An Improvement for Hiding Data in Audio Using Echo Modulation

Huynh Ba Dieu
International School, Duy Tan University
182 Nguyen Van Linh, Da Nang, VietNam
huynhbadieu@dtu.edu.vn

ABSTRACT

This paper presents an improved technique to hide information in audio based on echo modulation. The improvement is done by using a sequence of random binary numbers and by modifying the formula used for adding echo. The experiment shows that our method is more security while still remains the inaudibility of stego audio file and suitable for hiding data in audio.

KEYWORDS

Audio steganography, echo hiding, cepstrum, audibility, random number generator.

1 INTRODUCTION

With the advancement of the technologies and the internet, storage and communication via digital data has gained a lot of significance. Due to this problem, the chances of digital information being illegal intercepted, modified and ravaged are being improved, which makes the information security becomes an important and urgent issue. Many techniques such as encryption and steganography are already used in this regard.

Conventional encryption algorithms permit only authorized users to access encrypted digital data. Once such data are decrypted, however, there is no way in prohibiting its illegal copying and distribution.

The term steganography is the technique of embedding secret information in a communication channel in such a manner that the very existence of the information is concealed [1], [2]. The aim is to embed and deliver secret messages in digital data without any suspiciousness. The secret message might be a text, an image, or any data that can be represented in bit stream form.

A steganography system, in general, is expected to meet three key requirements, namely, imperceptibility of embedding, correct recovery of embedded information, and large payload. Some degradation in the perceptual quality of the stego-signal from that of the original host signal may be acceptable.

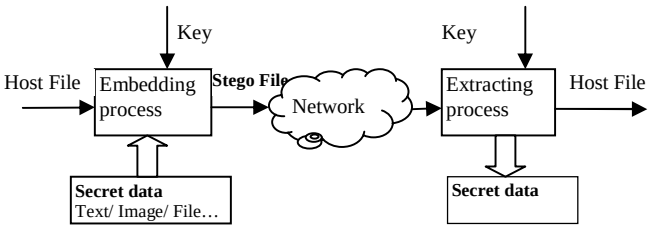

Figure 1. A diagram of the steganography system

Steganography techniques have been successfully applied on text files, images, audio and video files. Although, steganography in audio is a field not much explored. Audio steganography relies on the imperfection of the human auditory system. The human auditory system (HAS) is much more sensitive than the human visual system (HVS), so the space of frequency domain or time domain in audio signals where data can be embedded imperceptibly is limited.

In this paper we present a scheme for hiding data in audio using echo modulation. This method is an improvement of the ones developing by J. A. R. Chavez et al [3].

In following section, we present some common methods used in audio steganography. In section 3, the proposed method is presented. In section 4, the experimental results are discussed. In section

5, the work is summarized and proposes areas for further research.

2 METHODS FOR AUDIO STEGANOGRAPHY

There are many steganographic techniques for hiding secret data or messages in audio in a way that the modifications made to the audio file are perceptually indiscernible [4]. Several recent methods necessitate previous familiarity with signal processing techniques, Fourier transform, and other high level mathematics areas.

2.1 Least significant bit (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. 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 [2].

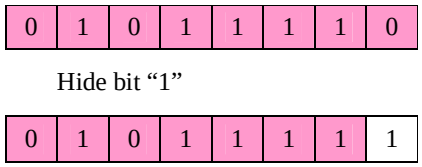

Figure 2. Illustration for LSB hiding technique

2.2 Parity coding

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 [5].

There are two main disadvantages associated with the use of methods like LSB coding or parity coding. The human ear is very sensitive and can often detect even the slightest bit of noise introduced into a sound file, although the parity coding method does come much closer to making the introduced noise inaudible. Both methods are not robust. The embedded information would be lost if stego audio is re-sampled.

2.3 Phase coding

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.

A characteristic feature of phase coding is the low data transmission rate owing to the fact that the secret message is encoded only in the first segment of the audio signal. On the contrary, an increase in the length of the segment would have a ripple effect by altering the phase relations between the frequency components of the segment; thereby making detection easier. Hence, the phase coding method is normally used only when a small amount of data (e.g., watermark needs to be masked) [1].

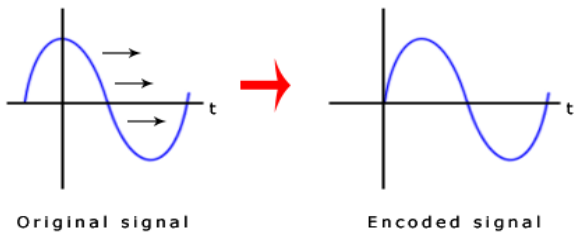

Figure 3. Phase coding procedure.

2.4 Spread Spectrum coding

The basic Spread Spectrum (SS) coding method randomly spreads the bits of the secret data message across the frequency spectrum of the audio signal. However, unlike LSB coding, the SS coding method spreads the secret message using a code that is independent of the actual cover signal.

The SS coding method can perform better than LSB coding and phase coding techniques by virtue of a moderate data transmission rate coupled with a high level of robustness against steganalysis techniques. However, like the LSB coding method, the SS method can introduce noise to the audio file [5].

2.5 Echo hiding

In echo hiding, information is embedded in a sound file by introducing an echo into the discrete signal. Like SS coding, echo hiding allows for a higher data transmission rate and provides superior robustness when compared to the noise-inducing methods.

To hide the data, three parameters of the echo are varied: amplitude, decay rate, and offset (delay time) from the original signal. All three parameters are set below the human hearing threshold so the echo is not easily resolved. In addition, offset is varied to represent the binary message to be encoded. One offset value represents a binary one, and a second offset value represents a binary zero.

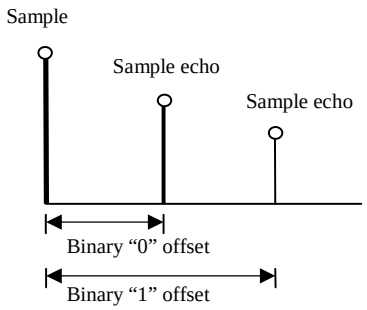

Figure 4. Echo coding procedure.

To hide the data, three parameters of the echo are varied: amplitude, decay rate, and offset (delay time) from the original signal. All three parameters are set below the human hearing threshold so the echo is not easily resolved. In addition, offset is varied to represent the binary message to be encoded. One offset value represents a binary one, and a second offset value represents a binary zero.

3 PROPOSED ALGORITHM

3.1 Previous work

In [3], Rios Chavez et al developed a scheme for embedding data in audio signal. The steps for embedding processes are:

1. Audio is divided into frames of size N.
2. Two vectors are generated: one is using for hiding bit “1” and another using for hiding bit “0”.

$$y(n)=\begin{cases} 0.99*x(n) & w(i)=0 \\ 0.98*x(n) + 0.1*x(n-d_1) & w(i)=1 \end{cases} \quad (1)$$

The extraction of watermark (secrete message in our scheme) in [3] is no blind. The algorithm consist to compare whether is each frame of watermarked signal less than or equal to each frame of the host signal. If it is true then removes a bit “0”, otherwise removes a bit “1”.

3.2 Analysis of the method in [3]

There are two disadvantages in [3]. First, in formula (1), if the value of x(n) is equal to value of x(n-d1), so the sum 0.98*x(n) + 0.1*x(n-d1) = 0.99 x(n). When doing extraction, we will take bit “0” instead bit “1”. Moreover, we have to spend more time for calculating when embedding bit “0” while it makes more difference between the host audio and the stego-audio. We can use copying instead of calculating a new value.

The original echo hiding technique is a bind technique. Extraction of embedded data is performed by calculating the cepstrum of each

frame, expressed by the following:

$$Re (IF (ln (|F(X)|))) \quad (2)$$

In the above expression, X is the vector of samples in one frame, F() is Fourier transformation, IF() is inverse Fourier transformation, ln() is natural logarithm and Re() takes the real part of complex number.

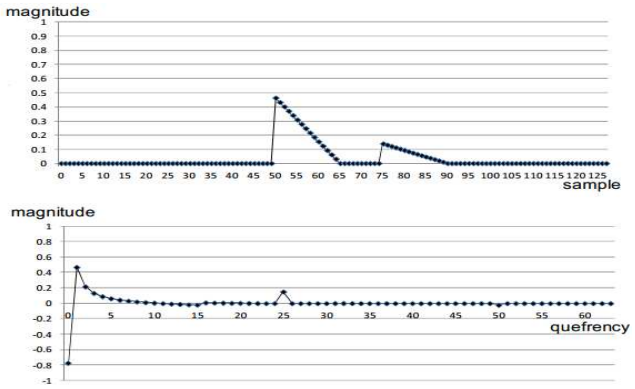

Figure 5. Echo and its corresponding quefrency.

If the data is embedded in the frame, we can observe the spike at its corresponding quefrency with means delay in cepstrum space [6]. This is the second disadvantage of [3]. When the steganalyst know the size of the frames, he has chance to get hidden message.

3.3 Proposed Scheme

The proposed scheme will deduct the disadvantage in [3] by modifying (1) and using a random binary sequence R for embedding. Base on the value of R_i, we can add echo to the host audio or copy the host audio for hiding the secrete bit W_i.

$$\text{If } R_i = \begin{cases} 1: & \text{copy for hiding bit 1,} \\ & \text{add echo for hiding bit 0.} \\ 0: & \text{copy for hiding bit 0,} \\ & \text{add echo for hiding bit 1.} \end{cases}$$

The sequence R, having same length with W, is generated base on the congruent generator [7].

First, we generate the sequence X with x_i is calculated by (3):

$$x_i = (ax_{i-1} + 2a) \bmod N \quad (3)$$

with a is a prime number, x_0 is assigned with seed and N is the upper limit of x_i.
After that, the value of R_i is calculated by (4):

$$R_i = \begin{cases} 1: \text{ if } (x_i \bmod 6) > 2 \\ 0: \text{ if } (x_i \bmod 6) <= 2 \end{cases} \quad (4)$$

The key is used for generating sequence R is a pair (seed, a).

In proposes scheme, base on the value of R_i, we can use copying host data for hiding bit "1" or "0". Similar, with adding echo, we can hide bit "1" or bit "0".

To remove the ambiguity when extract data if using (1), we propose the following formula:

$$y(n) = \begin{cases} 1.1*x(n) \\ \quad \text{if } x(n)=x(n-d_1) \\ 0.99*x(n) + 0.1*x(n-d_1) \\ \quad \text{if } x(n)<> x(n-d_1) \end{cases} \quad (5)$$

3.3.1 Algorithm for Embedding

Input:
Host audio file.
Secret message contains L bits.
Key is a pair of integers (seed, a).
Output:
Stego audio file.
The steps for the embedding process are:

1. Using key to generate sequence R follow (3) and (4)
2. Dividing the host audio into frames with size 1024. If the number of frames < L then exit.
3. Each frame is copied or added echo based on the value of R_i and W_i and use (5).
4. Concatenation the frames back together to create the stego audio file.

3.3.2 Algorithm for Extraction

Input:

Host audio file.
Stego audio file.
Key is a pair of integers (seed, a).
Length of secret message L.

Output:

Secret message W contains L bits.

The steps for the extracting process are:

1. Using key to generate sequence R follow (3) and (4).
2. Dividing the host audio and the stego audio into frames with size 1024.
3. Comparing the frame of the host audio with the frame of the stego audio and based on R_i to extract bit "1" or "0" to form secret message W, using (5). Loop until L bits are extracted.

3.3.3 Evaluation of the proposed approach

For embedding and extracting secret message W, we need to generate a sequence of bits R with the time complexity is O(L). By using (5), we extract bits with no ambiguity as using (1) when the value of x(n) is equal to value of x(n-d1). With using sequence R generated by key, the algorithm is more security event the steganalyst knows the size of frames and the method for hiding.

4 EXPERIMENTAL RESULT

In the experiment, we use a pair (7, 9137) as key to generate sequence R. By using (3) and (4), we have the first 16 values of R like this:

{1, 0, 0, 1, 1 ,0,1, 1, 1, 1, 0, 0, 0 , 1, 1, 0……}

The host audio file is a wav file, time length 58 second. The host audio file is sampled at a frequency of 44.1 Hz. We choose d1 = 441 to remain the delay of echo within 1ms.

The secret message W consist 1024 bits, is the image data of Tan University Logo file.

Figure 6. DuyTan university logo

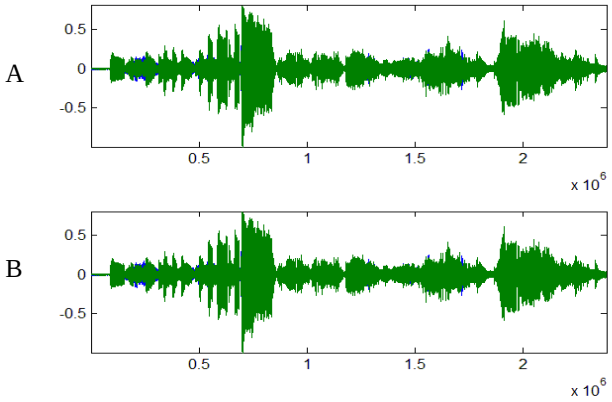

Figure 7 Host audio (A) and stego audio(B)

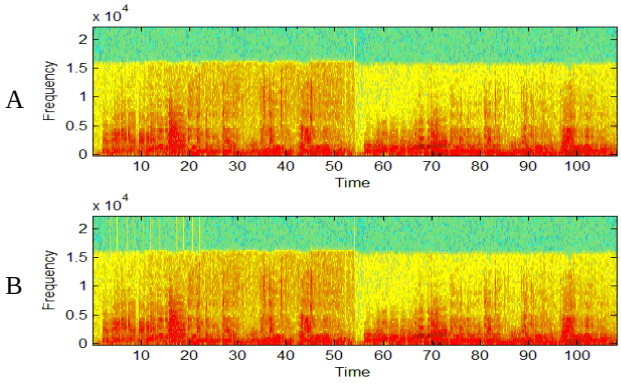

Figure 8. Spectrum of host audio (A) and stego audio (B)

Listening test results show that the distortion in the stego audio is imperceptible from the original audio. We use the Objective Difference Grade (ODG) [8] to test inaudibility with 100 people. Table I shows the percentage of test.

Table 1. Percentage of The ODG

Impairment Description	ODG(%)
Imperceptible	99
Perceptible, but not annoying	1
Slightly annoying	0
Annoying	0
Very annoying	0

The proposed scheme is used mainly for hiding data in audio files, not for watermarking. The robustness of the proposed scheme are based on the robustness of the echo hiding technique.

5 CONCLUSION

We have proposed a scheme for hiding data in audio using echo modulation. The experiment shows that our method is more security while still remains the inaudibility of stego audio file and suitable for hiding data in audio. In the future research, we plan to increase the payload and the robustness of this scheme by variegating the size of each frame and using error correction code.

6 REFERENCES

1. N. Cvejic, Digital Audio Watermaking Techiques and Technologies: Applications and Benchmark, Information Science Reference, Hershey NewYork, 2007.
2. M. Asad, J. Gilani and A. Khalid, “An enhanced least significant bit modification technique for audio steganography”, ICCNIT, pp. 143-147, July 2011.
3. J.A. R. Chavez, C. A. Ruiz and S. A G. Sanchez, “Audio watermarking of wav files by echo modulation”, CONIELECOMP, pp. 350-354, Feb 2012.
4. F. Djebbar, B. Ayad, H. Hamamz and K. A. Meraim, “A view on latest audio steganography techniques”, IIT, pp. 409- 414, April 2011.
5. N. Meghanathan and L. Nayak, “Steganalysis algorithms for detecting the hidden information in image, audio and video cover media”, IJNSA, vol 2, pp. 43-55, January 2010.
6. T. Shiro, H. Yamasaki, K. Yoda and Y. Watanabe, “Energy efficient echo hiding extarction method based on fine grain intermittent power control”, SAS, pp. 1-6, Feb 2012.
7. D. E. Knuth, The Art of Computer Programming, Volume 2: Seminumerical Algorithms, Addison-Wesley, 1997
8. Object difference grade. https://en.wikipedia.org/wiki/Objective_difference_grade (accessed Oct 30, 2013).