Audio Steganography

Bibek Shrestha [a], Binod Manandhar [b], Sugam Dhakal [c], Diwan Bahadur Kumal [d], Sharmila Bista [e]

[a, b] *Department of Electonics and Computer Engineering, National College of Engineering, Lalipur, Nepal*

Corresponding Email : [a] wiwek99@gmail.com, [b]bindmdr@gmail.com, [c] sagar.dhakal999@gmail.com, [d] diwankumal52@gmail.com, :

[e] sharmilabista10@gmail.com

Abstract— **Hiding information is an art and science of communication technology. Using Steganography information can be hidden within other information that it cannot be detected, but only by its intended recipient. Nowadays for hiding information there are different multimedia systems like audio. Embedding secret message in sound is a difficult process. There are varying techniques for embedding information in audio. In this paper we will attempt the general principles of hiding secret information using audio technique of least significant bit and an overview of its functions.**

Keywords—Audio Steganography, Security, Secret data transmission, Least Significant bit.

I. INTRODUCTION

Steganography is the technique of hiding information in some media for the safe communication. This technique relies on encoding messages in transport layers in such a manner that the existence of the message is unknown to an observer. The aim of Audio Steganography is to hide a message in some cover media and to obtain new data which is indistinguishable from the original message, by people in such a way that any eavesdropper cannot detect the presence of the original message in new data. Computers and Networks, has made information hiding in Covert channels and text possible. Now days audio files are available everywhere and moreover; today's technology allows the copying and redistribution of audio files over the Internet at a very low or almost no cost. So it is necessary to have methods that confine access to these audio files and also for its security. Audio Steganography is one of the solutions..

In Audio Steganography, the weakness of the Human Auditory System (HAS) is used to hide information in the audio [1]. Many programs are available in the internet that uses Steganography to hide the secret information. The media's that use digitally embedding message are plain text, hypertext, audio/video, still image and network traffic. There exists a large variety of Steganography techniques with varying complexity and possessing, some with strong and weak aspects. Information hiding in text is the most popular method of Steganography. It is used to hide a secret message in every nth character or altering the amount of white space after lines or between words of a text message [2]. It is used in the initial decade of the internet era. But it is not used frequently because the text files have a small amount of redundant data.

This technique lacks in payload capacity and robustness. To hide data in audio files, the secret message is embedded into the digitized audio signal. The audio data hiding method provides the most effective way to protect privacy. A key aspect of embedding text in audio files is that, no extra bytes are generated for embedding. Hence it is more comfortable to transmit the huge amount of data using audio signal. Embedding the secret messages in digital sound is usually a very difficult process [3].

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].

II. LSB TECHNIQUE OF AUDIO STEGANOGRAPHY

The Least Significant Bit Coding (LSB): LSB is one of the earliest techniques in the information hiding of digital audio, as well as other media types. In this technique LSB of binary sequences of each sample of digitized audio file is replaced with the binary equivalent of secret message [9]. 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 the binary equivalent of the letter, A [8].

Advantages**:** It is the simplest form to embed information in a digital audio file. It allows a 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) [9]. This method is more widely used, as modifications to LSB's usually do not create audible changes to the sounds.

Disadvantage: It has considerably low robustness against attacks.

III. WAVE FORM AUDIO FORMAT

Wavelet domain [13] is suitable for frequency analysis because of its multi-resolution properties that provides access to both most significant parts and details of spectrum. Wavelet domain techniques work with wavelet coefficient. Upon applying the inverse transform, the stegano signal can be reconstructed.

Advantage: It has high data hiding capacity and transparency. Disadvantage: Lossy data retrieval.

WAV Format

WAV (or WAVE), short for Waveform audio format, is a Microsoft and IBM audio file format standard for storing audio on PCs. It is a variant of the RIFF bit stream format method for storing data in "chunks". A WAVE file is often just a RIFF file with a single "WAVE" chunk which consists of two sub-chunks -- a "fmt" chunk specifying the data format and a "data" chunk containing the actual sample data.[13]

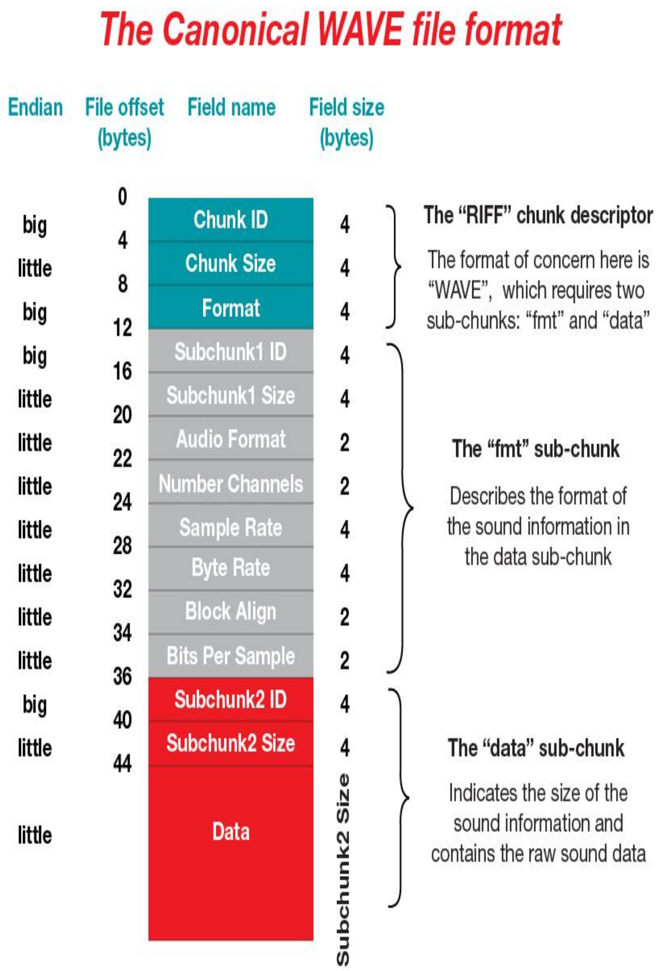

Figure 6 The Canonical Wave File Format

The canonical WAVE format starts with the RIFF header [2]:

0-4 Chunk ID — Contains the letters "RIFF" in ASCII form

4-4 Chunk Size — 36 + SubChunk2Size, or more precisely:

4 + (8 + SubChunk1Size) + (8 + SubChunk2Size)

This is the size of the rest of the chunk

Following this number. This is the size of the

Entire file in bytes minus 8 bytes.

Two fields not included in this count: Chunk ID and Chunk Size. 8-4 Format contains the letters "WAVE". The "WAVE" format consists of two sub chunks: "fmt " and "data". The "fmt" subchunk describes the sound data's format: 12-4 Subchunk1ID Contains the letters "fmt ". 16-4 Subchunk1Size 16 for PCM.

This is the size of the rest of the Sub chunk which follows this number:

20-2 Audio Format — PCM = 1(i.e. Linear quantization)

Values other than 1 indicate some

Form of compression.

22-2 NumChannels — Mono = 1, Stereo = 2, etc.

24-4 SampleRate — 8000, 44100, etc.

28-4 ByteRate = SampleRate* NoChannels* BitsPerSample/8

32-2 BlockAlign = Number Channels * BitsPerSample/8

The number of bytes for one sample including all channels.

34-2 BitsPerSample — 8 bits = 8, 16 bits = 16, etc.

The "data" subchunk contains the size of the data and the actual sound:

36-4 Subchunk2ID Contains the letters "data"

40-4 Subchunk2Size No. Samples * NoChannels* BitsPerSample/8

This is the number of bytes in the data. You can also think of this as the size of the read of the subchunk following this number.

44 * Data The actual sound data.

IV. AUDIO STEGANOGRAPHY

Audio stenography is masking, which exploits the properties of the human ear to hide information unnoticeably. An audible, sound can be inaudible in the presence of another louder audible sound. This property allows selecting the channel in which to hide information.

The audio files may be modified for hiding data like other digital media like image, text or video. The methods that embeds data in sound files use the properties of the Human Auditory System (HAS). The HAS perceives the additive random noise and the perturbations in an audio file can also be detected. While the HAS have a large dynamic range, but it has a small different range. As a result, loud sounds tend to mask out quiet sounds. And there are also some distortions that are so common that the HAS ignores them. When we observe the audio wave file before embedding and after embedding secret data respectively in Fig.1 and Fig.3 the human auditory system can't recognize the small change. The digital sound is obtained from the analog sound by converting it to digital domain. This process implies two sub processes: sampling and quantization. Sampling is the process in which the analogue values are only captured at regular time intervals.

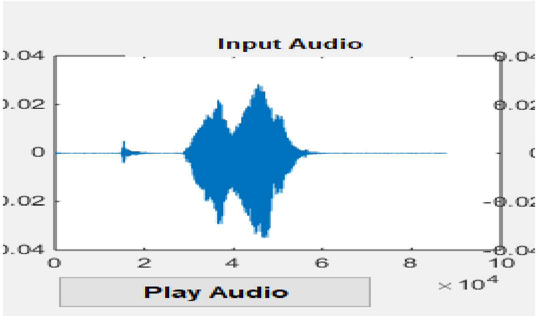

Figure 7 Before embedding the text

Quantization converts each input value into one of a discrete value. Popular sampling rates for audio include 8 kHz, 9.6 kHz, 10 kHz, 12 kHz, 16 kHz, 22.05 kHz and 44.1 kHz. The most popular file formats for kHz and 44.1 kHz. The most popular file formats for sounds are the Windows Audiovisual (WAV) and the Audio Interchange File Format (AIFF). There are also compression algorithms such as the International Standards Organization Motion Pictures.

Expert Group-Audio (ISO MPEG-AUDIO). When developing a data hiding method for audio, one of the first considerations is the likely environments the sound signal will t ravel between encoding and decoding. The two main areas of modification to be considered are, the storage environment or digital representation of the signal that will be used and the transmission pathway the signal might travel [3, 4]. In order to conceal secret messages successfully, a variety of methods for embedding information in digital audio have been introduced. These

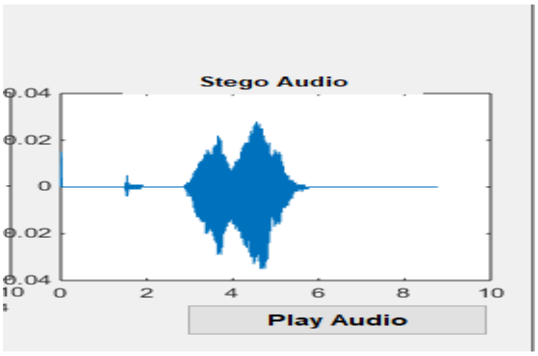

Figure 8 After Embedding the text

methods range from rather simple algorithms that insert information in the form of signal noise to more powerful methods that exploit sophisticated signal processing techniques to hide information. This section of the paper is organized as follows. First, the clarification of the Audio Environment. Secondly, this section describes as one of the wide range of techniques that have been used in Audio Steganography.

V. TECHNIQUE USED FOR DATA HIDING IN AUDIO

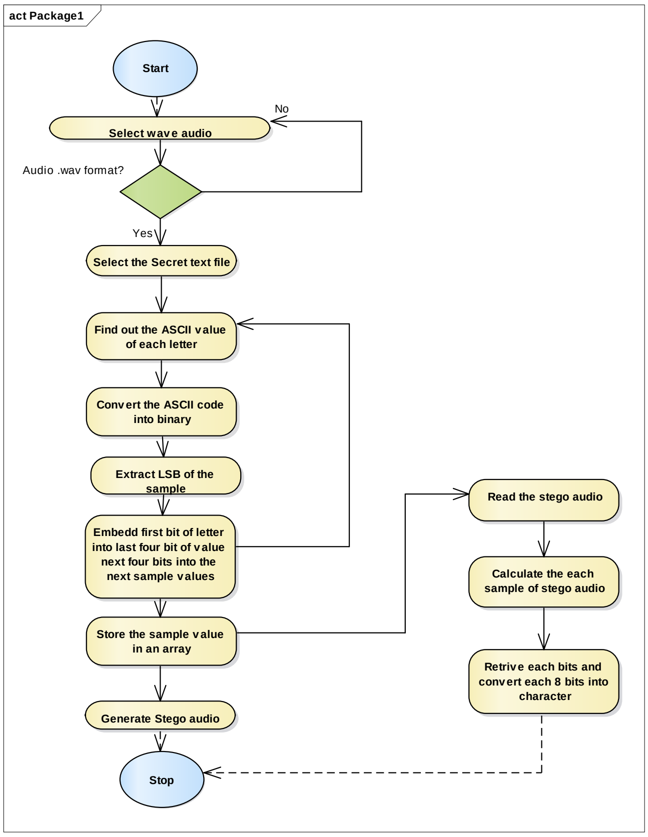

Figure 9 Flow Chart of LSB method

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 [14]. The following Fig.4 demonstrates how the message is encoded in 8-bit sample using the LSB method.

In LSB coding, the ideal data transmission rate is 1 kbps per 1 kHz. However, in some implementations of LSB coding, 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.

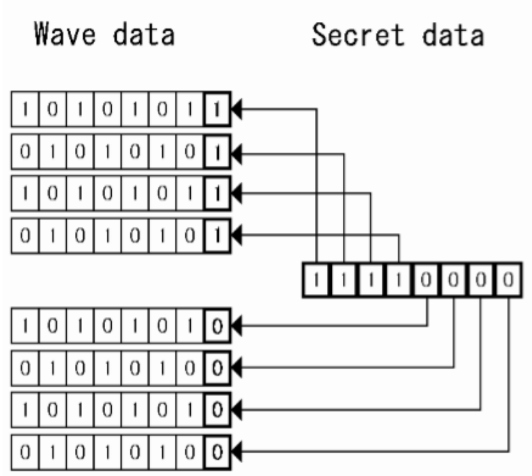

Figure 10 Low Bit Encoding

To extract a secret message from an LSB encoded sound file, the receiver needs access to the sequence of sample indices used in the embedding process. Normally, the length of the secret message to be encoded is smaller than the total number of samples in a sound file. One must decide then on how to choose the subset of samples that will contain the secret message and communicate that decision to the receiver. One trivial technique is to start at the beginning of the sound file and perform LSB coding until the message has been completely embedded, leaving the remaining samples unchanged. This creates a security problem, however in that the first part of the sound file will have different statistical properties than the second part of the sound file that was not modified. One solution to this problem is to pad the secret message with random bits so that the length of the message is equal to the total number of samples. Yet now the embedding process ends up changing far more samples than the transmission of the secret required. This increases the probability, that a would-be attacker will suspect secret communication. A more sophisticated approach is to use a pseudo random number generator to spread the message over the sound file in a random manner. One popular approach is to use the random interval method, in which a secret key possessed by the sender is used as a seed in a pseudo random number generator to create a random sequence of sample indices. The receiver also has access to the secret key and knowledge of the pseudo random number generator, allowing the random sequence of sample indices to be reconstructed. Checks must be put in place, however, to prevent the pseudo random number generator from generating the same sample index twice. If this happened, a collision would occur where a sample already modified with part of the message is modified again. The problem of collisions can be overcome by keeping track of all the samples that have already been used. Another approach is to calculate the subset of samples via a pseudo random permutation of the entire set using a secure hash function. This technique ensures that the same index is never generated more than once. There are two main disadvantages associated with the use of methods like LSB coding. The human ear is very sensitive and can often detect even the slightest bit of noise introduced into a sound file, second disadvantage, however, is that this is not robust. If a sound file embedded with a secret message using either LSB coding was resampled the embedded information would be lost. Robustness can be improved somehow by using a redundancy technique while encoding the secret message. However, redundancy techniques reduce the data transmission rate significantly.

A) Proposed Method

The Advantages of proposed method:

- Imperceptibility – The imperceptibility is one of the important factors of a steganography system, as its strength system depends on its ability to be unnoticed by the human senses (HAS/HVS).
- Robustness - Robustness defines how strong this technique against changes is. It measures the capability of the embedded secret data against different types of changes intended or unintended.
- Payload capacity - Payload capacity is the amount of embedded data that can be hidden into a particular innocent cover medium relative to the size of this medium.

B) Embedding Secret Text in Audio

The process of embedding the text in audio file, here we select audio file i.e. Windows Audio-Visual (WAV) file to input data. The file has 8-bit linear quantized digital audio samples. The header and data parts are separated. The secret text is placed in the header part and the LSB positions of the data element in alternate samples of LSB data element. Thus, Stego audio file is created. It’s a WAV file having hidden text, but there is no change in audibility of the Cover file.

1) Embedding Process

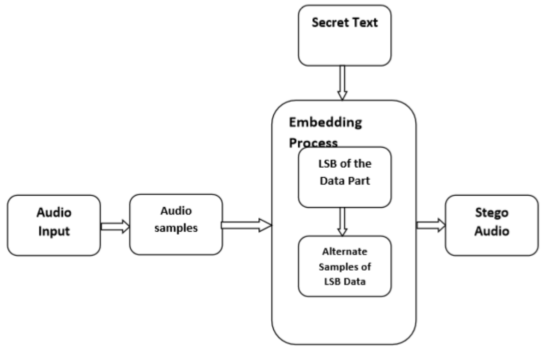

Figure 11 Block Diagram for Embedding

- The technique used is LSB coding. The audio file consists of data in bytes.
- To encode the message, we first find the length of the string.
- The offset in the original file, from which the encoding process must start.
- By default, it is set to 50. This is done because, the WAV file has a header. In the initial offsets and if that header is tampered with, the destination file will not be able to access its header in the appropriate format.
- Encode that length which can be up to 256 Characters into the 1st 8 bytes of the audio file. This will assist us in the decoding process. Take each character from the message string, convert it into byte and change the LSB of the next 8 bytes of the audio file as per each of the bit of the character type.
- Repeat the same procedure till the message string gets exhausted.
- Thus, on writing byte after byte to the new file, we get a new audio file ―Embedded .wav having a message hidden into it which can be sent to the receiver without any fear of eavesdropper.

C) Extraction of secret text from Stego Audio

The Process of extracting the hidden text from a WAV audio file is as follows. The input file is Stego audio (WAV) file and then separated header and data parts. The header consists of size of secret text. Store LSB of data part and perform a left shift of the previous bit. Then convert binary to ASCII values. Thus, the secret text can be extracted.

1) Extraction Process

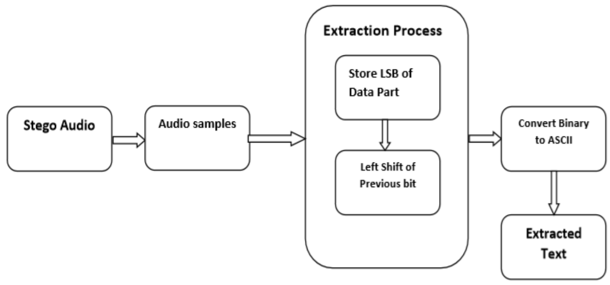

Figure 12 Block Diagram for Extraction

- Select the audio file ―Embedded.wav which has the message hidden in it.
- From the selected offset that was specified at the sending side (i.e. 500) , take the LSB of the next 8 bytes to get the length of the message (that was encoded in the first 8 bytes from the given offset) which will help us to get the encoded message only from the next 8 * length bytes of audio file.
- Create a byte from the LSB of the next consecutive 8 bytes and go on printing each of the character of the message string in the text box.
- Continue this process till the length of the string is reached. Finally, we get the hidden message from the received audio file into the provided text box.
- Thus, we have achieved the process of decoding a message from the audio file.

VI. Result

In this manner the secret text can be hidden and regenerated in audio signal using LSB method, which is difficult even to intruder's imagination of getting the information.

VII. Conclusion

Here we have proposed an efficient audio Steganography system, in which the LSB technique is used to get high data hiding capacity and low perceptibility. So, by using this technique the capacity of data hiding has increased and the clarity of the covering medium (.wav audio), remains unchanged even after hiding text. This can also be implemented on image Steganography.

Future Scope: Security for this system can be boosted by using a concept called Cryptography (i.e., encryption and decryption) to the text data. The system can be further developed to hide a secret image in covert audio as well as in two dimensional and three-dimensional signals.

Acknowledgement

The authors would like to acknowledge the support provided by National College of Engineering. They would also like to extend appreciation to the reviewers for their valuable positive suggestions.

References

[1] Shahreza S.S. and Shalmani M.T.M., "Adaptive wavelet domain Audio Steganography with high capacity and low error rate", in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp: 1729 – 1732, 2008.

[2] Bethany Delman, 'Genetic Algorithms in Cryptography Published in July 2004. https://ritdml.rit.edu/handle/1850/263, Accessed on 22nd Jan 2012

[3] Darrell Whitley,'A Genetic Algorithm Tutorial', Computer Science Department, Colorado State University, Fort Collins, CO 80523.

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

[5] RJ Anderson, Stretching the Limits of Steganography, Information Hiding, Springer Lecture Notes in Computer Science, 1174 (1996) 39-48.

[6] Scott. Craver, On Public-key Steganography in the Presence of an Active Warden, Proceedings of 2nd International Workshop on Information Hiding., (1998) 355-368.

[7] 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.

[8] Soumyendu Das, Subhendu Das, Bijoy Bandyopadhyay and Sugata Sanyal, "Steganography and Steganalysis:different_Steganograpy_Steganalysis_diff er ent_approaches.pdf.

[9] "Audio steg: methods", Internet publication on www.snotmonkey.com http://www.snotmonkey.com/work/school/405/methods . htm. Accessed on 22nd Jan 2012

[10] N. Cvejic, T. Seppanen, "Increasing Robustness of LSB Audio Steganography using a novel embedding method", in Proc. IEEE Int. Conf Info. tech.: Coding and Computing, Vol. 2, pp.533-537, April 2004.

[11] Samir K Bandyopadhyay, Debnath Bhattacharyya, Debashis Ganguly, Swarnendu Mukherjee and Poulami Das, "A Tutorial Review on Steganography" http://www.jiit.ac.in/jiit/ic3/IC3_2008/IC32008/APP2_2 1.pdf, Accessed on 22nd Jan 2012

[12] K. Gopalan and S. Wenndt, "Audio Steganography for covert data transmission by imperceptible tone insertion ", Proceedings of Communications Systems and Application, IEEE, 2011.

[13] N. Cvejic, T. Seppanen, "A wavelet domain lsb Insertion algorithm for high capacity audio Steganography ", IEEE International Conference on Acoustics, speech and Signal Processing, ICASSP 2008.

[14] P. Ramesh Yadav, V. Usha Shree, K. Padmapriya, Hiding Data in Audio Using Audio Steganography, International Journal of Computer Applications in Engineering Sciences, ISSN: 2231-4946.