J. Basic. Appl. Sci. Res., 3(12)71-81, 2013
© 2013, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Changing Slope Method: A Novel Technique for Digital Audio Steganography

Muhammad Zeeshan Muzaffar[a,d,*], Ijaz Mansoor Qureshi[a,b,d], Atta-ur-Rahman[c,d], Muhammad Tahir Naseem[c,d]

[a]School of Engineering & Applied Sciences (SEAS),ISRA University, Islamabad Campus, Pakistan
[b]Department of Electronic Engineering, Air University, Islamabad, Pakistan
[c]Barani Institute of Information Technology (BIIT), PMAS-AA University Rawalpindi, Pakistan
[d]Institute of Signals, Systems and Softcomputing (ISSS), Islamabad, Pakistan

Received: September 25 2013
Accepted: October 26 2013

ABSTRACT

Steganography is a technique to transmit the secret information over public channels. The secret information can be hidden in different types of cover signals like video, image, audio and text. In the proposed scheme, audio signal is used to hide the secret information. The theme of the scheme used is first developing different groups of equal length of all the possible amplitudes of cover audio signal. All amplitude levels of all groups have some specific meaning and number of possible amplitudes in each group is related with the possible secret bits attached with each sample. On the other hand, secret data bits are also kept in groups of equal length called chunks and one chunk is embedded in one sample of cover audio. Main Contribution towards this research work is the algorithm that is used to embed chunks of secret message in cover audio by using Changing Slope Method (CSM). Changing slope means the change of slope of the line joining any two consecutive samples of cover audio. The benefit achieved here is the value of Perceptual Evaluation of Speech Quality (PESQ) which is 4.492. This value shows that graphs of original audio and stego signal are very close.
KEYWORDS: Steganography; Changing Slope Method (CSM).

I. INTRODUCTION

Secret data transmission is an evergreen research area. Steganography is one of the techniques used for secret data transmission. For this purpose, a cover signal is used which carries the secret information by using some kind of steganography method.

There are four basic kinds of cover signals. These are audio, video, images and text. Different researchers use different kinds of cover signals with different types of information signals.

In literature various techniques has been proposed for digital steganography [1].These techniques can be mainly categorized into two. Some of the techniques are implemented in time domain and some of them are in the frequency domain. Time domain methods include Low bit encoding, echo hiding while frequency domain includes Phase coding and Spread Spectrum hiding [2]. Low bit encoding techniques has been widely used in digital steganography [3][4].Deepak D. et al.[5]uses the Low bit encoding for hiding the secret message. He used different channels of a wave (.wav) file to store different characters of same message. Then he creates more randomness by scrambling bits of each information character before placing each character bit in LSB's (Least significant bit)of their respective channel bytes. This scrambling fashion is already known at receiver side to build up original character from the received signal. In [6], another form of Low bit encoding is visited. In this paper, a PKE algorithm is used to embed secret information and achieved security for hidden information in this manner.

Ashwini Mane et al. also give a LSB technique for data hiding [7]. They make an algorithm for data embedding and used this algorithm for offline data security. The algorithm embeds secret data in audio LSB by using a secret key and store stego file on drive. Then retrieval of the secret information is only possible by knowing the key.

In echo hiding method information is embedded in the echo part of the cover audio signal [6][8]. The echo is nothing but the resonance added to the cover audio signal and hence, the problem with the additive noise is avoided here. While using echo hiding three parameters are to be considered. First is the initial amplitude, second is the offset (delay) and third is decay rate, so that echo is not audible. The main disadvantage of this method is lenient detection and low detection ratio. Due to its low embedding rate and low security no researches are going on echo hiding technique.

Phase coding is another method used by researchers to embed secret data into the phase of cover audio. Using frequency domain Phase coding, Salah *et al.*[9] hide the watermark information in a WAV file by reading the WAV

***Corresponding Author:** Muhammad Zeeshan Muzaffar, Postgraduate Research Student, School of Engineering & Applied Sciences (SEAS), ISRA University, Islamabad Campus, I-10, Islamabad, 46000, Pakistan. Email: zeebwp@gmail.com

file, first separates the header information and takes the FFT of the rest of the data part by using Butterfly method. After that, every result has the real and imaginary parts. Then by using these real and imaginary parts, phase of those results can be calculated. Then modifying these phases and to embed information. Then recalculate the real and imaginary parts according to new phases and take the IFFT using direct method. At last attach the header which was split in the start and write a new wave (.wav) file which contains information. In [10], after split the header of audio file, rest of the data portion is divided into segments of equal lengths. The length of segments is equal to the size of message to be encoded. Then take the DFT of each segment and get the matrix of phases as a result. Use static phase coding in which fix phase $\pi/2$ is added if secret data bit is 1 and fix phase $\pi/2$ is subtracted if secret data bit is 0.

Researchers also use different Genetic algorithms for secure data transmission with audio as cover signal. Krishna Bhowal et al.[11] uses Genetic algorithm for image transmission by using cover audio signal. They used the technique to replace some bits of audio sample with some of image bits and then apply the Genetic algorithm on the remaining bits and find the closest guess of original audio sample. In this way, they took the benefit of HAS and achieved imperceptibility for cover audio.

Some researchers use different type of time domain algorithms which actually choose some amplitude after some calculation on cover signal and the new amplitude also contain their secret data. Like in [12], a Mod 4 method is used to change the amplitude of the cover signal and during this process, secret data is also embedded. Like in example, they take the cover signal sample's amplitude as 23 and after applying the algorithm, output modified amplitude becomes 22 which also carry the two information bits 10.

In this paper, we used a new method to embed information into audio signals. We name the set of all possible values of amplitudes of audio signals as "audio sample space". An algorithm is proposed to subdivide this sample space into subspaces and embed information into these subspaces. On the other hand, an algorithm for decoding on receiver side is also proposed. The algorithm has the capability to work on real time systems and provide sufficient security at commercial level.

Rest of the paper is organized as: section II describes the System Model; section III describes the proposed algorithm for data embedding and retrieval and their details; section IV contains the simulation results and section V contains the conclusion.

II. SYSTEM MODEL

Let $\mathbf{A} = \{a_1, a_2, \dots, a_{\boldsymbol{n}}\}$ be the given cover audio signal, where $\boldsymbol{n}$ be the total number of audio samples. Each audio sample contains$\boldsymbol{q}$ bits.

Let $\mathbf{I} = \{i_1, i_2, \dots, i_{\boldsymbol{m}}\}$ be the total information that we want to embed into given cover audio signal, where $\boldsymbol{m}$ be the total number of information bits. So we divide these information bits into chunks of $\boldsymbol{k}$ bits, where $\boldsymbol{k}$ is the total number of information bits attached to each sample of cover audio. So, there are total of $\boldsymbol{b} = \boldsymbol{m}/\boldsymbol{k}$ information chunks, where $\boldsymbol{m}$is multiple of $\boldsymbol{k}$and $\boldsymbol{b} < \boldsymbol{n}$. So we can represent those chunks as $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots \mathbf{c}_{\boldsymbol{b}}\}$, where $\mathbf{c}_{\boldsymbol{l}}$ contains bits of information $\mathbf{I}$from $(\boldsymbol{l}-1)\boldsymbol{k}+1$ to $\boldsymbol{lk}$, where $\boldsymbol{l} \le \boldsymbol{b}$ This division of bits we termed as *horizontal division* of information bits. Now each chunk $\mathbf{c}_{\boldsymbol{l}}$represents a binary number of $\boldsymbol{k}$bits.

Another type of division used in the proposed algorithm is *vertical division* which is performed on actual sample space of cover audio signal. In this division we divide quantization levels into groups of equal height. There are a total of $2^{\boldsymbol{q}}$ quantization levels and we divide these levels into groups containing $2^{\boldsymbol{k}}$ quantization levels each. So there will be a total of $\boldsymbol{p} = 2^{\boldsymbol{q}-\boldsymbol{k}}$ vertical divisions. These vertical groups can be represented as $\mathbf{N} = \{\mathbf{N}_{\mathbf{1}}, \mathbf{N}_{\mathbf{2}}, \dots, \mathbf{N}_{\boldsymbol{p}}\}$, where any $\mathbf{N}_{\boldsymbol{i}}$ contain $2^{\boldsymbol{k}}$ consecutive quantization levels from $2^{\boldsymbol{k}}(\boldsymbol{i}-1)$ to $2^{\boldsymbol{k}}\boldsymbol{i}-1$ and $\boldsymbol{i} = 1,2, \dots, \boldsymbol{p}$.Each $\mathbf{N}_{\boldsymbol{i}}$is distinct from each other such that

$$\mathbf{N_1} \cap \mathbf{N_2} \cap \dots . \cap \mathbf{N_p} = \boldsymbol{\phi}$$

And

$$\mathbf{N_1} \cup \mathbf{N_2} \cup \dots . \cup \mathbf{N_p} = \mathbf{N}$$

Where

$$\mathbf{N}_{\boldsymbol{i}} = \{\mathbf{N}_{\boldsymbol{i},\mathbf{0}}, \mathbf{N}_{\boldsymbol{i},\mathbf{1}}, \dots, \mathbf{N}_{\boldsymbol{i},2^{\boldsymbol{k}}-\mathbf{1}}\}$$

Figure1 shows the quantization level division for $k = \mathbf{2}$, where each level named as in terms of $\mathbf{N}_{\boldsymbol{i}}$'s. $\mathbf{N}_{\boldsymbol{i},\boldsymbol{j}} = \left(2^{\boldsymbol{k}}(\boldsymbol{i}-1)+\boldsymbol{j}\right)$th level of our cover audio sample space, where$1 \le \boldsymbol{i} \le \boldsymbol{p}$ and $0 \le \boldsymbol{j} \le 2^{\boldsymbol{k}}-1$. e.g. for $\boldsymbol{k} = 2$, $\mathbf{N}_{\mathbf{2},\mathbf{3}}$represents

$$\mathbf{N}_{\mathbf{2},\mathbf{3}} = (2^2(2-1)+3) = 7th\ level$$

As shown in Figure1 below.

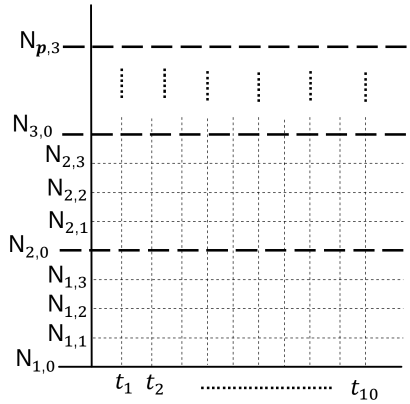

Figure 1: Vertical division of quantization levels for $k = 2$

Each $\mathbf{N}_{i,j}$ will represent a specific information which is binary value of $\boldsymbol{j}$. e.g. $\mathbf{N_{1,2}}, \mathbf{N_{2,2}}, \dots, \mathbf{N}_{p,2}$are all the levels which represents binary information 10 and this is binary code of 2.

III. PROPOSED ALGORITHM

Our proposed algorithm maps each sample to amplitude which represents the hidden information and HAS (human auditory system) cannot sense any type of change. The algorithm has the following number of steps shown in Figure2 box below.

This algorithm is applied to an offline cover audio file. So, in step1, step2 and step3 just read the cover audio file and collect all necessary information to process further on the algorithm.

As each sample is gray coded in PCM (Pulse Coded Modulation) audio, so to find the actual level number, we convert it into decimal in Step4 and Step5. This conversion is applied to whole samples of cover audio so that each sample can be mapped on its appropriate$N_{i,j}$. Now we have the cover audio in the form of $N_{i,j}$'s as

$$A = \{N_{i_1,j_1}, N_{i_2,j_2}, \dots, N_{i_n,j_n}\}$$

Where

$$0 \le N_{i_1,j_1}, N_{i_2,j_2}, \dots, N_{i_n,j_n} \le 2^k p - 1$$

Step7 to Step15 are recursive. In these steps we hide our information chunks between samples of cover audio signal. Step7 computes the meanof every two consecutive samples of cover audio and we use this mean for computing the new mean that will also contain our hidden information chunks. This new mean is represented by nmean as shown in Step11 and Step12.

In Step8, r is calculated by using mod operator, e.g $10 \bmod 3 = 1$ which is actually just remainder. In Step11 and Step12, $c_{dx}$is the decimal equivalent of $c_x$which is xth chunk of information.

In Step13, we use two points $N_{i_x,j_x}$and nmean to find the updated value of $N_{i_{x+1},j_{x+1}}$. We use two point form of line for this purpose as shown below

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$

$$\frac{N_{i_{x+1},j_{x+1}} - N_{i_x,j_x}}{\text{nmean} - N_{i_x,j_x}} = 2$$

$$N_{i_{x+1},j_{x+1}} = 2\left(\text{nmean} - N_{i_x,j_x}\right) + N_{i_x,j_x}$$

After updating$N_{i_{x+1},j_{x+1}}$, algorithm will run for $N_{i_{x+1},j_{x+1}}$and $N_{i_{x+2},j_{x+2}}$ and this process continues up till last sample.

We can see algorithm working by taking some example. Let us suppose that,$N_{i_x,j_x} = 6$, $N_{i_{x+1},j_{x+1}} = 23$, $c_x = 10$ and $c_{dx} = 2$. At Step7, $\text{mean} = 14.5$,then $r = 0.5$ and raw form of $\text{nmean} = 14$ at Step9. Now as $\text{mean} > 0$, so we will move to Step12. At this step, $\text{nmean} = 15$. So, updated $N_{i_{x+1},j_{x+1}} = 24$. Now we will repeat for next sample updating with this new value. At last, after completing samples updating, before sending file to receiver, again convert each sample into its gray code PCM .WAV file as our cover audio was also gray coded. Then send it to receiver via some media.

Begin

Step1: Read the cover audio file

Step2: $\boldsymbol{k}$ = number of bits containing any $\mathbf{c}_l$

Step3: $\boldsymbol{n}$ = no of audio samples available

Step4: Convert each gray coded sample into its binary equivalent.

Step5: Convert each binary coded sample into its decimal equivalent.

Step6: $\boldsymbol{x} = 1$

Step7: $mean = (\mathbf{N}_{i_x,j_x} + \mathbf{N}_{i_{x+1},j_{x+1}})/2$

Step8: $\boldsymbol{r} = mean\ mod\ 2^{\boldsymbol{k-1}}$

Step9: $nmean = mean - \boldsymbol{r}$

Step10: if $mean < 0$, then go to Step11, otherwise go to Step12

Step11: $nmean = nmean - 0.5\mathbf{c}_{\mathrm{d}x}$,Go to Step13.

Step12: $nmean = nmean + 0.5\mathbf{c}_{\mathrm{d}x}$,Go to step 13.

Step13: $\mathbf{N}_{i_{x+1},j_{x+1}} = 2(nmean - \mathbf{N}_{i_x,j_x}) + \mathbf{N}_{i_x,j_x}$

Step14: $\boldsymbol{x} = \boldsymbol{x} + 1$

Step15: if $\boldsymbol{x} \leq \boldsymbol{n} - 1$ go to step 7. Otherwise go to step 16.

Step16:- convert each decimal audio sample into gray code and save file.

Step17: send wav file to receiver.

End

Figure 2: Algorithm for embedding the information

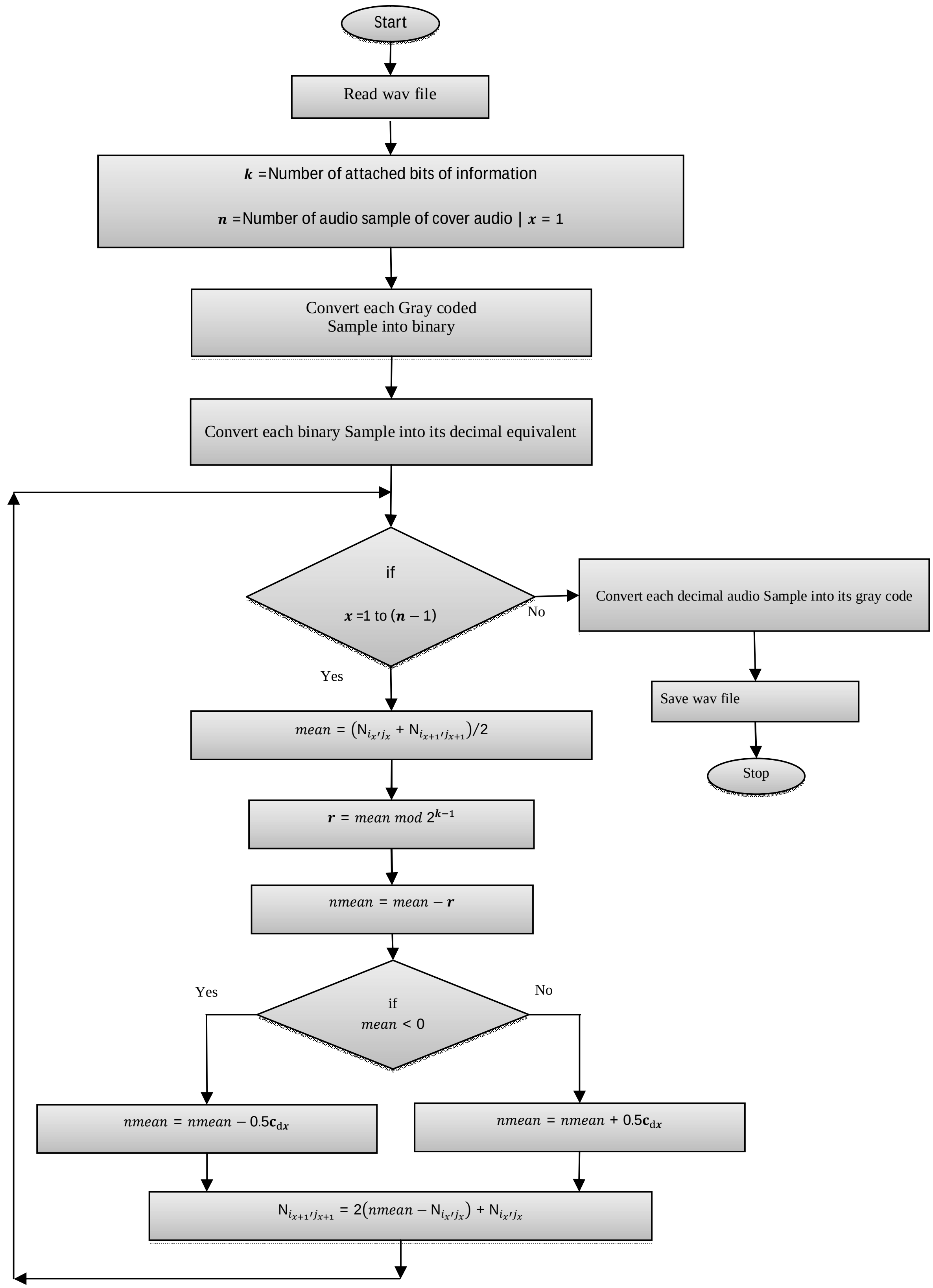

Figure 3: Flow Chart for information embedding

Now let us see how we can recover our information chunks back on receiver side.

In Figure4 below, Step1 to Step6 is the basic information collection steps which are necessary to run the algorithm. Step7 computes the mean because our information resides in the mean. r is remainder. Step9 computes c_{dx}which is decimal value of xth information chunk. When these chunks are computed for all received audio samples, then we will convert all chunks into binary. Subsequently, we will arrange these chunks and our required info will be recovered.

We can see algorithm working on receiver side by taking same example. Now $N_{i_x,j_x} = 6$, $N_{i_{x+1},j_{x+1}} = 24$. At Step7, mean = 15,then r = 1 and $\mathbf{c}_{\mathrm{d}\boldsymbol{x}} = 2$ at Step9. Now the binary form of $\mathbf{c}_{\mathrm{d}\boldsymbol{x}} = \mathbf{c}_{\mathrm{x}} = \mathbf{10}$. This is how we can retrieve the embedded information according to the algorithm given below.

Begin

Step1: Read received wav file

Step2: $\boldsymbol{k}$ = number of bits containing any $\mathbf{c}_{\boldsymbol{l}}$

Step3: $\boldsymbol{n}$ = no of audio samples available

Step4: Convert each gray coded sample of wav files into binary

Step5: Convert each binary sample into its decimal equivalent

Step6: $\boldsymbol{x} = 1$

Step7: $mean = \left(\mathbf{N}_{i_x,j_x} + \mathbf{N}_{i_{x+1},j_{x+1}}\right)/2$

Step8: $\boldsymbol{r} = mean \bmod 2^{\boldsymbol{k-1}}$

Step9: $\mathbf{c}_{\mathrm{d}\boldsymbol{x}} = |2\boldsymbol{r}|$

Step10: $\boldsymbol{x} = \boldsymbol{x} + 1$

Step11: if $\boldsymbol{x} \leq \boldsymbol{n} - 1$

Then goto step 7

Step12: Convert each value info vector into its binary equivalent and arrange in on sequence.

Step13: Show information

End

Figure 4: Algorithm for information extraction

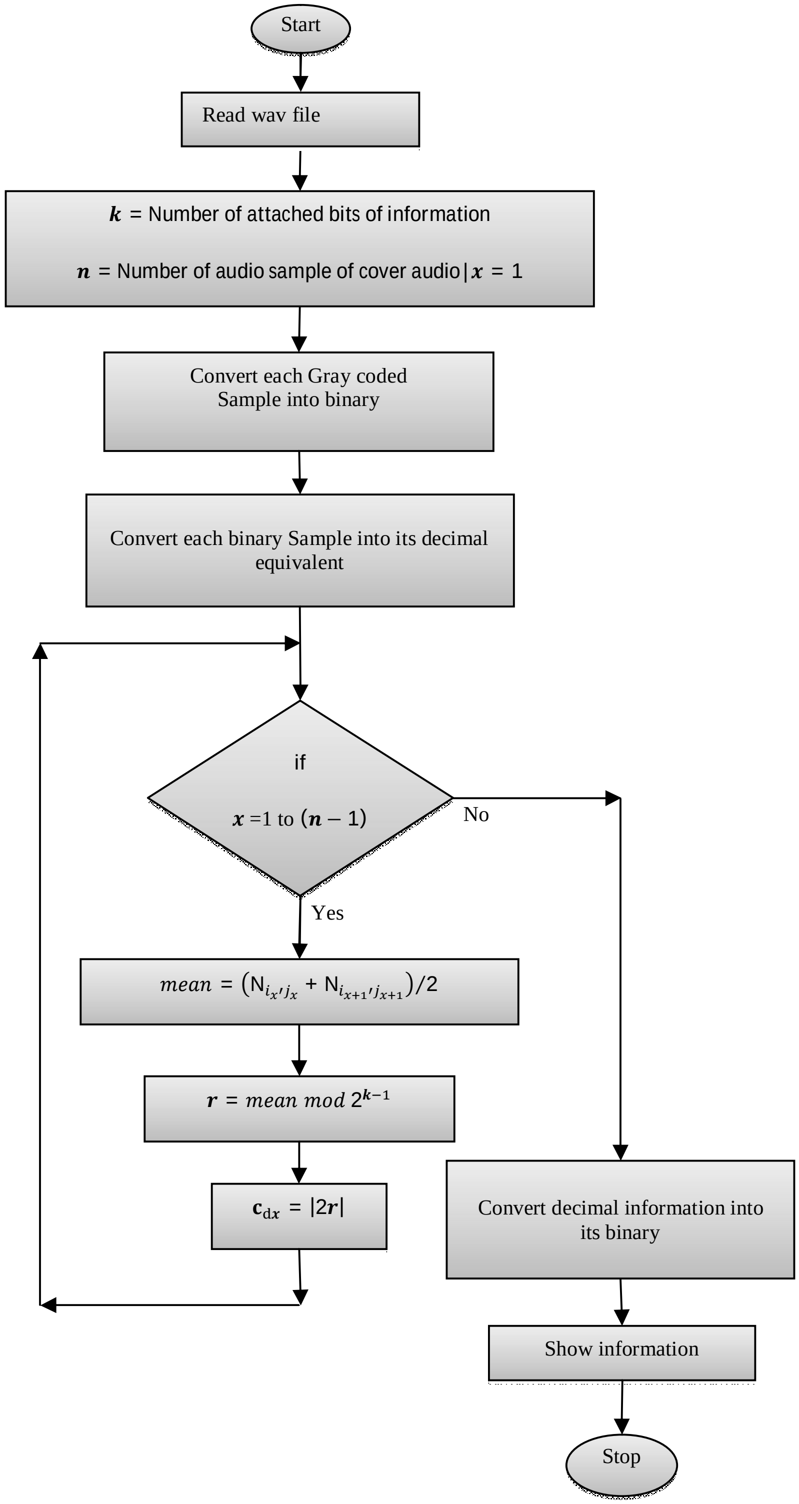

Figure 5: Flow Chart for information retrieval

IV. RESULTS

In this section, the results of our proposed algorithm are presented. For this purpose, a PCM coded Wave file is used shown in Figure6. It contains $n = 99001$ samples and $q = 16$ bits. So, there are a total of $2^q = 2^{16} = 65536$

levels in the whole signal sample space. We used $m = 198000$ of information bits attached secretly by using our proposed algorithm. So, $k = 2$and $b = 99000$.

So, there will be a total of $p = 2^{q-k} = 16384$ vertical divisions and $N_{p,2^{k-1}} = 65535$. So, we represent our chunks as

$$C = \{c_{1,1}c_{1,2}, c_{2,1}c_{2,2}, \dots c_{b,1}c_{b,2}\}$$

Each new signal level which is modified after processing through our algorithm can differ a maximum of $2^k - 1$ from the original level of the audio signal. This creates a very close graph to the original one if $k \ll q$ as shown in Figure6, original and stego signal are overlapped with each other and seems to be as one audio signal. To see the difference between both of the levels we zoom in our graph up till the area shown in the form of rectangle in Figure6. After zooming in the area shown in Figure6, the result becomes Figure7. But in Figure7, not seen any major difference and graph is seems to be one again. Now we zoom again up to the rectangle shown in Figure7 and result after zooming becomes Figure8.

In Figure8, we can see two graphs but they are still very close. So, the zooming factor shown in Figure8 used again and Figure9 is to be formed. In this figure, we can see the difference between both graphs. This difference is so small that to see that difference, we use various levels of zooming. That's why, such a small difference cannot be detected by Human auditory system (HAS).

Figure 9 shows the cross correlation between the original audio and the stego one. This cross correlation also seems to be as auto correlation of same audio signal and this thing also shows the imperceptibility of our stego algorithm.

Table 1: PESQ values for different values of *k*

k	**PESQ**
1	4.496798
2	4.494659
3	4.489622
4	4.470681
5	4.432074
6	4.254979

And the last one and the reliable one technique up till so for to judge the human audio imperceptibility is Perceptual Evaluation of Speech Quality (PESQ) algorithm [13][14] which is an International Telecommunication Union (ITU) standard. PESQ values normally vary between 1.0 and 4.5 [13]. Value of PESQ will be less than 1.0 in worst cases and more the grater values of PESQ shows the more best quality of audio signals. Our PESQ values for different cases are also shown in Table1 and result is very good for lower values of k.

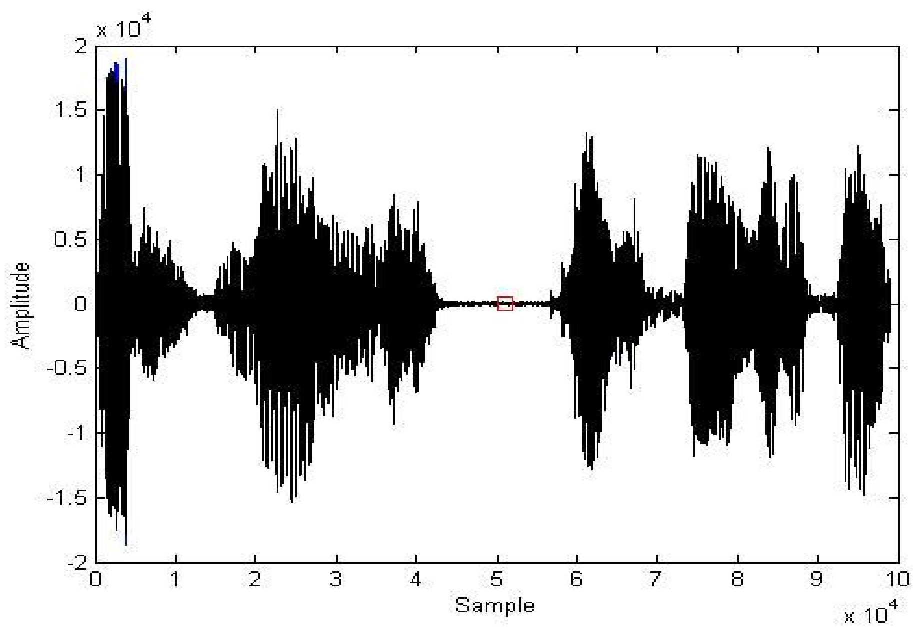

Figure 6 : Original vs.Stego Signal for q=16, k=2

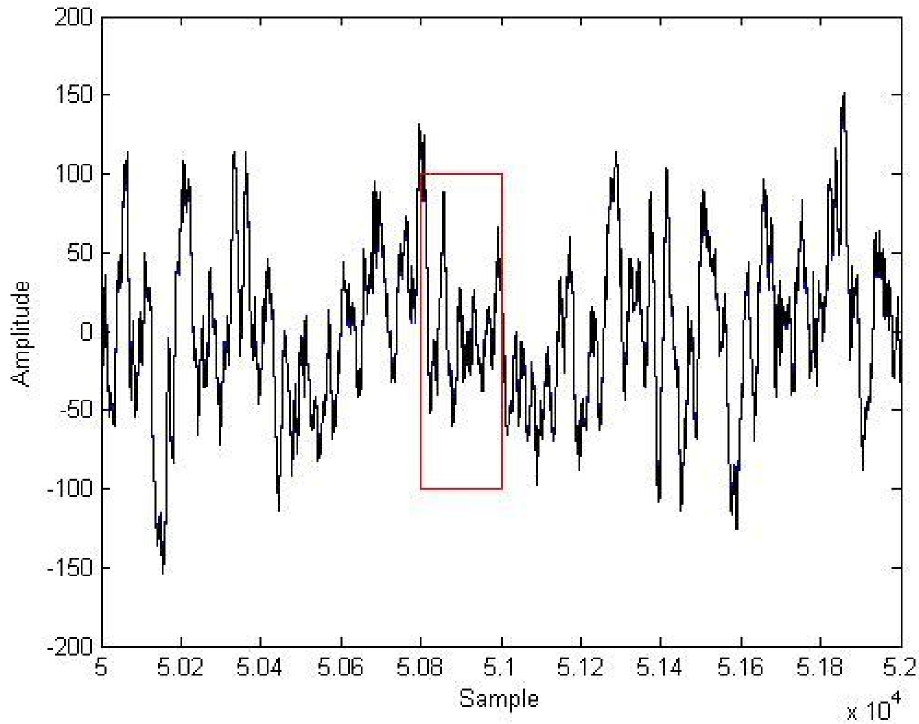

Figure 7 : Zoom area shown in rectangle in Figure 6

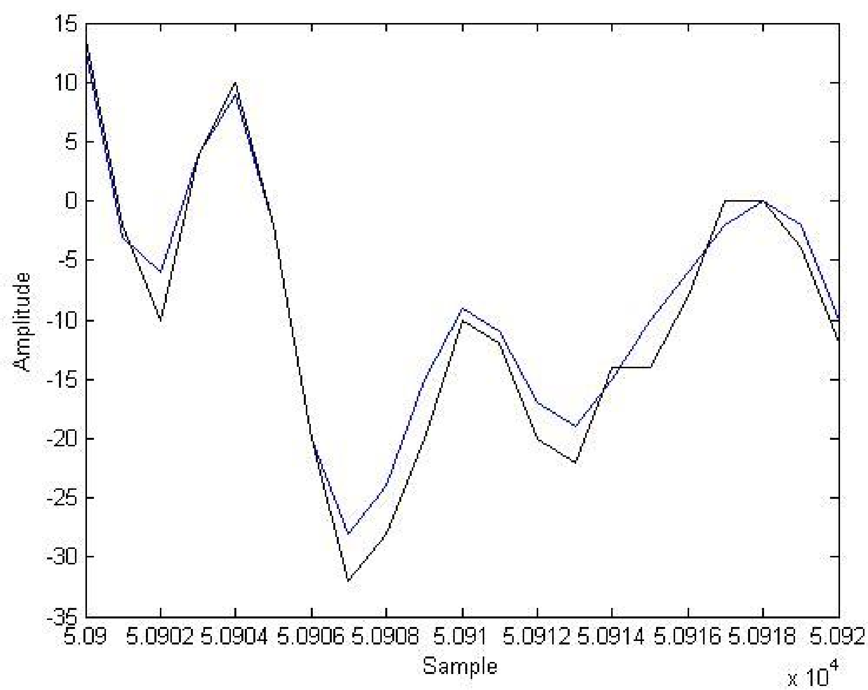

Figure 8: Zoom area shown in rectangle in Figure 7

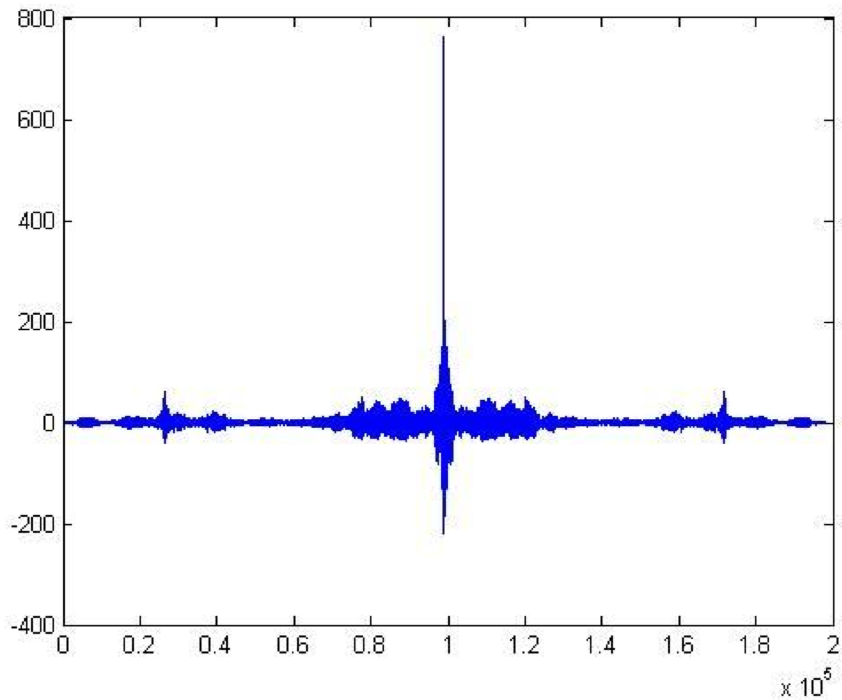

Figure 9: Cross Correlation between Original audio and Stego one

V. CONCLUSIONS

The benefits achieved in our proposed scheme are as follows.

- Firstly, the information is not hidden into the amplitudes but the means of consecutive amplitudes. If one amplitude changes, then only the leading and trailing information chunks will be destroyed and this error will not propagate further.
- Secondly, the average PESQ values for the existing networks are 3.8 [15]. Whereas, the PESQ estimates for our proposed scheme is 4.496 for $k = 1$, 4.494 for $k = 2$ and 4.489 for $k = 3$ which is very much close to 4.5 the maximum value. The measures of PESQ also show that human auditory system cannot detect the change in stego signal.
- Thirdly, the cross correlation also seems just like the autocorrelation and this thing also reflect that original and stego voices are very close to each other.
- Fourthly, the size of the carrier and stego files are the same. So, the memory also reflects that no data is embedded into the file.
- Lastly the graph of stego signal is very much close to the original audio. For that reason, the stego file is very imperceptible not only for HAS but also from the graph shown.

VI. ACKNOWLEDGMENT

This research work was sponsored by Higher Education Commission (HEC), of Pakistan.

REFERENCES

[1] M.L.Mat Kiah*et al.*"A review of audio based Steganography and digital watermarking" *International journal of the physical sciences,* vol. 6(16), pp. 3837-3850, August, 2011.

[2] Fatiha Djebbar *et al.*"Comperitive study of digital audio steganography technique". *EURASIP journal on audio, speech and music processing.* 2012

[3] Muhammad Asad, Junaid Gilani, Adnan Khalid "An Enhanced Least Significant Bit Modification Technique for Audio Steganography".2011.

[4] Singh, Pardeep Kumar, R. K Aggrawal. " Enhancement of LSB based Steganography for Hiding Image in Audio." *International journal on Computer Science and Engineering* 2.05 (2010): 1652-P1658.

[5] Deepak D. *et al.*"Efficient Method to Increase Robustness in Audio Steganography".*International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.* vol. 1, Issue 6, December 2012.

[6] Bhagyashri A. Patil and Vrishali A. Chakkarwar. "Review of an improved Audio Steganography Technique over LSB through Random Based Approach". *IOSR journal of computer engineerin.* Vol. 9, Issue 1, 2013.

[7] Ashwini Mane, Gajanan Galsetwar and Amutha Jeyakumar."Data hiding technique: Audio Steganography LSB technique".*International Journal of Engineering Research and Applications.* Vol. 2, Issue 3, pp. 1123-1125. 2012.

[8] F. Djebbar, B. Ayad, H. Hassmam, and K. Abed-Meraim, "A view on latest audio steganography techniques", *International Conference on Innovations in Information Technology (IIT), IEEE*, 2011.

[9] Salah S. Alrawi *et al.* "Watermarking in WAV files Based on Phase Coding".*Engineering and Technology journal,*Vol. 29, No. 4, Pages 770-780, 2011.

[10] Samir Kumar Bandyopadhyay, Barnali Gupta Banik."LSB Modification and Phase Encoding Technique of Audio Steganography Revisited " *International Journal of Advanced Research in Computer and Communication Engineering.* Vol. 1, Issue 4, June 2012.

[11] Krishna Bhowal, D. Sarkar, S. Biswas and P. P. Sarkar"Secured Image transmission with GA based Audio Steganography". *IEEE annual conference india(INDICON).* Pages 1-4, 2011.

[12] Souvik Bhattacharyya*et al*."Audio Steganography Using Mod 4 Method(M4M)". *Journal of Computing, New York ,USA*, Vol. 3, Issue 8, August 2011.

[13] ITU-T Recommendation P.862. Perceptual Evaluation of Speech Quality (PESQ), an Objective Method for End-To-End Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs, February 2001.

[14] ITU-T Recommendation P.862.3, "Application guide for objective quality measurement based on Recommendation P.862, P.862.1 and P.862.2", November 2005.

[15] The PESQ Algorithm as the Solution for Speech Quality Evaluation on 2.5G and 3G Networks, 2009