 and msgx3=msg1 and msgx4=msg0)
- cvr = cvr – r+6;
- Else If(msgx1=msg0 and msgx2=msg1 and msgx3=msg1 and msgx4=msg1)
- cvr = cvr – r+7;
- Else If(msgx1=msg1 and msgx2=msg0 and msgx3=msg0 and msgx4=msg0)
- cvr = cvr – r+8;
- Else If(msgx1=msg1 and msgx2=msg0 and msgx3=msg0 and msgx4=msg1)
- cvr = cvr – r+9;
- Else If(msgx1=msg1 and msgx2=msg0 and msgx3=msg1 and msgx4=msg0)
- cvr = cvr – r+10;
- Else If(msgx1=msg1 and msgx2=msg0 and msgx3=msg1 and msgx4=msg1)
- cvr = cvr – r+11;
- Else If(msgx1=msg1 and msgx2=msg1 and msgx3=msg0 and msgx4=msg0)

- cvr = cvr – r+12;
- Else If(msgx1=msg1 and msgx2=msg1 and msgx3=msg0 and msgx4=msg1)
- cvr = cvr – r+13;
- Else If(msgx1=msg1 and msgx2=msg1 and msgx3=msg1 and msgx4=msg0)
- cvr = cvr – r+14;
- Else If(msgx1=msg1 and msgx2=msg1 and msgx3=msg1 and msgx4=msg1)
- cvr = cvr – r+15;
- If sgn = -1 then cvr = cvr * -1;
- Set the value of cvr at a(m,n);
- **End For loop**

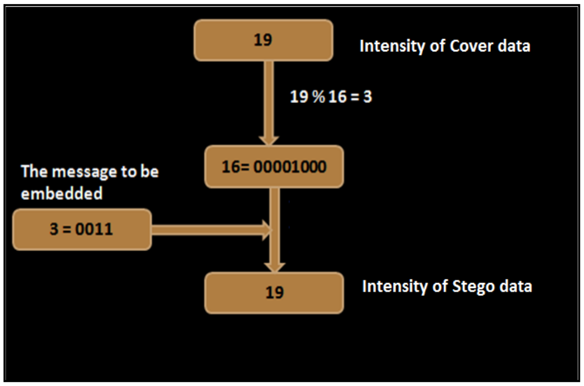


Figure 6: A snapshot of data embedding process.

4.2 Data Extraction Method

Mod 16 Method for Audio (M4MA) Receiving Algorithm is described as: Input: Sampled Audio Matrix (a), Message size

- Input: Stego Audio Data Matrix(a), Message size
- Initialize m = n = x = count =1;binmsg1=’’;
- **Begin for loop** starting with i=1, incrementing 4 and till entire message is retrieved
- let r be the remainder after dividing v by 16;
- v = value of a(m,n)
- **if**(r==BI) where BI varies from 0 to 15 **then** concatenate the 4 bit binary equivalent of BI to binmsg1.
- **End if**
- **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**
- Initialize msgx=msg1=’’;k=0;
- **Begin for Loop** with i=1, incrementing 1 and till Message size
- **Begin for Loop** with j=1, incrementing 1 and till 8
- Increment k by 1;
- msgx(j)= character equivalent of (binmsg1(k));
- **End For loop**
- **End For Loop**

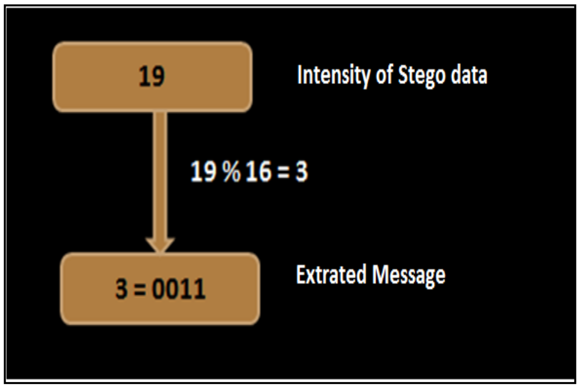


Figure 7: A snapshot of data extraction process.

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 M16MA 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.

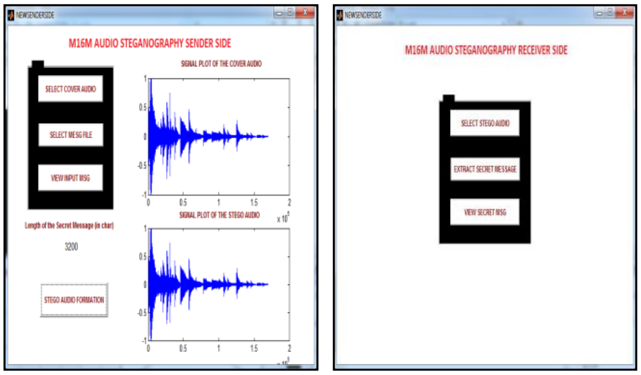


Figure 8: GUI of the proposed steganography system.

5.1 Computer Algorithm

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

Sender Side

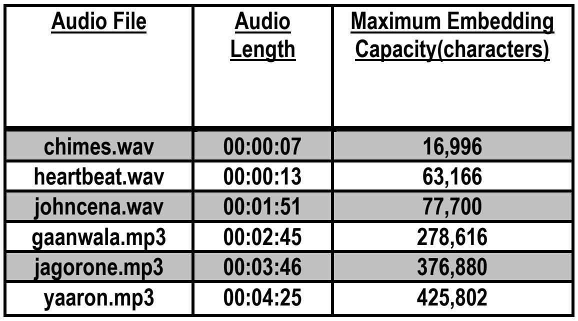
- Select the cover audio from the set of audio files.
- Select the secret message in text form.
- Embed the secret message through the M16MA sending method to generate the Stego Audio.

Receiver Side

- Select the stego audio from the set of audio files.
- Extract the secret message through the M16MA 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 data hiding 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. 9 shows the length and maximum embedding capacity of each of the audio files.



Audio File	Audio Length	Maximum Embedding Capacity(characters)
chimes.wav	00:00:07	16,996
heartbeat.wav	00:00:13	63,166
johncena.wav	00:01:51	77,700
gaanwala.mp3	00:02:45	278,616
jagorone.mp3	00:03:46	376,880
yaaron.mp3	00:04:25	425,802

Figure 9: Maximum Embedding Capacity varying with format and size of the audio.

6.1 Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE) of a signal

The peak signal-to-noise ratio (PSNR) is the ratio between a signal's maximum power and the power of the signal's noise. Engineers commonly use the PSNR to measure the quality of reconstructed signals that have been compressed. Signals can have a wide dynamic range, so PSNR is usually expressed in decibels, which is a logarithmic scale. In statistics, the **mean squared error (MSE)** of an estimator is one of many ways to quantify the difference between values implied by an estimator and the true values of the quantity being estimated. MSE is a risk function, corresponding to the expected value of the **squared error loss** or **quadratic loss**. MSE measures the average of the squares of the "errors." The error is the amount by which the value implied by the estimator differs from the quantity to be estimated.

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 [17-20], 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 1in 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 there is a series of n measurements of X and Y written as x_i 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

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{X})(y_i - \bar{Y})}{(n-1)S_x S_y}$$

$\bar{X}$ and $\bar{Y}$ are the sample means of X and Y, S_x and S_y are the sample standard deviations of X and Y.

6.3 Comparison of M16MA 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 do not come much closer to making the introduced noise inaudible.

Proposed M16MA method in Audio steganography has been designed keeping in mind to overcome the above mentioned short comings.

- Embedding capacity largely increased by mapping of four bits at a time instead of one.
- Similarity measure between cover audio and stego audio has been inducted here through correlation method and this method is capable of producing stego audio with minimum or zero degradation.

Audio File	Audio Length	Maximum Embedding Capacity(characters)		Data Size 100	Data Size 500	Data Size 1000	Data Size 2500	Data Size 5000	Data Size 10000	Data Size 20000	Data Size 50000
chimes.wav	00:00:07	16,996	PSNR	57.6082	51.6096	48.6593	44.7342	41.6145	38.1638	N.A	N.A
			MSE	0.1128	0.4489	0.8854	2.1861	4.4836	9.9243	N.A	N.A
			CORREL-ATION	1.0000	1.0000	0.9994	0.9989	0.9980	0.9973	N.A	N.A
heartbeat.wav	00:00:13	63,166	PSNR	62.6636	56.7838	53.9258	49.7569	47.0730	44.1819	41.2801	37.3739
			MSE	0.0352	0.1364	0.2633	0.6877	1.2758	2.4825	4.8425	11.9041
			CORREL-ATION	1.0000	1.0000	1.0000	0.9996	0.9987	0.9976	0.9876	0.9743
johncena.wav	00:01:51	77,700	PSNR	61.8796	55.8404	53.8343	50.5305	47.8982	45.0435	42.0621	38.0744
			MSE	0.0422	0.1695	0.2689	0.5755	1.0550	2.0358	4.0445	10.1308
			CORREL-ATION	1.0000	1.0000	1.0000	0.9991	0.9984	0.9980	0.9872	0.9763
gaanwala.mp3	00:02:45	278,616	PSNR	70.6899	63.8019	60.8880	56.8660	53.8713	50.8857	47.8550	43.8315
			MSE	0.0055	0.0271	0.0530	0.1338	0.2667	0.5303	1.0656	2.6911
			CORREL-ATION	1.0000	1.0000	1.0000	0.9995	0.9988	0.9982	0.9974	0.9963
jagorone.mp3	00:03:46	376,880	PSNR	72.0019	65.1138	62.1999	58.1780	55.1832	52.1976	49.1348	45.0646
			MSE	0.0041	0.0200	0.0392	0.0989	0.1971	0.3920	0.7936	2.0259
			CORREL-ATION	1.0000	1.0000	1.0000	0.9993	0.9987	0.9979	0.9971	0.9963
yaaron.mp3	00:04:25	425,802	PSNR	71.4752	64.2170	61.8149	58.3409	55.4204	52.5417	49.5647	45.6034
			MSE	0.0046	0.0246	0.0428	0.0953	0.1867	0.3622	0.7188	1.7895
			CORREL-ATION	1.0000	1.0000	1.0000	0.9989	0.9982	0.9974	0.9970	0.9965

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

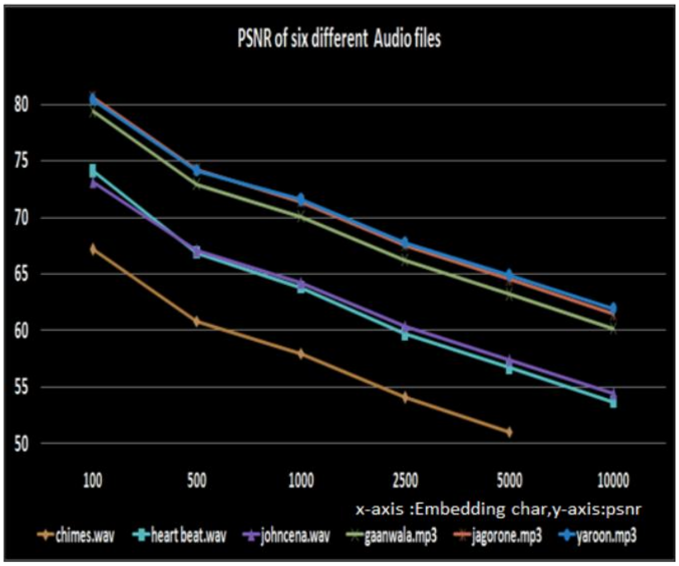


Figure 11: PSNR Plot of different audio files.

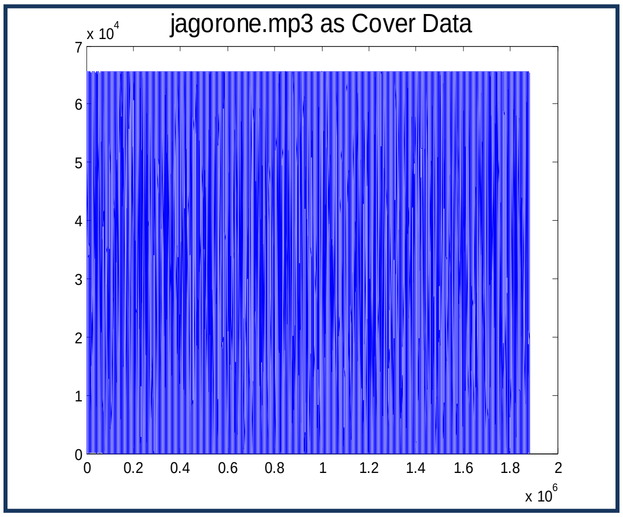


Figure 12: Signal plotting of the cover audio jagorone.mp3.

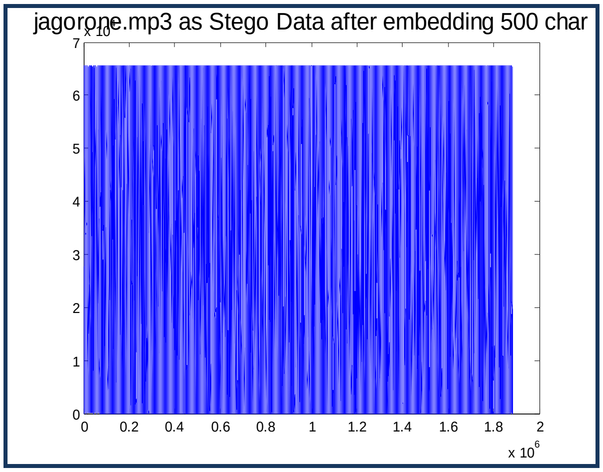


Figure 13: Signal plotting of the jagorone.mp3after embedding 500 char.

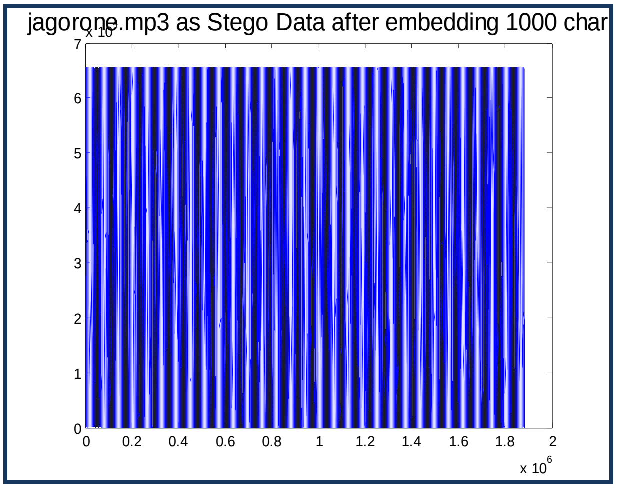


Figure 14: Signal plotting of the jagorone.mp3after embedding 1000 char.

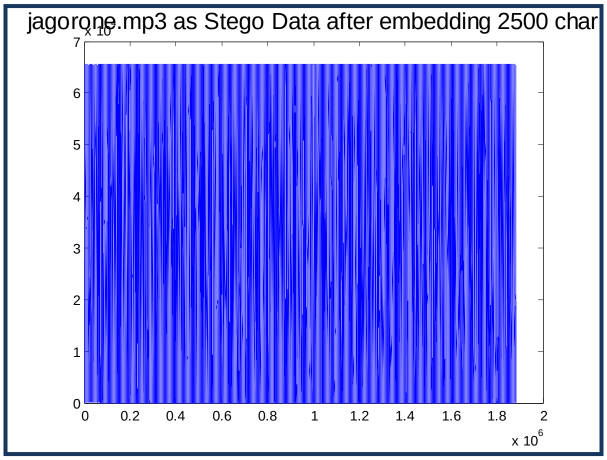


Figure 15: Signal plotting of the jagorone.mp3after embedding 2500 char.

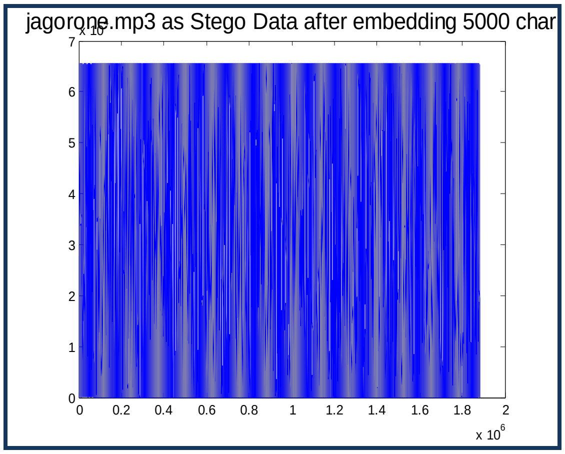


Figure 16: Signal plotting of the jagorone.mp3after embedding 5000 char.

7. CONCLUSION

In this paper authors 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. The proposed method can produce stego audio at various embedding rate with minimum or zero degradation which can be seen through the figures [12-16]. This method can avoid steganalysis also. 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.

8. 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 CommunicationTechnologies,(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 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 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.

[17] S. Dowdy and S. Wearden. Statistics for research. Wiley. ISBN 0471086029, page 230, 1983.

[18] M. A. Jaro. Advances in record linking methodology as applied to the 1985 census of tampa florida. Journal of the American Statistical Society. 84:414–420, 1989.

[19] M. A. Jaro. Probabilistic linkage of large public health data file. Statistics in Medicine 14 (5-7)., pages 491–498, 1995.

[20] W. E. Winkler. The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04., 1999.

[21] Poulami Dutta, Debnath Bhattacharyya, and Tai-hoon Kim Data Hiding in Audio Signal: A Review at International Journal of Database Theory and Application, Vol. 2, No. 2, June 2009.