INFORMATION HIDING IN AUDIO SIGNALS

project/btech/ece@nssce/1014/16.22

PROJECT REPORT
As a partial fulfilment of the curriculum

by

RAHUL R NAIR (NSAKEEC066)
SREERAG S (NSAKEEC086)
VISHNU K (NSAKEEC097)
VISHNU S DEV (NSAKEEC099)

under the guidance of

Dr.Sindhu.R
Head Of The Department

Department of Electronics & Communication Engineering
NSS College OF Engineering, Palakkad
March 2014

NSS College OF Engineering, Palakkad

Department of Electronics & Communication Engineering

CERTIFICATE

Certified that project work titled INFORMATION HIDING IN AUDIO SIGNALS is a bonafied work carried out in eighth semester by Rahul R Nair, Sreerag S, Vishnu K and Vishnu S Dev during the academic year of 2013-2014 under the guidance Dr.Sindhu.R,Head of the department as part of the partial fulfillment for the award of Bachelor of Technology in ELECTRONICS AND COMMUNICATION ENGINEERING from University of Calicut and no part of this work has been submitted earlier for the award of any degree.

PROJECT GUIDE
Dr.Sindhu R

HEAD OF DEPARTMENT
Dr. Sindhu . R

ACKNOWLEDGEMENT

We express our sincere piece of gratitude to Dr. Sindhu . R ,Head Of Electronics And Communication Department for her moral support and encouragement as Project Guide

We would like to express our deep sense of gratitude to all the faculity members, Department of Electronics and Communication, for being a great force behind all our efforts through their advice, guidance, and encouraging nature through out the project duration.

We acknowledge the help of Prof. Sidharth.N and for his guidance and motivation throughout the project.

Finally we would like express our gratitude towards our parents for their moral and financial support for making this project a success

RAHUL R NAIR (NSAKEEC066)
SREERAG S (NSAKEEC086)
VISHNU K (NSAKEEC097)
VISHNU S DEV (NSAKEEC099)

March 2014
Department of Electronics & Communication Engineering
NSS College Of Engineering,Palakkad

ABSTRACT

Data hiding, a form of steganography, is one of the emerging techniques that embeds secret data into a digital media and thus ensures secured data transfer. In this project, the steganographic method used, is based on audio steganography which is concerned with embedding secret data in an audio file.

Data hiding in media, including images, video, and audio, as well as in data files is currently of great interest both commercially, mainly for the protection of copyrighted digital media, and to the government and law enforcement in the context of information systems security and covert communications.

Embedding secret messages using audio signal in digital format is now the area in focus. There exist numerous steganography techniques for hiding information in audio medium.

In this project, a new model ISS-IHAS - embedding message in audio signal that embeds the message like the existing systems but with strong encryption that gains the full advantages of cryptography. Using this technique it is possible to conceal the full existence of the original message and the results obtained from the proposed model is compared with other existing on the basis of Signal To Noise ratio and MER.

The techniques that are in consideration are LSB, Parity, Phase, Echo and Spread spectrum Encoding schemes. Each of them are attempted to be implemented and demonstatred with the aid of MATLAB.

Contents

List of Figures

Chapter 1

INTRODUCTION

A number of different cover objects (signals) can be used to carry hidden messages. Data hiding in audio signals exploits imperfection of human auditory system known as audio masking. In presence of a loud signal (masker), another weaker signal may be inaudible, depending on spectral and temporal characteristics of both masked signal and masker Masking models are extensively studied for perceptual compression of audio signals. In the case of perceptual compression the quantization noise is hidden below the masking threshold, while in a data hiding application the embedded signal is hidden there. Data hiding in audio signals is especially challenging, because the human auditory system operates over a wide dynamic range.

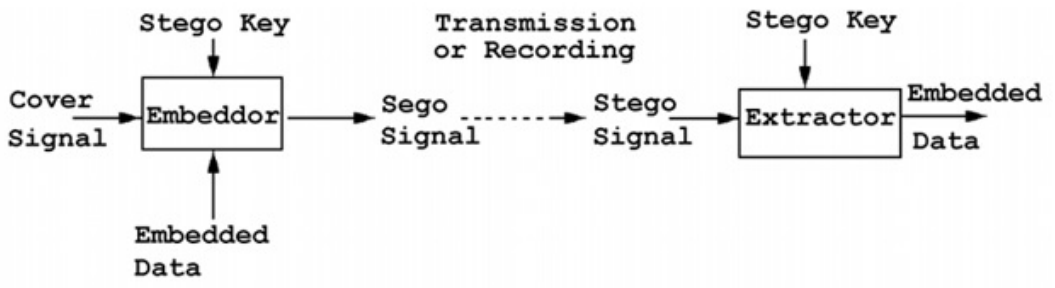


Figure 1. Block diagram of data hiding and retrieval.

The human auditory system perceives over a range of power greater than one billion to one and a range of frequencies greater than one thousand to one. Sensitivity to additive random noise is also acute. The perturbations in a sound file can be detected as low as one part in ten million (80 dB below ambient level). However, there are some âholesâ available. While the human auditory system has a large dynamic range, it has a fairly small differential range. As a result, loud sounds tend to mask out quiet sounds. Additionally, the human auditory system is unable to perceive absolute phase, only relative phase. Finally, there are some environmental distortions so common as to be ignored by the listener in most cases.

Many number of algortihms are available for the above described purpose and out of them most used ones are taken here for implementation and analysis.

Chapter 2

LSB CODING

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.One of the earliest techniques studied in the information hiding of digital audio (as well as other media types) is LSB coding. In this technique LSB of binary sequence of each sample of digitized audio file is replaced with binary equivalent of secret message. For example if we want to hide the letter 'A' (binary equivalent 1000001) into a digitized audio file where each sample is represented with 16 bits, then LSB of 7 consecutive samples (each of 16 bit size) is replaced with each bit of binary equivalent of the letter'A' .

It is the simplest way to embed information in a digital audio file. It allows large amount of data to be concealed within an audio file, use of only one LSB of the host audio sample gives a capacity equivalent to the sampling rate which could vary from 8 kbps to 44.1 kbps (all samples used). This method is more widely used as modifications to LSBs usually not create audible changes to the sounds.It has considerably low robustness against attacks.

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. 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.

Steps for Data Embedding

1. Read the cover audio signal.
2. Read the audio signal to be embedded, its size less than the size of the cover audio signal and convert it into binary sequence of bits.
3. Every message bit is embedded into the LSBs of the cover audio after processing.
4. Processing is done as follows:
    - If the message bit to be embedded is 0, then adjust or flip the LSB such that the XORing of LSB and next to LSB is 0.
    - If the message bit to be embedded is 1, then adjust or flip the LSB such that the XORing of LSB and next to LSB is 1.
5. The modified cover audio samples are then written to the file forming the stego audio signal.

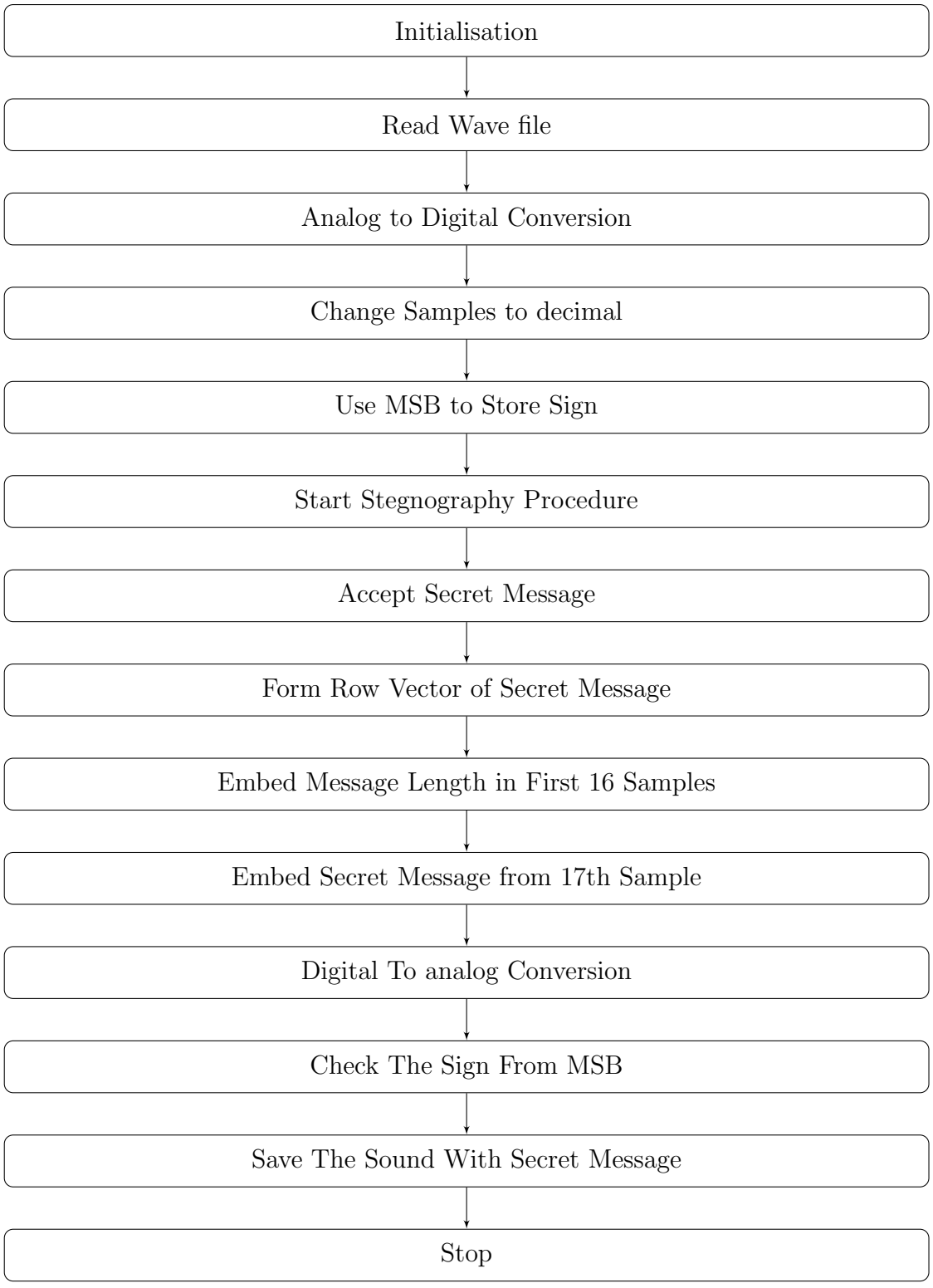
Steps for Data Extraction/Retrieval:

1. The Stego audio file is read.
2. Retrieval of the message bit is done by XORing the LSB and the bit next to LSB. If the result of XORing is 0, then the message bit is 0. If the result of XORing is 1, then the message bit is 1.
3. After every such 16 message bits are retrieved, they are converted to their decimal equivalents.
4. Finally the secret signal is reconstructed.

2.1 Encoder Implementation

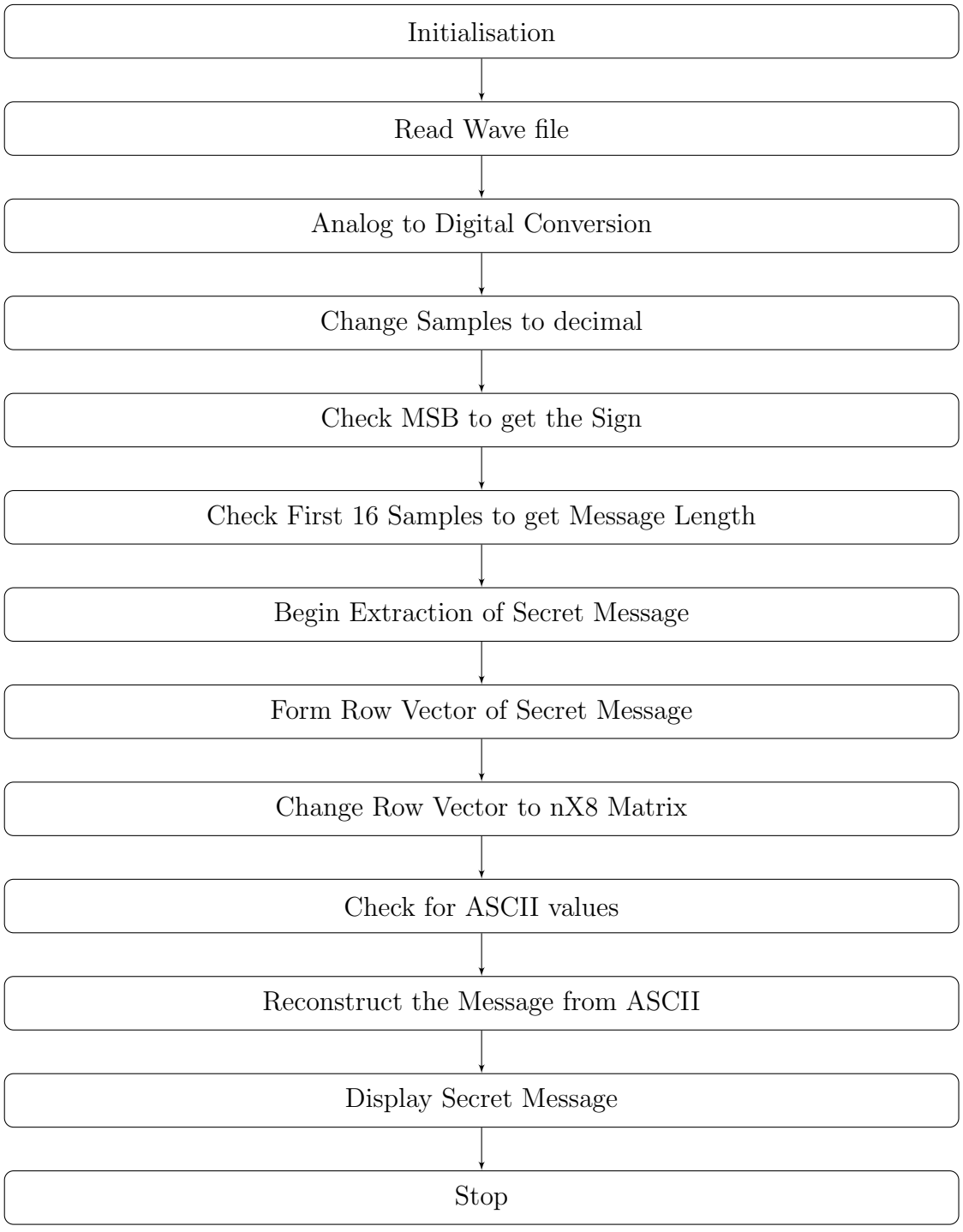
Being the simplest algorithm of all,the code is developed without any complexity.Initially the audio wave file is read,they are sampled to data samples based on the lenght of the audio signal,then they are converted to equivalent binary form,the resulting matrix is converted to decimal.The MSB of the sample is used to store the sign.To start the stegnography procedure we accept the message signal, it is then

transformed into a row vector. The length of the message is emedded in the first 16 samples, embedding of secret message starts from the 17th sample onwards.To regenerate the audio signal it is then converted to its analog equivalent.



2.2 Decoder Implementation

The decoder in effect performs just the reverse of what the encoder does. It first reads the audio signal with secret message change the samples to binary equivalents.Now the MSB of each sample is checked to determine the sign.The scanning of first 16 wave data samples gives an idea about the length of the message. After determining the length the upper limit for scanning of message is determined and the extraction procedure starts,a row vector of extracted bits is formed and is converted to a nX8 matrix.The ASCII equivalents of these values is compared and the secret message is displayed in command window.



2.3 Simulation Result

The following is a screenshot of the Simulated result of the LSB coding technique.It shows the plot of an audio signal before and after the encoding procedure

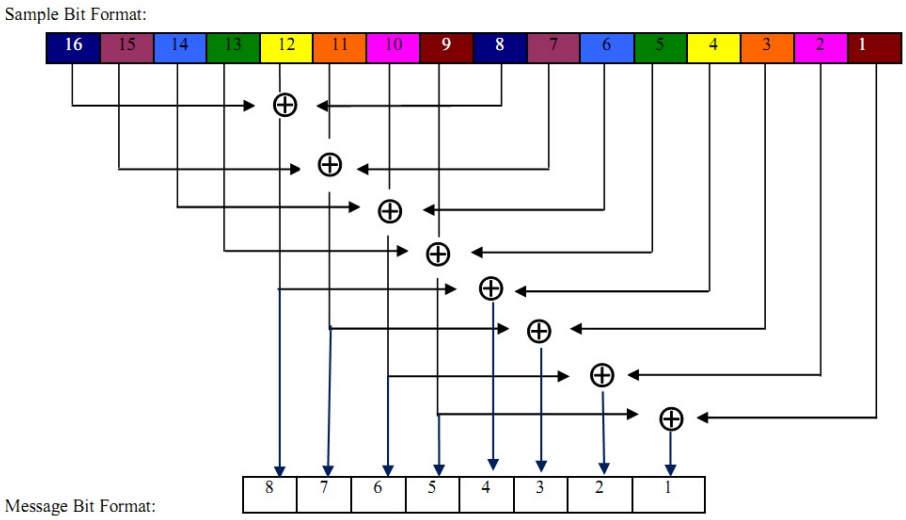


Figure 2.1: Diagramatic representation of procedure for using XOR with LSB's

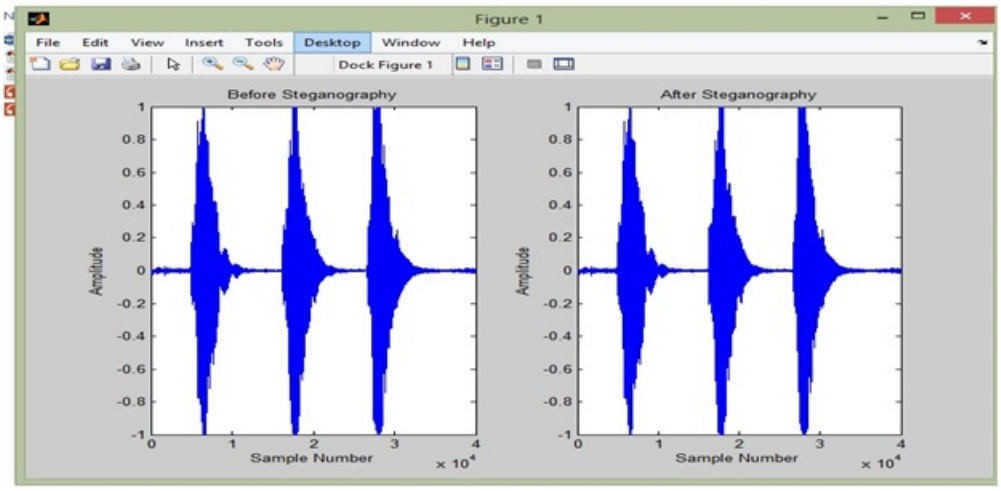


Figure 2.2: LSB Coding result

2.4 Performance

The performance of the code is evaluated in terms of SNR and MER with varying cover bits and embedded bits.

Message length was kept constant and the cover bits count was kept on varying, and it was found that even for a large increase in cover bits the MER varies only in the range of 2.4 to 2.5, which implies that the code is not effected by errors,however the SNR varies significantly.When the cover bits were 178.1KB the SNR was 48.1792db and for 79KB SNR was 50.649db, which means length of cover bits is in inverse propotion with SNR.

The Inference from MER plot is that MER changes in propotion with cover bit count but there is a only a very less variation for MER even for large increase in cover bit count.

Audio(.wav)	Cover Bits(KB)	Msg length	MER	SNR(db)
Button	225	24	2.4233	46.93
Beep	178.1	24	2.4756	48.93
One	79	24	2.5642	50.649

Table 2.1: By varying cover bits

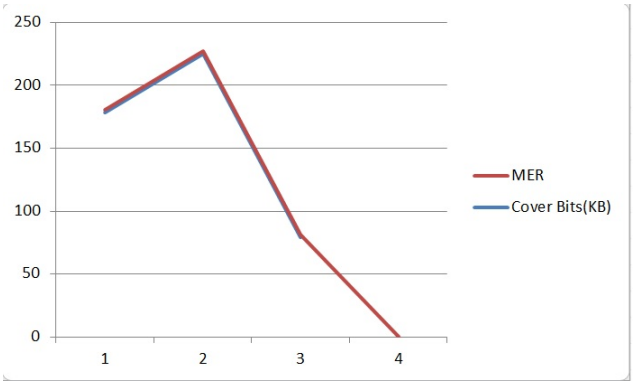


Figure 2.3: MER plot

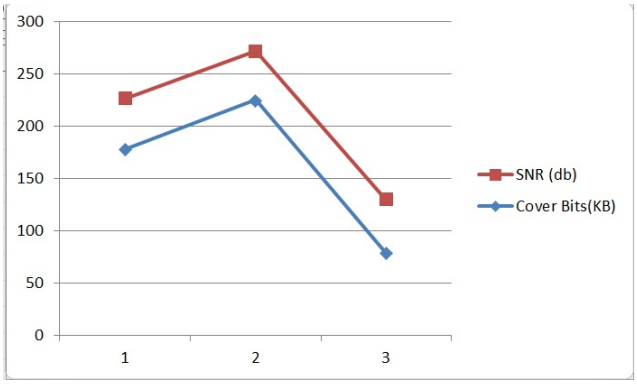


Figure 2.4: SNR plot

The SNR plot shows that there is a reasonably large change in SNR for change in cover bits, while comparing the values with the constant cover bit table its quite evident that SNR is in inverse propotion with length of cover bits.

In this particular analysis cover bits is varied keeping the embedded bits as constant.When the message length was 1000 the SNR was found to be 55.45db, also the MER was found as 3.1542, where as when message length reduced to 24 the SNR also reduced to 48.1792db and MER reduced to 2.4233 which implies SNR and MER varies in propotion with message length.

The cover bits are kept constant but having higher value,the emededded bit count is chosen to be 24,168,1000.When the message length was 24 MER was recorded as 2.4756 and SNR was 46.93 but as we incresed the count to 1000, MER was recorded as 2.5642 and SNR as 51.4107 which implies SNR and MER varies in propotion with message length.

As the sampling rate was reduced the SNR was decreasing propotionally,LSB

Audio(.wav)	Cover Bits(KB)	Msg length	MER	SNR(db)
Beep	178.1	24	2.4233	48.1792
Beep	178.1	168	2.7747	50.2414
Beep	178.1	1000	2.1542	55.45

Table 2.2: Using Beep.wav

Audio(.wav)	Cover Bits(KB)	Msg length	MER	SNR(db)
Button	225	24	2.4756	46.93
Button	225	168	2.5236	47.93
Button	225	1000	1.1544	55.4107

Table 2.3: Using Button.wav

being a very popular and common technique used for stegnography it was analysed it different situations.The embedded bits were taken to be constant and and cover bits were varied,Mean error rate changed propotionally to cover bits which implies that the the code with hidden message gets destroyed very easily if only a small message is hidden in a relatively large number of cover bits as this technique proposes the hiding of message bits in the least significant bit,if any of the LSB of any sample is effected the whole message gets corrupted.The SNR performance of the code is relatively good,as the sampling rate is increased we can expect the best out of this code.As there is no complexity in algorithm, in security point of view the code is advised to consider for confidential data hiding.More over this coding is at its best when the hiding rate is limited to 16kbps.

Audio(.wav)	**Cover Bits(KB)**	**Msg length**	**MER**	**SNR(db)**
One	79	24	2.5642	50.649
One	79	168	2.8024	52.7337
One	79	1000	3.0582	54.3946

Table 2.4: Using One.wav

Chapter 3

PARITY CODING

One of the prior works in audio data hiding technique is parity coding technique. Instead of breaking a signal down into individual samples, the parity coding method breaks a signal down into separate regions of samples and encodes each bit from the secret message in a sample region's parity bit. If the parity bit of a selected region does not match the secret bit to be encoded, the process flips the LSB of one of the samples in the region. Thus, the sender has more of a choice in encoding the secret bit, and the signal can be changed in a more unobtrusive fashion. Figure 3.1, shows the parity coding procedure.

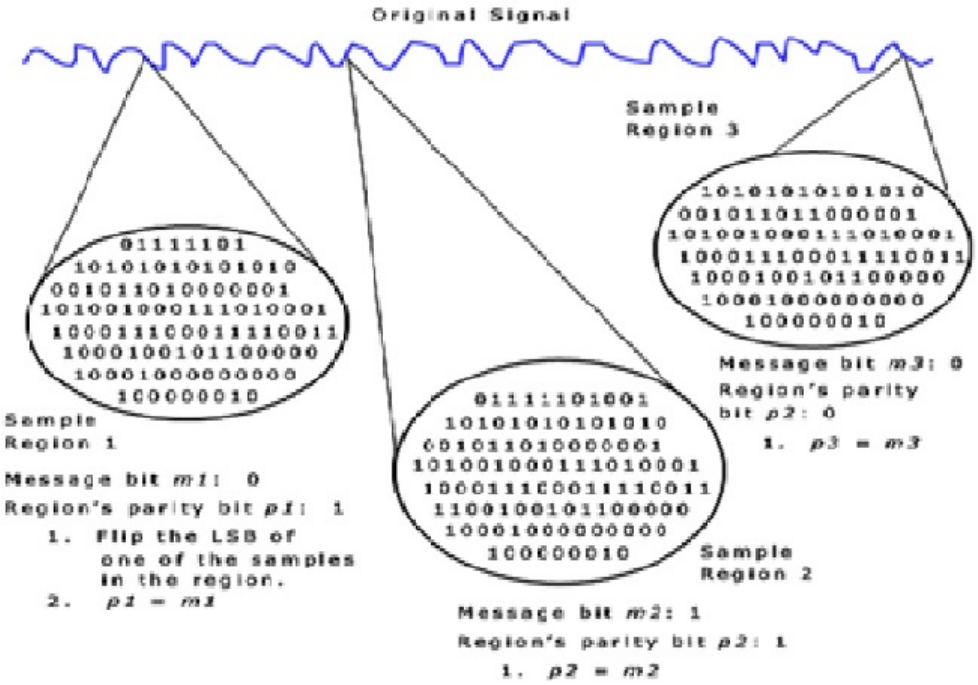


Figure 3.1: Parity Coding Procedure

Instead of breaking a signal down into individual samples,the parity coding method breaks a signal down into separate regions of samples and encodes each bit from the secret message in a sample region's parity bit. If the parity bit of a

selected region does not match the secret bit to be encoded, the process flips the LSB of one of the samples in the region. Thus, the sender has more of a choice in encoding the secret bit, and the signal can be changed in a more unobtrusive fashion. Using the parity coding method, the first three bits of the message 'HEY' are encoded in the following figure. Even parity is desired. The decoding process extracts the secret message by calculating and lining up the parity bits of the regions used in the encoding process. Once again, the sender and receiver can use a shared secret key as a seed in a pseudorandom number generator to produce the same set of sample regions.

Steps for Data embedding:

1. Read the cover audio signal.
2. Read the audio signal to be embedded, its size less than the size of the cover audio signal and convert it into binary sequence of message bits.
3. Depending upon the value of the message bit to be embedded (0/1), the LSB of the sample of the cover audio signal is modified or unchanged.
4. If the message bit to be embedded is 0, then the LSB of the sample of the cover audio signal is modified or unchanged such that the parity of the sample after embedding of this message bit is even.
5. If the message bit to be embedded is 1, then the LSB of the sample of the cover audio signal is modified or unchanged such that the parity of the sample after embedding of this message bit is odd.
6. The modified cover audio samples are then written to the file forming the stego audio signal

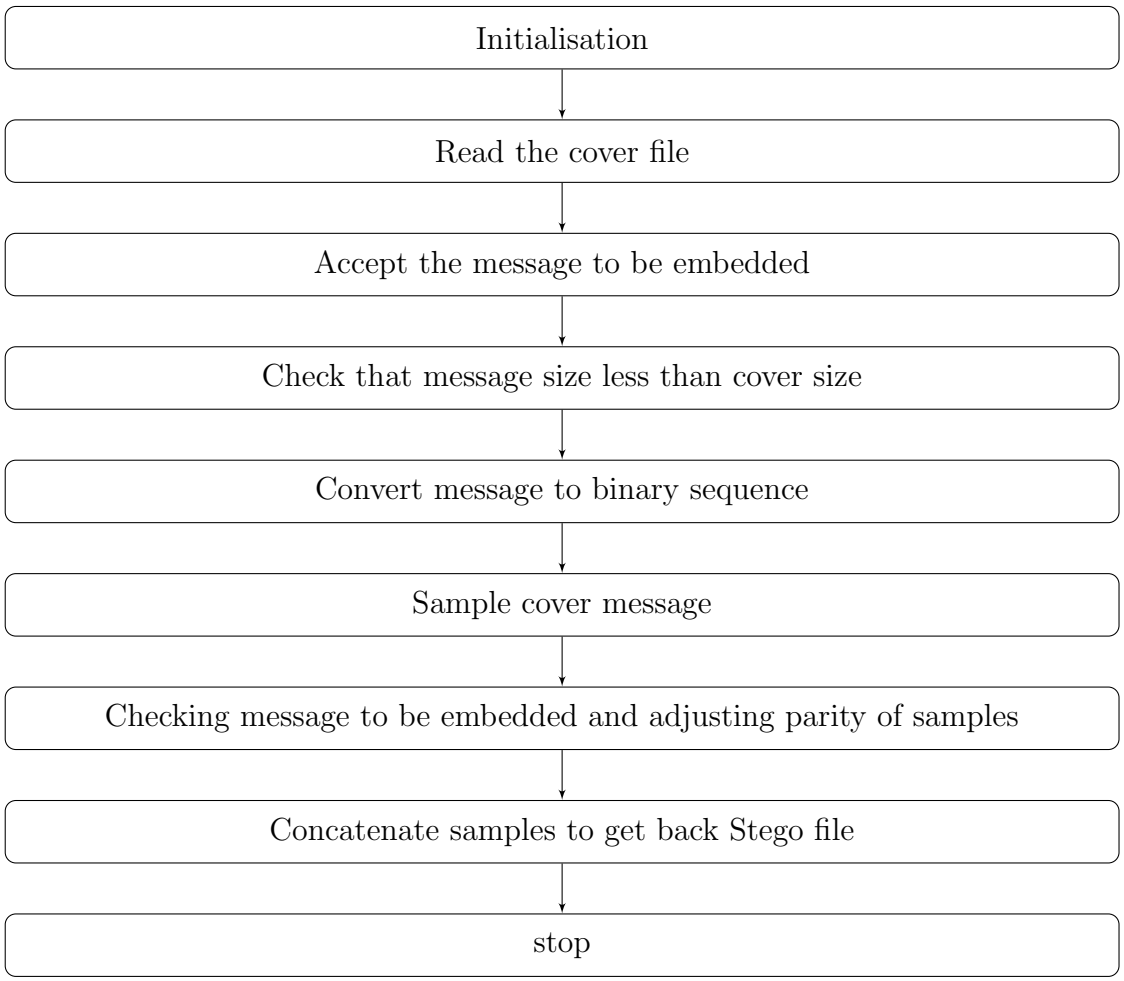
Steps for Data Extraction/Retrieval:

1. The Stego audio file is read.
2. The parity of every sample of the stego is checked.
3. If the parity is even, then the message bit retrieved is 0.
4. If the parity is odd, then the message bit retrieved is 1.
5. After every such 16 message bits are retrieved, they are converted to decimal equivalents.
6. Finally the secret message is reconstructed.

The sender has more of a choice in encoding the secret bit, and the signal can be changed in a more unobtrusive manner.This method like LSB coding is not robust in nature.

3.1 Encoder Implementation

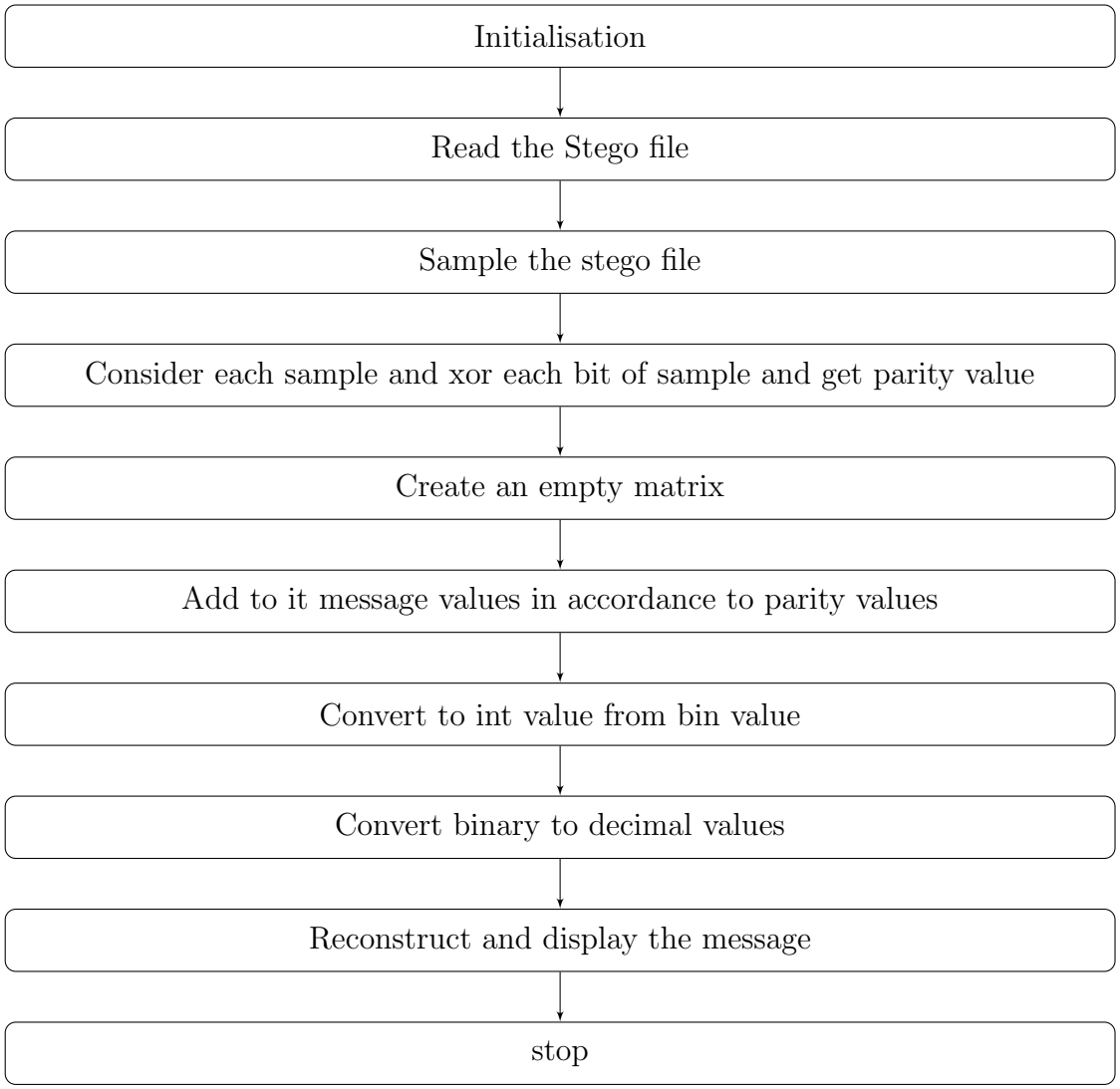
The parity coder is actually an improved version of LSB coder.The parity code is developed with a GUI for interactions.When the code is excecutes the MATLAB M-fig file gets initialised in the background before we actually see the GUI.To encode a message audio file of wave format is read and the message to embed is taken as input via the GUI.Before the actual encoding starts the code determines the length of the message and length of the cover audio and finds out whether the message can be embedded in the intended cover signal.If it is compatible the message is converted to binary sequence and is broken down to many samples.The embedding procedure then starts and data is hided by adjusting the parity of the samples.Concatenate the samples to obtain audio signal back.



3.2 Decoder Implementation

As in the encoder, the GUI is initialised first.Then the cover signal with hidden message is read, to get the parity value of each sample, cover signal is analysed after sampling it, an empty matrix is initialised and message value in accordance with the parity value is recorded in that matrix,convert these integer values to binary

values and then binary to decimal equivalents, the ASCII values corresponding each decimal is recorded and the message is displayed.



3.3 Simulation Result

The following is a screenshot of the simulated result of the parity coding technique.It shows the plot of an audio signal before and after the encoding procedure

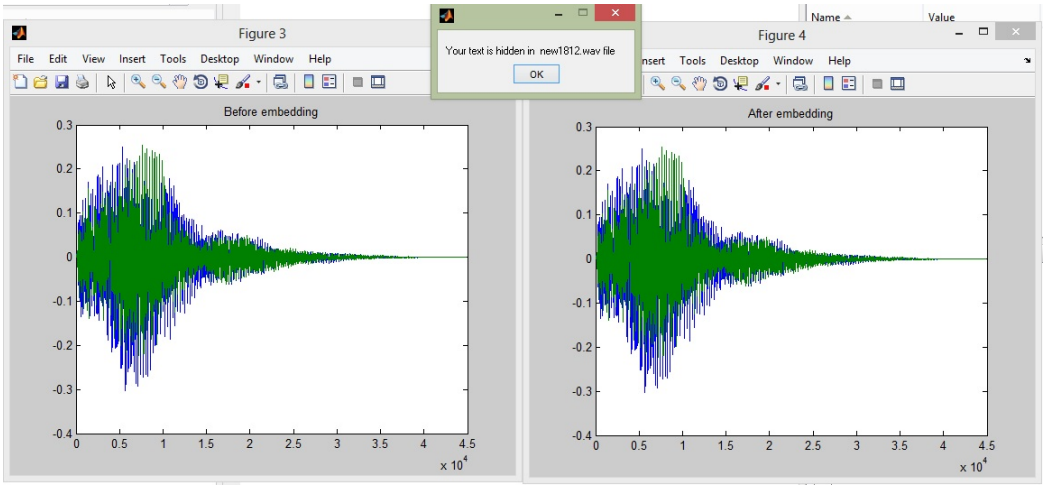


Figure 3.2: Parity Coding result

3.4 Performance

The performance of the code is evaluated in terms of SNR and MER with varying cover bits and embedded bits.

Audio(.wav)	Cover Bits(KB)	Msg length	MER	SNR(db)
Button	225	24	3.7229	54.9655
Beep	178.1	24	2.9294	52.1792
One	79	24	2.837	51.2007

Table 3.1: By varying cover bits

Message length was kept constant and the cover bits count was kept on varying, and it was found that for a large increase in size of cover bits the MER varies only in the range of 3.7 to 2.8, which implies that the code is very much effected by errors,which makes it not suitable for long diatance transmission, SNR also varies significantly for increase in cover bit count.When the cover bits were 178.1KB the SNR was 51.328db and for 79KB SNR was 51.2007db, which means length of cover bits is in direct propotion with SNR.

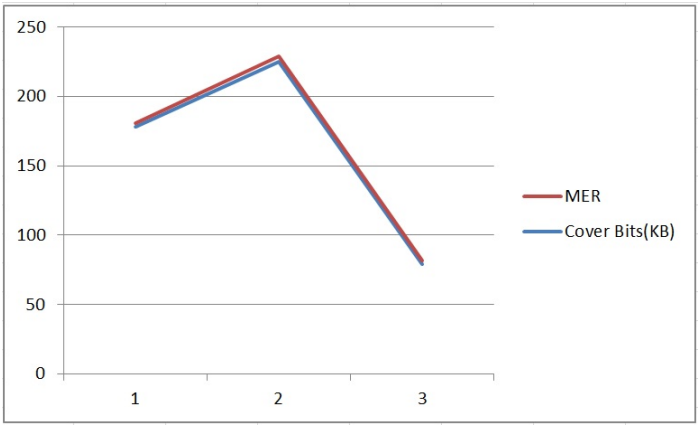


Figure 3.3: MER plot

The Inference from MER plot is that MER changes in propotion with cover bit count but there is a only a slight variation for MER even for large increase in cover bit count and it is in direct propotion with cover bit count.The variation is also very large even for a slight change in cover bits.

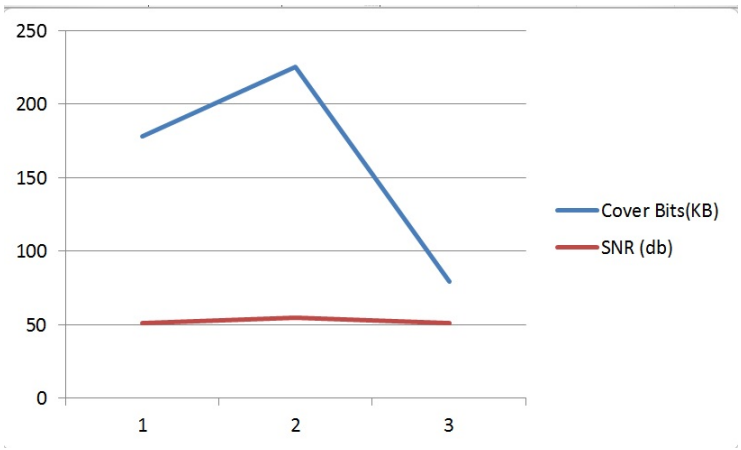


Figure 3.4: SNR plot

The SNR plot shows that there is a reasonably large change in SNR for change in cover bits,while comparing the values with the constant cover bit table its quite evident that SNR is in direct propotion with length of cover bits.

Audio(.wav)	**Cover Bits(KB)**	**Msg length**	**MER**	**SNR(db)**
Beep	178.1	24	2.9294	51.2416
Beep	178.1	168	3.1361	52.7834
Beep	178.1	1000	3.4071	53

Table 3.2: Using Beep.wav

In this particular analysis cover bits is varied keeping the embedded bits as constant.When the message length was 1000 the SNR was found to be 53db,also the

MER was found as 3.4071,whereas when message length reduced to 24 the SNR also reduced to 51.2416 and MER reduced to 2.9294 which implies SNR and MER varies in propotion with message length, in comparison with LSB the MER and SNR values has increased.

Audio(.wav)	**Cover Bits(KB)**	**Msg length**	**MER**	**SNR(db)**
Button	225	24	3.7229	54.9655
Button	225	168	3.8211	55.1002
Button	225	1000	4	55.4454

Table 3.3: Using Button.wav

The cover bits are kept constant but having higher value, the emededded bit count is chosen to be 24,168,1000.When the message length was 24, MER was recorded as 3.7229 and SNR was 54.9655 but as we incresed the count to 1000, MER was recorded as 4 and SNR as 55.4454 which implies SNR and MER varies in propotion with message length,in comparison with LSB the MER and SNR values has increased.

Audio(.wav)	**Cover Bits(KB)**	**Msg length**	**MER**	**SNR(db)**
One	79	24	2.837	51.2007
One	79	168	3.0401	51.7988
One	79	1000	3.3766	52.6132

Table 3.4: Using One.wav

Parity Coding is nothing but an improed version of LSB,rather than hiding the message in a LSB of a sample, the sample as a whole is used for hiding th message bit.From the first table its quite evident that as the size of the cover bits increases there is a noticable increase in SNR value.This implies that as the number of samples increases the performance of the code also increases.The MER here shows a realatively good range of values than the LSB which reveals that the chnces of error getting into the hdden message is less.In security point of view its least secure of all the algorithms mentioned in this comparison on the grounds that if we know that there is message embedded in the signal it can be decoded very easily using any general logic, if the sample size is known the work is even more easy.Here while doing the comparison it was noted that as the sample size is reduced,the number of samples increses which eventually leads to an increase in SNR.

Chapter 4

PHASE CODING

The phase coding method works by substituting the phase of an initial audio segment with a reference phase that represents the data. The phase of subsequent segments is adjusted in order to preserve the relative phase between segments. Phase coding, when it can be used, is one of the most effective coding methods in terms of the signal-to perceived noise ratio.When the phase relation between each frequency component is dramatically changed, noticeable phase dispersion will occur. However, as long as the modification of the phase is sufficiently small (sufficiently small depends on the observer; professionals in broadcast radio can detect modifications that are imperceivable to an average observer), an inaudible coding can be achieved. 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.

Phase coding is explained in the following procedure:

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

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

3. Phase differences between adjacent segments are calculated.

4. 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:

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

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

6. 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 (consider Fig 4.1 for phase coding procedure).

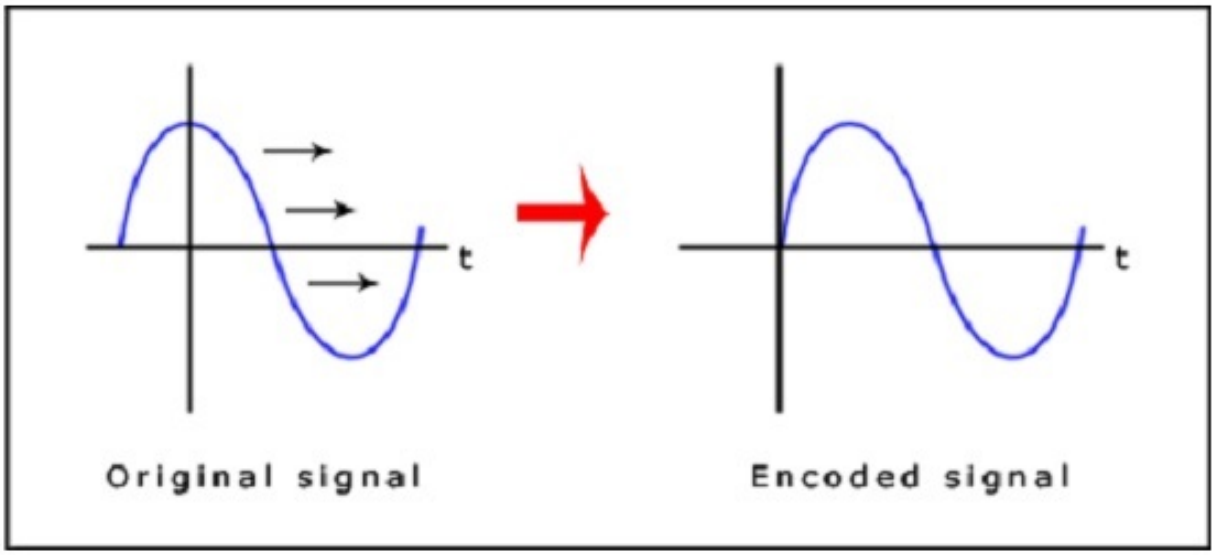


Figure 4.1: The signals before and after Phase coding procedure.

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

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 drawback 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, needs to be concealed.

In a normal communication channel, it is often desirable to concentrate the information in as narrow a region of the frequency spectrum as possible in order to conserve available bandwidth and to reduce power. The basic spread spectrum technique, on the other hand, is designed to encode a stream of information by spreading the encoded data across as much of the frequency spectrum as possible. This allows the signal reception, even if there is interference on some frequencies. While there are many variations on spread spectrum communication, we concentrated on Direct Sequence Spread Spectrum encoding (DSSS). The DSSS method spreads the signal by multiplying it by a chip, a maximal length pseudorandom sequence modulated at a known rate. Since the host signals are in discrete-time format, we can use the sampling rate as the chip rate for coding. The result is that the most difficult problem in DSSS receiving, that of establishing the correct start and end of the chip quanta for phase locking purposes, is taken care of by the discrete nature of the signal.

Phase coding, when it can be used, gives the best signal-to-perceived noise ratio. Phase discontinuity of an audio signal is perceptible when the phase relation between each frequency component of the signal is dramatically changed. Thus, inaudible phase coding can only be achieved by keeping the modification of the phase sufficiently small and slowly varying.

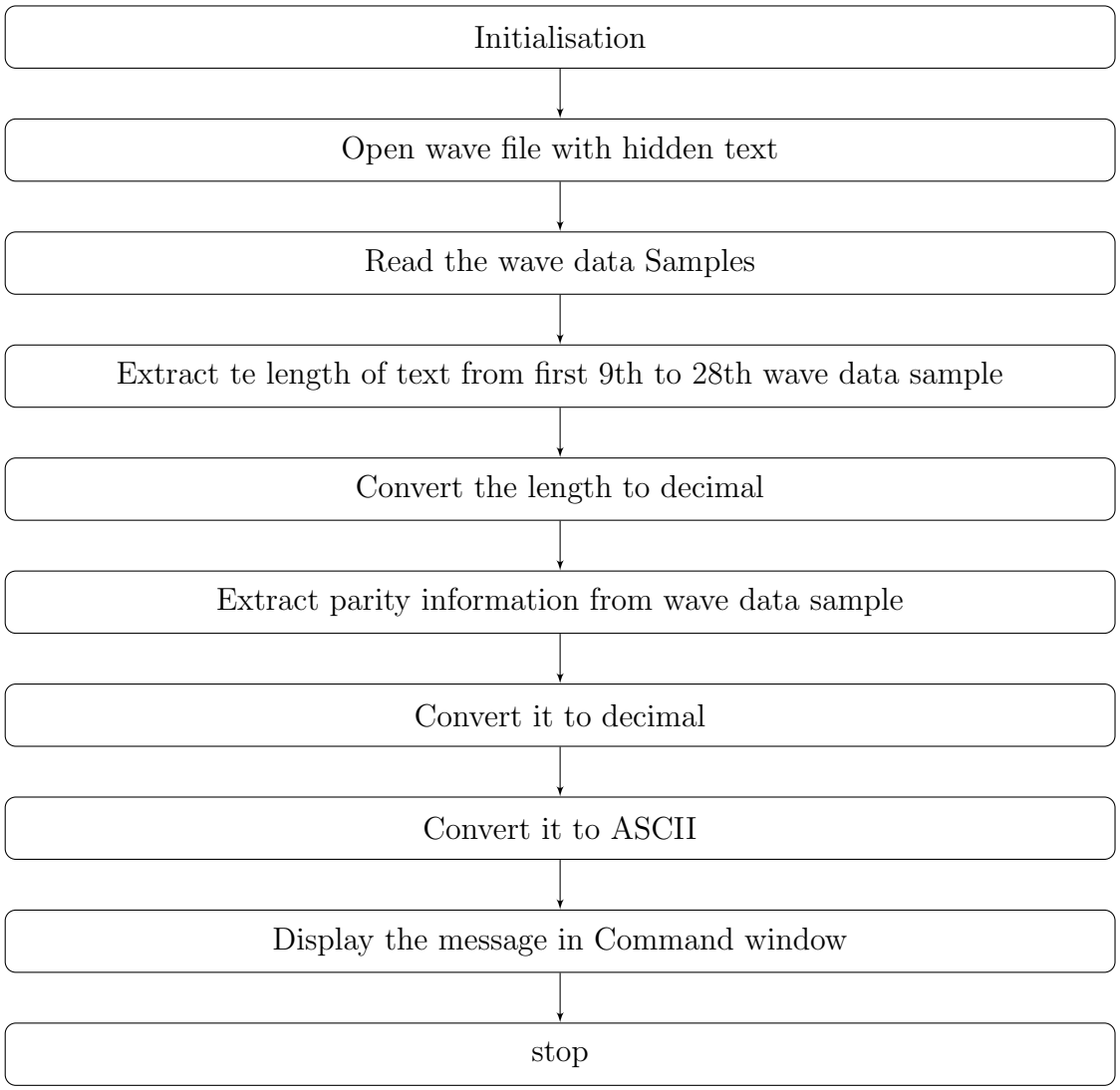
It is basically a method for hiding data in audio files that employs the manipulation of the phase of selected spectral components of the host audio file. This method has been demonstrated in uncompressed audio files and also has been demonstrated to survive MP3 compression and decompression. In the following we describe the method in more detail. Naturally occurring audio signals such as music or voice contain a fundamental frequency and a spectrum of overtones with well-defined relative phases. When the phases of the overtones are modulated to create a composite waveform different from the original, the difference will not be easily detected. Thus, the manipulation of the phases of the harmonics in an overtone spectrum of voice or music may be exploited as a channel for the transmission of hidden data.

4.1 Encoder Implementation

Since Phase coing was done in a GUI, initialisation is required just before the GUI itself appears and it is done as in parity coding,using the GUI the audio which is used as cover is accepted.It initially samples the audio 41st to 43rd byte includes the length of data sample,Copy the 16-bit wave data sample from the 44th byte.Convert it to double and then to binary,now take the message convert it to a coloumn vector.Hide the identity in first 8 wave data sample by modulating the phase of the cover audio signal,the binary length of the message is embedded in 9th to 28th data sample.The message's equivalent binary is embedded from 29th position of wave sample,then a new wave file is opened in writting mode the header of the original cover and the message embedded cover is copied in to this new file and hence the audio is reconstructed.

4.2 Decoder Implementation

Decoding is relatively simple,the audio file with secret message is read and is made into various wave data samples.Extract the length of the text from first 9th to 28th wave data samples,convert it to decimal equivalents,extract the message info from phase changes in data samples,Convert the message to decimal equivalents and compare with ASCII values and display message on the command window.



4.3 Simulation Result

The following is a screenshot of the simulated result of the phase coding technique.It shows the plot of an audio signal before and after the encoding procedure.

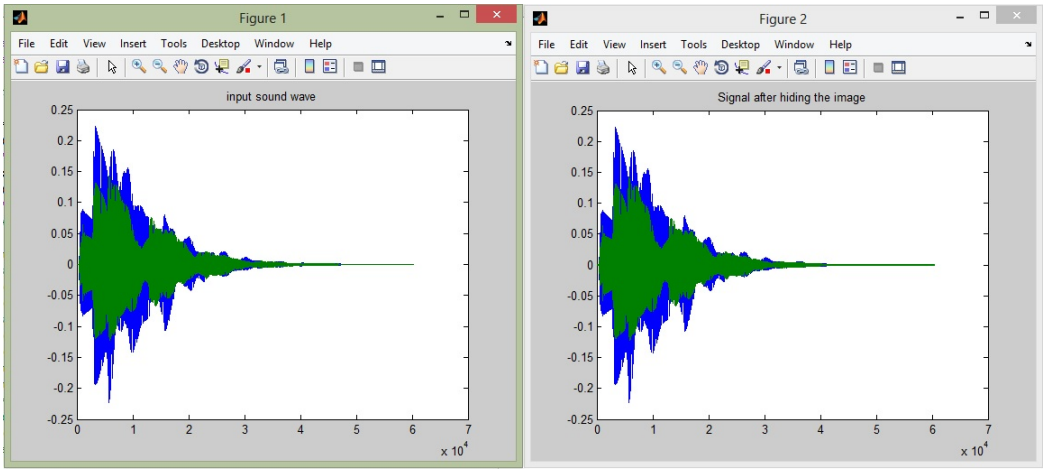


Figure 4.2: Phase Coding result

4.4 Performance

The performance of the code is evaluated in terms of SNR and MER with varying cover bits and embedded bits.

Audio(.wav)	Cover Bits(KB)	Msg length	MER	SNR(db)
Button	225	24	3.7229	54.9655
Beep	178.1	24	2.9294	48.1792
One	79	24	2.837	51.2007

Table 4.1: By varying cover bits

Message length was kept constant and the cover bits count was kept on varying, and it was found that even for a large increase in cover bits the MER varies only in the range of 7.7 to 7.9,which implies that the code is not effected by errors,however the SNR varies significantly.When the cover bits were 178.1KB the SNR was 73.4107dB and for 79KB SNR was 69.1113dB,which means length of cover bits is in direct propotion with SNR.The high values of SNR clearly reveals that the code is superior in performance in comparison to other codes in the study.

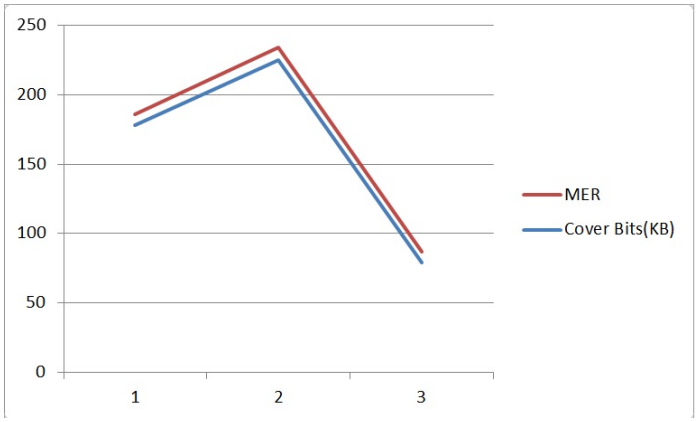


Figure 4.3: MER plot

The Inference from MER plot is that MER changes in propotion with cover bit count but there is relatively good variation in MER even for slight increase in cover bit count.

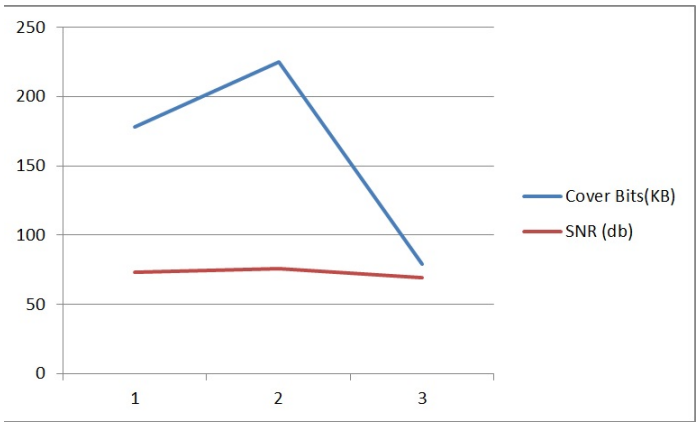


Figure 4.4: SNR plot

The SNR plot shows that there is a reasonably large change in SNR for change in cover bits,while comparing the values with the constant cover bit table its quite evident that SNR is in direct propotion with length of cover bits. This plot gives the highest SNR values for the records,the variations are better than the LSB but less than parity.

Audio(.wav)	**Cover Bits(KB)**	**Msg length**	**MER**	**SNR(db)**
Beep	178.1	24	7.9571	73.107
Beep	178.1	168	8.3756	74.9825
Beep	178.1	1000	8.9	77.0835

Table 4.2: Using Beep.wav

In this particular analysis cover bits is varied keeping the embedded bits as constant.When the message length was 1000 the SNR was found to be 77.0835db,also the MER was found as 8.9,whereas when message length reduced to 24 the SNR also

reduced to 73.107 and MER reduced to 7.9571 which implies SNR and MER varies in propotion with message length.

Audio(.wav)	**Cover Bits(KB)**	**Msg length**	**MER**	**SNR(db)**
Button	225	24	8.8444	75.7322
Button	225	168	9.0102	77.1587
Button	225	1000	9.9932	80.386

Table 4.3: Using Button.wav

The cover bits are kept constant but having higher value,the emededded bit count is chosen to be 24,168,1000.When the message length was 24 MER was recorded as 8.8444 and SNR was 75.7332 but as we incresed the count to 1000, MER was recorded as 9.9932 and SNR as 80.386 which implies SNR and MER varies in propotion with message length.

Audio(.wav)	**Cover Bits(KB)**	**Msg length**	**MER**	**SNR(db)**
One	79	24	7.7002	61.1113
One	79	168	7.9278	71.626
One	79	1000	8.2951	74.4863

Table 4.4: Using One.wav

Phase coding is known to be the most efficient stegnographic coding system known till date and now thats experimentally proved in this comparison.As this coding changes the phase of the cover signal corresponding to the message to embed,the change in phases are not recognisable as they are beyond the audible limit of human ear.the changes in the values of SNR in high ranges implies this coding in signal processing and data retrieval need not require the original cover signal as such, some amount of noise and corruption in samples are allowable within certain limits for efficient decoding of message signals.It is the most idle one for long distance transmissions.The MER in the first table for different cover signals implies the effect of receiving an error signal is very low,the relative signal strength is a high values which makes it easy for processing,the capacity for embedding secret message is very low for this type of coding as it is directly propotional to the lenght of cover signal.how ever it capable of operating at high data rates upto range of 333kbps.

Chapter 5

SPREAD SPECTRUM CODING

In a normal communication channel, it is often desirable to concentrate the information in as narrow a region of the frequency spectrum as possible in order to conserve available bandwidth and to reduce power. The basic spread spectrum technique, on the other hand, is designed to encode a stream of information by spreading the encoded data across as much of the frequency spectrum as possible. This allows the signal reception, even if there is interference on some frequencies. While there are many variations on spread spectrum communication, we concentrated on Direct Sequence Spread Spectrum encoding (DSSS). The DSSS method spreads the signal by multiplying it by a chip, a maximal length pseudorandom sequence modulated at a known rate. Since the host signals are in discrete-time format, we can use the sampling rate as the chip rate for coding. The result is that the most difficult problem in DSSS receiving, that of establishing the correct start and end of the chip quanta for phase locking purposes, is taken care of by the discrete nature of the signal. Consequently,a much higher chip rate, and therefore a higher associated data rate, is possible. Without this, a variety of signal locking algorithms may be used, but these are computationally expensive.

Procedure: In DSSS, a key is needed to encode the information and the same key is needed to decode it. The key is pseudorandom noise that ideally has flat frequency response over the frequency range, i.e., white noise. The key is applied to the coded information to modulate the sequence into a spread spectrum sequence.

The DSSS method: The code is multiplied by the carrier wave and the pseudorandom noise sequence, which has a wide frequency spectrum. As a consequence, the spectrum of the data is spread over the available band. Then, the spread data sequence is attenuated and added to the original file as additive random noise (see Figure 4). DSSS employs bi-phase shift keying since the phase of the signal alternates each time the modulated code alternates (see Figure 2.5).For decoding, phase values f0 and f0 + p are interpreted as a â0â or a â1,â which is a coded binary string. Spread Spectrum is shown in Figure 2.5

In the decoding stage, the following is assumed::

1. The pseudorandom key is maximal (it has as many combinations as possible

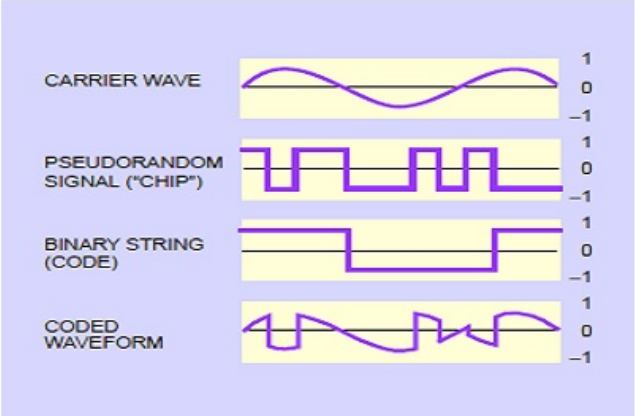


Figure 5.1: Spread spectrum information encoded by the direct sequence method .

and does not repeat for as long as possible). Consequently it has a relatively flat frequency spectrum.

2. The key stream for the encoding is known by the receiver. Signal synchronization is done, and the start/stop point of the spread data is known

3. The following parameters are known by the receiver: chip rate, data rate, and carrier frequency.

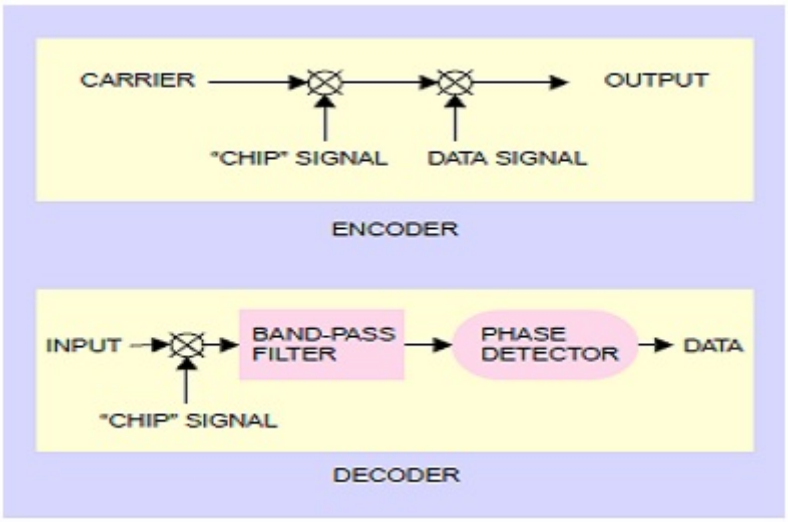


Figure 5.2: Spread spectrum information encoded by the direct sequence method.

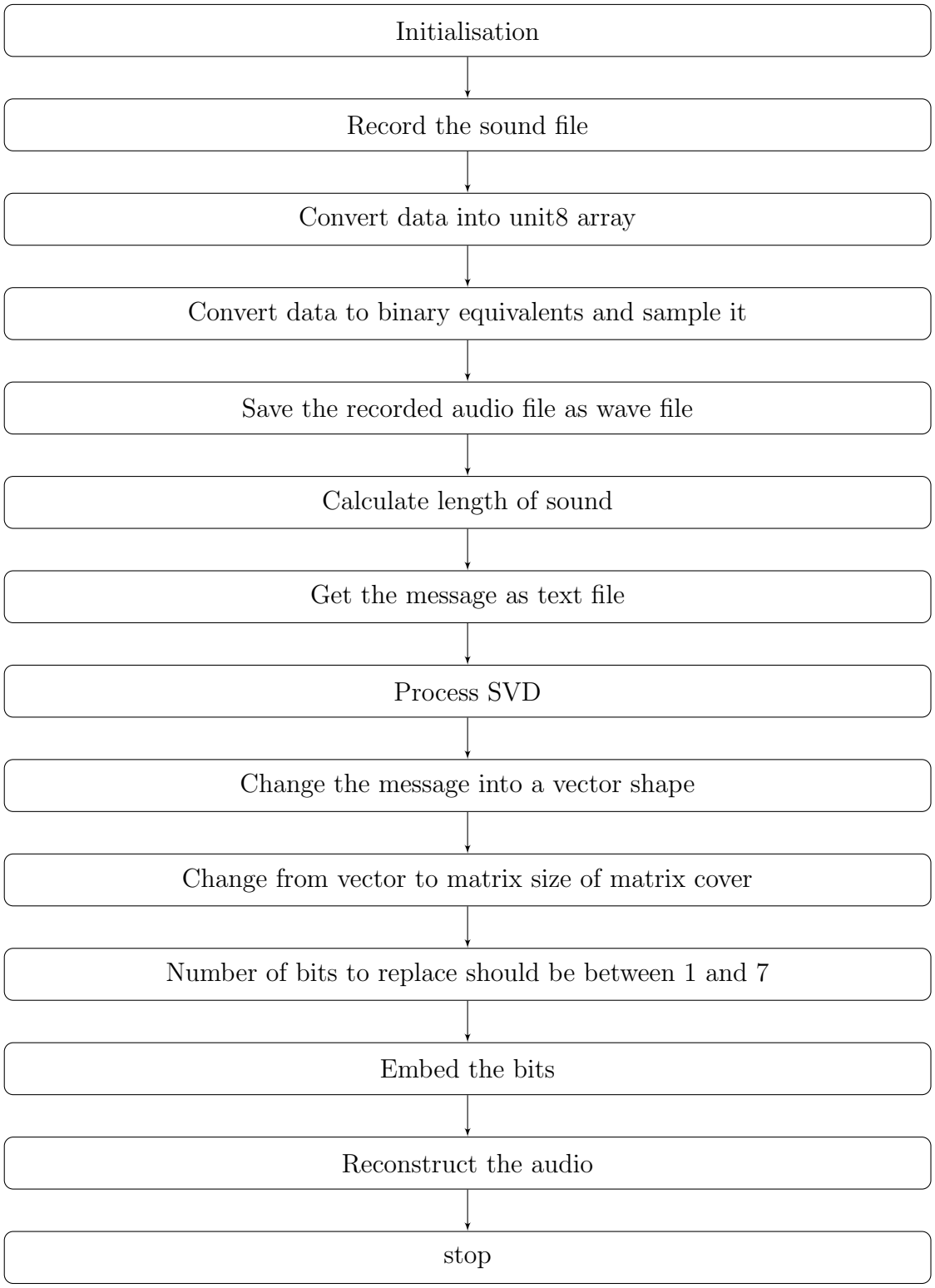
In the context of audio steganography, the basic spread spectrum (SS) method attempts to spread secret information across the audio signal's frequency spectrum as much as possible. This is analogous to a system using an implementation of the LSB coding that randomly spreads the message bits over the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

5.1 Encoder Implementation

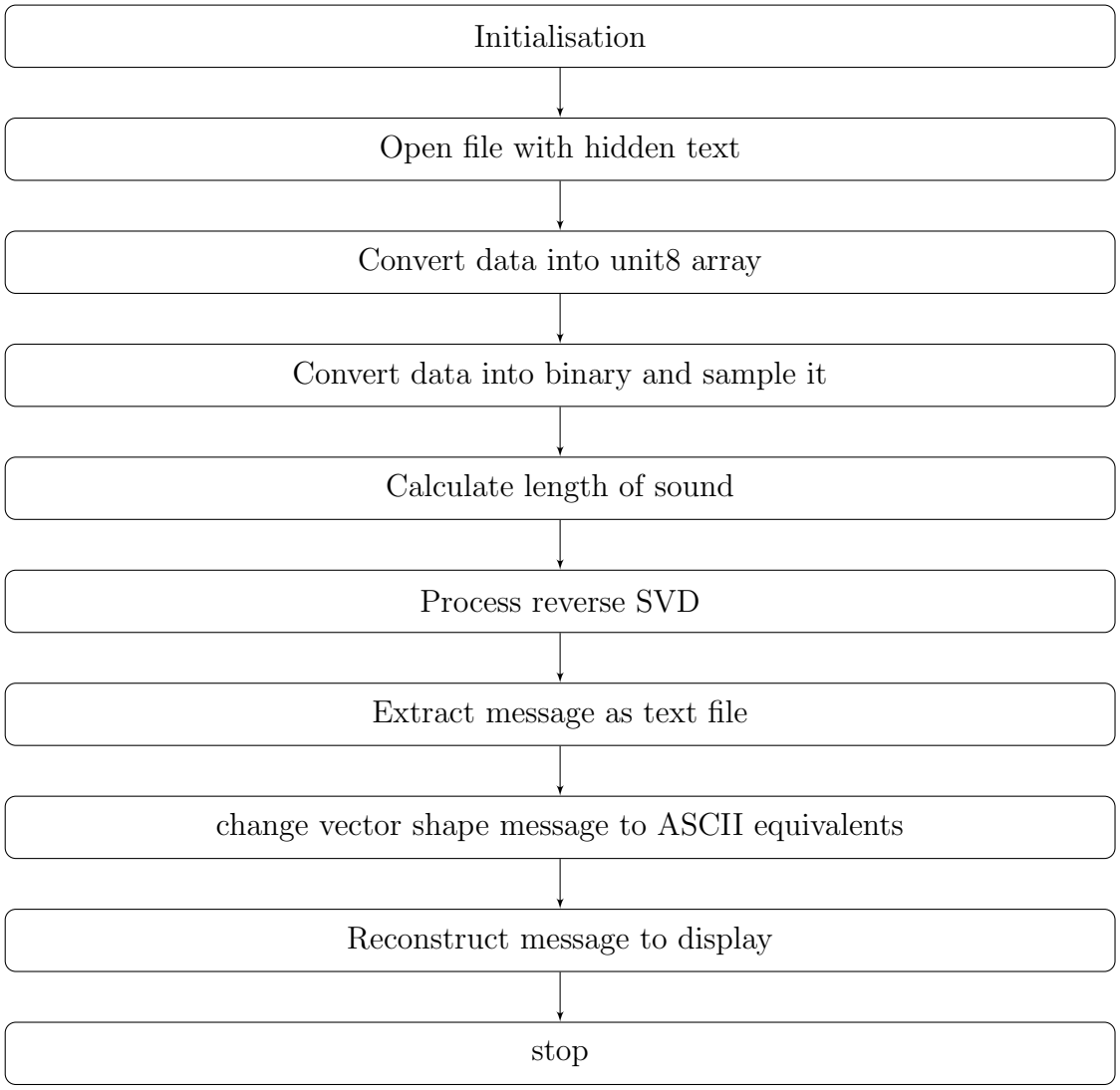
Being a code with the GUI as we execute the file the GUI itself initialises and then reads the audio cover file.It samples the audio cover signal into different data wave samples, these samples are treated as an array in integer form and is converted to double form for display.Save the recorded audio file as wave file with header, calculate the length of the sample. Start the development of encryption key which is nothing but a PN sequence and the process is popularlly called as security variable development,save the message in a text file, read that and convert that into a vector matrix, the number of bits to replace or multiply should be between 1 and 7. Embed the bits into the audio and reconstruct the audio.



5.2 Decoder Implementation

As in the encoder the GUI is initialised and then the audio with the hidden text is read,sampled into smaller data samples and is converted then to its equivalent in-

teger array,the double values are calculated from these integer values,so process the reverse SVD,either the PN sequence is extracted or is generated in synchronous with transmitter,extract message as text file, get its ASCII equivalents to reconstruct the message.



5.3 Simulation Result

The following is a screenshot of the Simulated result of the spread spectrum coding technique.It shows the plot of an audio signal before and after the encoding procedure

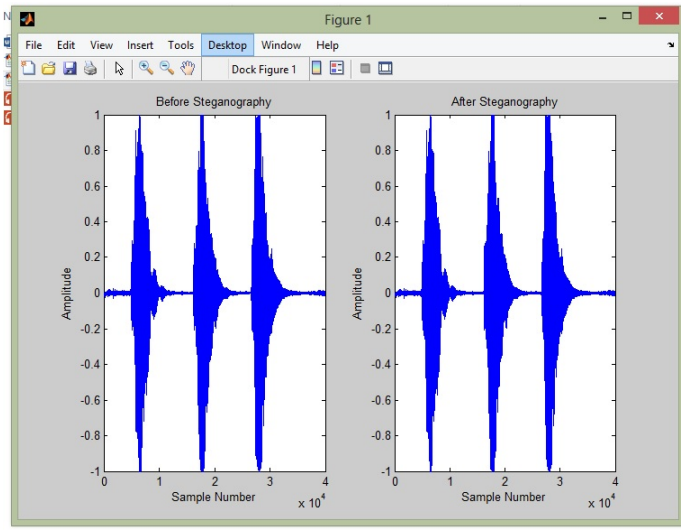


Figure 5.3: Spread Spectrum Coding result

5.4 Performance

The performance of the code is evaluated in terms of SNR and MER with varying cover bits and embeded bits

Audio(.wav)	Cover Bits(KB)	Msg length	BER
One	79	24	0.125
One	79	168	0.0893
One	79	1000	7.2

Table 5.1: Spread Spectrum Performance

In spread spectrum coding technique the performance is usually measured in terms of BER(Bit Error Rate).The values of BER can be analysed as follows when a message of 1000 bits is embedded into the cover signal,there is a chance of 7 bits getting an error while spreading and despreading of message signal.There will changes in the plots of cover image before and after the embedding process.This coding is less affected by noise and highly secure.As we are encrypting with a PN sequence only the intented person can decrypt the hidden message.The only drawback that we noticed was there will be relative change in size and length of the message signal and before encoding process,it is also vulnerable for time scale modifications.The hiding rate has to be limited to 20bps.

Chapter 6

MULTILEVEL IMPERCEPTIBLE DATA HIDING

Cryptography is an important component of secure information and communications systems and a variety of applications have been developed that incorporate cryptographic methods to provide data security. Cryptography is an effective tool for ensuring both the confidentiality and integrity of data.

Background study for this code development:

1. Basic concepts of cryptography, cryptology, cryptosystem and fundementatal concepts of caeser cipher, features and break analysis and various related cipher like mono, homo and PolyGram and other substitution ciphers.
2. Detailed study about transposition, substitution, transformation and other related Encryption types symmetric and asymmetric algorithms and related key concepts.
3. Gathered information about different ciphers like block cipher and stream cipher methodology related issues, challenges and other features and draw backs of this system.
4. Various attacking methods especially for cipher text, concentrated on cryptanalysis and brute force attack.
5. Mathematical concepts of substitutions, permutation, modulus functions, factorization concept and related issues.
6. Study about Block size, Key size, Number of rounds, Sub key generation algorithm, Round functions, Fast software encryption or decryption Braking analysis
7. Differentiate various attacking methods for cipher text in the form Cipher text, Known plaintext, Chosen plaintext, Chosen cipher text, Chosen text and analysis of various essential ingredients of symmetric system, secret key, cipher text, encryption and decryption and algorithm development.

8. Various key concepts private keys, public keys, session keys, master keys and proposed genetic keys.

9. Study about symmetric and asymmetric algorithms like, DES, AES and other related concepts, it was analyzed in various ways performance, time taken analysis, processing power and other issues based on cryptography aspects.

10. Done base work based on the different analysis of various substitution ciphers, exiting algorithms, related issues of attacking cipher text, features and international journals published recently on the web and other related articles and books.

The simplified algorithm for coding is defined as

1. Read and scale the Audio file in to a 2D image matrix arrangement

2. Convert message to 8bit and decompose cover image to RGB components.

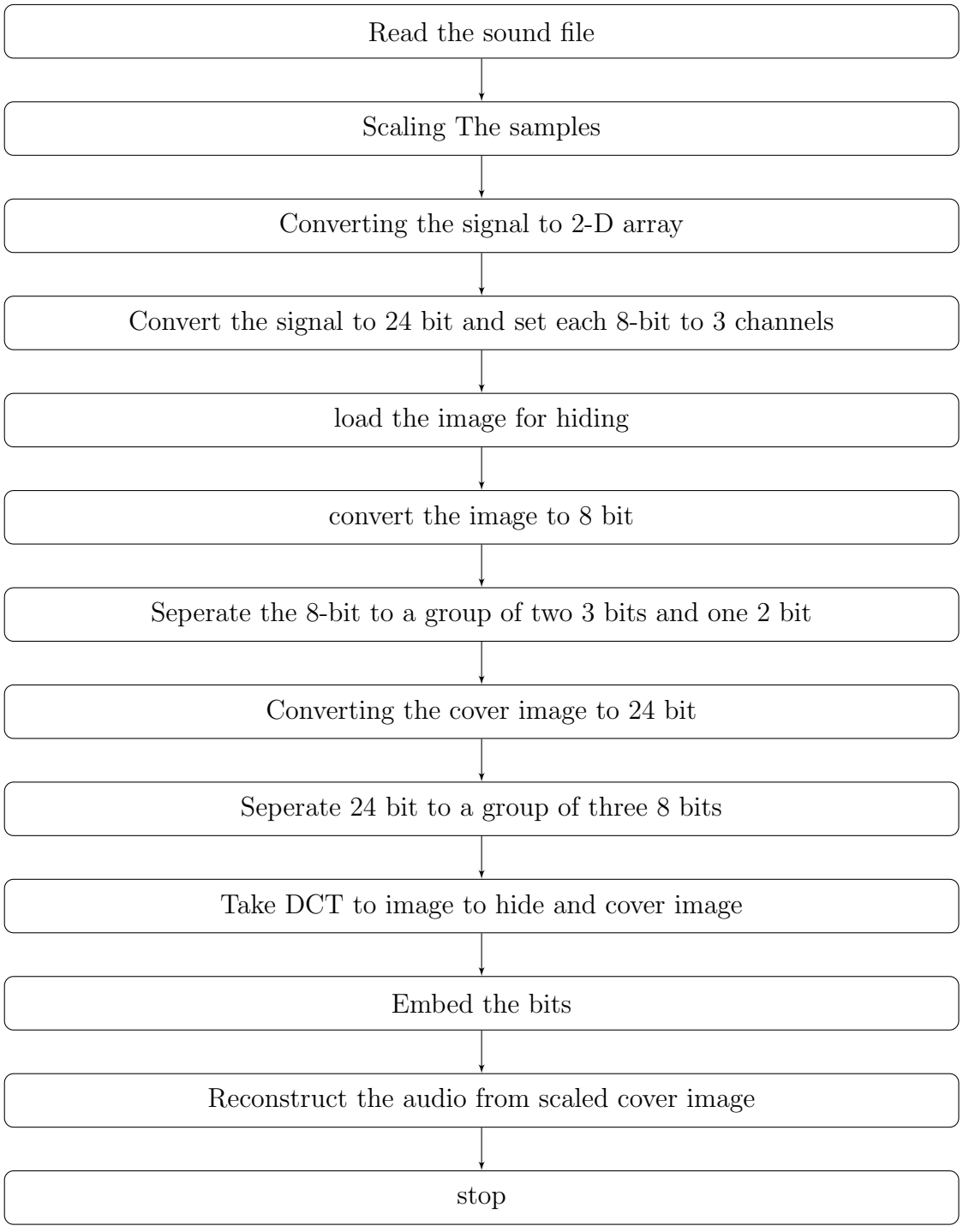
3. Split 8bit message in to pair of 3,3,2 bits and embed in to R,G,B planes respectively.

4. Concatenate back the RGB planes to image and scale the image back to Audio range.

5. For Decoding convert the audio to Image and split the image to RGB planes as in previous steps.

6.1 Encoder Implementation

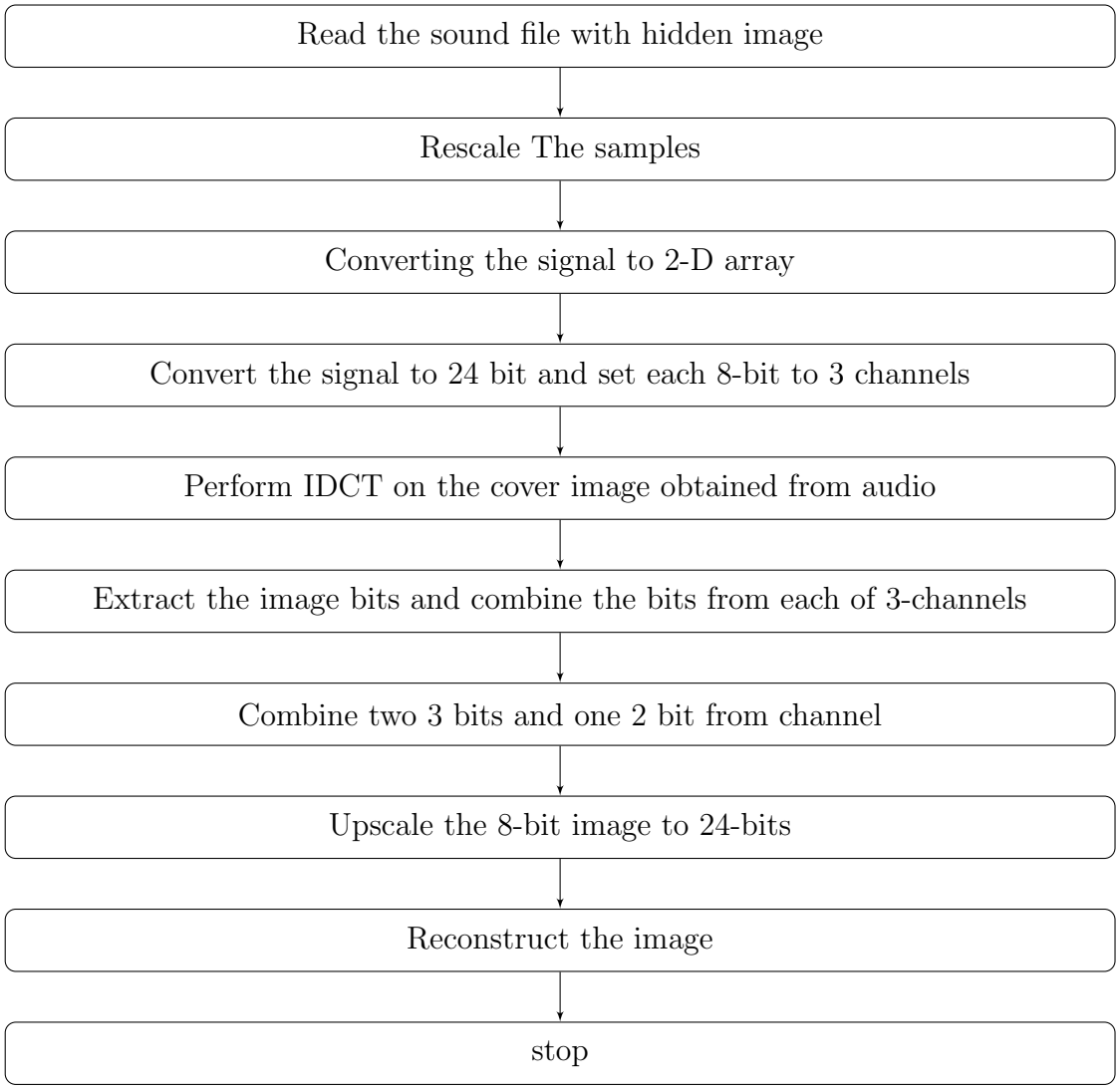
Read the audio file,scale it to a particular range such that it is eliminates the negative part of the signal,now convert itinto a 2-D array,this can e written as an image, hence an image equivalent to audio is obtained,to make it a colour image convert the signal to a 24-bit such that we can set each 8-bit to one of RGB channels.Load the image for hiding,covert the image to 8-bit which is nothing but grayscaling the image ,hide two 3-bits to two of the planes of RGB and the remaining 2-bit to the un allocated plane of RGB.now convert the cover image to 24-bit,seperate the 24-bits to a group of 8-bits.The message has been hidden to cover and the audio can be retrieved after the upscaling the samples.



6.2 Decoder implementation

The audio with the hidden message is read,rescale the sample and convert it to an equivalent image just as we did in the encoder part,the equivalent image is splitted into 3 channels of 8-bits each ,extract the image bits from the cover audio(image) and combine the bits from each of 3-channels .Combine the two 3-bit and one 2-bit info to get the binary form of the hidden message,by appending it to 24-bits it can be

easily converted to the decimal equivalents,this can help to reconstruct the hidden data which is an image in our case.



6.3 Simulation Result

The following is a screenshot of the Simulated result of the MIDH technique.It shows the plot of an audio signal before and after the encoding procedure

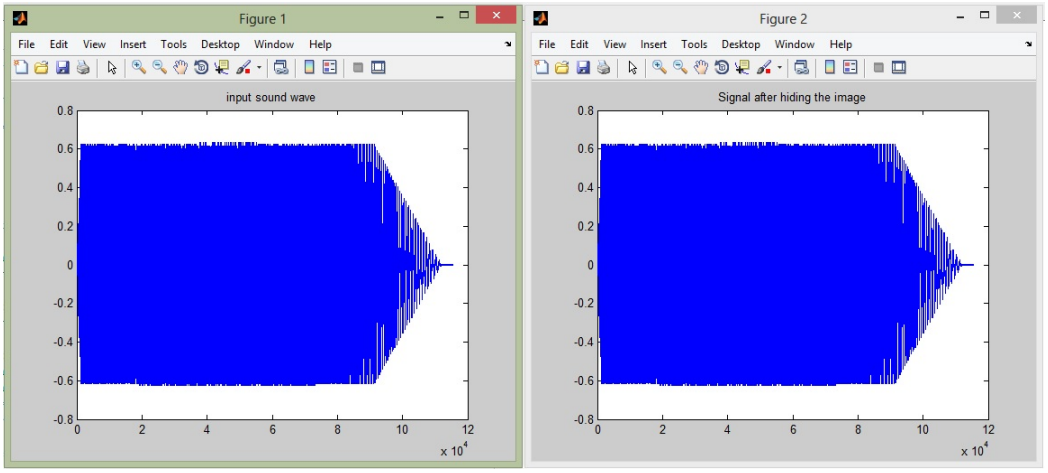


Figure 6.1: MIDH Coding result

6.4 Performance

The performance of the code is evaluated in terms of SNR and MER with varying cover bits and embeded bits.

Audio(.wav)	Cover Bits(KB)	Msg length	Delta	S.Pwr	N.Pwr	SNR(db)
Button	225	24	6.0075	3.1455	3.3103	110.576
Button	225	168	6.2686	3.1836	3.3215	111.744
Button	225	1000	6.5436	3.5078	3.5683	114.958

Table 6.1: MIDH Performance

This coding is developed with the intention of high SNR and relatively low MER,which actually paves to a new efficient and robust algorithm taking the best of all the existing algorithms.Clearly reveals the SNR performance of this code is awesome in all means.It avoids lossy data compressions, the MER shows its less effected by noise even if higher number of bits are added to it. SNR values reveals its the most apt one for signal processing and long distance data transmission.

Chapter 7

RESULTS AND CROSS COMPARISON

The results obtained after the implementation of various algorithms are shown for the sake of comparison

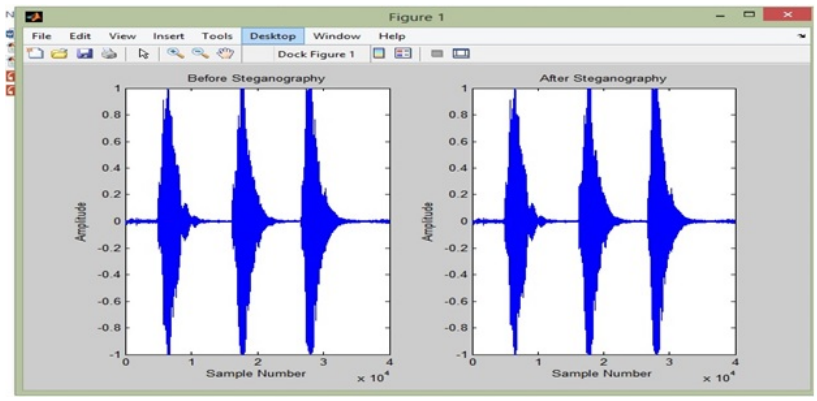


Figure 7.1: LSB Coding result

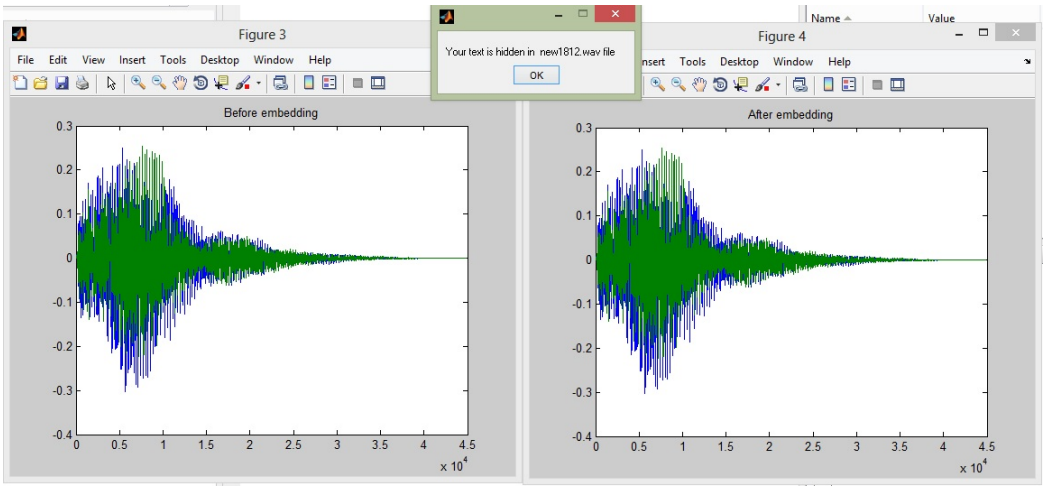


Figure 7.2: Parity Coding result

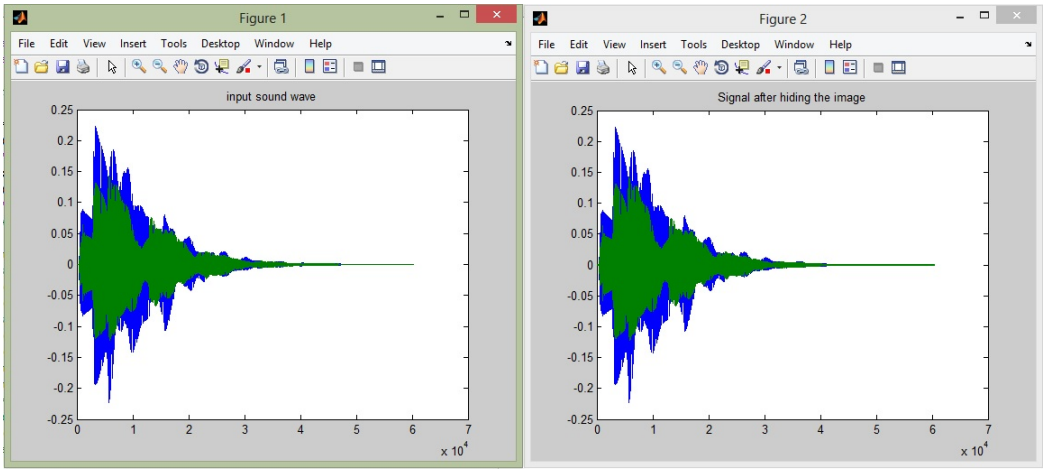


Figure 7.3: Phase Coding result

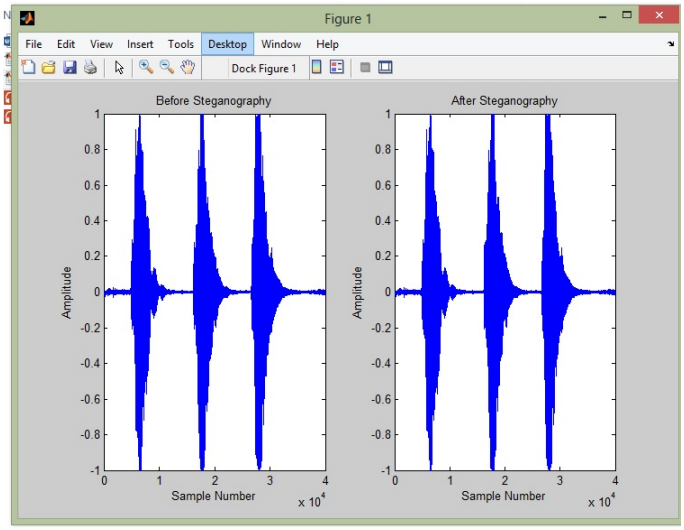


Figure 7.4: Spread Spectrum Coding result

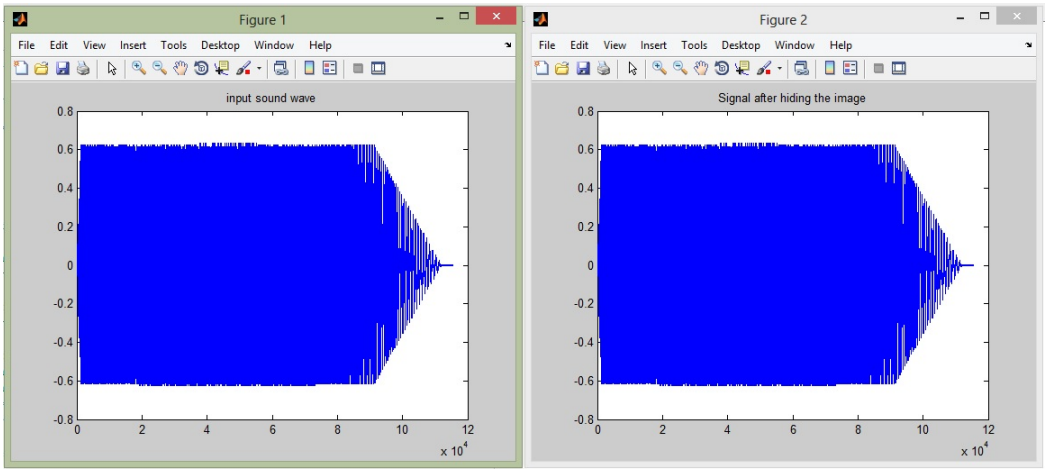


Figure 7.5: MIDH Coding result

Coding tech	Audio.wav	Cover Bits(KB)	Msg len	MER	SNR(db)
LSB	Beep	178.1	24	2.4756	48.1792
Parity	Beep	178.1	24	2.9294	51.2416
Phase	Beep	178.1	24	7.9571	73.4107
Spread Spectrum	Beep	178.1	24	**	.1250(BER)
MIDH	Beep	178.1	24	6.0075	110.576

Table 7.1: Cross Comparison

The one site view at the plot and table of SNR and MER of the popular excisting algorithms shows that phase coding is definitely top notch.MIDH, the new algorithm is superior than that as it is having a SNR of 110.576 dB where the highest SNR only second to it is 73 .4107 dB.The MER of MIDH is in the range of 6.0075 but for that of phase is 7.9571 which implies the noise is least effected by errors,high SNR make it apt for signal processing applications.In short the newly developed Robust and effective MIDH technique is better than phase coding which was found to be the best among the known algorithms.

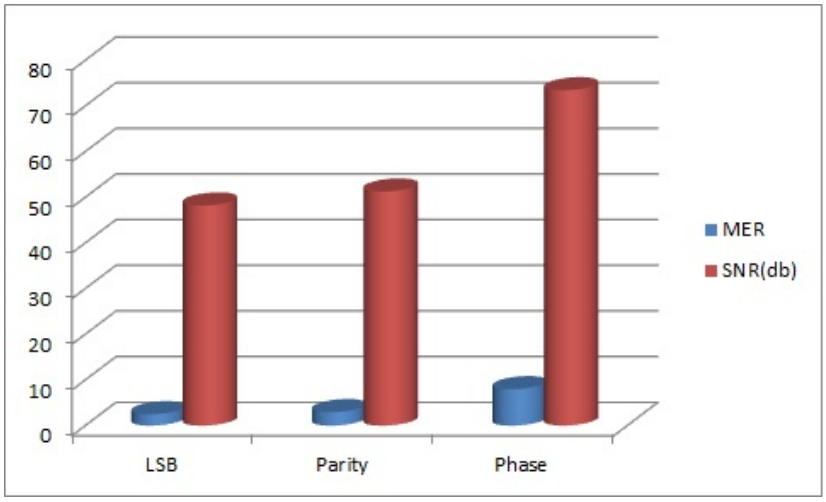


Figure 7.6: Cross Comparison Plot

The cross comparison reveals that among the known algorithms phase coding is superior in terms of SNR,the MER is also showing a high rate and very low capacity for hiding pulls it back from coming in as the most efficient code, since the phase changes are not audible to our ears in securuty point of view also it is top notch,from our test capacity is propotional to length of carrier,Since MER value is high its the idle one for short distance transmission.In case of LSB SNR decreases with sampling rate,so themessage gets corrupted very easily,but simplicity to impliment and relative good range of SNR makes it a very popular one used these days.In parity coding as the cover bits increased there is relative increase in SNR,but MER better than LSB,Low security is its major drawback.Since Spread spectrum is analysed in terms of BER,there is no standard as such to compare it with the other algorithms, but its decoding is very complex and even a slight error can corrupt the message.A large

cover file is required for data embedding.In security point of view it is highly secure.

Chapter 8

CONCLUSION AND FUTURE SCOPE

We have introduced a robust method of imperceptible audio data hiding. This system is to provide a good, efficient method for hiding the data from hackers and sent to the destination in a safe manner. This proposed system will not change the size of the file even after encoding and also suitable for any type of audio file format. Thus we conclude that audio data hiding techniques can be used for a number of purposes other than covert communication or deniable data storage , information tracing and finger printing, tamper detection. As the sky is not limit so is not for the development. Man is now pushing away its own boundaries to make every thought possible. So similarly these operations described above can be further modified as it is in the world of Information Technology. After designing any operation every developer has a thought in his mind that he could develop it by adding more features to it.

This proposed system provides an efficient method for hiding the data from the unauthorised ones,but still further analysis in practical conditions especially in unguided mode of data transmission are necessary for the approval of MIDH as an algorithm for commercial purposes,there is also scope for investigation on the grounds of sample rate, hiding rate, amplification, re-sampling, imperceptibility, filtering, quantisation, noise addition, transcoding etc....

Bibliography

[1] Data hiding via phase manipulation of audio signals (University of Rochester NY USA).

[2] ISS-IHAS model with reference to proposed e - cipher method (International Journal of advanced science and applications Vol.2 No.6 2011).

[3] Information hiding in audio signals (International Journal of advanced science and applications vol7. No.9 2010).

[4] Data hiding in multimedia using VLSI Technology (International conference on VLSI, Communication and instrumentation proceedings 2011).

[5] Data hiding in audio signal: A review (International journal of database theory and application 2009).

[6] An efficient technique for data hiding in audio signals (American academic and scholarly research journal special issue sep 2012).

[7] Analysis and Implementation of Distinct Steganographic Methods - Ãnal TATAR, Department of Information Systems Security.

[8] Steganography and Steganalysis: Different Approaches - Soumyendu Das, Subhendu Das, Bijoy Bandyopadhyay - Institute of Radio physics Electronics.

[9] Stegnographic Methods - Jozsef LENTI, Department of Control Engineering and Information Technology.