IMPLEMENTATION OF STEGANO-AUDIO SYSTEM OVER LAN

Zin May Win, Tin Mar Kyi
Computer University (Mandalay)
cosmosflower.mail@gmail.com

ABSTRACT

Information hiding technique is a new kind of secret communication technology. Steganography is the art and science of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message. Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images and audio files are the most popular because of their frequency on the Internet. Cryptography focuses on keeping the contents of a message secret, and steganography focuses on keeping the existence of a message secret. In audio steganography the information is hidden exclusively in audio files. This paper used wave steganography and techniques for secure information. This system is offered the LSB algorithms in audio steganography to illustrate the security potential of steganography for business and personal use.

1. INTRODUCTION

Steganography is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. Today steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels [4, 6].

To hide information in audio files similar techniques are used as for image files. One different technique unique to audio steganography is masking, which exploits the properties of the human ear to hide information unnoticeably. A faint, but audible, sound becomes inaudible in the presence of another louder audible sound. This property creates a channel in which to hide information. Although nearly equal to images in steganographic potential, the larger size of meaningful audio files makes them less popular to use than images.

Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size.

This paper is included five sections including Introduction in this section. Section 2 describes theoretical background that includes: steganography, Method of Audio Steganography and different type of audio steganography, Section 3 describes Least Significant Bit (LSB), Data Embedding Algorithm, and Data Extraction Algorithm. Section 4 discusses system flow diagram for encoding and decoding. Section 5 explains the experimental results in step by step processing with figure. Finally, Section 6 presented the main conclusions, advantages and limitation of the system.

2. STEGNOGRAPHY

Steganography is a useful tool that allows covert transmission of information over an overt communications channel. Combining covert channel exploitation with the encryption methods of substitution ciphers and/or one time pad cryptography, steganography enables the user to transmit information masked inside of a file in plain view. The hidden data is both difficult to

detect and when combined with known encryption algorithms, equally difficult to decipher.

There exist two types of materials in steganography: message and carrier. Message is the secret data that should be hidden and carrier is the material that takes the message in it [4, 6].

Steganographic algorithms can be characterized by a number of defining properties. Three of them, which are most important for audio steganographic algorithms, are defined below.

Fig. 1 below shows the different categories of file formats that can be used for steganography techniques.

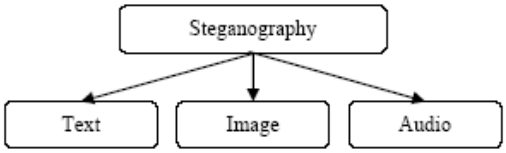


Figure 1. Steganography types diagram

2.1. Audio steganography

In audio steganography, secret message is embedded into digitized audio signal which result slight altering of binary sequence of the corresponding audio file. There are several methods are available for audio steganography. We are going to have a brief introduction on some of them [5].

2.1.1 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. Sampling technique followed by Quantization converts analog audio signal to digital binary sequence. In this technique LSB of binary sequence of each sample of digitized audio file is replaced with binary equivalent of secret message [7].

2.1.2 Phase Coding

Human Auditory System (HAS) can't recognize the phase change in audio signal as easy it can recognize noise in the signal. The phase coding method exploits this fact. This technique encodes the secret message bits as phase shifts in the phase spectrum of a digital signal, achieving an inaudible encoding in terms of signal-to- noise ratio [7].

2.1.3 Spread Spectrum

There are two approaches are used in this technique: the direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS). Direct-sequence spread spectrum (DSSS) is a modulation technique used in telecommunication. As with other spread spectrum technologies, the transmitted signal takes up more bandwidth than the information signal that is being modulated [7].

2.1.4 Echo Hiding

In this method the secret message is embedded into cover audio signal as an echo. Three parameters of the echo of the cover signal namely amplitude, decay rate and offset from original signal are varied to represent encoded secret binary message. They are set below to the threshold of Human Auditory System (HAS) so that echo can't be easily resolved [7].

3. LEAST SIGNIFICANT BIT

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover audio. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an audio is changed to a bit of the secret message. When using an audio, a bit of sample can be used, since they are each represented by a byte. In other words, one can store 3 bits in each sample [1, 2].

(00101101 0001110 11011100)
(10100110 1100010 00001100)
(11010010 1010110 01100011)

When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the audio, the resulting grid is as follows:

(00101101 00011101 11011100)

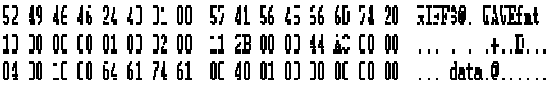
(10100111 11000101 00001100)
(11010011 10101100 01100011)

Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. On average, only half of the samples (bits) in an audio will need to be modified to hide a secret message using the maximum cover size.

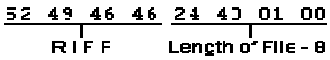
Changing the LSB of a sample is resulted in small changes in the intensity of the audio. These changes cannot be perceived by the human ear - thus the message is successfully hidden. With a well-chosen audio, one can even hide the message in the least as well as second to least significant bit and still not see the difference [1, 2].

3.1. The Wave File Format

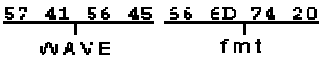
Wave file in a HEX editor



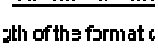
Every Wave file starts with the text "RIFF", followed by the Int32 length of the entire file:



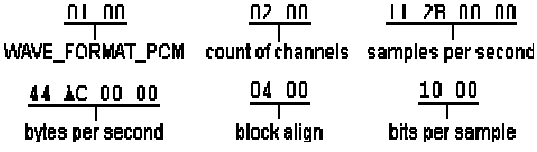
The next fields contain Wave data and open the format chunk:



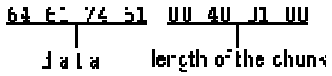
The length of the following format chunk must be 16.



The format is being specified by a WAVEFORMATEX structure:



The format chunk can be followed by some extra information. Then the interesting parts begin with the data chunk.



The data chunk contains all the Wave samples. That means the rest of the file is pure audio data. Little changes might be hearable, but won't destroy the file [10].

3.2. The MIDI File Format

When the sequencer (MIDI keyboard) saves the recorded messages, it places a header at the beginning of the file, and headers at the beginning of each track. Every header contains two fields for *type* and *length*. The type can be "MThd" for a file header, or MTrk for a track header. After the file header, the first track header has to follow [9]**:**

- The length of a track header specifies the count of bytes until the next track header begins.
- These bytes are system and MIDI messages.
- System messages have the type 0xFF, a subtype byte, and a length byte.
- Usually a file starts with a couple of non-midi messages, followed by Control Change messages, and then Program Change and Note On/Off messages:
- The end of every track is marked by an End of Track event.

3.3. Compressed Audio

Uncompressed audio files are very large. Here is the formula:

File size = SR × BD × L × C (1)

Where, SR is the sample rate, BD is the bit depth, L is the length of audio (in seconds), C is number of channels and SR × BD itself is called *bit rate*. That is why various audio compression algorithms have been researched. Most commonly used are MPEG audio layer III, a.k.a MP3. It can compress that 10 MB audio into about 1 MB. Other common alternatives are AAC (advanced audio coding) and WMA (windows media audio).

Equation (1) above only applies to uncompressed audio. For compressed ones, the formula is much simpler:

File size = bit rate × length of audio (2)

If the sound card only accepts uncompressed PCM, the CPU has to *decompress* the compressed audio first, before sending it to the sound card. When a compressed audio such as MP3 is being played on a media player, the media player will choose a suitable *codec* (compressor - decompressor) to decompress the audio. Every compression format has its own codec [3].

3.4. Data Embedding Algorithm

The choice of embedding algorithm in the most cases is driven by the results of the steganographic channel robustness analysis. Audio file is partitioned into frames of 90 ms duration. Then a random sample is selected using the key and secrete data is added into the lowest bit of the selected sample by the following procedure [5]:

- Read one byte of the message stream.
- For each bit in (message)
- Read a byte from the key
- Skip a couple of samples.
- Read one sample from the wave stream.
- Get the next bit from the current message byte.
- Place it in the last bit of the sample.

3.5. Data Extraction Algorithm

Firstly key is used to locate the sample in which secrete data is added.

- Read a byte from the key.
- Skip a couple of samples.
- Read one sample from the wave stream.
- Then the data bit is extracted from the lowest bit of the sample and put in data array [5].

4. SYSTEM FLOW DIAGRAM

This system composed of three major tasks. These are wave steganography, midi steganography and comparison of wave and midi steganography. In wave steganography, the system accepts three inputs from user such as text data, key and wave audio file. Then the system text data to stream and scan wave samples from wave file. By using least significant method the system is created stego wave file according to the key as shown in figure2.

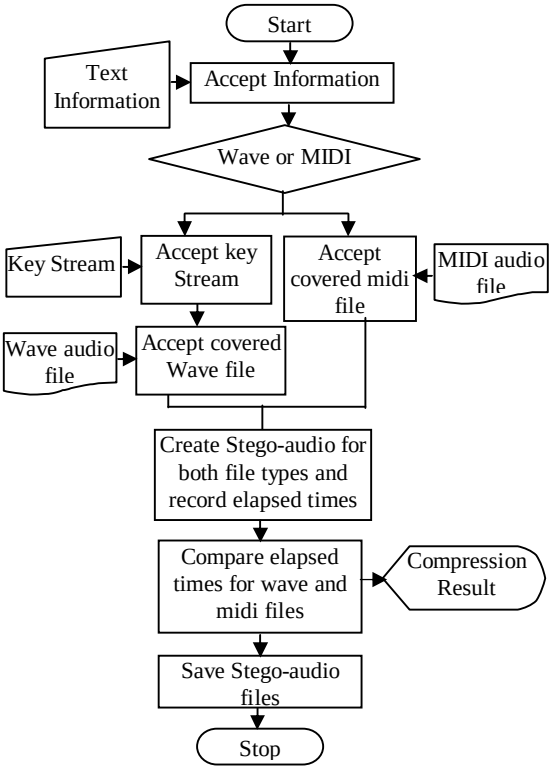


Figure 2. Steganographic encode flow diagram

In midi steganography, all are the same processes as wave steganography except the key file. The key file cannot use in midi steganography because the data is hidden between of “note on” and “note off” intervals that are randomly existed in figure 3.

Extract data from stego wave and midi processes are exactly the same reverse processes which used during the stego wave and midi file created. Compare the creation of stego wave and midi file used the same source text data and record the elapsed time for both processes.

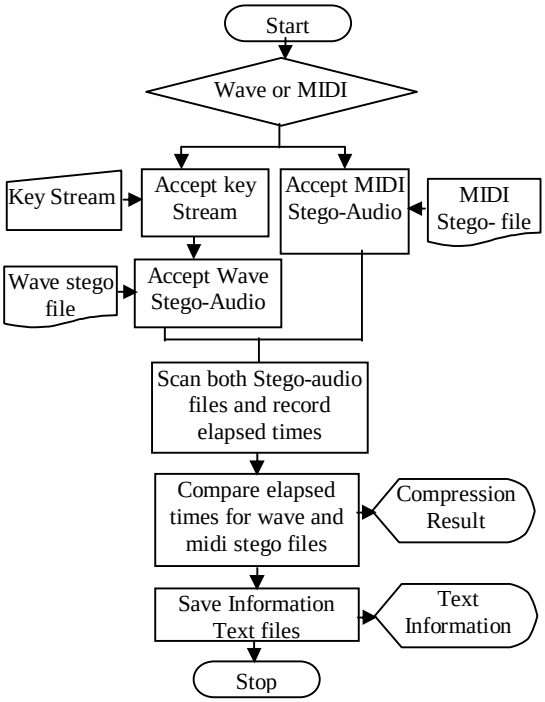


Figure 3. Steganographic Decode flow diagram

5. EXPERIMENTAL RESULT

This system gives the security in data communication. If the user wants to communicate the important message to another user, they will be protected of message by using this system. This system used user's text data as a stream in audio steganography and offered the user to two types of audio file types. The wave file hiding message process are shown in figure 4 and the midi file hiding message are shown in figure 5 respectively.

This hiding message can be sent to another user in local area network. The second user can extract the message form audio file using this system in figure 6. In the audio steganograhpic, the hiding message in audio and if unauthorized person can not know this message until opening this audio file. So, this system can be operated secure message communication process over LAN.

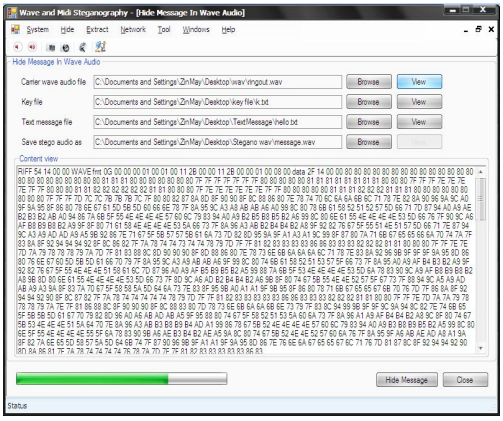


Figure 4. Hiding message in wave audio

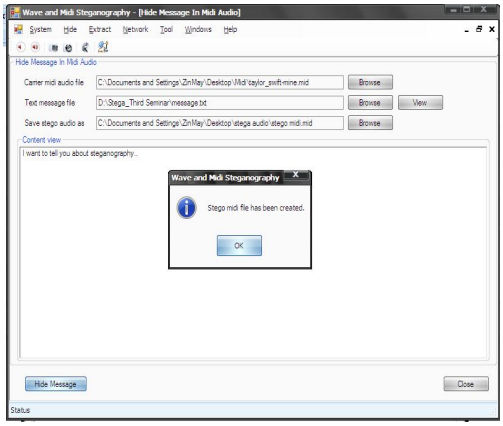


Figure 5. Hiding message in midi audio

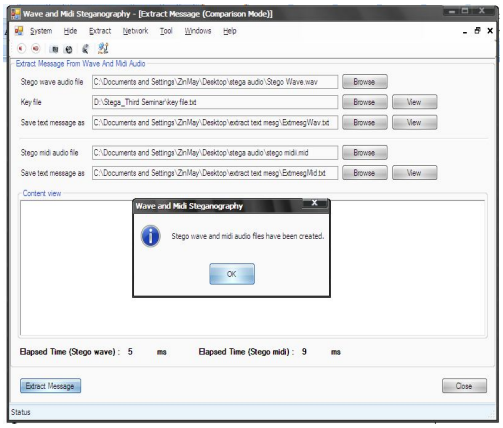


Figure 6. Comparison of elapsed time for extracting wave and midi stego audio

This process is shown in figure 7. This system was pleasurable from the start and only got more interesting as we went on developing it. This system had learned that while implementing Audio steganography is important, thinking of how to detect and attack it and the methods.

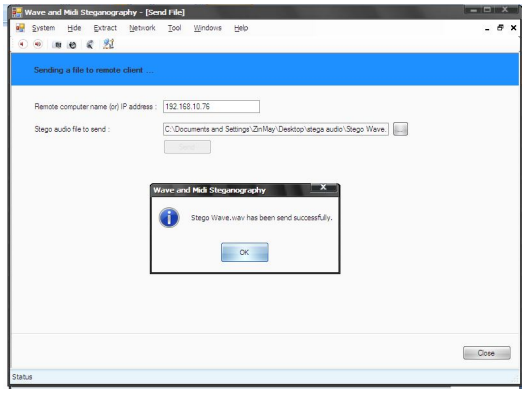


Figure 7. Sending wave and midi stego audio

6. CONCLUSION

Data hiding in the least significant bits (LSBs) of audio samples in the time domain is one of the simplest algorithms with very high data rate of additional information. The LSB watermark encoder usually selects a subset of all available host audio samples chosen by a secret key. The substitution operation on the LSBs is performed on this subset, where the bits to be hidden substitute the original bit values.

In LSB coding, the least significant bits of a sample are replaced with 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.

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.

REFERENCES

[1] Deshpande Neeta, Kamalapur Snehal, and Daisy Jacobs, “Implementation of LSB Steganography and its Evaluation for Varius Bits”, 2001.

[2] Dr. D. Mukhopadhyay,A. Mukherjee,S. Ghosh,S. Biswas, and P.Chakarborty, An Approach for Message hiding using Substitution Techniques and Audio Hiding in Steganography, IEEE 2005.

[3] F.A.P Petitcolas, R.J.Anderson, and M.G.Kuhn, “Information Hiding A Survey,” Proc.IEEE, vol.87, nol7, a999, pp.1062-1078.

[4] Ingemar J. Cox, Ton Kalker, Georg Pakura and Mathias Scheen “Information Transmission and Steganography”, springer, vol.3710/2005, pp.15-29.

[5] J. Johnston and K-Brandenburg, “Techniques for data hiding”, IBM Systems Journal, Volume 39, Issue 3-4, July 2000, PP.547-568

[6] K. M. Leung, “Introduction to Steganography”, Polytechnic University, Department of Computer and Information Science, November, 2004.

[7] M. Toth, “Steganography that is Hiding Information into Pictures and Other Media”, HTI, October, 2003.

[8] Muhalim Mohamed Amin, Subariah lbrahim, Mazleena Salleh, Mohd Rozi Katmin, “Information Hiding Using Steganography”, Department of Computer System & Allam Mousa and Ahmad Hamad.

[9] Noto, M., “MIDI Stego: Hiding Text in MIDI files”.

[10] Poulami Cutta, Debnath Bhattacharyya, and Tai-hoon Kim, Data Hiding in Audio Signal: A Review, International Journal of Database Theory and Application, vol.2, no.2, June 2009.